



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



**RANCANG BANGUN SISTEM INTERAKSI SUARA
NPC ADAPTIF BERBASIS *MULTI-TASK NATURAL
LANGUAGE UNDERSTANDING* PADA GIM THE
ELDER SCROLLS V: SKYRIM DENGAN
ARSITEKTUR *HYBRID CLIENT-CLOUD***

SKRIPSI
**POLITEKNIK
NEGERI
JAKARTA**

ANDI RIZKI MAHESA 2207411004

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



**RANCANG BANGUN SISTEM INTERAKSI SUARA
NPC ADAPTIF BERBASIS *MULTI-TASK NATURAL
LANGUAGE UNDERSTANDING* PADA GIM THE
ELDER SCROLLS V: SKYRIM DENGAN
ARSITEKTUR *HYBRID CLIENT-CLOUD***

SKRIPSI

**Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk
Memperoleh Diploma Empat Politeknik**

ANDI RIZKI MAHESA 2207411004

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Andi Rizki Mahesa
NIM : 2207411004
Jurusan/Program Studi : T.Informatika dan Komputer/Teknik Informatika
Judul skripsi : Rancang Bangun Sistem Interaksi Suara NPC Adaptif Berbasis *Multi-Task Natural Language Understanding* Pada Gim The Elder Scrolls V: Skyrim Dengan Arsitektur *Hybrid Client-Cloud*

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 01 Juli 2026

Yang membuat pernyataan



Andi Rizki Mahesa

NIM. 2207411004



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Skripsi diajukan oleh:

LEMBAR PENGESAHAN

: Andi Rizki Mahesa

: 2207411004

: Teknik Informatika

: Rancang Bangun Sistem Interaksi Suara NPC Adaptif Berbasis Multi-Task Natural Language Understanding pada Gim The Elder Scrolls V: Skyrim dengan Arsitektur Hybrid Client-Cloud

Telaah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal 17, Bulan Juni, Tahun 2026 dan dinyatakan **LULUS**.

Disahkan Oleh

Pemimbing I : Rizki Elisa Nalawati, S.T., M.T. ()

Penguji I : Euis Oktavianti, S.Si., M.T.I. ()

Penguji II : Susana Dwi Yulianti, S.Kom., M.Kom. ()

Penguji III : Malisa Huzaifa, S.Kom., M.T. ()

Mengetahui,

Jurusan Teknik Informatika dan Komputer
Ketua



Dr. Anita Hidayati, S.Kom., M.Kom.

NIP. 197908032003122003



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, karunia, dan kemudahan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Interaksi Suara NPC Adaptif Berbasis *Multi-Task Natural Language Understanding* Pada Gim *The Elder Scrolls V: Skyrim* Dengan Arsitektur *Hybrid Client-Cloud*”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan pada Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

Penulis menyadari bahwa penyusunan skripsi ini tidak dapat berjalan dengan baik tanpa bantuan, arahan, doa, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Anita Hidayati, S.Kom., M.Kom., selaku Ketua Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta.
2. Ibu Euis Oktavianti, S.Si., M.T.I., selaku Kepala Program Studi Teknik Informatika Politeknik Negeri Jakarta.
3. Ibu Rizki Elisa Nalawati, S.T., M.T., selaku dosen pembimbing yang telah memberikan arahan, masukan, bimbingan, dan dukungan selama proses penyusunan skripsi ini.
4. Seluruh dosen Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta yang telah memberikan ilmu, pengalaman, dan pembelajaran selama masa perkuliahan.
5. Seluruh responden, penguji, dan volunteer yang telah membantu proses pengujian sistem melalui survei, wawancara, UAT, SUS, maupun pengujian langsung terhadap sistem yang dikembangkan.
6. Komunitas Nexus Mods dan Reddit r/skyrimmods yang telah menjadi ruang publikasi, pengujian, serta sumber masukan dalam pengembangan sistem interaksi suara NPC pada penelitian ini.
7. Orang tua dan keluarga penulis yang selalu memberikan doa, dukungan, semangat, dan motivasi selama proses perkuliahan hingga penyusunan skripsi ini.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

8. Arjuna Dwi Damara Putra dan teman-teman penulis lainnya yang telah memberikan dukungan, bantuan, semangat, serta menjadi tempat berbagi selama proses pengerjaan skripsi.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan, baik dari sisi teknis sistem maupun penyusunan laporan. Oleh karena itu, penulis terbuka terhadap kritik dan saran yang membangun agar penelitian ini dapat dikembangkan lebih baik pada masa mendatang. Penulis berharap skripsi ini dapat memberikan manfaat bagi pembaca, khususnya dalam bidang pengembangan gim, pemrosesan bahasa alami, dan integrasi sistem kecerdasan buatan pada lingkungan permainan digital.





© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Andi Rizki Mahesa
NIM : 2207411004
Jurusan/Program Studi : T.Informatika dan Komputer / Teknik Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

Rancang Bangun Sistem Interaksi Suara NPC Adaptif Berbasis *Multi-Task Natural Language Understanding* Pada Gim The Elder Scrolls V: Skyrim Dengan Arsitektur *Hybrid Client-Cloud*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini, Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 01 Juli 2026

Yang Menyatakan



Andi Rizki Mahesa

NIM. 2207411004



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**RANCANG BANGUN SISTEM INTERAKSI SUARA NPC ADAPTIF
BERBASIS MULTI-TASK NATURAL LANGUAGE UNDERSTANDING
PADA GIM THE ELDER SCROLLS V: SKYRIM DENGAN
ARSITEKTUR HYBRID CLIENT-CLOUD**

ABSTRAK

Sistem dialog NPC pada gim The Elder Scrolls V: Skyrim masih banyak bergantung pada pilihan dialog statis, sehingga interaksi pemain dengan NPC cenderung terbatas, repetitif, dan kurang fleksibel dalam situasi permainan yang dinamis. Penelitian ini bertujuan merancang dan membangun sistem interaksi suara NPC adaptif berbasis Multi-Task Natural Language Understanding dengan arsitektur Hybrid Client-Cloud. Sistem dikembangkan dengan memadukan Client-STT berbasis Faster-Whisper, Server Cloud NLU berbasis RoBERTa, Fuzzy String Matching, FastAPI, PapyrusUtil, serta respons suara berbasis Baked Audio. Metode penelitian menggunakan pendekatan Mixed Methods melalui pengujian Black Box Testing, komparasi model NLU, User Acceptance Testing (UAT), System Usability Scale (SUS), dan profiling latensi. Hasil pengujian menunjukkan bahwa sistem memperoleh tingkat keberhasilan Black Box Testing sebesar 97,96% dari 98 kasus uji, skor SUS sebesar 79,67, dan skor UAT rata-rata sebesar 4,09 dari 5. Model RoBERTa juga menunjukkan performa NER yang lebih baik dibandingkan DistilBERT. Berdasarkan hasil tersebut, sistem yang dikembangkan mampu mendukung interaksi suara NPC secara fungsional, responsif, dan dapat diterima oleh pengguna.

Kata kunci: *Fuzzy String Matching; Hybrid Client-Cloud; Multi-Task Natural Language Understanding; NPC; Speech-to-Text*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR.....	iii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR KODE PROGRAM	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.4.1 Tujuan	4
1.4.2 Manfaat	5
1.5 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Permainan Digital, <i>Role-Playing Games</i> , dan <i>Non-Playable Character</i> .	7
2.1.1 Permainan Digital sebagai Sistem Interaktif.....	8
2.1.2 <i>Role-Playing Games</i> dan <i>Open-world Game</i>	8
2.1.3 <i>Non-Playable Character</i> sebagai <i>Game Agent</i>	9
2.2 Sistem Dialog Statis dan Keterbatasan Interaksi NPC Konvensional....	10
2.2.1 <i>Static Dialogue Trees</i>	11
2.2.2 Keterbatasan Dialog Menu dalam Interaksi <i>Real-time</i>	12
2.2.3 Kebutuhan Interaksi NPC yang Lebih Natural	12
2.3 Interaksi Suara sebagai Antarmuka Perintah Pemain.....	13
2.3.1 Antarmuka Suara dalam Sistem Interaktif	14
2.3.2 Alur Input Suara menjadi Perintah Sistem.....	14
2.3.3 Tantangan Interaksi Suara dalam Lingkungan Permainan.....	14



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1.4	<i>Speech-to-Text</i> dalam Sistem Interaksi Suara	15
2.4.1	<i>Automatic Speech Recognition</i>	16
2.4.2	Whisper dan Model STT Modern	16
2.4.3	Kesalahan Transkripsi dan Dampaknya terhadap Sistem	17
2.5	<i>Natural Language Understanding</i> dalam Pemrosesan Perintah Pemain	17
2.5.1	<i>Intent Classification</i>	18
2.5.2	<i>Named Entity Recognition</i>	19
2.5.3	<i>Sentiment Analysis</i>	19
2.6	<i>Multi-Task Learning</i> pada Model <i>Natural Language Understanding</i> ...	20
2.6.1	Konsep <i>Multi-Task Learning</i>	21
2.6.2	<i>Multi-Task Learning</i> untuk <i>Intent</i> , <i>Entity</i> , dan <i>Sentiment</i>	21
2.6.3	Keunggulan dan Keterbatasan <i>Multi-Task Learning</i>	22
2.7	Transformer dan RoBERTa dalam Pemahaman Bahasa Alami.....	22
2.7.1	Transformer Encoder untuk Pemrosesan Bahasa.....	23
2.7.2	<i>Fine-Tuning</i> Model Bahasa.....	24
2.7.3	RoBERTa untuk Tugas Klasifikasi dan Ekstraksi Informasi.....	24
2.8	Arsitektur <i>Hybrid Client-Cloud</i> dan <i>Computation offloading</i>	25
2.8.1	Konsep <i>Client</i> , <i>Server</i> , dan <i>Cloud</i>	26
2.8.2	<i>Computation offloading</i>	26
2.8.3	Pembagian Beban Kerja pada Sistem Interaksi Suara	27
2.9	<i>Fuzzy String Matching</i> sebagai Mekanisme Toleransi Kesalahan	27
2.9.1	<i>Exact matching</i> dan <i>Approximate Matching</i>	28
2.9.2	<i>Levenshtein Distance</i> dan <i>Similarity Ratio</i>	29
2.9.3	Penerapan <i>Fuzzy Matching</i> pada Entitas Permainan.....	29
2.10	Interoperabilitas Sistem AI dengan <i>Game engine</i>	29
2.10.1	<i>Game engine</i> sebagai Lingkungan Eksekusi Permainan.....	31
2.10.2	Pertukaran Data antara Sistem Eksternal dan <i>Game engine</i>	32
2.10.3	Papyrus, SKSE64, dan PapyrusUtil dalam Modding Skyrim.....	32
2.11	<i>Voice Cloning</i> , <i>Baked Audio</i> , dan Sinkronisasi Gerakan Bibir.....	34
2.11.1	<i>Text-to-Speech</i> , <i>Voice conversion</i> , dan <i>Voice Cloning</i>	35
2.11.2	<i>Baked Audio</i> sebagai Strategi Reduksi Latensi.....	35
2.11.3	Sinkronisasi Audio dan <i>Lip-sync</i> pada Karakter NPC	36
2.12	Metode Pengujian Sistem	36
2.12.1	<i>Black Box Testing</i>	37



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.12.2	<i>User Acceptance Testing</i>	38
2.12.3	<i>System Usability Scale</i>	38
2.12.4	<i>Profiling</i> Latensi dan Performa Sistem	38
2.13	Penelitian Terdahulu	39
2.13.1	Ringkasan Penelitian Terdahulu	39
2.13.2	Analisis Gap Penelitian	41
2.13.3	Posisi Penelitian yang Dilakukan	42
BAB III METODE PENELITIAN		44
3.1	Rancangan Penelitian	44
3.2	Tahapan Penelitian	44
3.3	Objek Penelitian	48
BAB IV HASIL DAN PEMBAHASAN		49
4.1	Analisis Kebutuhan Sistem	49
4.1.1	Kebutuhan Fungsional	49
4.1.2	Kebutuhan Non-Fungsional	49
4.1.3	Kebutuhan <i>Hardware</i>	50
4.1.4	Kebutuhan <i>Software</i>	52
4.2	Perancangan Sistem	53
4.2.1	Arsitektur Sistem	53
4.2.2	Perancangan Pemrosesan Suara Klien (<i>Client-STT</i>)	55
4.2.3	Perancangan <i>Natural Language Understanding (Cloud NLU)</i>	57
4.2.4	Perancangan Modifikasi Komponen <i>Game engine</i>	59
4.2.5	Perancangan Audio Generatif (<i>Baked Audio</i>)	60
4.3	Implementasi Sistem	62
4.3.1	Implementasi Pemrosesan Suara Klien (<i>Client-STT</i>)	63
4.3.2	Implementasi <i>Data Engineering</i> dan Pelatihan NLU	67
4.3.3	Implementasi <i>Server Cloud NLU</i>	78
4.3.4	Implementasi Skrip Modifikasi <i>Game engine</i>	89
4.3.5	Implementasi Integrasi Audio Generatif (<i>Baked Audio</i>)	114
4.4	Pengujian Sistem	133
4.4.1	Deskripsi Pengujian	134
4.4.2	Prosedur Pengujian	137
4.4.3	Data Hasil Pengujian	148
4.5	Pembahasan	187



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.5.1 Efisiensi Arsitektur <i>Hybrid Client-Cloud</i> dan Optimasi Perangkat Keras	188
4.5.2 Sinkronisasi Multimodal NLU-Papyrus dan Interoperabilitas <i>Game engine</i>	190
4.5.3 Keterbatasan Sistem dan Temuan Lapangan	191
BAB V PENUTUP	194
5.1 Kesimpulan.....	194
5.2 Saran	195
DAFTAR PUSTAKA	196
DAFTAR RIWAYAT HIDUP	199
LAMPIRAN	200





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2. 1 Alur Umum Interaksi Suara sebagai Perintah Pemain	13
Gambar 2. 2 Arsitektur Umum <i>Hybrid Client-Cloud</i> pada Sistem Interaksi Suara	25
Gambar 3. 1 Diagram Tahapan Penelitian	45
Gambar 4. 1 Topologi Arsitektur <i>Client-Cloud</i>	54
Gambar 4. 2 Diagram Alur Logika <i>Push-to-Talk</i> dan Transkripsi Lokal	55
Gambar 4. 3 Sequence Diagram Eksekusi Skrip Papyrus	59
Gambar 4. 4 <i>Pipeline</i> Pra-Produksi <i>Voice Cloning</i>	61
Gambar 4. 5 Eksekusi Terminal <i>Client-STT</i> saat Proses Transkripsi Konteks Konfigurasi Lokal Klien	67
Gambar 4. 6 Hasil Eksekusi Generasi Data dan Distribusi EDA	69
Gambar 4. 7 Metrik Pelatihan, Kurva <i>Loss</i> , dan Akurasi dari Terminal Google Colab	74
Gambar 4. 8 Black-Box Inference Testing pada Text Mode dan Simulasi STT Lowercase	74
Gambar 4. 9 Log Terminal Konsol Google Colab Pasca-Eksekusi INT8 Dynamic Quantization	78
Gambar 4. 10 Langkah eksekusi Docker build dan pembuatan citra kontainer di terminal Ubuntu VPS	80
Gambar 4. 11 Eksekusi Docker Build Dan Peluncuran Awal FastAPI Pada Terminal VPS Ubuntu DomaiNesia.....	81
Gambar 4. 12 Log aktivitas REST API pada terminal VPS saat sukses memproses skema JSON <i>Payload</i>	89
Gambar 4. 13 Diagram Alur Logika Siklus Hidup Event OnUpdate() Papyrus dengan Mekanisme Polling Loop dan Early Return Protocol.....	94
Gambar 4. 14 Struktur visual file JSON <i>SeranaContext</i> dan <i>SeranaCommand</i> di dalam direktori penyimpanan lokal <i>game Client</i>	96
Gambar 4. 15 Visualisasi perubahan ekspresi wajah NPC <i>Serana</i> yang bergeser secara dinamis: (a) <i>Serana</i> tersenyum ketika command dengan Sentimen Positif; (b) <i>Serana</i> sedih ketika command dengan Sentimen Negatif	99
Gambar 4. 16 NPC melakukan pergerakan otomatis menghampiri pemain (ForceGreet) akibat terjadinya lonjakan akumulasi skor afeksi.....	105
Gambar 4. 17 Antarmuka penukaran barang Inventory	114
Gambar 4. 18 Tool Bethesda Archive Extractor dan Yakitori Audio Converter	116
Gambar 4. 19 Antarmuka RVC Web UI Step 1 dan Step 2a-2b untuk pemrosesan dataset <i>Serana_Golden_v1</i> pada RunPod.....	120
Gambar 4. 20 Antarmuka RVC Web UI Step 3 untuk penyiapan hyperparameters dan pengekseskuan pelatihan model	120
Gambar 4. 21 UI ElevenLabs API/Studio dengan konfigurasi model <i>eleven_v3</i> , pemilihan <i>Voice Elise</i> , dan parameter sintesis	123
Gambar 4. 22 Antarmuka tab Batch Conversion RVC Web UI yang menampilkan jalur direktori lokal input, pemilihan file indeks <i>Serana_Golden_v1</i> , dan penyesuaian parameter inferensi	128



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4. 23 Proses batch validation spesifikasi format audio WAV 44100Hz Mono menggunakan tool audio editor dan struktur direktori simetris berkas .wav dan .lip.....	130
Gambar 4. 24 UI Topic Info <i>Dialogue</i> Editor pada Creation Kit yang menampilkan gerbang kondisi nilai SeranaAudioPointer dan SeranaAffinityScore	131
Gambar 4. 25 Perbandingan Konvergensi Loss Function Pelatihan Model <i>Multi-Task Learning</i> : (a) Baris Atas: Grafik Baseline DistilBERT; (b) Baris Bawah: Grafik Model Final RoBERTa-base.....	176
Gambar 4. 26 Ringkasan Log Akhir Evaluasi Dua Tahap Model <i>Multi-Task Learning</i> pada Google Colab: (a) Kolom Kiri: Model Baseline DistilBERT Base Cased; (b) Kolom Kanan: Model Final RoBERTa-base.....	178
Gambar 4. 27 Grafik Distribusi Nilai <i>System Usability Scale</i> Berdasarkan Adjective Ratings	182
Gambar 4. 28 Kurva Komparasi Komponen Latensi <i>Client</i> STT vs <i>Cloud</i> NLU	186
Gambar 4. 29 Grafik Distribusi Umpan Balik Negatif Responden Berdasarkan Kluster Hambatan Operasional	193

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR TABEL

Tabel 2. 1 Perbandingan Konsep Dasar dalam Lingkungan Permainan Digital.....	7
Tabel 2. 2 Perbandingan Dialog Statis dan Interaksi Berbasis Bahasa Alami.....	11
Tabel 2. 3 Jenis Kesalahan Transkripsi dan Dampaknya terhadap Sistem.....	15
Tabel 2. 4 Hubungan Tugas NLU dengan Pemrosesan Perintah Pemain.....	18
Tabel 2. 5 Perbandingan Model Terpisah dan <i>Multi-Task Learning</i> dalam NLU	20
Tabel 2. 6 Peran Transformer dan RoBERTa dalam Sistem <i>Natural Language Understanding</i>	23
Tabel 2. 7 Contoh Penerapan <i>Fuzzy String Matching</i> pada Entitas Permainan	28
Tabel 2. 8 Komponen Interoperabilitas Sistem AI dengan <i>Game engine</i>	30
Tabel 2. 9 Perbandingan Teknologi Respons Suara dalam Sistem NPC	34
Tabel 2. 10 Ringkasan Metode Pengujian Sistem.....	36
Tabel 2. 11 Ringkasan Penelitian Terdahulu	40
Tabel 4. 1 Spesifikasi <i>Hardware Client</i> (Klien Lokal).....	50
Tabel 4. 2 Spesifikasi <i>Hardware Server</i> (VPS Domainsia)	51
Tabel 4. 3 Spesifikasi Komputasi Pra-Produksi (RunPod).....	51
Tabel 4. 4 <i>Skema Payload</i> REST API.....	57
Tabel 4. 5 Parameter Konfigurasi Audio dan Variabel Global <i>Client-STT</i>	63
Tabel 4. 6 Parameter dan Konfigurasi Model	71
Tabel 4. 7 Perbandingan Metrik Ukuran Berkas dan Efisiensi Kompresi Model NLU	77
Tabel 4. 8 Matriks Kode Sinyal Otonom Lingkungan dan Parameter Pemicu Skrip Papyrus.....	106
Tabel 4. 9 Spesifikasi Teknis Parameter Akustik Dataset Vokal Primer Karakter Serana.....	116
Tabel 4. 10 Analisis Komparasi Arsitektur Teknologi Kloning Suara RVC v1 vs RVC v2	118
Tabel 4. 11 Parameter Konfigurasi Hyperparameters Pelatihan Model RVC v2 Serana_Golden_v1	120
Tabel 4. 12 Distribusi Matriks Variasi 424 Berkas <i>Baked Audio</i> Berdasarkan Kategori <i>Intent</i> , <i>TargetType</i> , dan Jumlah Varian per Tier Afeksi	124
Tabel 4. 13 Parameter Konfigurasi Inferensi Massal RVC v2 Tahap Batch Conversion	128
Tabel 4. 14 Ringkasan Parameter Metodologi dan Instrumen Operasional Pengujian Sistem Terdistribusi Serana <i>Cloud AI</i>	136
Tabel 4. 15 Ringkasan Prosedur Pengujian Sistem.....	138
Tabel 4. 16 Lingkungan Pengujian <i>Black Box Testing</i>	139
Tabel 4. 17 Matriks Sebaran Cakupan Distribusi Kasus Uji <i>Black Box Testing</i>	140
Tabel 4. 18 Karakteristik Responden Pengujian	146
Tabel 4. 19 Parameter Pengukuran Performa Sistem.....	147
Tabel 4. 20 Hasil <i>Black Box Testing</i> - Skenario 1: Movement_GoTo.....	150
Tabel 4. 21 Hasil <i>Black Box Testing</i> - Skenario 2: Movement_Follow.....	151
Tabel 4. 22 Hasil <i>Black Box Testing</i> - Skenario 3: Movement_Wait	153
Tabel 4. 23 Hasil <i>Black Box Testing</i> - Skenario 4: Combat_Attack	154

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 24 Hasil <i>Black Box Testing</i> - Skenario 5: <i>Combat_Tactical</i>	156
Tabel 4. 25 Hasil <i>Black Box Testing</i> - Skenario 6: <i>Social_Chat</i>	157
Tabel 4. 26 Hasil <i>Black Box Testing</i> - Skenario 7: <i>Social_Status</i>	159
Tabel 4. 27 Hasil <i>Black Box Testing</i> - Skenario 8: <i>Inventory_Trade</i>	160
Tabel 4. 28 Hasil <i>Black Box Testing</i> - Skenario 9: <i>System_Dismiss</i>	161
Tabel 4. 29 Hasil <i>Black Box Testing</i> – Skenario 10: <i>Interact_Environment</i>	162
Tabel 4. 30 Hasil <i>Black Box Testing</i> - Skenario 11: <i>Pemicu Sinyal Otonom Lingkungan</i>	165
Tabel 4. 31 Hasil <i>Black Box Testing</i> - Skenario 12: <i>Regulasi Status Affinity Lokal</i>	166
Tabel 4. 32 Hasil <i>Black Box Testing</i> - Skenario 13: <i>Validasi Jembatan Data Payload</i>	168
Tabel 4. 33 Hasil <i>Black Box Testing</i> - Skenario 14: <i>Penanganan Kasus Batas ..</i>	169
Tabel 4. 34 Rekapitulasi Hasil Eksperimen <i>Black Box Testing</i>	171
Tabel 4. 35 Matriks Komparasi Performa Akhir Model <i>DistilBERT Base Cased</i> vs <i>RoBERTa-base</i>	174
Tabel 4. 36 Hasil Tabulasi Skor Kuesioner 10 Pertanyaan <i>System Usability Scale (SUS)</i>	179
Tabel 4. 37 Hasil Evaluasi Dimensi Penerimaan Pengguna - <i>User Acceptance Testing (UAT)</i>	182
Tabel 4. 38 Matriks Hasil <i>Profiling Latensi Arsitektur Hybrid Client-Cloud</i>	185

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR KODE PROGRAM

Kode Program 4. 1 Pengalamatan Direktori Dinamis dan Inisialisasi Jalur Data	64
Kode Program 4. 2 Inisialisasi Model Faster-Whisper Terkuantisasi INT8 berbasis CPU	65
Kode Program 4. 3 Implementasi Fungsi Perekaman Audio Berbasis <i>Push-to-Talk</i>	66
Kode Program 4. 4 Implementasi Kamus Entitas dan Kerangka Kalimat Dasar..	68
Kode Program 4. 5 Konfigurasi LLM gemini-2.5-flash dan Injeksi Few-Shot Prompting.....	68
Kode Program 4. 6 Implementasi <i>Multi-Task Learning</i> dan Loss Function.....	70
Kode Program 4. 7 Kompilasi Arsitektur Multi-Task PyTorch ke Grafik ONNX FP32	75
Kode Program 4. 8 Implementasi Operasi Pembersihan Struktur Grafis (Graph Healer).....	76
Kode Program 4. 9 Konfigurasi Dockerfile Isolasi <i>Runtime Environment Server Cloud NLU</i>	79
Kode Program 4. 10 Skrip Inisialisasi Environment Jaringan FastAPI dan Pemuatan Engine ONNX INT8	81
Kode Program 4. 11 Implementasi Kelas Pydantic BaseModel dan Inisialisasi <i>Endpoint HTTP POST Asinkron</i>	82
Kode Program 4. 12 Implementasi Logika Post-Processing <i>Fuzzy String Matching</i> dan Protokol Failsafe.....	84
Kode Program 4. 13 Potongan Kode Post-Processing FormID Binding, <i>Sentiment Shift</i> , dan Konstruksi Outbound <i>Payload</i>	88
Kode Program 4. 14 Struktur Deklarasi Properti Tiga Lapisan Utama pada <i>SeranaAI_CoreScript.psc</i>	90
Kode Program 4. 15 Deklarasi Variabel State Tracker untuk Protokol Sensor Deteksi Perubahan Delta	91
Kode Program 4. 16 Peristiwa OnInit() dan Logika Failsafe Pemulihan Kendali Alias Aktor.....	92
Kode Program 4. 17 Blok Penanganan Early Return untuk State Machine Dialog ForceGreet pada OnUpdate()	93
Kode Program 4. 18 Logika Eksekusi Penulisan Telemetri Outbound pada <i>SeranaContext.json</i>	95
Kode Program 4. 19 Protokol Invalidasi Cache dan Pembacaan Inbound <i>Payload SeranaCommand.json</i>	96
Kode Program 4. 20 Logika Penanganan Ekspresi Wajah Berdasarkan Muatan Sentimen Awal <i>Cloud</i>	97
Kode Program 4. 21 Implementasi Pusat Percabangan Data (Action Demultiplexer) untuk 10 Fitur Utama Mod	98
Kode Program 4. 22 Implementasi Fungsi HandleMovement() Berbasis Protokol Smart Splitter NLU dan Proyeksi Trigonometri	101
Kode Program 4. 23 Pipa Pemrosesan Polling Dual-Scanner Navigasi dan Protokol Pembersihan Otomatis XMarker pada OnUpdate()	102

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kode Program 4. 24 Fungsi Pemetaan Kluster Hubungan Afeksi dan Logika Clamping Akumulasi Skor	103
Kode Program 4. 25 Protokol Pemicuan ForceGreet Otonom dan Mekanisme Penyelamat Timeout 45 Detik pada OnUpdate()	105
Kode Program 4. 26 <i>Pipeline</i> Polling Sensor Sinyal Otonom Lingkungan Sekuensial pada OnUpdate()	109
Kode Program 4. 27 Implementasi Pipa Pemindaian Tiga Lapis Penargetan Musuh pada Fungsi HandleCombat()	110
Kode Program 4. 28 Skrip Penanganan Fungsi HandleInteract() dan Blok Polling Sensor Jarak Proximity Scanner pada OnUpdate()	111
Kode Program 4. 29 Skrip Implementasi Fungsi Kontrol Pengikut Dawnguard dan Overriding Modifikasi Combat Style	113
Kode Program 4. 30 Rekaman Berkas Log Terminal Pipa <i>Runtime</i> Pelatihan RVC v2 Serana_Golden_v1 pada Infrastruktur <i>Cloud</i>	122
Kode Program 4. 31 Potongan Struktur Skema Multi-Level Nested Dictionary pada papyrus_Voicelines.json	132

POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Mekanisme interaksi pada sistem kecerdasan agen permainan tipe *Role-Playing Games* (RPG) dunia terbuka (*open-world*) saat ini secara umum masih bergantung pada desain pohon dialog statis (*static dialogue trees*). Ketergantungan terhadap percabangan teks konvensional tersebut menyebabkan karakter *Non-Playable Character* (NPC) follower cenderung merespons masukan pemain secara kaku, repetitif, dan kurang adaptif terhadap konteks permainan (Park et al., 2023). Kondisi ini mengurangi imersi serta ilusi otonomi dalam permainan karena NPC tidak memiliki kemampuan untuk mengingat riwayat interaksi sebelumnya maupun menyesuaikan respons sosial secara organik terhadap perubahan situasi pemain (Zheng et al., 2024). Akibatnya, interaksi yang terbentuk cenderung bersifat deterministik dan kurang mampu merepresentasikan perilaku sosial yang dinamis sebagaimana interaksi antar manusia.

Problem teknis dan operasional tersebut turut diperkuat oleh hasil survei pendahuluan yang dilakukan terhadap 18 pemain aktif *Role-Playing Games* dan *The Elder Scrolls V: Skyrim* secara spesifik. Hasil observasi menunjukkan bahwa mayoritas responden menilai sistem dialog tradisional terasa repetitif dan kurang fleksibel dalam mendukung kebutuhan interaksi permainan modern. Responden menyampaikan bahwa ketergantungan pada masukan manual yang mengharuskan pemain membuka menu dialog di tengah situasi genting, seperti pertempuran, sering kali memutus momentum permainan serta mengganggu fokus visual pemain terhadap lingkungan sekitar. Temuan awal ini menunjukkan adanya kebutuhan terhadap mekanisme interaksi yang lebih natural, cepat, dan mampu merespons konteks permainan secara lebih dinamis tanpa mengganggu alur permainan yang sedang berlangsung.

Sebagai respons terhadap permasalahan tersebut, pemanfaatan teknologi *Natural Language Understanding* (NLU) berbasis *Deep Learning* dipandang sebagai pendekatan yang mampu memahami konteks bahasa manusia secara lebih fleksibel dibandingkan sistem dialog berbasis aturan tetap. Melalui pendekatan ini, sistem



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dapat menafsirkan instruksi perintah, mengenali objek atau target yang dimaksud pemain, serta mengidentifikasi muatan emosional dari ucapan yang diberikan. Kebutuhan untuk mempertahankan karakteristik suara asli NPC mendorong penggunaan teknologi *Voice Cloning* agar respons verbal yang dihasilkan tetap sesuai dengan identitas karakter dalam permainan. Integrasi komponen *Speech-to-Text* (STT), NLU, dan penghasil respons suara secara bersamaan pada perangkat lokal berpotensi membebani kapasitas komputasi pengguna (Shen, 2025). Penggunaan model transkripsi suara modern seperti Whisper juga diketahui memerlukan sumber daya komputasi yang relatif besar untuk mempertahankan akurasi pengenalan suara pada berbagai kondisi lingkungan dan variasi pelafalan pengguna (Radford *et al.*, 2023). Kondisi tersebut berisiko menimbulkan *CPU-bound bottleneck* yang dapat menurunkan *frame rate* dan memicu fenomena *stuttering* selama sesi permainan berlangsung.

Mengatasi hambatan tersebut, penelitian ini mengusulkan penerapan arsitektur komputasi terdistribusi *Hybrid Client-Cloud*. Pendekatan *computation offloading* memungkinkan sebagian beban pemrosesan yang bersifat intensif dialihkan menuju infrastruktur *cloud* sehingga perangkat klien dapat mempertahankan stabilitas performa yang lebih baik (Zhang, Yi and Ma, 2024). Dalam penelitian ini, komputer lokal difokuskan untuk menangani proses penangkapan sinyal suara, pemrosesan transkripsi lokal, dan interaksi langsung dengan lingkungan permainan, sedangkan proses penalaran bahasa alami dijalankan pada infrastruktur *Server Cloud*. Komunikasi antara kedua komponen dilakukan melalui mekanisme pertukaran data asinkron untuk menjaga keandalan sistem, meminimalkan dampak kesalahan transkripsi, serta mempertahankan respons interaksi antara pemain dan NPC agar tetap berada dalam batas *near real-time* tanpa mengganggu stabilitas perenderan grafis pada sisi pengguna.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem interaksi suara berbasis arsitektur *Hybrid Client-Cloud* pada NPC Serana dalam permainan *The Elder Scrolls V: Skyrim*. Sistem yang dikembangkan mengintegrasikan komponen *Speech-to-Text*, *Natural Language Understanding* berbasis *Multi-Task Learning*, mekanisme toleransi kesalahan terhadap hasil transkripsi, serta respons suara berbasis *Baked Audio* hasil

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Voice Cloning untuk menghasilkan interaksi yang lebih natural, responsif, dan imersif dibandingkan pendekatan dialog konvensional.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan, perumusan masalah dalam penelitian ini berfokus pada pengembangan arsitektur komputasi terdistribusi dan pemrosesan bahasa alami, yang didefinisikan melalui pertanyaan penelitian sebagai berikut:

1. Bagaimana merancang arsitektur komputasi terdistribusi *Hybrid Client-Cloud* yang mampu memisahkan dan memindahkan beban kerja komputasi pemrosesan bahasa alami dari mesin gim (*game engine*) lokal secara asinkron dan stabil?
2. Bagaimana mengonstruksi alur pemrosesan (*pipeline*) *Natural Language Understanding* (NLU) berbasis *Multi-Task Learning* yang mampu melakukan klasifikasi intent, identifikasi entitas yang menjadi referensi objek atau lokasi dalam permainan, serta analisis sentimen secara simultan dalam satu siklus pemrosesan?
3. Bagaimana memitigasi risiko penurunan akurasi identifikasi entitas ketika sistem dihadapkan pada kesalahan ejaan atau cacat pelafalan pada hasil transkripsi suara sebelum perintah diproses oleh sistem permainan?
4. Bagaimana mengintegrasikan aset respons vokal hasil *Voice Cloning* berbasis *Baked Audio* beserta berkas sinkronisasi gerakan bibir karakter ke dalam sistem, sehingga respons interaksi yang dihasilkan tetap imersif?

1.3 Batasan Masalah

Agar ruang lingkup penelitian ini terarah dan berfokus pada penyelesaian masalah yang telah dirumuskan, batasan masalah ditetapkan sebagai berikut:

1. Penelitian ini dibatasi pada implementasi sistem interaksi suara pada permainan *The Elder Scrolls V: Skyrim Special Edition* dengan karakter *Non-Playable Character* (NPC) Serana sebagai subjek interaksi utama. Sistem menggunakan arsitektur *Hybrid Client-Cloud*, di mana proses transkripsi suara dilakukan pada perangkat klien dan proses *Natural Language Understanding* dilaksanakan pada *Server Cloud*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2. Sistem *Natural Language Understanding* (NLU) dibatasi pada pelaksanaan tiga tugas utama, yaitu klasifikasi intent (*Intent Classification*), pengenalan entitas bernama (*Named Entity Recognition*), dan analisis sentimen (*Sentiment Analysis*), yang diintegrasikan ke dalam satu arsitektur *Multi-Task Learning*.
3. Mekanisme perbaikan kesalahan identifikasi entitas permainan akibat inakurasi transkripsi suara dibatasi pada penggunaan algoritma *Fuzzy String Matching* yang merujuk pada basis data entitas permainan yang telah ditentukan sebelumnya.
4. Keluaran respons verbal NPC dibatasi pada pemanggilan aset audio statis (*Baked Audio*) hasil *Voice Cloning* yang telah diproduksi pada fase pra-produksi dan disinkronkan dengan berkas animasi gerakan bibir karakter (*lip-sync*), serta tidak melibatkan proses sintesis suara generatif secara *real-time* selama permainan berlangsung.

1.4 Tujuan dan Manfaat

Uraian mengenai tujuan yang ingin dicapai serta manfaat yang diharapkan dari penelitian ini.

1.4.1 Tujuan

Tujuan penelitian pembuatan sistem interaksi NPC cerdas ini adalah sebagai berikut:

1. Merancang dan mengimplementasikan arsitektur komputasi terdistribusi *Hybrid Client-Cloud* untuk melakukan pengalihan beban kerja (*computation offloading*) model kecerdasan buatan dari perangkat keras pengguna, sehingga interaksi suara antara pemain dan *Non-Playable Character* (NPC) dapat berlangsung stabil tanpa memicu degradasi kinerja atau penurunan *frame rate* pada mesin gim lokal.
2. Mengimplementasikan *pipeline Natural Language Understanding* berbasis model *Multi-Task Learning* di dalam infrastruktur *Server Cloud*, dengan tujuan agar sistem mampu melakukan klasifikasi niat, identifikasi entitas, dan analisis sentimen secara simultan dalam satu sesi inferensi.
3. Membangun dan mengintegrasikan mekanisme toleransi kesalahan (*fault-tolerance*) berbasis algoritma *Fuzzy String Matching* pada lapisan jembatan data

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

asinkron, guna meningkatkan tingkat keandalan dan akurasi identifikasi *Named Entity Recognition* apabila terjadi inakurasi transkripsi dari model *Speech-to-Text*.

4. Mengintegrasikan subsistem *Voice Cloning* menggunakan koleksi data audio statis (*Baked Audio*) beserta mekanisme sinkronisasi gerakan bibir karakter (*lip-sync*) ke dalam logika perilaku internal NPC, guna menghasilkan respons verbal berlatensi rendah yang tetap selaras dengan profil akustik orisinal dan konteks interaksi tanpa membebani proses sintesis audio secara *real-time*.

1.4.2 Manfaat

1. Memberikan pengalaman bermain yang lebih imersif melalui peningkatan ilusi kecerdasan karakter pendamping, yang dimungkinkan oleh penggantian sistem pohon dialog statis (*static dialogue trees*) dengan interaksi masukan vokal bebas.
2. Meningkatkan fleksibilitas interaksi pemain dengan NPC melalui perintah suara.
3. Memberikan mekanisme toleransi kesalahan terhadap hasil transkripsi suara melalui *Fuzzy String Matching*.
4. Menjadi referensi pengembangan sistem *Game AI* yang mengintegrasikan *Speech-to-Text*, *Natural Language Understanding*, *data bridge*, modifikasi *game engine*, dan *Baked Audio* dalam satu sistem interaktif berbasis *Hybrid Client-Cloud*.

1.5 Sistematika Penulisan

Laporan skripsi ini disusun ke dalam lima bab utama secara sistematis, mulai dari identifikasi permasalahan, landasan teori, metode penelitian, hasil dan pembahasan, hingga penarikan kesimpulan dan rekomendasi pengembangan pada penelitian selanjutnya.

Berikut adalah urutan pengorganisasian dokumen skripsi yang diterapkan:

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang permasalahan terkait keterbatasan interaksi NPC, perumusan masalah yang difokuskan pada infrastruktur terdistribusi, batasan ruang lingkup, serta tujuan dan manfaat penelitian. Bagian ini ditutup dengan penjelasan sistematika penulisan untuk memberikan gambaran umum mengenai struktur keseluruhan dokumen laporan akhir.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB II TINJAUAN PUSTAKA

Bab ini memuat jabaran landasan teori pendukung serta ulasan mengenai penelitian-penelitian terdahulu yang relevan dengan topik pengembangan sistem. Literatur tersebut difungsikan sebagai dasar argumentasi teknis sekaligus instrumen komparasi ilmiah untuk menjustifikasi keputusan rekayasa yang diterapkan pada sistem.

BAB III METODE PENELITIAN

Bab ini menjelaskan rincian rancangan penelitian berbasis metode campuran (*Mixed Methods*) dan alur kerja pengembangan yang digunakan untuk menyelesaikan permasalahan. Penjabaran di dalam bab ini mencakup definisi objek penelitian, tahapan rekayasa perangkat lunak, teknik pengumpulan data, hingga prosedur analisis data eksperimental.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memuat substansi teknis proyek yang meliputi analisis kebutuhan, perancangan arsitektur, implementasi sistem hulu-hilir, serta tabulasi data hasil pengujian secara menyeluruh. Evaluasi kritis turut dipaparkan pada sub-bab pembahasan untuk menganalisis efisiensi pemindahan beban komputasi menuju lapisan *Server Cloud*, serta mendokumentasikan keterbatasan aktual yang ditemukan selama sistem beroperasi di lingkungan gim.

BAB V PENUTUP

Bab ini mengungkapkan simpulan akhir secara logis, singkat, dan objektif terhadap penyelesaian masalah yang telah diuraikan pada bagian hasil dan pembahasan. Bagian ini juga memuat saran teknis dan operasional yang direkomendasikan untuk pengembangan maupun penyempurnaan kerangka kecerdasan buatan gim pada penelitian masa mendatang.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB II

TINJAUAN PUSTAKA

2.1 Permainan Digital, *Role-Playing Games*, dan *Non-Playable Character*

Permainan digital merupakan sistem perangkat lunak interaktif yang memungkinkan pemain melakukan tindakan, menerima respons dari sistem, dan berinteraksi dengan lingkungan virtual. Dalam konteks penelitian ini, permainan digital tidak hanya dipahami sebagai media hiburan, tetapi juga sebagai lingkungan komputasi yang memiliki aturan, objek, karakter, sistem input, sistem respons, dan logika internal. Sebelum membahas sistem interaksi suara, perlu dijelaskan terlebih dahulu konsep permainan digital, karakteristik *Role-Playing Games* (RPG), serta peran *Non-Playable Character* (NPC) sebagai entitas yang menjadi pusat interaksi pemain.

Sub-bab ini membahas tiga konsep dasar yang saling berkaitan. Pertama, permainan digital dipahami sebagai sistem interaktif. Kedua, RPG dan *open-world game* dijelaskan sebagai konteks permainan yang memberikan kebebasan eksplorasi dan interaksi. Ketiga, NPC dijelaskan sebagai agen permainan yang dikendalikan oleh sistem dan berperan dalam membangun pengalaman bermain. Pembahasan ini menjadi dasar untuk memahami mengapa peningkatan kualitas interaksi NPC diperlukan dalam penelitian ini.

Tabel 2. 1 Perbandingan Konsep Dasar dalam Lingkungan Permainan Digital

Konsep	Pengertian Umum	Peran dalam Penelitian
Permainan Digital	Sistem perangkat lunak interaktif yang memungkinkan pengguna melakukan tindakan dan menerima respons dari lingkungan virtual.	Menjadi lingkungan utama tempat sistem interaksi suara dijalankan.
<i>Role-Playing Games</i> (RPG)	Genre permainan yang menempatkan pemain sebagai karakter tertentu dengan	Menjadi konteks genre karena penelitian dilakukan pada permainan berbasis peran.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	kemampuan, misi, dan perkembangan karakter.	
<i>Open-world Game</i>	Desain permainan yang memberikan kebebasan eksplorasi kepada pemain dalam dunia virtual yang luas.	Menjelaskan mengapa interaksi NPC perlu fleksibel dan adaptif terhadap berbagai situasi pemain.
<i>Non-Playable Character (NPC)</i>	Karakter virtual yang dikendalikan sistem permainan, bukan pemain secara langsung.	Menjadi subjek utama interaksi suara dalam penelitian, yaitu NPC Serana.

2.1.1 Permainan Digital sebagai Sistem Interaktif

Permainan digital (*digital game*) merupakan sistem interaktif yang menerima masukan dari pemain, memproses tindakan tersebut melalui aturan permainan, kemudian menghasilkan respons dalam bentuk perubahan kondisi dunia virtual. Berbeda dari media pasif, permainan digital menempatkan pemain sebagai bagian dari sistem karena tindakan pemain dapat memengaruhi jalannya permainan. (Gallotta *et al.*, 2024) menjelaskan bahwa perkembangan kecerdasan buatan dalam permainan digital mendorong hadirnya sistem yang lebih adaptif, terutama dalam membangun agen virtual dan interaksi yang lebih natural.

Dalam penelitian ini, permainan digital diposisikan sebagai lingkungan eksekusi tempat sistem interaksi suara dijalankan. Artinya, sistem yang dikembangkan tidak berdiri sendiri, melainkan harus bekerja bersama komponen permainan seperti karakter, audio, animasi, skrip, dan logika interaksi. Sistem AI yang dikembangkan perlu dirancang agar mampu berkomunikasi dengan lingkungan permainan tanpa mengganggu stabilitas permainan yang sedang berjalan.

2.1.2 Role-Playing Games dan Open-world Game

Role-Playing Games (RPG) merupakan genre permainan yang menempatkan pemain sebagai karakter tertentu yang dapat berkembang melalui eksplorasi, penyelesaian misi, interaksi sosial, dan peningkatan kemampuan. Pada permainan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

RPG berbasis *open-world*, pemain memiliki kebebasan lebih besar untuk menjelajahi dunia permainan, menentukan urutan aktivitas, serta berinteraksi dengan berbagai karakter dan objek secara fleksibel. Karakteristik ini membuat RPG *open-world* memiliki variasi interaksi yang lebih luas dibandingkan permainan dengan alur linear.

Menurut (Gallotta *et al.*, 2024), meningkatnya kompleksitas dunia permainan modern mendorong kebutuhan terhadap agen virtual yang lebih adaptif terhadap tindakan pemain. Dalam konteks penelitian ini, *The Elder Scrolls V: Skyrim Special Edition* dipilih karena memiliki karakteristik RPG *open-world*, sistem NPC pendamping, serta lingkungan permainan yang memungkinkan pengujian sistem interaksi suara pada karakter NPC secara langsung.

2.1.3 Non-Playable Character sebagai Game Agent

Non-Playable Character (NPC) merupakan karakter virtual yang berada di dalam lingkungan permainan dan dikendalikan oleh sistem, bukan oleh pemain secara langsung. NPC berfungsi sebagai komponen penting yang mendukung jalannya permainan melalui penyediaan informasi, pemberian misi, interaksi sosial, hingga pendamping dalam aktivitas eksplorasi maupun pertempuran. Keberadaan NPC memungkinkan dunia permainan menjadi lebih hidup karena pemain dapat berinteraksi dengan entitas virtual yang memiliki peran tertentu dalam struktur permainan.

Dalam perspektif kecerdasan buatan, NPC dapat dipandang sebagai *game agent* yang menerima informasi dari lingkungan, melakukan proses pengambilan keputusan, kemudian menghasilkan tindakan sebagai respons terhadap kondisi yang dihadapi. Karakteristik agen tersebut menjadikan NPC sebagai salah satu komponen utama dalam implementasi kecerdasan buatan pada permainan digital. Seiring perkembangan teknologi AI, NPC tidak lagi hanya dirancang untuk menjalankan perilaku yang sepenuhnya statis, tetapi mulai dikembangkan agar mampu menampilkan perilaku yang lebih adaptif, kontekstual, dan responsif terhadap tindakan pemain.

Menurut (Zheng *et al.*, 2024), salah satu keterbatasan utama NPC konvensional adalah tidak adanya mekanisme memori yang memadai untuk menyimpan serta



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

memanfaatkan riwayat interaksi sebelumnya. Akibatnya, NPC cenderung memberikan respons yang sama terhadap situasi yang serupa meskipun pemain telah berinteraksi berulang kali. Kondisi tersebut menyebabkan perilaku NPC terasa kurang alami dan mengurangi tingkat imersi yang dirasakan pemain selama permainan berlangsung.

Perkembangan penelitian terkini menunjukkan bahwa peningkatan kualitas NPC diarahkan pada kemampuan memahami konteks interaksi, mempertahankan konsistensi perilaku, serta menyesuaikan respons berdasarkan pengalaman yang diperoleh selama permainan. Pendekatan ini bertujuan untuk menciptakan *game agent* yang lebih menyerupai perilaku sosial manusia sehingga interaksi yang terjadi antara pemain dan NPC dapat berlangsung secara lebih natural dan meyakinkan. Dalam penelitian ini, NPC diposisikan sebagai *game agent* yang menerima perintah suara dari pemain, memahami makna perintah melalui komponen *Natural Language Understanding* (NLU), dan menghasilkan respons perilaku maupun respons verbal sesuai konteks yang diberikan pemain.

2.2 Sistem Dialog Statis dan Keterbatasan Interaksi NPC Konvensional

Sistem dialog pada NPC konvensional umumnya menggunakan struktur percakapan yang telah ditentukan sebelumnya oleh pengembang. Pemain memilih salah satu opsi dialog yang tersedia, kemudian NPC memberikan respons sesuai cabang dialog yang telah dirancang. Pendekatan ini stabil dan mudah dikontrol, tetapi memiliki keterbatasan karena pemain hanya dapat berinteraksi melalui pilihan yang sudah disediakan.

Sub-bab ini membahas sistem dialog statis pada NPC, keterbatasannya dalam interaksi *real-time*, serta kebutuhan terhadap bentuk interaksi yang lebih natural. Pembahasan ini penting karena penelitian ini berangkat dari keterbatasan dialog menu konvensional dan mencoba menawarkan alternatif berupa perintah suara berbasis *Speech-to-Text* dan *Natural Language Understanding*.

Tabel 2.2 menunjukkan bahwa dialog statis memiliki keunggulan dari sisi kontrol narasi, tetapi memiliki keterbatasan dalam fleksibilitas dan adaptasi konteks. Sebaliknya, interaksi berbasis bahasa alami memungkinkan pemain memberikan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

masuk secara lebih bebas, meskipun membutuhkan sistem pemrosesan bahasa yang lebih kompleks agar perintah dapat dipahami dengan tepat.

Tabel 2. 2 Perbandingan Dialog Statis dan Interaksi Berbasis Bahasa Alami

Aspek	Dialog Statis	Interaksi Berbasis Bahasa Alami
Bentuk input	Pilihan menu atau teks yang telah disediakan	Kalimat atau perintah bebas dari pengguna
Fleksibilitas interaksi	Terbatas pada cabang dialog yang tersedia	Lebih fleksibel karena sistem dapat menafsirkan variasi kalimat
Adaptasi terhadap konteks	Rendah, karena respons telah ditentukan sebelumnya	Lebih tinggi, bergantung pada kemampuan sistem memahami konteks
Kesesuaian untuk situasi <i>real-time</i>	Kurang sesuai karena pemain perlu membuka menu dialog	Lebih sesuai karena perintah dapat diberikan secara langsung
Dampak terhadap imersi	Dapat terasa repetitif dan mekanis	Berpotensi meningkatkan kesan natural dan responsif

2.2.1 Static Dialogue Trees

Static dialogue trees merupakan struktur dialog bercabang yang berisi kumpulan pilihan percakapan dan respons NPC yang telah ditentukan sebelumnya. Dalam sistem ini, alur percakapan bersifat tetap karena pemain hanya dapat memilih opsi dialog yang tersedia. Pendekatan ini banyak digunakan dalam permainan RPG karena mampu menjaga konsistensi narasi, memudahkan pengujian, dan mengurangi risiko respons yang tidak sesuai dengan konteks cerita.

Namun, sistem dialog statis memiliki keterbatasan dalam menghadirkan interaksi yang fleksibel. (Hochreiter, Kriglstein and Wallner, 2026) menjelaskan bahwa interaksi NPC dalam permainan video masih banyak bergantung pada *scripted dialogue*, sehingga respons NPC cenderung terbatas pada dialog yang telah ditulis



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sebelumnya. Dalam konteks penelitian ini, keterbatasan tersebut menjadi dasar perlunya sistem interaksi yang memungkinkan pemain memberikan perintah secara lebih bebas melalui suara.

2.2.2 Keterbatasan Dialog Menu dalam Interaksi *Real-time*

Dialog menu memiliki keterbatasan ketika digunakan dalam situasi permainan yang membutuhkan respons cepat. Pemain perlu membuka menu dialog, membaca pilihan yang tersedia, lalu memilih salah satu opsi. Proses tersebut dapat mengganggu alur permainan, terutama pada kondisi eksplorasi atau pertempuran yang menuntut fokus dan tindakan cepat.

Dalam permainan *open-world*, pemain sering menghadapi situasi yang berubah secara dinamis. Dialog menu tidak selalu mampu menyesuaikan kebutuhan pemain secara langsung karena pilihan yang tersedia sudah dibatasi oleh sistem. Misalnya, pemain mungkin ingin memberi instruksi spesifik kepada NPC seperti menyerang musuh di lokasi tertentu, tetapi sistem dialog konvensional hanya menyediakan pilihan umum. Dialog menu kurang ideal untuk interaksi yang bersifat cepat, kontekstual, dan berbasis perintah langsung.

2.2.3 Kebutuhan Interaksi NPC yang Lebih Natural

Kebutuhan terhadap interaksi NPC yang lebih natural muncul karena pemain tidak hanya membutuhkan respons yang benar secara teknis, tetapi juga respons yang terasa sesuai dengan konteks permainan. (Park et al., 2023) menunjukkan bahwa agen virtual yang memiliki kemampuan memori, refleksi, dan perencanaan dapat menghasilkan perilaku sosial yang lebih believable. Hal ini menunjukkan bahwa NPC yang lebih kontekstual dapat meningkatkan kualitas pengalaman pemain.

Dalam penelitian ini, interaksi yang lebih natural diarahkan melalui penggunaan perintah suara. Pemain dapat memberikan instruksi dalam bentuk bahasa alami, kemudian sistem memproses perintah tersebut untuk menghasilkan *Intent*, *entity*, dan *sentiment*. Dengan pendekatan ini, NPC diharapkan tidak hanya merespons melalui dialog statis, tetapi dapat menerima perintah pemain secara lebih fleksibel sesuai konteks permainan.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.3 Interaksi Suara sebagai Antarmuka Perintah Pemain

Interaksi suara merupakan bentuk antarmuka yang memungkinkan pengguna memberikan instruksi kepada sistem melalui ucapan. Dalam konteks permainan digital, suara dapat digunakan sebagai media input alternatif selain tombol, menu, atau pilihan dialog. Pendekatan ini relevan untuk sistem interaksi NPC karena pemain dapat memberikan perintah secara langsung tanpa harus selalu membuka menu dialog.

Sub-bab ini membahas interaksi suara sebagai antarmuka perintah pemain. Pembahasan difokuskan pada konsep antarmuka suara, alur perubahan input suara menjadi perintah sistem, serta tantangan penerapannya dalam lingkungan permainan. Landasan ini diperlukan karena penelitian ini menggunakan suara pemain sebagai input awal yang kemudian diproses melalui *Speech-to-Text* dan *Natural Language Understanding*.



Gambar 2. 1 Alur Umum Interaksi Suara sebagai Perintah Pemain

Gambar 2.1 menunjukkan bahwa interaksi suara tidak langsung dipahami oleh sistem permainan. Ucapan pemain terlebih dahulu harus diubah menjadi teks melalui proses *Speech-to-Text*, kemudian dianalisis oleh komponen *Natural Language Understanding* untuk memperoleh makna perintah. Hasil pemahaman tersebut selanjutnya digunakan oleh sistem permainan untuk menentukan respons NPC yang sesuai.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.3.1 Antarmuka Suara dalam Sistem Interaktif

Antarmuka suara atau *Voice User Interface* merupakan antarmuka yang memungkinkan pengguna berinteraksi dengan sistem menggunakan ucapan. (Deshmukh and Chalmeta, 2024) menjelaskan bahwa antarmuka suara dapat meningkatkan naturalitas interaksi karena ucapan merupakan bentuk komunikasi yang umum digunakan manusia. Dalam sistem interaktif, antarmuka suara dapat mengurangi ketergantungan pengguna terhadap interaksi berbasis teks atau tombol.

Dalam penelitian ini, antarmuka suara digunakan agar pemain dapat memberikan perintah kepada NPC secara lebih langsung. Misalnya, pemain dapat mengucapkan perintah untuk mengikuti, menunggu, menyerang musuh, atau bergerak menuju lokasi tertentu. Dengan pendekatan ini, suara tidak hanya berfungsi sebagai input tambahan, tetapi menjadi bagian utama dari mekanisme interaksi antara pemain dan NPC.

2.3.2 Alur Input Suara menjadi Perintah Sistem

Agar suara pemain dapat dipahami oleh sistem, input suara perlu melewati beberapa tahap pemrosesan. Tahap pertama adalah *Speech-to-Text*, yaitu proses mengubah ucapan menjadi teks. Setelah itu, teks hasil transkripsi diproses oleh komponen *Natural Language Understanding* untuk menghasilkan informasi terstruktur seperti *Intent*, *entity*, dan *sentiment*. (Radford et al., 2023) menunjukkan bahwa model *Automatic Speech Recognition* modern seperti Whisper mampu melakukan transkripsi ucapan dengan memanfaatkan data audio berskala besar.

Dalam konteks penelitian ini, alur tersebut digunakan untuk mengubah ucapan pemain menjadi perintah yang dapat dieksekusi oleh NPC. Sebagai contoh, ucapan “Serana, attack the bandit in Solitude” dapat diproses menjadi *Intent* berupa perintah menyerang dan *entity* berupa target atau lokasi yang relevan. Hasil pemrosesan tersebut kemudian dikirim ke sistem permainan agar NPC dapat memberikan respons sesuai perintah pemain.

2.3.3 Tantangan Interaksi Suara dalam Lingkungan Permainan

Penerapan interaksi suara dalam permainan memiliki beberapa tantangan. Hasil transkripsi dapat dipengaruhi oleh kualitas mikrofon, kebisingan lingkungan, pelafalan pemain, dan beban sistem saat permainan sedang berjalan. Sistem perlu

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

memberikan respons dalam waktu yang cukup cepat agar interaksi tidak terasa lambat atau mengganggu alur permainan.

(Deshmukh and Chalmeta, 2024) menekankan bahwa keberhasilan antarmuka suara tidak hanya bergantung pada kemampuan mengenali ucapan, tetapi juga pada kualitas pengalaman pengguna. Dalam penelitian ini, tantangan tersebut diatasi melalui pembagian proses antara *Client* dan *Server Cloud*, penggunaan model *Speech-to-Text* lokal, serta pemrosesan NLU yang menghasilkan keluaran terstruktur. Dengan demikian, interaksi suara diharapkan dapat berjalan lebih stabil dan tetap sesuai dengan kebutuhan permainan.

1.4 *Speech-to-Text* dalam Sistem Interaksi Suara

Speech-to-Text merupakan komponen yang digunakan untuk mengubah ucapan pengguna menjadi teks. Dalam sistem interaksi suara, tahap ini menjadi proses awal sebelum sistem dapat memahami maksud perintah pemain. Tanpa proses transkripsi, suara pemain tidak dapat diproses lebih lanjut oleh komponen *Natural Language Understanding* karena model NLU membutuhkan masukan dalam bentuk teks.

Sub-bab ini membahas konsep *Automatic Speech Recognition*, model STT modern seperti Whisper, serta dampak kesalahan transkripsi terhadap sistem. Pembahasan ini penting karena kualitas hasil *Speech-to-Text* berpengaruh langsung terhadap keberhasilan sistem dalam mengenali perintah, target, lokasi, maupun konteks ucapan pemain.

Tabel 2. 3 Jenis Kesalahan Transkripsi dan Dampaknya terhadap Sistem

Jenis Kesalahan Transkripsi	Contoh Kasus	Dampak terhadap Sistem
Kesalahan ejaan	“whitrun” alih-alih “Whiterun”	Entitas lokasi berpotensi gagal dikenali.
Kesalahan nama objek	“livr” alih-alih “lever”	Sistem dapat gagal mengeksekusi aksi terhadap objek.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kesalahan nama musuh	“bndit” alih-alih “bandit”	Target serangan NPC dapat tidak terdeteksi.
Gangguan suara lingkungan	Musik keras atau noise keyboard	Transkripsi dapat gagal atau kehilangan kata kunci utama.
Ambiguitas pelafalan	Kata dengan bunyi mirip	Sistem dapat memilih entitas yang tidak sesuai.

Tabel 2.3 menunjukkan bahwa hasil transkripsi tidak selalu identik dengan ucapan asli pemain. Kesalahan kecil pada kata yang berfungsi sebagai entitas, seperti nama lokasi, objek, atau target musuh, dapat menyebabkan sistem salah memahami perintah. Sistem interaksi suara tidak hanya memerlukan model STT, tetapi juga mekanisme lanjutan untuk memitigasi dampak kesalahan transkripsi.

2.4.1 Automatic Speech Recognition

Automatic Speech Recognition (ASR) merupakan teknologi yang digunakan untuk mengenali ucapan manusia dan mengubahnya menjadi teks. ASR berperan sebagai jembatan antara input audio dan pemrosesan bahasa alami. (Ahlawat, Aggarwal and Gupta, 2025) menjelaskan bahwa perkembangan *Deep Learning* telah meningkatkan kemampuan sistem ASR dalam mengenali ucapan dengan lebih baik pada berbagai kondisi audio.

Penelitian ini menggunakan ASR sebagai tahap awal untuk menangkap perintah suara pemain. Hasil transkripsi dari proses ASR kemudian digunakan sebagai input bagi model NLU. Dengan demikian, keberhasilan ASR menjadi faktor penting karena kesalahan pada tahap ini dapat memengaruhi hasil pemahaman perintah pada tahap berikutnya.

2.4.2 Whisper dan Model STT Modern

Whisper merupakan salah satu model *Speech-to-Text* modern yang dikembangkan untuk melakukan transkripsi ucapan pada berbagai kondisi audio. Radford et al. (2023) menjelaskan bahwa Whisper dilatih menggunakan data audio berskala besar sehingga memiliki kemampuan generalisasi yang baik dalam tugas pengenalan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

suara. Kemampuan ini membuat Whisper relevan digunakan sebagai dasar pemrosesan suara dalam sistem interaktif.

Penelitian ini menggunakan Whisper untuk mengubah perintah suara pemain menjadi teks sebelum diproses oleh model NLU. Namun, penggunaan Whisper tetap memiliki batasan, terutama ketika berhadapan dengan istilah khusus dalam permainan, nama lokasi, nama karakter, atau kondisi audio yang kurang ideal. Oleh karena itu, hasil transkripsi tetap memerlukan pemrosesan lanjutan agar dapat digunakan secara tepat oleh sistem permainan.

2.4.3 Kesalahan Transkripsi dan Dampaknya terhadap Sistem

Kesalahan transkripsi dapat terjadi ketika sistem STT salah mengenali kata yang diucapkan pemain. Kesalahan tersebut dapat berupa salah ejaan, hilangnya sebagian kata, perubahan nama lokasi, atau perubahan nama target. Kesalahan kecil pada nama entitas dapat berdampak besar karena sistem mungkin gagal mengenali lokasi, objek, atau musuh yang dimaksud oleh pemain.

Sebagai contoh, perintah “go to Solitude” dapat gagal diproses apabila hasil transkripsi tidak sesuai dengan nama lokasi yang tersimpan dalam basis data permainan. Kesalahan seperti ini menunjukkan bahwa sistem interaksi suara tidak cukup hanya bergantung pada STT, tetapi juga memerlukan mekanisme pendukung seperti *Natural Language Understanding* dan *Fuzzy String Matching*. Dengan demikian, kesalahan transkripsi dapat dikurangi dampaknya sebelum perintah dieksekusi oleh NPC.

2.5 *Natural Language Understanding* dalam Pemrosesan Perintah Pemain

Natural Language Understanding (NLU) merupakan bagian dari pemrosesan bahasa alami yang berfokus pada pemahaman makna dari teks. NLU digunakan setelah proses *Speech-to-Text* untuk mengubah hasil transkripsi menjadi informasi yang lebih terstruktur. Informasi tersebut diperlukan agar sistem tidak hanya membaca teks, tetapi juga memahami maksud perintah pemain.

Penelitian ini menggunakan NLU untuk menghasilkan tiga keluaran utama, yaitu *Intent*, *entity*, dan *sentiment*. *Intent* digunakan untuk mengetahui maksud perintah, *entity* digunakan untuk mengenali objek atau target dalam permainan, sedangkan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sentiment digunakan untuk membaca kecenderungan emosi atau sikap pemain. Ketiga keluaran tersebut menjadi dasar bagi sistem dalam menentukan respons NPC.

Tabel 2. 4 Hubungan Tugas NLU dengan Pemrosesan Perintah Pemain

Tugas NLU	Fungsi Utama	Contoh Input Pemain	Output yang Diharapkan
<i>Intent Classification</i>	Mengidentifikasi maksud perintah pemain	“Serana, attack the dragon”	<i>Intent</i> : attack
<i>Named Entity Recognition</i>	Mengenali entitas target, objek, lokasi, atau karakter	“Go to Whiterun”	<i>Entity</i> : Whiterun
<i>Sentiment Analysis</i>	Mengenali kecenderungan emosional ucapan pemain	“Good job, Serana”	<i>Sentiment</i> : positive
Kombinasi NLU	Mengubah kalimat bebas menjadi data terstruktur	“Serana, attack the bandit in solitude”	<i>Intent</i> : attack, <i>Entity</i> : bandit, Location: solitude

Tabel 2.4 menunjukkan bahwa NLU berperan sebagai lapisan pemahaman yang menghubungkan teks hasil transkripsi dengan tindakan yang dapat diproses oleh sistem permainan. Tanpa NLU, sistem hanya memiliki teks, tetapi belum mengetahui tindakan apa yang harus dijalankan, objek apa yang dimaksud, dan konteks komunikasi seperti apa yang diberikan oleh pemain.

2.5.1 *Intent Classification*

Intent Classification merupakan proses untuk mengidentifikasi maksud utama dari suatu kalimat. *Intent* digunakan untuk menentukan tindakan apa yang diinginkan pengguna. (Perdana and Adikara, 2025) menjelaskan bahwa *Intent Classification*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dapat digunakan bersama tugas NLU lain untuk memahami maksud pengguna secara lebih terstruktur.

Penelitian ini menggunakan *Intent Classification* untuk mengenali jenis perintah pemain, seperti mengikuti, menunggu, menyerang, membuka objek, atau bergerak menuju lokasi tertentu. Sebagai contoh, ucapan “Serana, attack the bandit in Solitude” dapat diklasifikasikan sebagai perintah menyerang. Hasil klasifikasi ini kemudian digunakan oleh sistem permainan untuk menentukan aksi NPC yang sesuai.

2.5.2 Named Entity Recognition

Named Entity Recognition (NER) merupakan proses untuk mengenali entitas penting dalam teks, seperti nama lokasi, karakter, objek, atau target. NER diperlukan karena perintah pemain tidak hanya berisi tindakan, tetapi juga objek atau tujuan dari tindakan tersebut. (Annisa, Perdana and Adikara, 2024) menunjukkan bahwa NER dapat dikombinasikan dengan *Intent Classification* untuk membangun pemahaman bahasa yang lebih lengkap.

Penelitian ini menggunakan NER untuk mengenali entitas permainan yang muncul dalam perintah pemain. Misalnya, pada kalimat “go to Solitude”, sistem perlu mengenali “Solitude” sebagai entitas lokasi. Hasil NER membantu sistem menentukan objek, target, atau lokasi yang harus digunakan dalam proses eksekusi perintah NPC.

2.5.3 Sentiment Analysis

Sentiment Analysis merupakan proses untuk mengidentifikasi kecenderungan emosi, sikap, atau opini dalam suatu teks. Sentimen dapat digunakan untuk membaca apakah ucapan pemain bersifat positif, negatif, atau netral. (He, 2024) menjelaskan bahwa model berbasis Transformer seperti RoBERTa dapat digunakan untuk tugas klasifikasi sentimen karena mampu memahami konteks teks secara lebih baik.

Penelitian ini menggunakan *Sentiment Analysis* sebagai informasi tambahan untuk menentukan respons verbal NPC. Misalnya, ucapan apresiatif dari pemain dapat diarahkan pada respons yang lebih positif, sedangkan ucapan bernada negatif dapat

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menghasilkan respons yang berbeda. Dengan demikian, *Sentiment Analysis* tidak digunakan untuk menentukan aksi utama NPC, tetapi untuk memperkaya konteks respons verbal agar interaksi terasa lebih natural.

2.6 *Multi-Task Learning* pada Model *Natural Language Understanding*

Multi-Task Learning adalah pendekatan *Machine Learning* yang melatih satu model untuk menyelesaikan beberapa tugas sekaligus, seperti memproses *intent*, *entity*, dan *sentiment* dalam satu alur kerja. Pendekatan ini sangat relevan untuk sistem interaksi suara yang membutuhkan pemahaman utuh terhadap maksud, objek, dan konteks ucapan pemain. Dalam penelitian ini, *Multi-Task Learning* digunakan pada model NLU untuk menghasilkan keluaran terstruktur dari transkripsi suara secara ringkas tanpa memerlukan banyak model terpisah, sehingga menjadi dasar bagi sistem untuk menerjemahkan bahasa alami pemain menjadi data yang dapat digunakan oleh NPC.

Tabel 2. 5 Perbandingan Model Terpisah dan *Multi-Task Learning* dalam NLU

Aspek	Model Terpisah	<i>Multi-Task Learning</i>
Struktur model	Setiap tugas menggunakan model berbeda	Beberapa tugas diproses dalam satu model
Alur inferensi	Berurutan atau paralel dengan beberapa model	Satu kali inferensi menghasilkan beberapa keluaran
Kompleksitas sistem	Lebih tinggi karena banyak model perlu dikelola	Lebih ringkas karena arsitektur terintegrasi
Konsistensi representasi	Representasi tiap tugas dapat berbeda	Tugas berbagi representasi bahasa yang sama
Relevansi dalam penelitian	Kurang efisien untuk sistem respons cepat	Sesuai untuk menghasilkan <i>Intent</i> , <i>entity</i> , dan <i>sentiment</i> dalam satu siklus pemrosesan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2.5 menunjukkan bahwa *Multi-Task Learning* memiliki keunggulan dari sisi integrasi proses karena beberapa tugas NLU dapat dijalankan dalam satu model. Pendekatan tersebut relevan pada penelitian ini karena sistem perlu memproses perintah pemain secara cepat, tetapi tetap menghasilkan keluaran yang cukup lengkap untuk menentukan tindakan NPC.

2.6.1 Konsep *Multi-Task Learning*

Multi-Task Learning adalah pendekatan pembelajaran yang memanfaatkan representasi bersama untuk menyelesaikan lebih dari satu tugas. Model dilatih agar dapat mempelajari pola umum dari data, kemudian menggunakan pola tersebut untuk menghasilkan keluaran yang berbeda sesuai kebutuhan tugas. (Perdana and Adikara, 2025) menjelaskan bahwa pendekatan multi-task dapat digunakan dalam pemrosesan bahasa alami untuk menangani *Intent Classification* dan *Named Entity Recognition* secara terintegrasi.

Penelitian ini menggunakan konsep tersebut karena perintah pemain memiliki beberapa unsur yang perlu dipahami secara bersamaan. Misalnya, sistem perlu mengetahui jenis perintah, target atau lokasi yang disebutkan, serta konteks emosional dari ucapan pemain. Dengan *Multi-Task Learning*, proses pemahaman bahasa dapat dilakukan dalam satu alur yang lebih terpadu.

2.6.2 *Multi-Task Learning* untuk *Intent*, *Entity*, dan *Sentiment*

Sistem interaksi suara, *Intent*, *entity*, dan *sentiment* memiliki peran yang saling melengkapi. *Intent* menentukan tindakan utama yang diinginkan pemain, *entity* menunjukkan objek atau lokasi yang menjadi tujuan perintah, sedangkan *sentiment* memberikan informasi tambahan mengenai kecenderungan emosi atau sikap pemain. Ketiganya diperlukan agar sistem dapat memahami perintah secara lebih utuh.

(Annisa, Perdana and Adikara, 2024) menunjukkan bahwa penggabungan *Intent Classification* dan NER dapat digunakan untuk membangun sistem NLU yang lebih terstruktur. Pendekatan tersebut diperluas dengan menambahkan *Sentiment Analysis* sebagai keluaran tambahan. Sebagai contoh, perintah “Serana, attack the bandit in Solitude” dapat menghasilkan *Intent* menyerang, *entity* berupa target atau lokasi, dan *sentiment* yang menunjukkan nada ucapan pemain.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.6.3 Keunggulan dan Keterbatasan *Multi-Task Learning*

Keunggulan utama *Multi-Task Learning* adalah kemampuannya menyatukan beberapa tugas dalam satu model, sehingga alur pemrosesan menjadi lebih sederhana dibandingkan menggunakan model terpisah untuk setiap tugas. Pendekatan ini juga dapat membantu model memanfaatkan representasi bahasa yang sama untuk memahami maksud, entitas, dan konteks ucapan. Hal ini penting karena hasil pemrosesan harus dikirim kembali ke sistem permainan secara cepat dan terstruktur.

Namun, *Multi-Task Learning* juga memiliki keterbatasan. Setiap tugas dapat memiliki karakteristik data dan tingkat kesulitan yang berbeda, sehingga proses pelatihan perlu menjaga keseimbangan antar tugas. Pendekatan multi-task dibatasi pada keluaran yang relevan dengan interaksi NPC, yaitu *Intent*, *entity*, dan *sentiment*. Dengan batasan tersebut, model tetap diarahkan pada kebutuhan sistem permainan dan tidak diperluas menjadi sistem percakapan bebas.

2.7 Transformer dan RoBERTa dalam Pemahaman Bahasa Alami

Transformer merupakan arsitektur model bahasa yang banyak digunakan dalam pemrosesan bahasa alami modern. Arsitektur ini mampu memahami hubungan antar kata dalam sebuah kalimat melalui mekanisme attention, sehingga model dapat menangkap konteks teks secara lebih baik dibandingkan pendekatan berbasis urutan konvensional. Kemampuan memahami konteks penting karena perintah pemain dapat memiliki bentuk kalimat yang bervariasi.

Penelitian ini menggunakan arsitektur Transformer sebagai dasar pemahaman bahasa alami melalui model RoBERTa. RoBERTa dipilih karena mampu digunakan untuk tugas klasifikasi teks dan ekstraksi informasi, seperti *Intent Classification*, *Named Entity Recognition*, dan *Sentiment Analysis*. Dengan demikian, pembahasan Transformer dan RoBERTa diperlukan untuk menjelaskan dasar model bahasa yang digunakan dalam sistem NLU. Tabel 2.6 menunjukkan bahwa Transformer dan RoBERTa berperan sebagai fondasi model bahasa dalam sistem NLU. Pada penelitian ini, teks hasil transkripsi suara tidak diproses hanya berdasarkan pencocokan kata, tetapi dianalisis menggunakan model bahasa yang

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mampu memahami konteks kalimat untuk menghasilkan keluaran berupa *Intent*, *entity*, dan *sentiment*.

Tabel 2. 6 Peran Transformer dan RoBERTa dalam Sistem *Natural Language Understanding*

Komponen	Fungsi Umum	Peran dalam Penelitian
<i>Transformer Encoder</i>	Mempelajari representasi teks secara kontekstual	Menjadi dasar pemrosesan teks hasil transkripsi suara
<i>Fine-Tuning</i>	Menyesuaikan model bahasa dengan dataset dan tugas tertentu	Digunakan untuk melatih model pada label <i>Intent</i> , <i>entity</i> , dan <i>sentiment</i>
RoBERTa	Model bahasa berbasis <i>Transformer</i> untuk pemahaman teks	Digunakan sebagai model NLU untuk menghasilkan keluaran terstruktur
Klasifikasi Teks	Mengelompokkan teks ke dalam label tertentu	Digunakan pada <i>Intent Classification</i> dan <i>Sentiment Analysis</i>
Ekstraksi Informasi	Mengambil informasi penting dari teks	Digunakan pada <i>Named Entity Recognition</i>

2.7.1 Transformer Encoder untuk Pemrosesan Bahasa

Transformer Encoder merupakan bagian dari arsitektur Transformer yang bertugas membentuk representasi teks berdasarkan konteks kata dalam kalimat. Melalui mekanisme *self-attention*, model dapat memperhatikan hubungan antar kata tanpa harus memproses kalimat secara berurutan. Pendekatan ini memungkinkan model memahami makna kata berdasarkan konteks penggunaannya.

(Yulianti *et al.*, 2024) menunjukkan bahwa model berbasis Transformer dapat digunakan dalam tugas ekstraksi entitas karena mampu menangkap konteks teks secara lebih baik. Kemampuan tersebut diperlukan agar sistem dapat memahami perintah pemain yang mengandung tindakan, target, dan lokasi. Misalnya, pada

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

perintah “go to Solitude”, model perlu memahami bahwa “Solitude” merujuk pada entitas lokasi dalam permainan.

2.7.2 *Fine-Tuning* Model Bahasa

Fine-Tuning merupakan proses pelatihan lanjutan terhadap model bahasa yang telah dilatih sebelumnya agar dapat menyesuaikan diri dengan tugas atau domain tertentu. Model pra-latih umumnya telah memiliki pemahaman bahasa umum, tetapi tetap perlu disesuaikan ketika digunakan pada konteks khusus seperti perintah dalam permainan. Proses ini penting karena sistem perlu mengenali istilah, pola kalimat, dan entitas yang spesifik terhadap domain *game*.

Fine-Tuning dilakukan agar model dapat mengenali pola perintah pemain dan menghasilkan keluaran yang sesuai dengan kebutuhan sistem. Model tidak hanya diarahkan untuk memahami teks secara umum, tetapi juga untuk menghasilkan *Intent*, *entity*, dan *sentiment*. Dengan demikian, *Fine-Tuning* menjadi tahap penting agar model RoBERTa dapat digunakan dalam sistem interaksi suara NPC.

2.7.3 RoBERTa untuk Tugas Klasifikasi dan Ekstraksi Informasi

RoBERTa merupakan model bahasa berbasis Transformer yang dikembangkan sebagai pengembangan dari BERT. Model ini dapat digunakan untuk berbagai tugas pemrosesan bahasa alami, termasuk klasifikasi teks dan ekstraksi informasi. (Ouaddi *et al.*, 2025) menunjukkan bahwa RoBERTa dapat digunakan untuk tugas *Intent Classification*, sedangkan He *et al.* (2024) menunjukkan penerapan RoBERTa dalam tugas *Sentiment Analysis*.

Penelitian ini menggunakan RoBERTa sebagai dasar model NLU untuk menghasilkan tiga keluaran utama, yaitu *Intent*, *entity*, dan *sentiment*. *Intent* digunakan untuk menentukan tindakan NPC, *entity* digunakan untuk mengenali target atau lokasi, sedangkan *sentiment* digunakan sebagai konteks tambahan untuk respons verbal. Dengan kemampuan tersebut, RoBERTa berperan sebagai komponen utama dalam pemahaman perintah pemain sebelum hasilnya dikirim ke sistem permainan.

Hak Cipta :

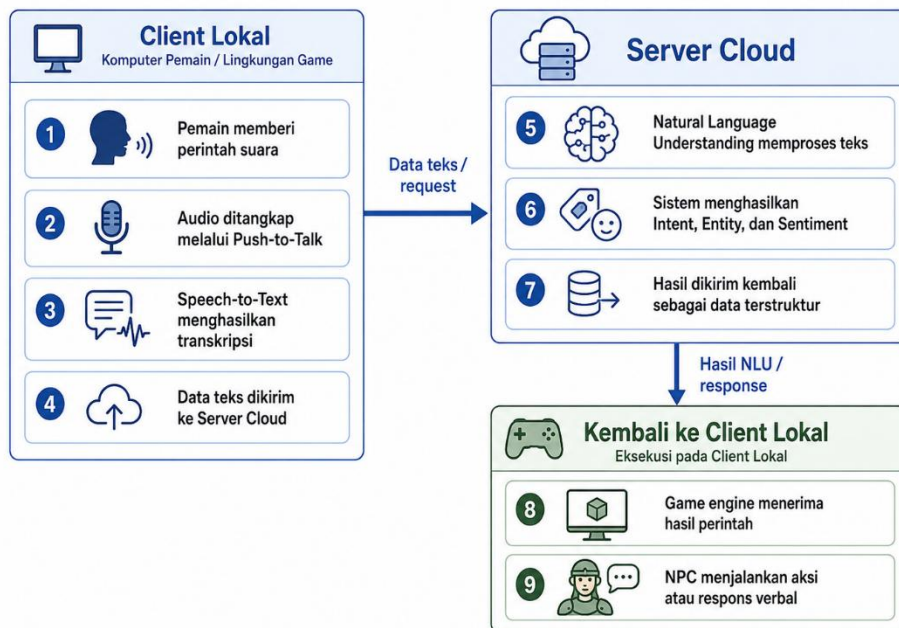
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.8 Arsitektur Hybrid Client-Cloud dan Computation offloading

Arsitektur *Hybrid Client-Cloud* merupakan pendekatan komputasi yang membagi proses kerja antara perangkat lokal pengguna sebagai *Client* dan infrastruktur jarak jauh sebagai *Server Cloud*. Pendekatan ini digunakan ketika suatu sistem membutuhkan keseimbangan antara respons cepat pada perangkat lokal dan kemampuan pemrosesan yang lebih besar pada *Server*. Arsitektur tersebut digunakan karena sistem interaksi suara membutuhkan pemrosesan permainan di sisi klien dan pemrosesan bahasa alami di sisi *Server*.

Sub-bab ini membahas konsep *Client*, *Server*, dan *cloud*, kemudian menjelaskan *computation offloading* serta pembagian beban kerja pada sistem interaksi suara. Pembahasan ini penting karena sistem yang dikembangkan tidak sepenuhnya berjalan secara lokal, tetapi memisahkan proses berdasarkan karakteristik beban komputasi dan kebutuhan respons dalam permainan.

Arsitektur Umum Hybrid Client-Cloud pada Sistem Interaksi Suara



Gambar 2. 2 Arsitektur Umum *Hybrid Client-Cloud* pada Sistem Interaksi Suara

Gambar 2.2 menunjukkan bahwa proses pemrosesan tidak seluruhnya dilakukan pada perangkat lokal. Komputer pemain tetap menjalankan permainan dan proses awal seperti penangkapan suara serta transkripsi, sedangkan proses pemahaman



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

bahasa alami yang lebih berat dialihkan ke *Server Cloud*. Dengan pembagian tersebut, sistem diharapkan dapat menjaga stabilitas permainan sambil tetap menyediakan kemampuan pemrosesan bahasa alami yang lebih kompleks.

2.8.1 Konsep *Client*, *Server*, dan *Cloud*

Client merupakan perangkat atau aplikasi yang digunakan pengguna untuk menjalankan sistem dan mengirimkan permintaan pemrosesan. *Client* merujuk pada komputer pemain yang menjalankan permainan, menangkap suara, melakukan transkripsi awal, serta menerima hasil pemrosesan dari *Server*. Sementara itu, *Server* merupakan sistem yang menyediakan layanan pemrosesan atau pengelolaan data bagi *Client*.

Cloud computing memungkinkan proses tertentu dijalankan pada infrastruktur jarak jauh yang memiliki sumber daya lebih fleksibel dibandingkan perangkat lokal. (Andriulo *et al.*, 2024) menjelaskan bahwa kombinasi *Edge computing* dan *cloud computing* dapat digunakan untuk membagi proses komputasi berdasarkan kebutuhan latensi dan kapasitas pemrosesan. Penelitian ini menggunakan konsep tersebut untuk memisahkan proses yang dekat dengan permainan dari proses AI yang lebih berat.

2.8.2 *Computation offloading*

Computation offloading merupakan strategi memindahkan sebagian tugas komputasi dari perangkat lokal ke sistem lain yang memiliki sumber daya lebih besar. Strategi ini digunakan untuk mengurangi beban perangkat lokal, terutama ketika sistem harus menjalankan proses yang berat. (Zhang, Yi and Ma, 2024) menjelaskan bahwa *computation offloading* dapat dilakukan dengan mempertimbangkan jenis tugas, kebutuhan sumber daya, dan kondisi sistem.

Computation offloading diterapkan dengan memindahkan proses *Natural Language Understanding* ke *Server Cloud*. Komputer lokal tetap menjalankan permainan, menangkap suara, menjalankan *Speech-to-Text*, dan mengeksekusi respons NPC. Dengan pembagian tersebut, proses pemahaman bahasa alami tidak membebani perangkat lokal secara berlebihan, sehingga permainan tetap dapat berjalan lebih stabil.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.8.3 Pembagian Beban Kerja pada Sistem Interaksi Suara

Pembagian beban kerja dilakukan agar setiap komponen sistem menjalankan proses yang paling sesuai dengan karakteristiknya. Proses yang berkaitan langsung dengan pemain dan *game engine* dijalankan pada sisi klien, sedangkan proses yang membutuhkan pemrosesan model bahasa dijalankan pada *Server Cloud*. Pendekatan ini membantu menjaga stabilitas permainan sekaligus tetap memungkinkan sistem menggunakan kemampuan AI.

Sisi klien menangani input suara, transkripsi, komunikasi dengan *game engine*, dan eksekusi respons NPC. Sementara itu, *Server Cloud* menangani proses NLU untuk menghasilkan *Intent*, *entity*, dan *sentiment*. Hasil pemrosesan tersebut kemudian dikirim kembali ke klien dalam bentuk data terstruktur agar dapat digunakan oleh sistem permainan. Dengan arsitektur ini, sistem interaksi suara dapat bekerja secara lebih terpisah, terarah, dan mudah diuji.

2.9 Fuzzy String Matching sebagai Mekanisme Toleransi Kesalahan

Fuzzy String Matching merupakan teknik pencocokan teks yang digunakan untuk menemukan kemiripan antara dua string meskipun tidak identik secara karakter. Berbeda dengan *exact matching* yang hanya menerima kecocokan sempurna, *Fuzzy String Matching* memungkinkan sistem mengenali teks yang memiliki kesalahan kecil, seperti huruf hilang, salah ejaan, atau variasi penulisan.

Penelitian ini menggunakan *Fuzzy String Matching* sebagai mekanisme toleransi kesalahan terhadap hasil transkripsi *Speech-to-Text*. Mekanisme ini penting karena nama lokasi, objek, atau target dalam permainan dapat gagal dikenali apabila hasil transkripsi tidak sama persis dengan data referensi. Dengan pencocokan berbasis kemiripan, sistem tetap dapat mengenali entitas permainan selama tingkat kemiripannya memenuhi ambang batas tertentu. Tabel 2.7 menunjukkan bahwa kesalahan kecil pada hasil transkripsi dapat menyebabkan entitas tidak dikenali apabila sistem hanya menggunakan pencocokan eksak. Dengan *Fuzzy String Matching*, teks yang tidak sempurna masih dapat dibandingkan dengan basis data entitas permainan untuk menemukan kandidat yang paling mirip.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2. 7 Contoh Penerapan *Fuzzy String Matching* pada Entitas Permainan

Hasil Transkripsi	Entitas Referensi	Jenis Kesalahan	Hasil yang Diharapkan
“whitrn”	Whiterun	Huruf hilang	Sistem tetap mengenali lokasi Whiterun
“dragn”	Dragon	Kesalahan ejaan	Sistem tetap mengenali target Dragon
“banded”	Bandit	Bunyi kata mirip	Sistem mengarahkan ke entitas Bandit jika skor kemiripan memenuhi batas
“river wood”	Riverwood	Spasi tambahan	Sistem tetap mengenali lokasi Riverwood
“levr”	Lever	Huruf hilang	Sistem tetap mengenali objek Lever

2.9.1 *Exact matching* dan *Approximate Matching*

Exact matching merupakan metode pencocokan string yang hanya menghasilkan kecocokan apabila teks masukan sama persis dengan data referensi. Metode ini sederhana dan cepat, tetapi kurang fleksibel ketika digunakan pada sistem berbasis suara karena hasil transkripsi dapat mengandung kesalahan. Misalnya, lokasi “Solitude” dapat gagal dikenali apabila hasil transkripsi mengalami perubahan bentuk kata atau ejaan.

Approximate Matching digunakan untuk mengatasi keterbatasan tersebut dengan menghitung tingkat kemiripan antara teks masukan dan data referensi. (Bosker, 2021) menunjukkan bahwa *Fuzzy String Matching* dapat digunakan untuk mencocokkan hasil transkripsi dengan bentuk referensi ketika terdapat variasi atau kesalahan pada teks. Penelitian ini menggunakan pendekatan tersebut untuk membantu sistem mengenali entitas permainan yang tidak tertulis secara sempurna pada hasil transkripsi.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.9.2 *Levenshtein Distance* dan *Similarity Ratio*

Levenshtein Distance merupakan metode yang mengukur jarak antara dua string berdasarkan jumlah operasi minimum yang diperlukan untuk mengubah satu string menjadi string lainnya. Operasi tersebut dapat berupa penambahan, penghapusan, atau penggantian karakter. Semakin kecil jarak antara dua string, semakin tinggi tingkat kemiripannya.

Similarity Ratio digunakan untuk menyatakan tingkat kemiripan dalam bentuk skor. Skor ini dapat digunakan untuk menentukan apakah hasil transkripsi cukup mirip dengan salah satu entitas referensi. (Jiang and Cai, 2024) menjelaskan bahwa teknik text matching digunakan untuk mengukur kesamaan atau keterkaitan antara dua teks. Penelitian ini menggunakan konsep tersebut untuk membandingkan hasil transkripsi dengan basis data entitas permainan sebelum sistem menentukan hasil pencocokan yang paling sesuai.

2.9.3 Penerapan *Fuzzy Matching* pada Entitas Permainan

Entitas permainan dapat berupa nama lokasi, karakter, musuh, atau objek interaktif. Entitas tersebut perlu dikenali dengan tepat karena menjadi bagian penting dari perintah pemain. Apabila pemain mengucapkan “go to Solitude”, sistem perlu mengenali “Solitude” sebagai lokasi agar perintah dapat diproses dengan benar.

Fuzzy Matching diterapkan dengan membandingkan entitas hasil transkripsi atau hasil NLU dengan daftar entitas valid dalam permainan. Jika terdapat kandidat yang memiliki skor kemiripan di atas ambang batas, sistem dapat memilih entitas tersebut sebagai hasil koreksi. Namun, mekanisme ini tetap memiliki batasan karena pencocokan yang terlalu longgar dapat menyebabkan sistem memilih entitas yang salah. Oleh karena itu, penggunaan *Fuzzy String Matching* perlu didukung oleh ambang batas kemiripan dan validasi konteks agar hasilnya tetap akurat.

2.10 Interoperabilitas Sistem AI dengan *Game engine*

Interoperabilitas merupakan kemampuan dua atau lebih sistem yang berbeda untuk saling bertukar data dan bekerja secara terkoordinasi. Interoperabilitas diperlukan karena sistem kecerdasan buatan yang memproses bahasa alami tidak berjalan secara langsung di dalam *game engine*, melainkan berada pada komponen terpisah yang berkomunikasi dengan permainan melalui data terstruktur. Tanpa mekanisme

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

interoperabilitas, hasil pemrosesan AI seperti *Intent*, *entity*, dan *sentiment* tidak dapat diterjemahkan menjadi tindakan NPC di dalam lingkungan permainan.

Sub-bab ini membahas interoperabilitas sistem AI dengan *game engine* melalui tiga bagian utama. Pertama, dijelaskan konsep *game engine* sebagai lingkungan eksekusi permainan yang mengatur grafis, audio, logika, skrip, aset, dan interaksi pemain. Kedua, dibahas mekanisme pertukaran data antara sistem eksternal dan *game engine* sebagai dasar integrasi AI. Ketiga, dijelaskan peran Papyrus, SKSE64, dan PapyrusUtil dalam konteks modding Skyrim sebagai mekanisme teknis yang memungkinkan sistem eksternal berkomunikasi dengan logika internal permainan.

Tabel 2. 8 Komponen Interoperabilitas Sistem AI dengan *Game engine*

Komponen	Fungsi Umum	Peran dalam Penelitian
Sistem AI Eksternal	Memproses teks dan menghasilkan hasil pemahaman bahasa	Menghasilkan <i>Intent</i> , <i>entity</i> , dan <i>sentiment</i> dari perintah pemain
<i>Game engine</i>	Menjalankan lingkungan permainan, aset, audio, animasi, dan logika interaksi	Menjadi tempat NPC Serana menerima dan menjalankan perintah
Data Terstruktur	Menyimpan hasil pemrosesan dalam format yang dapat dibaca sistem lain	Menjadi media pertukaran antara AI dan sistem permainan
Skrip Lokal	Menjalankan logika pada sisi permainan	Membaca hasil AI dan menerjemahkannya menjadi aksi NPC
Middleware/Utilitas Modding	Menyediakan fungsi tambahan yang tidak tersedia secara bawaan	Membantu pertukaran data antara sistem eksternal dan skrip permainan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2.8 menunjukkan bahwa integrasi AI dengan *game engine* tidak hanya bergantung pada model kecerdasan buatan, tetapi juga memerlukan mekanisme penghubung yang memungkinkan hasil pemrosesan AI diterjemahkan menjadi tindakan dalam permainan. Dengan demikian, interoperabilitas menjadi lapisan penting yang menjembatani sistem AI dan lingkungan permainan.

2.10.1 *Game engine* sebagai Lingkungan Eksekusi Permainan

Game engine merupakan perangkat lunak inti yang digunakan untuk menjalankan permainan digital. Di dalamnya terdapat berbagai subsistem yang mengatur grafis, audio, fisika, skrip, animasi, antarmuka pengguna, serta logika permainan. Dengan kata lain, *game engine* berperan sebagai lingkungan eksekusi utama yang mengatur bagaimana dunia permainan bekerja dan bagaimana pemain dapat berinteraksi dengan elemen-elemen yang ada di dalamnya.

Game engine tidak hanya menampilkan visual kepada pemain, tetapi juga mengelola aturan dan respons sistem. Ketika pemain menekan tombol, memilih menu, menyerang musuh, atau berinteraksi dengan NPC, semua tindakan tersebut diproses melalui mekanisme yang tersedia di dalam *game engine*. Oleh karena itu, setiap sistem eksternal yang ingin memengaruhi perilaku NPC perlu memiliki jalur komunikasi yang dapat diterima oleh lingkungan eksekusi permainan.

(Gallotta *et al.*, 2024) menjelaskan bahwa penerapan AI dalam permainan tidak hanya berkaitan dengan kemampuan model menghasilkan respons, tetapi juga berkaitan dengan bagaimana model tersebut dapat diintegrasikan ke dalam lingkungan permainan. Hal ini menunjukkan bahwa tantangan AI dalam *game* tidak berhenti pada akurasi model, melainkan juga mencakup cara menghubungkan keluaran AI dengan mekanisme permainan yang berjalan secara aktual.

Game engine diposisikan sebagai lingkungan eksekusi yang menerima hasil pemrosesan AI dalam bentuk data terstruktur. Sistem AI tidak secara langsung mengendalikan NPC, tetapi mengirimkan informasi yang kemudian dibaca oleh skrip lokal di sisi permainan. Dengan pendekatan tersebut, proses pengambilan keputusan berbasis AI dapat dipisahkan dari proses eksekusi tindakan di dalam permainan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.10.2 Pertukaran Data antara Sistem Eksternal dan *Game engine*

Sistem AI eksternal tidak dapat secara otomatis menjalankan tindakan di dalam permainan tanpa mekanisme pertukaran data. Hal ini terjadi karena sistem AI dan *game engine* berada pada lingkungan eksekusi yang berbeda. Sistem AI memproses input dalam bentuk teks dan menghasilkan keluaran terstruktur, sedangkan *game engine* membutuhkan instruksi yang sesuai dengan logika internal permainan agar NPC dapat bergerak, menyerang, menunggu, mengikuti pemain, atau memberikan respons verbal.

Pertukaran data menjadi mekanisme yang memungkinkan dua sistem berbeda saling memahami. Data yang dikirimkan harus memiliki format yang jelas, konsisten, dan dapat dibaca oleh kedua sisi. Hasil pemrosesan NLU seperti *Intent*, *entity*, dan *sentiment* perlu dikemas dalam format terstruktur agar dapat digunakan oleh sistem permainan. Format terstruktur tersebut berperan sebagai jembatan antara makna bahasa alami dan perintah teknis yang dapat dieksekusi oleh NPC.

(Hochreiter, Kriglstein and Wallner, 2026) menunjukkan bahwa kualitas dialog NPC modern sangat berkaitan dengan persepsi pemain terhadap naturalitas dan keterlibatan karakter. Namun, agar dialog atau respons yang lebih natural dapat diwujudkan dalam permainan, keluaran AI harus dapat dihubungkan dengan sistem internal *game*. Dengan demikian, integrasi AI tidak cukup hanya menghasilkan teks atau label, tetapi harus mampu mengalirkan hasil pemrosesan tersebut ke lingkungan permainan yang benar-benar menjalankan aksi karakter.

Pertukaran data dilakukan secara asinkron agar sistem AI dan *game engine* dapat bekerja tanpa harus saling mengunci proses eksekusi. Pendekatan asinkron penting karena permainan tetap harus berjalan meskipun proses AI membutuhkan waktu pemrosesan atau komunikasi jaringan. Dengan mekanisme ini, *game engine* dapat membaca hasil AI ketika data telah tersedia, kemudian menerjemahkannya menjadi aksi NPC sesuai dengan struktur logika yang telah dirancang.

2.10.3 Papyrus, SKSE64, dan PapyrusUtil dalam Modding Skyrim

Pada The Elder Scrolls V: Skyrim Special Edition, logika permainan dan perilaku mod banyak dikendalikan melalui Papyrus, yaitu bahasa skrip yang digunakan untuk mengatur interaksi objek, karakter, quest, dan berbagai peristiwa di dalam



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

permainan. Papyrus memungkinkan pengembang mod menambahkan atau mengubah perilaku tertentu tanpa harus membangun ulang *game engine* secara langsung. Penelitian ini menggunakan Papyrus sebagai lapisan skrip lokal yang menerjemahkan hasil pemrosesan AI menjadi aksi yang dapat dijalankan oleh NPC.

Meskipun Papyrus menyediakan mekanisme scripting, kemampuannya tetap memiliki batasan karena berjalan dalam lingkungan yang terisolasi dari sistem eksternal. Skrip Papyrus tidak dirancang secara bawaan untuk menjalankan model AI modern atau melakukan komunikasi kompleks dengan sistem *cloud*. Oleh karena itu, diperlukan komponen tambahan agar skrip dalam permainan dapat membaca data yang dihasilkan oleh sistem AI eksternal.

SKSE64 atau Skyrim Script Extender merupakan ekstensi yang memperluas kemampuan scripting pada Skyrim. SKSE64 menyediakan akses ke fungsi tambahan yang tidak tersedia pada Papyrus bawaan. Sementara itu, PapyrusUtil merupakan utilitas yang menyediakan kemampuan untuk membaca dan menulis data terstruktur, termasuk data berbasis JSON. Dengan bantuan PapyrusUtil, skrip Papyrus dapat membaca hasil pemrosesan AI yang disimpan dalam format terstruktur dan menggunakannya sebagai dasar pengambilan keputusan NPC.

Penelitian ini menggunakan Papyrus, SKSE64, dan PapyrusUtil sebagai bagian dari mekanisme interoperabilitas antara sistem AI eksternal dan *game engine*. Sistem AI menghasilkan keluaran berupa *Intent*, *entity*, dan *sentiment*, kemudian hasil tersebut dikemas dalam data terstruktur yang dapat dibaca oleh skrip lokal. Skrip Papyrus selanjutnya menerjemahkan keluaran tersebut menjadi tindakan NPC, seperti mengikuti pemain, menyerang target, bergerak menuju lokasi tertentu, atau memutar respons verbal yang sesuai.

Pendekatan ini dipilih karena bersifat non-invasif dan sesuai dengan pola kerja modding Skyrim. Sistem AI tidak mengubah memori permainan secara langsung, melainkan berkomunikasi melalui data yang dibaca oleh skrip. Dengan demikian, interoperabilitas dapat dicapai tanpa memaksa model AI berjalan sepenuhnya di dalam *game engine*. Pembahasan teknis mengenai implementasi mekanisme ini dijelaskan lebih rinci pada BAB III dan BAB IV.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.11 Voice Cloning, Baked Audio, dan Sinkronisasi Gerakan Bibir

Respons suara sangat penting dalam interaksi NPC untuk membentuk identitas karakter dan meningkatkan imersi pemain melalui kecocokan suara, kecepatan putar, serta keselarasan visual. Oleh karena itu, penelitian ini menerapkan konsep *Voice Cloning*, *Baked Audio*, dan sinkronisasi gerakan bibir sebagai dasar respons verbal NPC. Pembahasan dalam sub-bab ini mencakup tiga aspek utama, yaitu perbedaan antara *Text-to-Speech*, *Voice Conversion*, dan *Voice Cloning*, penerapan *Baked Audio* sebagai strategi untuk mengurangi latensi, serta sinkronisasi audio dengan gerakan bibir agar interaksi tampak lebih natural di dalam gim.

Tabel 2. 9 Perbandingan Teknologi Respons Suara dalam Sistem NPC

Konsep	Pengertian Umum	Peran dalam Penelitian
<i>Text-to-Speech</i>	Teknologi yang mengubah teks menjadi suara sintetis.	Menjadi konsep dasar pembangkitan suara, tetapi tidak digunakan sebagai proses <i>real-time</i> dalam sistem final.
<i>Voice conversion</i>	Teknologi yang mengubah karakteristik suara sumber menjadi menyerupai suara target.	Digunakan sebagai dasar konseptual dalam proses pembentukan suara karakter.
<i>Voice Cloning</i>	Pendekatan untuk menghasilkan atau meniru karakteristik suara tertentu.	Digunakan untuk membentuk respons verbal yang selaras dengan identitas suara NPC.
<i>Baked Audio</i>	Aset audio yang telah diproduksi sebelum permainan dijalankan.	Digunakan untuk mengurangi latensi dan menghindari pemrosesan suara <i>real-time</i> .
<i>Lip-sync</i>	Sinkronisasi antara suara dan gerakan bibir karakter.	Digunakan agar respons verbal NPC tampak lebih natural secara visual.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2.9 menunjukkan bahwa respons suara dalam sistem NPC tidak hanya berkaitan dengan pembangkitan audio, tetapi juga dengan strategi penyimpanan aset dan penyelarasan visual. Pendekatan *Baked Audio* dipilih agar sistem tidak perlu menghasilkan suara baru secara *real-time* ketika permainan sedang berjalan.

2.11.1 Text-to-Speech, Voice conversion, dan Voice Cloning

Text-to-Speech (TTS) merupakan teknologi yang mengubah teks menjadi suara sintesis. Barakat et al. (2024) menjelaskan bahwa perkembangan *speech synthesis* berbasis *Deep Learning* telah meningkatkan kemampuan sistem dalam menghasilkan suara yang lebih natural dan ekspresif. Namun, dalam lingkungan permainan, proses sintesis suara secara langsung dapat menambah beban komputasi dan meningkatkan latensi respons.

Selain TTS, terdapat *Voice conversion*, yaitu proses mengubah karakteristik suara sumber agar menyerupai suara target. (Bargum, Serafin and Erkut, 2024) menjelaskan bahwa *Voice conversion* dapat digunakan untuk memodifikasi representasi suara sehingga karakteristik vokal tertentu dapat dipertahankan atau ditiru. Penelitian ini menggunakan *Voice Cloning* pada tahap pra-produksi untuk menghasilkan respons suara NPC yang sesuai dengan identitas karakter, bukan sebagai proses generatif *real-time* saat permainan berjalan.

2.11.2 Baked Audio sebagai Strategi Reduksi Latensi

Baked Audio merupakan aset suara yang telah diproduksi dan disimpan sebelum sistem dijalankan. Berbeda dengan TTS *real-time*, pendekatan ini tidak menghasilkan suara baru saat permainan berlangsung, tetapi memanggil file audio yang sudah tersedia sesuai konteks respons. Strategi ini digunakan agar sistem dapat memutar respons NPC secara lebih cepat dan stabil.

Baked Audio dipilih karena sistem tidak dirancang sebagai percakapan bebas generatif, melainkan sebagai sistem perintah suara dengan respons verbal yang telah disiapkan. Setelah sistem NLU menghasilkan *Intent*, *entity*, dan *sentiment*, sistem permainan dapat memilih audio yang sesuai dari kumpulan aset. Dengan demikian, proses respons suara tidak membebani perangkat lokal dengan sintesis suara tambahan saat permainan sedang berjalan.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.11.3 Sinkronisasi Audio dan *Lip-sync* pada Karakter NPC

Sinkronisasi gerakan bibir atau *lip-sync* merupakan proses penyesuaian antara audio ucapan dan gerakan mulut karakter. *Lip-sync* penting karena suara yang tidak selaras dengan gerakan bibir dapat mengurangi kesan natural pada NPC. Sebaliknya, ketika audio dan gerakan bibir selaras, respons verbal karakter akan terasa lebih hidup dan meyakinkan.

(Jiang and Cai, 2024) menjelaskan bahwa audio-driven facial animation berkaitan dengan pembentukan gerakan wajah berdasarkan sinyal audio. (Kannoja *et al.*, 2025) juga menunjukkan bahwa keselarasan audio dan visual merupakan bagian penting dalam sistem *Voice Cloning* dan dubbing berbasis AI. Sinkronisasi gerakan bibir dilakukan dengan menyiapkan data pendukung bersama aset audio, sehingga respons NPC tidak hanya terdengar sesuai, tetapi juga tampak selaras secara visual di dalam permainan.

2.12 Metode Pengujian Sistem

Metode pengujian sistem digunakan untuk memastikan bahwa sistem yang dikembangkan berjalan sesuai kebutuhan, dapat diterima oleh pengguna, serta memiliki performa yang layak digunakan. Pengujian tidak hanya berfokus pada keberhasilan fungsi teknis, tetapi juga pada pengalaman pengguna dan respons sistem ketika digunakan dalam lingkungan permainan.

Sub-bab ini membahas empat metode pengujian yang digunakan, yaitu *Black Box Testing*, *User Acceptance Testing* (UAT), *System Usability Scale* (SUS), serta *profiling* latensi dan performa. Keempat metode tersebut digunakan agar evaluasi sistem mencakup aspek fungsional, penerimaan pengguna, kegunaan sistem, dan kinerja teknis. Tabel 2.10 menunjukkan bahwa pengujian sistem dalam penelitian ini tidak hanya berfokus pada keberhasilan fungsi teknis, tetapi juga mencakup aspek penerimaan pengguna, kemudahan penggunaan, dan performa sistem. Hal ini diperlukan karena sistem interaksi suara harus mampu bekerja secara fungsional sekaligus tetap nyaman digunakan dalam konteks permainan *real-time*.

Tabel 2. 10 Ringkasan Metode Pengujian Sistem

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Metode Pengujian	Fokus Pengujian	Peran dalam Penelitian
<i>Black Box Testing</i>	Menguji fungsi sistem berdasarkan input dan output	Memastikan setiap skenario perintah suara menghasilkan respons NPC yang sesuai
<i>User Acceptance Testing</i>	Mengukur penerimaan pengguna terhadap sistem	Menilai apakah sistem dapat diterima oleh pengguna akhir
<i>System Usability Scale</i>	Mengukur tingkat kegunaan sistem secara subjektif	Menilai kemudahan penggunaan dan kenyamanan interaksi pengguna
<i>Profiling Latensi dan Performa</i>	Mengukur waktu respons dan penggunaan sumber daya	Mengevaluasi apakah sistem tetap responsif dan stabil saat dijalankan

2.12.1 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada hubungan antara input dan output sistem tanpa melihat struktur kode internal. Dalam metode ini, penguji memberikan masukan tertentu, kemudian membandingkan keluaran aktual dengan keluaran yang diharapkan. (Maulana, Susanto and Christianto, 2024) menjelaskan bahwa *Black Box Testing* dapat digunakan untuk menguji fungsi sistem dari sisi input dan output berdasarkan skenario yang telah dirancang.

Penelitian ini menggunakan *Black Box Testing* untuk menguji apakah perintah suara pemain dapat diproses menjadi tindakan NPC yang sesuai. Misalnya, perintah untuk mengikuti, menunggu, menyerang, atau berpindah ke lokasi tertentu harus menghasilkan respons yang sesuai dengan skenario pengujian. Dengan demikian, metode ini digunakan untuk memastikan bahwa alur input suara, hasil NLU, dan aksi NPC berjalan sesuai fungsi yang diharapkan.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.12.2 User Acceptance Testing

User Acceptance Testing (UAT) merupakan metode pengujian yang digunakan untuk mengetahui apakah sistem dapat diterima oleh pengguna akhir. UAT berfokus pada penilaian pengguna terhadap kesesuaian sistem dengan kebutuhan, kemudahan penggunaan, dan manfaat yang dirasakan setelah mencoba sistem. (Nik Ahmad and Megat Sazali, 2021) menggunakan UAT untuk mengevaluasi penerimaan pengguna terhadap aplikasi, sehingga metode ini relevan untuk menilai kelayakan sistem dari sudut pandang pengguna.

Penelitian ini menggunakan UAT untuk mengetahui tingkat penerimaan pengguna terhadap sistem interaksi suara NPC. Pengguna menilai apakah fitur perintah suara, respons NPC, dan alur interaksi sudah sesuai dengan kebutuhan permainan. Hasil UAT digunakan sebagai data pendukung untuk mengetahui apakah sistem yang dikembangkan dapat diterima sebagai alternatif dari dialog menu konvensional.

2.12.3 System Usability Scale

System Usability Scale (SUS) merupakan instrumen pengukuran usability yang digunakan untuk menilai tingkat kegunaan sistem berdasarkan persepsi pengguna. (Brooke, dalam Kurniawan, Nata and Royal, 2022), menjelaskan bahwa SUS dikembangkan sebagai kuesioner untuk mengukur usability sistem komputer dari sudut pandang subjektif pengguna. SUS banyak digunakan karena sederhana, menghasilkan skor dalam rentang 0–100, serta dinilai valid dan reliabel walaupun digunakan pada sampel kecil.

Penelitian ini menggunakan SUS untuk mengukur kemudahan penggunaan sistem interaksi suara NPC. Penilaian dilakukan setelah pengguna mencoba sistem, kemudian pengguna mengisi kuesioner SUS berdasarkan pengalaman mereka. Skor SUS digunakan untuk mengetahui apakah sistem berada pada kategori kegunaan yang dapat diterima, sehingga hasilnya dapat melengkapi pengujian fungsional dan UAT.

2.12.4 Profiling Latensi dan Performa Sistem

Profiling latensi dan performa sistem merupakan proses pengukuran untuk mengetahui kecepatan respons dan penggunaan sumber daya sistem. Latensi menjadi aspek penting karena pengguna mengharapkan NPC merespons perintah



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dalam waktu yang wajar. Jika respons terlalu lambat, interaksi dapat terasa tidak natural dan mengganggu pengalaman bermain.

Profiling dilakukan untuk mengukur waktu pemrosesan dari input suara hingga respons NPC, serta melihat penggunaan sumber daya seperti CPU, memori, dan kestabilan permainan. Pengujian ini diperlukan karena sistem menggunakan arsitektur *Hybrid Client-Cloud*, sehingga performa perlu dinilai dari sisi klien, Server, dan komunikasi antar komponen. Dengan demikian, *profiling* digunakan untuk memastikan bahwa sistem tidak hanya berjalan secara fungsional, tetapi juga cukup responsif dan stabil saat digunakan.

2.13 Penelitian Terdahulu

Penelitian terdahulu digunakan untuk mengetahui posisi penelitian yang dilakukan terhadap kajian sebelumnya yang relevan. Sumber terdahulu dipilih berdasarkan kedekatannya dengan sistem yang dikembangkan, yaitu sistem interaksi suara pada NPC dalam permainan *The Elder Scrolls V: Skyrim Special Edition* yang melibatkan *Speech-to-Text*, *Natural Language Understanding*, arsitektur *Hybrid Client-Cloud*, toleransi kesalahan transkripsi, serta respons suara berbasis *Baked Audio* dan *lip-sync*.

Sub-bab ini membahas ringkasan penelitian terdahulu, analisis gap penelitian, serta posisi penelitian yang dilakukan. Ringkasan penelitian terdahulu tidak mencakup seluruh sumber pada BAB II, melainkan difokuskan pada beberapa penelitian yang paling relevan dengan ruang lingkup sistem yang dikembangkan.

2.13.1 Ringkasan Penelitian Terdahulu

Beberapa penelitian sebelumnya telah membahas pengembangan kecerdasan buatan dalam *game*, agen virtual, pemrosesan suara, pemahaman bahasa alami, arsitektur komputasi terdistribusi, serta teknologi respons suara. Ringkasan penelitian terdahulu yang paling relevan dengan penelitian ini disajikan pada Tabel 2.11.

Tabel 2. 11 Ringkasan Penelitian Terdahulu

No	Peneliti	Fokus Penelitian	Metode/Teknologi	Keterbatasan	Relevansi terhadap Penelitian Ini
1	Park et al. (2023)	Pengembangan agen generatif yang mampu menampilkan perilaku sosial lebih <i>believable</i>	Memori, refleksi, dan perencanaan pada agen virtual	Tidak berfokus pada NPC dalam <i>game engine</i> komersial dan tidak menggunakan perintah suara	Menjadi dasar kebutuhan NPC yang lebih kontekstual dan tidak hanya bergantung pada respons statis
2	Hochreiter et al. (2026)	Persepsi pemain terhadap dialog NPC generatif dalam permainan video	Studi persepsi pemain terhadap kualitas dialog NPC	Fokus pada persepsi dialog, bukan implementasi sistem STT-NLU ke dalam <i>game</i>	Menguatkan keterbatasan <i>scripted dialogue</i> dan kebutuhan dialog NPC yang lebih natural
3	Radford et al. (2023)	Pengembangan model <i>Automatic Speech Recognition</i> Whisper	<i>Speech-to-Text</i> berbasis data audio skala besar	Tidak membahas penggunaan STT untuk mengendalikan NPC	Menjadi dasar penggunaan STT sebagai tahap awal pemrosesan perintah

Hak Cipta :

- Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
- Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

				dalam permainan	suara pemain
4	Perdana & Adikara (2025)	<i>Multi-Task Learning</i> untuk <i>Named Entity Recognition</i> dan <i>Intent Classification</i>	Model NLU multi-task	Tidak diterapkan pada domain permainan digital atau NPC	Menjadi dasar penggabungan <i>Intent</i> dan <i>entity</i> dalam satu model NLU
5	Zhang, Yi, & Ma (2024)	Strategi <i>computation offloading</i> berdasarkan jenis tugas komputasi	Survei pemindahan beban komputasi ke infrastruktur lain	Tidak spesifik membahas sistem interaksi suara pada <i>game</i>	Menjadi dasar pemindahan proses NLU ke <i>Server Cloud</i> agar beban lokal berkurang
6	Kannoja et al. (2025)	Pemanfaatan AI untuk <i>speech synthesis</i> , <i>Voice Cloning</i> , dan sinkronisasi audio-visual	STT, TTS, <i>Voice Cloning</i> , dan <i>lip-sync</i>	Tidak membahas integrasi respons suara ke NPC <i>Skyrim</i>	Menjadi dasar konsep respons suara NPC berbasis <i>Voice Cloning</i> dan sinkronisasi gerakan bibir

2.13.2 Analisis Gap Penelitian

Berdasarkan penelitian terdahulu pada Tabel 2.11, dapat diketahui bahwa masing-masing penelitian telah membahas komponen tertentu yang relevan dengan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

penelitian ini. (Park et al., 2023) dan (Hochreiter, Kriglstein and Wallner, 2026) menunjukkan pentingnya pengembangan agen virtual dan dialog NPC yang lebih natural. Namun, penelitian tersebut belum secara khusus membahas integrasi sistem interaksi suara ke dalam *game engine* komersial seperti The Elder Scrolls V: Skyrim Special Edition.

(Radford et al., 2023) telah menunjukkan kemampuan model *Speech-to-Text* modern dalam melakukan transkripsi suara, tetapi penelitian tersebut tidak membahas bagaimana hasil transkripsi digunakan sebagai perintah NPC. Perdana dan (Perdana and Adikara, 2025) telah membahas *Multi-Task Learning* untuk *Intent Classification* dan *Named Entity Recognition*, tetapi belum diterapkan pada domain permainan digital. Sementara itu, (Zhang, Yi and Ma, 2024) membahas *computation offloading*, tetapi tidak secara spesifik membahas pembagian beban kerja antara komputer pemain dan *Server Cloud* dalam sistem interaksi suara. (Kannojia et al., 2025) membahas *Voice Cloning* dan *lip-sync*, tetapi belum mengarah pada penggunaan aset suara statis dalam sistem NPC berbasis mod.

Dengan demikian, gap utama yang ditemukan adalah belum adanya penelitian yang secara terpadu menggabungkan interaksi suara, *Speech-to-Text*, Multi-Task NLU, arsitektur *Hybrid Client-Cloud*, toleransi kesalahan transkripsi, serta respons suara berbasis *Baked Audio* dan *lip-sync* untuk meningkatkan interaksi NPC pada permainan *open-world*.

2.13.3 Posisi Penelitian yang Dilakukan

Penelitian ini diposisikan sebagai pengembangan sistem interaksi suara untuk NPC Serana dalam permainan The Elder Scrolls V: Skyrim Special Edition. Berbeda dengan penelitian terdahulu yang umumnya hanya membahas salah satu aspek, penelitian ini mengintegrasikan beberapa komponen ke dalam satu sistem, yaitu *Speech-to-Text*, *Natural Language Understanding* berbasis *Multi-Task Learning*, *Fuzzy String Matching*, arsitektur *Hybrid Client-Cloud*, serta respons verbal berbasis *Baked Audio* hasil *Voice Cloning*.

Kontribusi utama penelitian ini terletak pada integrasi sistem AI eksternal dengan lingkungan permainan melalui mekanisme pertukaran data terstruktur. Suara pemain dikonversi menjadi teks, diproses oleh model NLU untuk menghasilkan



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Intent, *entity*, dan *sentiment*, kemudian hasilnya digunakan untuk menentukan aksi dan respons NPC. Dengan pendekatan tersebut, penelitian ini tidak hanya membahas model AI secara terpisah, tetapi juga menunjukkan bagaimana komponen AI dapat dihubungkan dengan *game engine* untuk menghasilkan interaksi NPC yang lebih natural, responsif, dan imersif.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB III METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini menerapkan pendekatan metode campuran (*Mixed Methods*) guna mengevaluasi kelayakan arsitektur *Hybrid Client-Cloud* pada sistem interaksi suara NPC secara menyeluruh. Pendekatan kualitatif difokuskan pada validasi fungsional perangkat lunak melalui *Black Box Testing* yang mencakup validasi fungsional seluruh 10 kelas *Intent* NLU, 7 sinyal otonom berbasis konteks lingkungan, sistem manajemen afeksi, integritas jembatan data JSON, serta penanganan kondisi batas (*Edge Cases*). Evaluasi kualitatif juga diterapkan untuk menganalisis umpan balik, persepsi, dan penerimaan pengguna akhir secara deskriptif melalui pelaksanaan *User Acceptance Testing* (UAT).

Sementara itu, pendekatan kuantitatif dikhususkan untuk mengukur efisiensi performa teknis serta metrik kebergunaan sistem secara objektif. Fokus analisis teknis meliputi pengukuran latensi (*Round-Trip Time*) dalam pemrosesan kecerdasan buatan antara klien dan *cloud*, serta pemprofilan perangkat keras (*hardware profiling*) untuk mengidentifikasi potensi *bottleneck* selama perekaman suara. Pengukuran kuantitatif ini mencakup penilaian tingkat kemudahan penggunaan sistem yang dihitung secara matematis melalui instrumen *System Usability Scale* (SUS). Pendekatan kuantitatif juga mencakup evaluasi komparatif performa model *Multi-Task Learning* antara arsitektur DistilBERT Base Cased sebagai model baseline sesuai proposal awal dan RoBERTa-base sebagai model final, menggunakan metrik Accuracy dan Macro F1-Score pada tiga tugas klasifikasi (*Intent*, *Sentiment*, NER) di bawah kondisi pengujian terkontrol yang identik. Sinergi dari kedua metode ini dirancang untuk menghasilkan validasi arsitektur sistem yang komprehensif dan terukur.

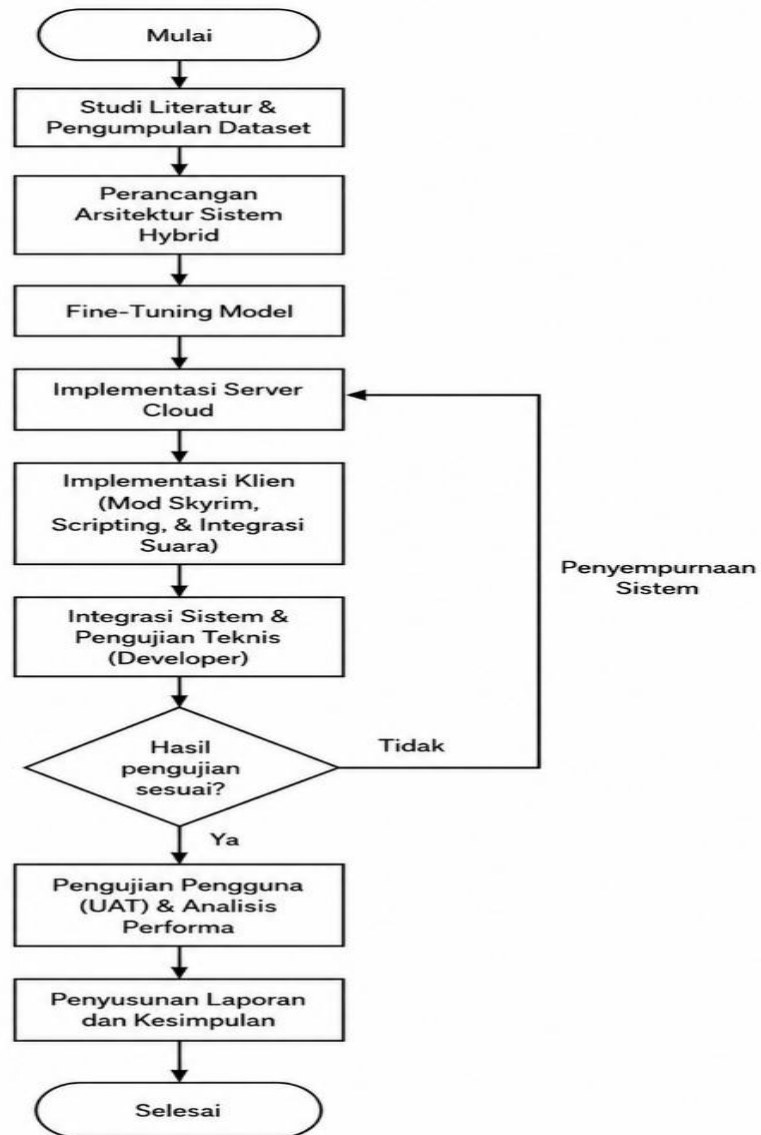
3.2 Tahapan Penelitian

Pelaksanaan penelitian ini dilakukan secara sistematis melalui serangkaian tahapan yang terstruktur. Secara visual, alur tahapan penelitian ini dapat dilihat pada Gambar 3.1. Berikut adalah penjelasan untuk masing-masing tahapan:



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 3. 1 Diagram Tahapan Penelitian

1. Studi Literatur & Pengumpulan Dataset

Tahap ini diawali dengan kajian literatur terkait komputasi *Client-Cloud*, pembelajaran mesin *Multi-Task Learning*, dan rekayasa modifikasi pada *Creation Engine*. Selanjutnya, dilakukan pengumpulan dataset yang terbagi menjadi dua ranah: ranah linguistik dan ranah audio. Pada ranah linguistik, dilakukan proses rekayasa data (*data engineering*) untuk mengekspansi 300 data dasar (*seed*) menjadi lebih dari 3.000 baris instruksi berlabel. Pada ranah audio, dilakukan ekstraksi suara murni NPC Serana dari dalam gim

dengan durasi total 14 menit 48 detik sebagai *Dataset* untuk proses *Voice Cloning*.

2. Perancangan Arsitektur Sistem *Hybrid Client-Cloud*.

Merancang infrastruktur komputasi hibrida yang mendistribusikan beban kerja pemrosesan. Sisi Klien (*Client*) di perangkat lokal dirancang untuk menangani penangkapan dan transkripsi suara lokal (*Client-STT*), penyimpanan berkas audio luring (*Baked Audio*), dan eksekusi instruksi di dalam gim. Sementara itu, sisi Awan (*Cloud*) didesain secara eksklusif sebagai penyedia layanan *Application Programming Interface* (API) yang hanya menerima teks dan mengembalikan muatan data (*payload*) berformat JSON tanpa melakukan sintesis suara dinamis guna meminimalisasi latensi.

3. *Fine-Tuning* Model dan Produksi *Baked Audio*

Tahap ini merupakan fase komputasi pra-produksi yang dilakukan di luar lingkungan *cloud* utama. Proses *Fine-Tuning* untuk model *Natural Language Understanding* (NLU) berarsitektur RoBERTa-base dieksekusi menggunakan platform Google Colab. Model ini dilatih agar mampu memprediksi *Intent*, *Entity*, dan *Sentiment* secara simultan, lalu dioptimalkan menggunakan kuantisasi INT8 ONNX. Pada saat yang sama, komputasi GPU *Cloud* di platform RunPod digunakan untuk melatih model Retrieval-based Voice conversion (RVC v2). Model ini digunakan untuk melakukan pemetaan karakteristik warna suara (*timbre mapping*) pada 424 baris dialog yang disintesis dari layanan ElevenLabs, sehingga menghasilkan ratusan berkas audio statis (*Baked Audio*) dengan karakteristik vokal yang identik dengan pengisi suara asli.

4. Implementasi *Server Cloud* (VPS Domainsia)

Membangun *Server* jarak jauh pada *Virtual Private Server* (VPS) Domainsia. *Server* ini dikonfigurasi menggunakan kerangka kerja FastAPI untuk mengeksekusi model NLU RoBERTa-base yang telah melalui proses *Fine-Tuning* di Google Colab. Logika di dalam *Server* diatur agar modul pemrosesan bahasa dapat memetakan hasil klasifikasi intent pemain dengan referensi *FormID* audio yang spesifik (misalnya: 0x03A91B). Hasil

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pemrosesan *Server* ini hanya berupa teks JSON ringan yang memuat instruksi aksi, nilai afeksi, dan kode *FormID*.

5. Implementasi Klien (*Client-STT & Integrasi Game engine*)

Memprogram sistem penangkapan interaksi pemain di komputer lokal. Menggunakan bahasa Python, diimplementasikan mekanisme *Push-to-Talk* yang terintegrasi dengan model *Faster-Whisper* (CPU-based) sebagai *Client-STT*. Transkripsi teks hasil tangkapan tersebut dikirim ke *Server VPS*, dan respons JSON yang diterima dibaca secara asinkron menggunakan modul *PapyrusUtil*. Skrip bahasa *Papyrus* kemudian mengeksekusi aksi navigasi atau pertarungan NPC di dalam gim, sekaligus memicu pemutaran berkas *Baked Audio* yang telah tersimpan di direktori penyimpanan lokal pemain.

6. Integrasi Sistem dan Pengujian Teknis (Developer)

Menghubungkan klien lokal dan *Server Cloud* untuk memvalidasi keseluruhan alur komunikasi data serta kinerja sistem. Pada tahap ini dilakukan evaluasi performa model *Natural Language Understanding* melalui pengujian Accuracy dan Macro F1-Score pada tugas klasifikasi *Intent*, *Sentiment*, dan *Named Entity Recognition*. Evaluasi dilakukan secara komparatif antara model DistilBERT Base Cased sebagai baseline dan model RoBERTa-base sebagai model final, termasuk pengujian menggunakan Adversarial STT Test Set untuk mensimulasikan keluaran transkripsi suara dunia nyata. Dilakukan *Black Box Testing* terhadap seluruh fitur sistem yang mencakup transisi afeksi, parameter navigasi, sistem pertarungan, dan integritas komunikasi JSON. Tahap ini juga mencakup *profiling* latensi *Round-Trip Time* (RTT) serta pemantauan utilisasi perangkat keras guna mengidentifikasi potensi *bottleneck* pada proses inferensi *Client-STT*.

7. Pengujian Pengguna (UAT) dan Analisis Performa.

Mengundang pengguna akhir (*end-User*) untuk melakukan pengujian operasional terhadap sistem otonomi AI. Pengumpulan data kualitatif dan evaluasi persepsi mengenai interaksi dilakukan melalui instrumen *User Acceptance Testing* (UAT). Secara bersamaan, evaluasi kuantitatif terkait



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

tingkat kemudahan penggunaan sistem diukur secara matematis menggunakan kuesioner standar *System Usability Scale* (SUS).

8. Penyusunan Laporan dan Kesimpulan

Mendokumentasikan keseluruhan penelitian, yang meliputi *data engineering* linguistik, produksi audio, implementasi arsitektur *Client-Cloud*, hingga hasil pengujian metrik teknis dan analisis SUS dari responden. Dokumentasi ini disusun secara komprehensif ke dalam buku laporan skripsi sebagai luaran akhir akademis, yang ditutup dengan penarikan kesimpulan secara objektif.

3.3 Objek Penelitian

Objek penelitian dalam skripsi ini adalah sistem interaksi suara berbasis arsitektur *Hybrid Client-Cloud* yang diterapkan pada karakter *Non-Playable Character* (NPC) Serana dalam gim *The Elder Scrolls V: Skyrim*. Sistem tersebut dirancang untuk memungkinkan pemain memberikan perintah menggunakan suara alami yang kemudian diproses melalui mekanisme *Speech-to-Text*, *Natural Language Understanding*, dan integrasi *game engine* secara otomatis.

Penelitian difokuskan pada evaluasi kemampuan sistem dalam memahami perintah pemain, mengendalikan perilaku NPC secara otonom, menghasilkan respons suara berbasis *Baked Audio*, serta mempertahankan performa interaksi yang responsif melalui distribusi beban komputasi antara perangkat klien dan *Server Cloud*. Evaluasi dilakukan terhadap aspek fungsionalitas, performa teknis, dan penerimaan pengguna untuk menilai kelayakan implementasi arsitektur yang diusulkan.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan krusial untuk mendefinisikan spesifikasi, batasan, dan kapabilitas yang harus dipenuhi oleh arsitektur *Client-Cloud* dalam menjalankan modifikasi otonomi NPC Serana. Berdasarkan observasi dan perancangan yang telah dilakukan, kebutuhan sistem ini diklasifikasikan ke dalam empat parameter utama, yaitu kebutuhan fungsional, kebutuhan non-fungsional, kebutuhan *hardware*, serta kebutuhan *software*.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan alur proses dan kapabilitas interaksi yang harus diakomodasi oleh sistem perangkat lunak. Pada sisi *Client*, sistem harus mampu merekam input suara pengguna secara lokal menggunakan mekanisme *Push-to-Talk* dan mentranskripsikannya menjadi teks melalui modul *Client-STT*. Teks tersebut kemudian dituntut untuk dapat ditransmisikan secara akurat ke *Server* jarak jauh. Pada sisi *Cloud*, sistem harus mampu memproses teks yang diterima menggunakan model *Multi-Task Learning* NLU untuk memprediksi *Intent*, mengekstraksi *Entity*, dan menganalisis *Sentiment* secara simultan. Selanjutnya, sistem wajib merangkum hasil prediksi tersebut ke dalam *payload* berformat JSON untuk dikembalikan ke sisi *Client*. Pada tahap akhir ekuilibrium fungsional, jembatan data PapyrusUtil harus mampu menginjeksikan data JSON tersebut ke dalam *game engine*, sehingga skrip Papyrus dapat mengeksekusi parameter navigasi (follow, wait, trade, dismiss, combat) sekaligus memicu pemutaran *Baked Audio* yang merujuk pada kode FormID secara sinkron.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dalam penelitian ini berfokus pada stabilitas performa, efisiensi komputasi, dan batasan arsitektur sistem, dengan prioritas utama meminimalkan latensi waktu respons bolak-balik (*Round-Trip Time*) demi menjaga imersi pemain. Untuk menjaga stabilitas tersebut selama pengujian, model *Client-STT* dikonfigurasi berjalan pada CPU agar tidak berbagi sumber daya dengan GPU yang digunakan untuk rendering gim, meskipun pengembangan berbasis akselerasi

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

GPU tetap dimungkinkan di masa depan. Sementara itu, model NLU pada sisi Server dioptimalkan menggunakan kuantisasi INT8 ONNX untuk meningkatkan efisiensi inferensi serta mempercepat waktu respons layanan API.

4.1.3 Kebutuhan *Hardware*

Kebutuhan *hardware* mencakup spesifikasi perangkat keras fisik dan komputasi virtual yang mendasari infrastruktur *Client-Cloud* serta tahap pra-produksi. Spesifikasi ini dijabarkan ke dalam bentuk tabel berdasarkan masing-masing lingkungan eksekusi.

1. Lingkungan *Client* (Lokal):

Spesifikasi pada sisi *Client* dibagi menjadi dua kategori: Spesifikasi Minimum yang menjamin model *Client-STT* (Faster-Whisper) dan *game engine* dapat berjalan secara paralel tanpa kendala thermal throttling, serta unit pengembangan (Development Unit) yang digunakan oleh peneliti selama proses rekayasa perangkat lunak dan pengujian performa.

Tabel 4. 1 Spesifikasi *Hardware Client* (Klien Lokal)

Komponen	Spesifikasi Minimum	Development Unit
Processor (CPU)	Intel Core i5 Gen 9 / AMD Ryzen 5 3600	AMD Ryzen 5 7500F
Graphics (GPU)	NVIDIA GTX 780 3GB /AMD R9 290 4GB (Sistem mod tidak menggunakan GPU)	Radeon RX 6600 8GB
Memory (RAM)	8 GB DDR4	32GB DDR5
Storage	128GB SSD (Sistem mod hanya menggunakan sekitar 300MB)	1TB SSD
Input Device	Standard Desktop/Laptop Microphone	Standard Desktop/Laptop Microphone
Connection	8Mbps Stable	20Mbps Stable

Spesifikasi minimum pada penelitian ini tidak ditentukan berdasarkan kemampuan sistem untuk sekadar berjalan, melainkan berdasarkan hasil pengujian performa

yang masih mampu memberikan pengalaman interaksi yang layak. Batas minimum prosesor Intel Core i5 Generasi ke-9 atau AMD Ryzen 5 3600 dipilih karena perangkat dengan kelas performa tersebut masih dapat menjalankan *game engine* Skyrim dan model Faster-Whisper secara bersamaan tanpa mengalami kegagalan operasional sistem, meskipun menghasilkan latensi yang lebih tinggi dibandingkan perangkat pengembangan utama.

2. Lingkungan Server (Cloud)

Infrastruktur Server jarak jauh menggunakan layanan *Virtual Private Server* (VPS) dari Domainsia. Spesifikasi ini dialokasikan secara khusus untuk menjamin eksekusi titik akhir *Application Programming Interface* (API) dan inferensi model *Natural Language Understanding* (NLU) *RoBERTa-base* tetap bersiaga (*loaded in memory*) dengan stabilitas *continuous uptime*.

Tabel 4. 2 Spesifikasi *Hardware Server* (VPS Domainsia)

Komponen	Spesifikasi Alokasi
Virtual CPU (vCPU)	2 CPU Core
Memory (RAM)	2GB RAM
Storage	30GB SSD
Bandwidth	Unlimited

3. Lingkungan Pra-Produksi (Pembuatan Model)

Tahap *Fine-Tuning* dan pemetaan karakteristik suara difasilitasi oleh komputasi GPU High-End yang disewa melalui layanan *Cloud* RunPod. Lingkungan ini dieksekusi murni secara luring di luar siklus *gameplay* utama untuk melatih arsitektur Retrieval-based Voice conversion (RVC v2).

Tabel 4. 3 Spesifikasi Komputasi Pra-Produksi (RunPod)

Komponen	Spesifikasi Alokasi
Graphics (GPU)	RTX A5000
Video RAM (VRAM)	24GB VRAM
System Memory	25GB RAM

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.1.4 Kebutuhan Software

Kebutuhan *software* merangkum seluruh tech stack, library, dan bahasa pemrograman yang memungkinkan sistem otonomi ini berjalan secara *hybrid*.

1. Perangkat Lunak Sisi *Client*:

- 1) The Elder Scrolls V: Skyrim: Lingkungan eksekusi utama berupa Creation Engine yang menjadi wadah implementasi NPC Serana dan tempat pemain berinteraksi secara visual.
- 2) SKSE64 (Skyrim Script Extender): Modifikasi injeksi memori pihak ketiga yang krusial untuk memperluas fungsionalitas dan kapabilitas eksekusi dari skrip *game engine* bawaan.
- 3) PapyrusUtil: Modul ekstensi (SKSE plugin) yang bertindak sebagai jembatan data asinkron. Ekstensi ini sangat krusial karena memungkinkan *game engine* untuk membaca dan mengurai respons berformat JSON yang dikirimkan oleh API Server.
- 4) Papyrus: Bahasa pemrograman bawaan (native) dari Creation Engine. Digunakan dalam pembuatan berkas SeranaAI_CoreScript.psc untuk mengendalikan logika pertarungan, navigasi, parameter nilai afeksi, dan pemicu berkas *Baked Audio*.
- 5) Python (Klien): Bahasa pemrograman tingkat tinggi yang menjalankan skrip `stt_client.py` untuk menangkap masukan perifer keyboard (*Push-to-Talk*) dan merekam suara menggunakan pustaka `sounddevice`.
- 6) Faster-Whisper: Pustaka kecerdasan buatan dari integrasi CTranslate2 yang dijalankan di Python klien untuk melakukan transkripsi suara menjadi teks secara offline di tingkat lokal.

2. Perangkat Lunak Sisi *Server*:

- 1) Ubuntu OS: Sistem operasi distribusi Linux yang berjalan pada arsitektur VPS Domainsia untuk menjamin stabilitas jaringan backend.
- 2) Python (Server): Lingkungan eksekusi untuk menjalankan berkas `nlu_server.py`.
- 3) FastAPI & Uvicorn: Kerangka kerja web (web framework) yang sangat cepat dan ringan, digunakan untuk merancang titik akhir REST API (API *endpoint*) yang melayani permintaan dari klien lokal.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4) ONNX *Runtime*: Kerangka inferensi lintas platform yang secara spesifik dieksekusi di *Server* untuk menjalankan model *Multi-Task Learning* RoBERTa-base berformat .onnx (kuantisasi INT8) agar respons klasifikasi (*Intent, Entity, Sentiment*) tercapai dalam hitungan milidetik.

3. Perangkat Lunak Pra-Produksi:

1) ElevenLabs & RVC v2: Platform sintesis kecerdasan buatan yang digunakan secara eksklusif sebelum modifikasi dirilis untuk menghasilkan 424 berkas *Baked Audio* dengan pemetaan karakteristik vokal yang akurat.

4. Perangkat Lunak *Deployment* dan Manajemen Mod:

Mod Organizer 2 (MO2): Perangkat lunak manajemen modifikasi berbasis *Virtual File System* (VFS) yang digunakan sebagai media instalasi resmi mod ini. MO2 dipilih karena kemampuannya mengisolasi berkas modifikasi dari direktori instalasi gim asli, mengelola urutan pemuatan plugin (load order), serta memastikan rantai dependensi SKSE64 > PapyrusUtil > Serana *CloudAI* terpenuhi secara otomatis tanpa risiko korupsi berkas instalasi vanilla.

4.2 Perancangan Sistem

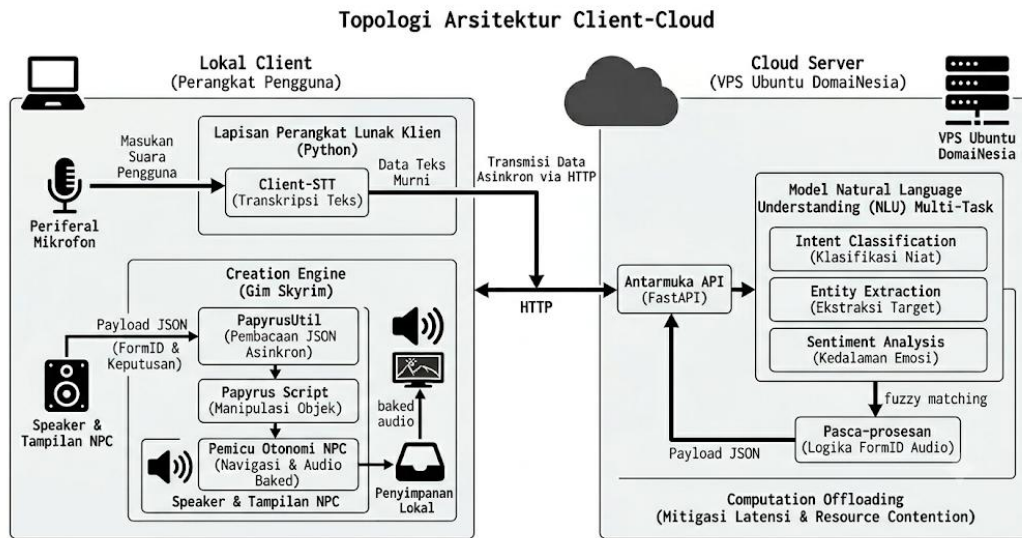
Tahapan perancangan sistem menguraikan topologi dan mekanisme kerja dari arsitektur *Client-Cloud* yang dibangun. Perancangan ini didasarkan pada kebutuhan untuk meminimalisasi latensi komputasi transmisi jaringan sekaligus mempertahankan stabilitas kinerja *game engine* secara lokal. Arsitektur sistem otonomi ini dibagi menjadi lima subsistem utama, yaitu perancangan arsitektur global, pemrosesan suara di sisi klien (*Client-STT*), pemrosesan bahasa di sisi *cloud* (*Cloud NLU*), modifikasi komponen *game engine*, serta rekayasa audio generatif.

4.2.1 Arsitektur Sistem

Arsitektur sistem pada penelitian ini dirancang menggunakan pendekatan *Hybrid Client-Cloud Architecture*. Pendekatan ini diimplementasikan secara eksplisit guna mengisolasi beban kerja pemrosesan kecerdasan buatan dari siklus grafis rendering pada *game engine*. Pemisahan komputasi melalui metode *computation offloading* ke infrastruktur *cloud* dirancang untuk mereduksi latensi transmisi secara signifikan serta mencegah terjadinya perebutan sumber daya komputasi (resource contention) pada perangkat keras lokal. Secara visual, topologi komunikasi data dua arah dari arsitektur *Client-Cloud* ini diilustrasikan pada Gambar 4.1.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 1 Topologi Arsitektur *Client-Cloud*

Alur transmisi data pada arsitektur ini diinisiasi dari lapisan perangkat keras fisik, di mana periferil mikrofon merekam masukan suara dari pengguna. Sinyal audio tersebut secara langsung disalurkan ke lapisan perangkat lunak klien yang dikonstruksi menggunakan bahasa pemrograman Python untuk ditranskripsi menjadi teks murni melalui modul *Client-STT*. Pengolahan awal di sisi lokal ini sengaja dirancang untuk meniadakan kewajiban pengunggahan berkas audio mentah, sehingga data yang ditransmisikan melalui protokol HTTP (*Hypertext Transfer Protocol*) menuju *Virtual Private Server* (VPS) Ubuntu jarak jauh hanya berupa teks asinkron yang sangat ringan.

Data teks selanjutnya diterima oleh antarmuka *Application Programming Interface* (API) berbasis FastAPI pada lingkungan komputasi *Server* untuk diproses oleh model *Natural Language Understanding* (NLU). Pemrosesan terpusat ini dikonfigurasi untuk menjalankan fungsionalitas inferensi *Multi-Task* guna mengklasifikasikan intent aksi (*Intent*), mengekstraksi target referensi (*Entity*), dan mengalkulasi kedalaman emosi (*Sentiment*). Seluruh keluaran linguistik tersebut kemudian diolah secara terprogram menggunakan logika pencocokan string (*Fuzzy Matching*) guna merumuskan kode referensi FormID audio secara deterministik. Keseluruhan parameter keputusan dari *Server* ini dikemas kembali menjadi *payload* berformat JSON dan dikirimkan ke komputer klien sebagai sebuah respons tertutup.

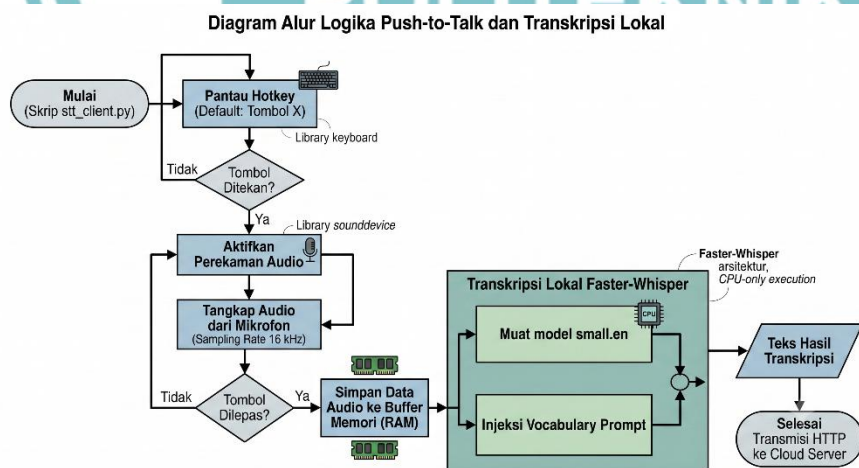
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Struktur data JSON yang diterima di perangkat lokal selanjutnya didistribusikan langsung ke dalam lapisan memori Creation Engine pada fase pemrosesan akhir. Modul jembatan data PapyrusUtil dieksekusi untuk melakukan pembacaan nested JSON tersebut secara asinkron guna mencegah interupsi pada kecepatan bingkai (*frame rate*) gim yang sedang berjalan. Skrip manipulasi objek berbasis bahasa Papyrus kemudian menerjemahkan parameter tersebut menjadi pemicu otonomi *Non-Playable Character* (NPC), yang secara serentak mengendalikan rute navigasi dan memanggil berkas suara statis (*Baked Audio*) dari direktori penyimpanan lokal berdasarkan referensi kode FormID yang diterima.

4.2.2 Perancangan Pemrosesan Suara Klien (*Client-STT*)

Mekanisme pemrosesan suara pada sisi klien dirancang untuk melakukan konversi sinyal audio menjadi teks secara lokal tanpa melibatkan transmisi data audio mentah ke jaringan internet. Perancangan ini bertujuan untuk meniadakan beban bandwidth yang besar dan meminimalisasi latensi pada tahap awal interaksi. Logika operasional dari subsistem ini dibangun menggunakan bahasa pemrograman Python melalui skrip *stt_client.py* yang mengintegrasikan model *Automatic Speech Recognition* (ASR). Diagram alur logika penangkapan suara dan transkripsi lokal ini diilustrasikan pada Gambar 4.2.



Gambar 4. 2 Diagram Alur Logika *Push-to-Talk* dan Transkripsi Lokal

Sistem penangkapan suara telah diimplementasikan menggunakan mekanisme *Push-to-Talk* (PTT) yang dipicu melalui *hotkey* spesifik pada *keyboard*. Fitur ini dirancang secara dinamis sehingga pengguna (*User*) dapat mengonfigurasi tombol



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pemicu sesuai preferensi, dengan tombol “X” sebagai konfigurasi awal (*default*). Perekaman audio hanya akan dilakukan selama tombol tersebut ditekan, yang dikendalikan melalui pustaka keyboard dan sounddevice pada Python. Audio ditangkap dengan parameter sampling rate sebesar 16.000 Hz untuk menjamin kompatibilitas frekuensi dengan standar masukan model transformer. Data audio yang tertangkap kemudian disimpan sementara dalam buffer memori sebelum diproses oleh mesin transkripsi.

Mesin transkripsi pada subsistem ini diimplementasikan menggunakan arsitektur *Faster-Whisper* dengan varian model *small.en* sebagai konfigurasi awal (*default*). Sistem ini dirancang secara fleksibel agar pengguna dapat menyesuaikan tingkat model yang digunakan, baik ke varian yang lebih tinggi maupun lebih rendah, sesuai dengan spesifikasi perangkat keras yang dimiliki. Panduan mengenai konfigurasi dan perubahan varian model ini telah didokumentasikan di dalam berkas *readme.txt*. Pemilihan model *default* ini didasarkan pada titik ekuilibrium antara kecepatan inferensi dan akurasi transkripsi untuk bahasa Inggris. Secara teknis, model dikonfigurasi dengan parameter `device="cpu"` dan `compute_type="int8"`. Penggunaan kuantisasi INT8 bertujuan untuk mereduksi ukuran model dan mempercepat kalkulasi matematis pada prosesor. Keputusan untuk mengeksekusi model secara murni pada *Central Processing Unit* (CPU) dirancang agar tidak mengganggu kinerja *Graphics Processing Unit* (GPU) yang dialokasikan sepenuhnya untuk proses rendering grafis pada gim *The Elder Scrolls V: Skyrim*.

Guna meningkatkan akurasi transkripsi terhadap terminologi spesifik di dalam gim, dirancang sebuah mekanisme injeksi *vocabulary prompt*. Fitur ini berfungsi memberikan panduan konteks kepada model agar dapat mengenali istilah unik (seperti nama lokasi, nama NPC, atau istilah sihir) yang secara normal dikategorikan sebagai *Out of Vocabulary* (OOV). Hasil akhir dari perancangan subsistem ini adalah untaian teks (string) yang siap ditransmisikan melalui protokol HTTP menuju *Server* untuk proses penalaran bahasa lebih lanjut.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.2.3 Perancangan *Natural Language Understanding (Cloud NLU)*

Infrastruktur *Natural Language Understanding (NLU)* dirancang untuk ditempatkan sepenuhnya pada sisi *Server* jarak jauh (*Virtual Private Server*) dengan memanfaatkan kerangka kerja *FastAPI*. Perancangan ini ditujukan untuk membangun antarmuka pemrograman aplikasi berbasis *REST API* yang ringan, asinkron, dan mampu menangani permintaan dari klien lokal dengan efisiensi tinggi. Subsistem ini dikonfigurasi untuk menerima masukan data tekstual dan data metrik afeksi dari klien melalui kelas skema penerima data (*ChatRequest*). Rancangan struktur skema muatan data (*payload*) permintaan dan respons pada titik akhir (*API endpoint*) dirinci pada Tabel 4.4.

Tabel 4. 4 Skema Payload REST API

Bagian Komunikasi	Kelompok Data	Nama Parameter	Tipe Data	Keterangan
<i>Request Body (Inbound ChatRequest)</i>	Input Linguistik	text	String	Untaian teks hasil transkripsi suara dari subsistem <i>Client STT</i> lokal klien.
	Status Afeksi	affinity_score	Integer	Nilai akumulasi skor hubungan emosional (<i>Score Affinity</i>) saat ini dari mesin permainan.
<i>Response Body (Outbound JSON Payload)</i>	FormID Audio	audio_formid	Integer	Kode referensi FormID untuk pemanggilan <i>Baked Audio</i> .
	Status Afeksi	sentiment_shift	Integer	Nilai perubahan skor afeksi NPC.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

FormID Target	target_formid	Integer	Nilai FormID hexadecimal objek atau lokasi hasil <i>Entity Extraction</i> .
Hasil Inferensi	<i>Intent</i>	String	Label hasil klasifikasi <i>Intent</i> .
Hasil Inferensi	<i>sentiment</i>	String	Label analisis <i>Sentiment</i> (“Positive”, “Negative”, atau “Neutral”).
Status Sistem	status	String	Status keberhasilan eksekusi <i>Pipeline</i> inferensi AI.
Tactical State	tactical_mode	String	Parameter gaya bertarung NPC selama combat.
Entity Result	target_name	String	Nama target hasil ekstraksi semantik <i>Named Entity Recognition</i> .

Mesin penalaran NLU di dalam *Server* menggunakan model *Multi-Task Learning* berarsitektur RoBERTa-base yang telah melalui optimasi kuantisasi INT8 berformat *ONNX Runtime*. Saat *Server* diinisialisasi, model kecerdasan buatan tersebut langsung dimuat ke dalam *Random Access Memory* (RAM) melalui sesi inferensi (*Inference Session*) guna memangkas waktu pemuatan ulang dan menjamin kecepatan eksekusi penalaran dalam orde milidetik. Model ini dirancang untuk memproses teks masukan secara paralel guna mengekstrak tiga parameter

Hak Cipta :

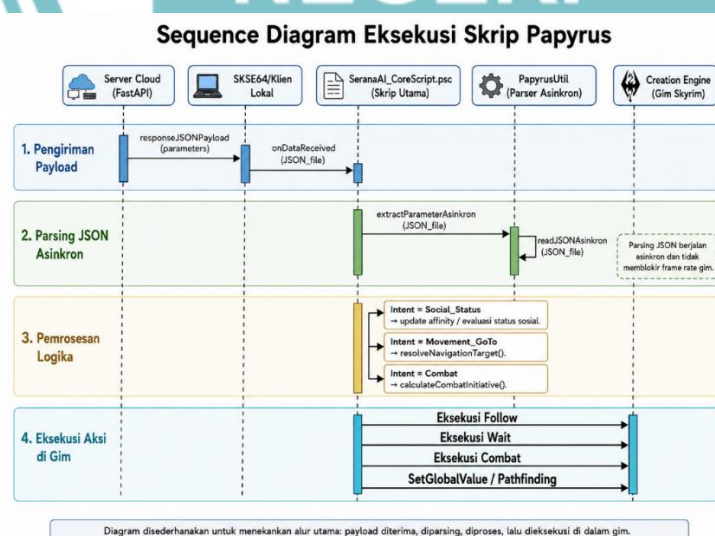
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

linguistik utama secara simultan, yaitu klasifikasi intent tindakan (*Intent*), ekstraksi entitas referensi (*Entity*), dan analisis afeksi percakapan (*Sentiment*).

Komponen krusial setelah tahap inferensi linguistik adalah implementasi algoritma pencocokan berbasis *Fuzzy Matching* menggunakan pustaka thefuzz di sisi *Server*. Algoritma ini dirancang untuk menjembatani hasil penalaran NLU yang bersifat dinamis dengan basis data permainan yang bersifat statis. Hasil kombinasi dari prediksi *Intent*, *Entity*, dan nilai afeksi akan dicocokkan secara fleksibel dengan kamus JSON untuk mencari baris dialog yang paling relevan. Setelah kecocokan ditemukan, sistem secara deterministik memetakan hasil tersebut menjadi kode referensi FormID audio spesifik (seperti contoh representasi nilai 0x000000). Hasil pemetaan tersebut beserta parameter instruksi aksi dan pembaruan nilai afeksi kemudian dikemas menjadi respons JSON tertutup untuk ditransmisikan kembali menuju klien lokal.

4.2.4 Perancangan Modifikasi Komponen *Game engine*

Integrasi parameter kecerdasan buatan dari infrastruktur *cloud* ke dalam ekosistem Creation Engine dirancang melalui modifikasi memori tingkat lanjut menggunakan Skyrim Script Extender (SKSE64). Modifikasi ini difokuskan pada perancangan arsitektur penerimaan data dan eksekusi motorik karakter di dalam gim. Alur kerja operasional dari lapisan skrip ini direpresentasikan melalui diagram urutan pada Gambar 4.3.



Gambar 4. 3 Sequence Diagram Eksekusi Skrip Papyrus



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Sistem saraf pusat dari otonomi NPC ini dikonstruksi melalui sebuah skrip utama bernama `SeranaAI_CoreScript.psc` yang dikembangkan menggunakan bahasa pemrograman Papyrus. Skrip ini dirancang untuk bekerja secara berdampingan dengan modul ekstensi `PapyrusUtil`. Peran `PapyrusUtil` di dalam perancangan ini sangat vital, yakni bertindak sebagai pengurai parsial (parser) yang membaca struktur data JSON dari *Server* secara asinkron tanpa memicu pembekuan *frame rate* pada klien lokal.

Setelah parameter JSON berhasil diekstraksi, skrip utama akan menerjemahkan nilai-nilai tersebut dan mendistribusikannya ke dalam serangkaian fungsi penanganan aksi (action handler). Fungsi-fungsi ini mencakup eksekusi rutinitas dasar pengikut seperti `HandleFollow()` untuk mode mengikuti, `HandleWait()` untuk mode siaga, serta `HandleCombat()` yang dirancang untuk mengambil alih inisiatif pertarungan secara dinamis berdasarkan kalkulasi ancaman. Bersamaan dengan eksekusi motorik tersebut, skrip juga dirancang untuk melakukan kalkulasi dan pembaruan memori hubungan sosial karakter secara terus-menerus melalui manipulasi variabel global `SeranaAffinityScore`.

Guna memberikan ilusi kognitif yang komprehensif, skrip Papyrus ini turut dilengkapi dengan kapabilitas sensor lingkungan (*context-aware*). Karakter dirancang untuk mampu mendeteksi variabel siklus waktu (*GameHour*) dan transisi cuaca lokal. Sementara itu, perancangan sistem navigasi dieksekusi melalui implementasi *Dual-Engine Pathfinding*. Mekanisme pergerakan tingkat tinggi ini tidak sekadar menggunakan kecerdasan buatan bawaan gim, melainkan diinjeksi dengan parameter referensi alias yang spesifik, yaitu `SeranaDestAlias` untuk target berupa benda atau entitas, dan `SeranaDestLocAlias` untuk target berbasis area atau lokasi. Hal ini memastikan karakter mampu melakukan pergerakan otonom yang presisi menuju koordinat yang ditentukan oleh *Server*.

4.2.5 Perancangan Audio Generatif (*Baked Audio*)

Perancangan komponen audio pada sistem otonomi ini secara eksplisit meniadakan proses sintesis *Text-to-Speech* (TTS) dinamis pada saat permainan berlangsung (run-time). Keputusan arsitektural ini diambil guna meminimalisasi latensi komputasi secara ekstrem dan memastikan respons suara karakter dieksekusi secara

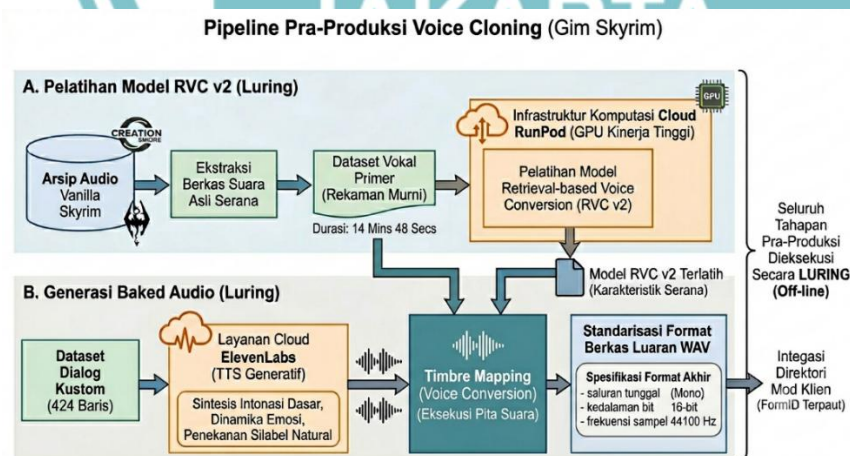
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

instan. Sebagai solusi efisiensi, diimplementasikan pendekatan *Baked Audio* yang diproduksi secara luring pada fase pra-produksi.

Keputusan untuk tidak menerapkan mekanisme *Text-to-Speech (TTS) real-time* didasarkan pada pertimbangan efisiensi latensi dan kompleksitas komputasi. Apabila pendekatan suara dinamis digunakan, maka setelah proses *Natural Language Understanding (NLU)* selesai dilakukan, sistem masih harus menjalankan tahapan sintesis suara (TTS) untuk menghasilkan audio dialog serta tahapan *Voice conversion* secara *real-time* guna mengubah karakteristik suara sintetis tersebut agar menyerupai suara asli karakter Serana. Dengan demikian, satu respons verbal memerlukan rangkaian inferensi tambahan di luar proses STT dan NLU yang telah ada. Akumulasi beberapa tahap inferensi berurutan tersebut berpotensi meningkatkan *Round-Trip Time (RTT)* secara signifikan serta menambah konsumsi sumber daya komputasi pada sisi *Server* maupun klien. Oleh karena itu, penelitian ini mengadopsi pendekatan *Baked Audio* yang diproduksi pada fase pra-produksi sehingga *Creation Engine* hanya perlu memanggil berkas suara yang telah tersedia berdasarkan referensi *FormID* yang dikirimkan oleh *Server*. Pendekatan ini dipilih karena mampu menghasilkan respons verbal yang konsisten, mempertahankan karakteristik suara Serana secara stabil, serta menghilangkan latensi tambahan yang umumnya muncul pada *pipeline TTS* dan *Voice conversion real-time*. Alur kerja pra-produksi dari rekayasa audio generatif ini diilustrasikan pada Gambar 4.4.



Gambar 4. 4 Pipeline Pra-Produksi Voice Cloning



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tahapan pra-produksi diawali dengan ekstraksi berkas suara asli karakter Serana secara langsung dari basis data bawaan (vanilla) *game engine*. Karena aset audio bawaan tersebut telah memiliki kualitas akustik yang murni dan terisolasi, tahapan prapemrosesan pembersihan noise reduction ditiadakan. Kumpulan data rekaman murni berdurasi 14 menit 48 detik tersebut ditetapkan sebagai Dataset Vokal Primer. Data latih ini dialokasikan untuk memfasilitasi komputasi pelatihan model Retrieval-based Voice conversion (RVC v2) menggunakan unit pemroses grafis berkinerja tinggi pada infrastruktur komputasi *cloud* RunPod guna menangkap parameter karakteristik vokal unik dari karakter target secara komprehensif.

Pada tahap produksi, kerangka kerja otonomi membutuhkan 424 baris variasi respons verbal untuk memfasilitasi pemetaan klasifikasi intent (*Intent*) pengguna. Sintesis awal dari baris dialog tersebut digenerasikan menggunakan layanan ElevenLabs guna membangun struktur intonasi dasar, dinamika emosi, dan penekanan silabel yang natural. Keluaran audio dari tahap sintesis ini kemudian diumpankan ke dalam model RVC v2 yang telah dilatih sebelumnya untuk mengeksekusi tahapan timbre mapping. Pemetaan parameter pita suara ini secara matematis mengubah karakteristik vokal bawaan ElevenLabs menjadi identik dengan rekaman pada Dataset Vokal Primer.

Sebagai batasan mutlak perancangan arsitektur *game engine*, seluruh berkas luaran dari proses timbre mapping ini dirancang agar dikonfigurasi ke dalam format WAV dengan spesifikasi saluran tunggal (Mono), kedalaman bit (bit depth) sebesar 16-bit, dan frekuensi sampel (*sample rate*) tepat di angka 44100 Hz. Standarisasi format ini dirancang agar berkas statis yang diintegrasikan ke dalam direktori modifikasi komputer klien dapat dibaca dan diputar secara langsung oleh Creation Engine berdasarkan parameter kode FormID yang ditransmisikan oleh *Server*, tanpa menimbulkan anomali komputasi suara.

4.3 Implementasi Sistem

Tahap implementasi sistem merupakan manifestasi fisik dan realisasi teknis dari perancangan arsitektur yang telah diuraikan pada sub-bab sebelumnya. Pada tahapan ini, seluruh kerangka konseptual diterjemahkan menjadi barisan instruksi

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

kode program (*source code*), konfigurasi infrastruktur perangkat keras dan perangkat lunak, serta penggabungan modul-modul komputasi terdistribusi.

Proses eksekusi dan integrasi dalam penelitian ini dibagi ke dalam lima ruang lingkup utama. Tahap pertama difokuskan pada implementasi pemrosesan suara di sisi klien lokal (*Client-STT*). Tahap kedua membedah eksekusi rekayasa data dan pelatihan model pemahaman bahasa yang dilakukan di lingkungan komputasi *cloud*. Tahap ketiga menguraikan peluncuran antarmuka pemrograman aplikasi (API) pada infrastruktur *Server* jarak jauh. Tahap keempat menjelaskan penanaman skrip motorik dan jembatan data ke dalam modifikasi mesin gim (*game engine*). Tahap terakhir mencakup integrasi ratusan berkas audio generatif ke dalam direktori permainan. Keseluruhan proses implementasi ini didokumentasikan secara empiris melalui pemaparan potongan kode riil dan penetapan parameter konfigurasi teknis yang dieksekusi selama masa pengembangan perangkat lunak.

4.3.1 Implementasi Pemrosesan Suara Klien (*Client-STT*)

Pemrosesan suara pada sisi klien dijalankan secara lokal di komputer pengguna menggunakan skrip Python mandiri bernama `stt_client.py`. Perangkat lunak ini memanfaatkan pustaka `sounddevice` dan `soundfile` untuk mengelola aliran audio mentah, pustaka `keyboard` untuk mendeteksi interaksi tombol fisik, serta pustaka `faster_whisper` sebagai mesin pengenalan suara otomatis atau *Automatic Speech Recognition* (ASR).

Subsistem *Client-STT* dirancang untuk berjalan secara asinkron dengan loop utama gim *Skyrim* agar tidak menyebabkan penurunan *frame rate* saat menangkap suara dan melakukan inferensi. Sebelum kode dieksekusi, konfigurasi audio dan penanganan masukan dikunci menggunakan konstanta global, dengan rincian parameter yang tertera pada Tabel 4.5.

Tabel 4. 5 Parameter Konfigurasi Audio dan Variabel Global *Client-STT*

Variabel / Parameter	Nilai Konfigurasi	Tipe Data	Deskripsi Teknis
RATE	16000	Integer	Frekuensi sampel (<i>sampling rate</i>) audio dalam satuan Hz yang sesuai dengan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Variabel / Parameter	Nilai Konfigurasi	Tipe Data	Deskripsi Teknis
			arsitektur masukan model <i>Transformer</i> .
CHANNELS	1	Integer	Saluran masukan tunggal (<i>Mono</i>) untuk mereduksi ukuran berkas <i>buffer</i> audio luring.
PTT_KEY	'x'	String / Char	Tombol pintas (<i>hotkey</i>) bawaan (<i>default</i>) pada <i>keyboard</i> sebagai pemicu penangkapan rekaman suara.
model_size	"small.en"	String	Varian bobot (<i>weight</i>) arsitektur model Whisper yang dimuat secara lokal.
device	"cpu"	String	Target unit pemrosesan perangkat keras untuk inferensi model transkripsi.
compute_type	"int8"	String	Format presisi matematis untuk menerapkan kuantisasi bobot model.

Langkah awal pada implementasi kode berfokus pada penentuan jalur direktori dinamis (*dynamic pathing*) untuk menjembatani data teks hasil transkripsi ke dalam sistem memori gim menggunakan metode pengiriman data Atomic Drop. Implementasi ini mutlak membutuhkan pelacakan lokasi absolut skrip klien secara otomatis menggunakan modul *os* bawaan Python, sehingga penulisan luaran data dapat dipaksa menuju folder penyimpanan *StorageUtilData* milik modul *Skyrim Script Extender (SKSE)* tanpa bergantung pada letak instalasi direktori gim di komputer pengguna. Potongan kode pengalamatan dinamis ini ditunjukkan pada Kode Program 4.1.

Kode Program 4. 1 Pengalamatan Direktori Dinamis dan Inisialisasi Jalur Data

```

1. # Kode 4.1 Pengalamatan Direktori Dinamis dan Inisialisasi Jalur Data
2. current_dir = os.path.dirname(os.path.abspath(__file__))
3. SKYRIM_MOD_DIR = os.path.join(current_dir, "..", "SKSE", "Plugins",
  "StorageUtilData")
4.
5. if not os.path.exists(SKYRIM_MOD_DIR):
6.     os.makedirs(SKYRIM_MOD_DIR)

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
7.  
8. TEMP_FILE_PATH = os.path.join(SKYRIM_MOD_DIR, "TempCommand.json")  
9. FINAL_FILE_PATH = os.path.join(SKYRIM_MOD_DIR, "SeranaCommand.json")
```

Berdasarkan Kode Program 4.1, alur penulisan dirancang berlapis menggunakan dua berkas JavaScript Object Notation (JSON) yang berbeda, yaitu TempCommand.json sebagai wadah penulisan sementara, dan SeranaCommand.json sebagai berkas final. Pemisahan ini merupakan inti dari implementasi teknik Atomic Drop, di mana skrip Python melakukan proses penulisan muatan data secara utuh pada berkas sementara terlebih dahulu, kemudian melakukan operasi penggantian nama berkas (file renaming) secara instan ke berkas final. Operasi penggantian nama berkas (rename) dipilih karena pada sebagian besar sistem berkas modern proses tersebut dieksekusi secara atomik, sehingga *game engine* hanya akan menemukan salah satu kondisi, yaitu berkas lama atau berkas baru yang telah selesai ditulis sepenuhnya. Mekanisme ini menjamin bahwa *game engine* Creation Engine tidak akan pernah membaca berkas JSON yang setengah tertulis atau korup akibat adanya tumpang tindih waktu akses memori (race condition).

Setelah jalur penyimpanan dipastikan tervalidasi oleh sistem operasi, tahapan berikutnya adalah melakukan pemuatan dan inisialisasi mesin kecerdasan buatan *Speech-to-Text* (STT) ke dalam memori lokal. Bobot model dari varian small.en diurai secara terprogram dengan memberlakukan restriksi komputasi murni berbasis prosesor melalui instruksi yang dimanifestasikan pada Kode 4.2.

Kode Program 4. 2 Inisialisasi Model Faster-Whisper Terkuantisasi INT8 berbasis CPU

```
1. # Kode 4.2 Inisialisasi Model Faster-Whisper Terkuantisasi INT8 berbasis CPU  
2. print(" [SYSTEM] Initializing Local STT Engine (Loading Whisper Model)...")  
3. model_size = "small.en"  
4. model = WhisperModel(model_size, device="cpu", compute_type="int8")  
5. print(" [SYSTEM] Audio input ready. Awaiting Voice command.")
```

Potongan kode pada Kode 4.2 memperlihatkan penerapan parameter `device="cpu"` dan `compute_type="int8"`. Injeksi parameter kuantisasi INT8 ini secara matematis mengonversi nilai bobot floating-point 32-bit menjadi representasi bilangan bulat 8-bit yang jauh lebih ringkas. Operasi ini berhasil menekan kebutuhan kapasitas *Random Access Memory* klien serta mempercepat waktu komputasi inferensi



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

transkripsi pada tingkat lokal, GPU diprioritaskan untuk kebutuhan perenderan visual gim Skyrim sehingga proses transkripsi diimplementasikan menggunakan CPU guna menghindari kompetisi sumber daya grafis selama permainan berlangsung.

Guna menangkap masukan vokal pengguna, skrip menerapkan fungsi penangkap aliran audio yang dikombinasikan dengan deteksi status penekanan tombol keyboard secara *real-time*. Fungsi `record_audio()` diimplementasikan untuk mengeksekusi perekaman secara dinamis menggunakan mekanisme *Push-to-Talk*, seperti yang dijabarkan pada potongan fungsi di Kode Program 4.3.

Kode Program 4.3 Implementasi Fungsi Perekaman Audio Berbasis *Push-to-Talk*

```
1. # Kode 4.3 Implementasi Fungsi Perekaman Audio Berbasis Push-to-Talk
2. def record_audio():
3.     print(f"\n[INFO] Hold the '{PTT_KEY}' key to speak, release to process...")
4.     keyboard.wait(PTT_KEY)
5.     print(" [RECORDING] Listening...")
```

Melalui implementasi fungsi pada Kode 4.3, periferal mikrofon dikondisikan berada dalam status siaga (idle) dan hanya akan melakukan blok perekaman aktif ketika pengguna menekan dan menahan tombol 'x'. Aliran data audio ditangkap dalam bentuk pecahan matriks (chunks) dengan kedalaman bit linear PCM 16-bit, lalu digabungkan kembali menggunakan fungsi pustaka numpy sebelum diekspor menjadi berkas fisik `temp_Voice.wav`. Berkas audio biner luring inilah yang kemudian diumpankan secara langsung ke fungsi inferensi `model.transcribe()` milik *Faster-Whisper*. Fungsi transkripsi turut mengirimkan muatan teks dari berkas luar `vocabulary.txt` ke dalam parameter `initial_prompt`, yang secara empiris meminimalkan galat kegagalan deteksi istilah asing atau *Out of Vocabulary* (OOV).



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
[INFO] Tahan tombol 'x' untuk bicara, lepaskan untuk memproses...
Mendengarkan... (Sedang Merekam)
Memproses suara...
Dragonborn: "Kill that wolf"
Mengirim ke Otak Serana (NLU) di Jakarta...
NLU Sukses!

-----
Breakdown Waktu:
- Waktu STT (Whisper) : 1396.05 ms
- Waktu NLU (API)      : 172.01 ms
TOTAL TRUE LATENCY    : 1568.06 ms
-----
Payload NLU berhasil ditembakkan ke Skyrim!
Target: SeranaCommand.json

-----
[INFO] Tahan tombol 'x' untuk bicara, lepaskan untuk memproses...
Mendengarkan... (Sedang Merekam)
Memproses suara...
Dragonborn: "Use your magic"
Mengirim ke Otak Serana (NLU) di Jakarta...
NLU Sukses!

-----
Breakdown Waktu:
- Waktu STT (Whisper) : 1427.56 ms
- Waktu NLU (API)      : 225.03 ms
TOTAL TRUE LATENCY    : 1652.59 ms
-----
Payload NLU berhasil ditembakkan ke Skyrim!
Target: SeranaCommand.json
-----
```

Gambar 4. 5 Eksekusi Terminal *Client-STT* saat Proses Transkripsi Konteks Konfigurasi Lokal Klien

Hasil luaran akhir dari seluruh rangkaian implementasi fungsional pada sisi klien ini adalah string teks hasil konversi ucapan yang telah dibersihkan dari simbol-simbol non-alfabetis menggunakan ekspresi reguler (Regular Expression). String teks bersih tersebut selanjutnya akan dilemparkan keluar dari subsistem lokal klien sebagai sebuah muatan data parameter menuju titik akhir REST API di lingkungan *Server Cloud* melalui perantara pustaka *Requests*.

4.3.2 Implementasi *Data Engineering* dan Pelatihan NLU

Implementasi komponen kecerdasan buatan pada penelitian ini dibagi menjadi tiga fase berurutan yang dieksekusi di dalam lingkungan komputasi *cloud* platform Google Colab. Ketiga fase tersebut meliputi rekayasa dan augmentasi data, proses pelatihan serta evaluasi model *Multi-Task Learning* RoBERTa-base, hingga kompilasi dan kuantisasi model ke format ONNX *Runtime*.

4.3.2.1 Rekayasa Data dan *Augmentasi*

Tahap awal implementasi NLU berfokus pada penyusunan dataset perintah pemain yang sesuai dengan konteks permainan *The Elder Scrolls V: Skyrim*. Dataset dibangun dari kombinasi kamus entitas dan template kalimat dasar. Kamus entitas mencakup kategori seperti lokasi, karakter, musuh, dan objek, sedangkan template kalimat disusun berdasarkan kelas *intent* dan *sentiment*. Struktur awal kamus dan template tersebut ditampilkan pada Kode Program 4.4.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kode Program 4. 4 Implementasi Kamus Entitas dan Kerangka Kalimat Dasar

```
1. # Kode 4.4 Implementasi Kamus Entitas dan Kerangka Kalimat Dasar
2. entities = {
    "location": ["Solitude", "Whiterun", "Riften"],
    "enemy": ["bandit", "dragon", "draugr"],
    "object": ["door", "lever", "chest"]
}

templates = [
    "Serana, go to {location}",
    "Serana, attack the {enemy} in Solitude",
    "Serana, open the {object}"
]
```

Dari kerangka tersebut, algoritma berbasis Regular Expression digunakan untuk mengekstraksi parameter dalam tanda kurung siku [...], lalu menggantinya dengan entitas acak dari kamus yang telah disusun. Proses ini juga menghasilkan tokenisasi otomatis dan pelabelan Named Entity Recognition (NER) menggunakan format BIO, yaitu B- (Begin) dan I- (Inside), sehingga keselarasan antara token dan tag tetap terjaga. Tahap ini menghasilkan lebih dari 300 baris data awal yang tervalidasi dan bebas dari kesalahan format.

Namun, jumlah tersebut belum mencukupi untuk menghasilkan variasi linguistik yang memadai bagi proses fine-tuning model Transformer. Oleh karena itu, data awal diperluas menggunakan model gemini-2.5-flash melalui API dengan konfigurasi temperature 0.7 dan keluaran berformat application/json. Proses augmentasi dilakukan menggunakan teknik Few-Shot Prompting dengan memanfaatkan sampel acak dari data awal sebagai referensi. Melalui mekanisme ini, sistem berhasil menghasilkan lebih dari 3.000 baris data baru yang lebih variatif, tetap konsisten dengan label intent, sentiment, dan tag NER, serta tetap sesuai dengan konteks permainan. Implementasi konfigurasi dan instruksi augmentasi ditampilkan pada Kode Program 4.5.

Kode Program 4. 5 Konfigurasi LLM gemini-2.5-flash dan Injeksi Few-Shot Prompting

```
1. # Kode 4.5 Konfigurasi LLM gemini-2.5-flash dan Injeksi Few-Shot Prompting
2. import google.generativeai as genai
3. from google.colab import Userdata
4. import json, random
5. # Inisialisasi API & Konfigurasi Model
6. api_key = Userdata.get('Gemini_API_Arjuna')
7. genai.configure(api_key=api_key)
8. model = genai.GenerativeModel(
9.     model_name='gemini-2.5-flash',
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

10. generation_config=genai.GenerationConfig(
11.     temperature=0.7,
12.     response_mime_type="application/json"
13. )
14. )
15. # Fungsi Generator Prompt (Mendukung Few-Shot Prompting & Injeksi Sentimen)
16. def get_dynamic_prompt(target_intent, target_sentiment, batch_size):
17.     # Penarikan acak seed data sebagai referensi Few-Shot
18.     samples = random.sample(seed_data, min(3, len(seed_data)))
19.     seed_examples_text = "\n[FEW-SHOT EXAMPLES FOR THIS INTENT]\n" +
20.         json.dumps(samples, indent=2)
21.     return f"""
22. Act as an Elite NLP Data Architect and Skyrim Lore Scholar.
23. Task: Generate EXACTLY {batch_size} unique NLU data rows for a Serana AI Mod in
24. a pure JSON Array format.
25. [CRITICAL INSTRUCTION]
26. 1. TARGET INTENT: Every row MUST have the "Intent" value EXACTLY as:
27. "{target_intent}".
28. 2. TARGET SENTIMENT: Every row MUST have the "sentiment" value EXACTLY as:
29. "{target_sentiment}".
30. [STRICT V5 CONSTRAINTS]
31. 1. LANGUAGE: 100% English. DO NOT output Indonesian.
32. 2. PUNCTUATION AS TOKENS: Periods (.), commas (,), question marks (?),
33. exclamation marks (!) MUST be separate tokens with "O" tags.
34. 3. NER TAGS ("The Golden Four"): ONLY use: "O", "B-Location", "I-Location", "B-
35. Character", "I-Character", "B-Enemy", "I-Enemy", "B-Item", "I-Item".
36. {seed_examples_text}
37. """

```



Gambar 4. 6 Hasil Eksekusi Generasi Data dan Distribusi EDA

Seluruh data generatif divalidasi oleh pengembang mod Skyrim yang aktif di Nexus Mods dan berpengalaman dalam pengembangan konten berbasis Creation Engine. Validasi ini dilakukan agar struktur kalimat yang dihasilkan tetap natural serta konsisten dengan karakter dan konteks permainan.

Data yang telah lolos validasi kemudian diproses melalui tahap pembersihan diagnostik (Drop Protocol). Tahap ini mengevaluasi integritas setiap baris data dengan menghapus anomali, seperti ketidaksesuaian jumlah token dan tag NER, kesalahan format BIO, serta ketiadaan entitas wajib pada kelas intent tertentu, misalnya entitas Item pada intent Inventory_Trade.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Sebagai tahap akhir, dilakukan augmentasi lowercase untuk menyesuaikan data dengan kemungkinan keluaran dari STT Whisper. Sebanyak 40% data latih dikloning dan dikonversi menjadi huruf kecil. Dataset akhir kemudian dibagi menggunakan stratified split dengan rasio 80% data latih dan 20% data uji, lalu disimpan sebagai `v5_train_data_cased_aug.json` dan `v5_test_data.json` untuk digunakan dalam pelatihan RoBERTa-base.

4.3.2.2 Pelatihan Model RoBERTa-base

Tahap utama pengembangan model NLU dilakukan melalui proses fine-tuning RoBERTa-base pada lingkungan komputasi cloud berbasis GPU. Model dirancang dengan pendekatan Multi-Task Learning untuk menghasilkan tiga keluaran secara simultan, yaitu intent, sentiment, dan Named Entity Recognition (NER).

Data dilatih menggunakan tokenizer RoBERTa berbasis Byte-Level BPE dengan parameter `add_prefix_space=True`. Label hanya ditempatkan pada sub-kata pertama, sedangkan sub-kata lain dan special tokens diberi label -100 agar diabaikan dalam perhitungan fungsi kerugian (loss function).

Class weights pada label *intent* dihitung menggunakan scikit-learn untuk mengurangi dampak ketidakseimbangan data. Bobot tersebut dikonversi ke tensor PyTorch dan digunakan dalam proses pelatihan. Struktur utama arsitektur model ditampilkan pada Kode Program 4.6.

Kode Program 4. 6 Implementasi Multi-Task Learning dan Loss Function

```
1. # Kode 4.6 Implementasi Arsitektur Multi-Task Learning dan Fungsi Loss Function
2. class SeranaRobertaMTL(RobertaPreTrainedModel):
3.     def __init__(self, config, num_Intents, num_sentiments, num_ner_tags,
Intent_weights):
4.         super().__init__(config)
5.         self.num_Intents = num_Intents
6.         self.num_sentiments = num_sentiments
7.         self.num_ner_tags = num_ner_tags
8.         # Ekstraktor fitur (Backbone)
9.         self.roberta = RobertaModel(config)
10.        self.dropout = nn.Dropout(config.hidden_dropout_prob)
11.        # Inisialisasi 3 linear layer (kepala klasifikasi)
12.        self.Intent_classifier = nn.Linear(config.hidden_size, num_Intents)
13.        self.sentiment_classifier = nn.Linear(config.hidden_size,
num_sentiments)
14.        self.ner_classifier = nn.Linear(config.hidden_size, num_ner_tags)
15.        self.Intent_weights = Intent_weights
16.        self.post_init()
17.        def forward(self, input_ids=None, attention_mask=None, Intent_labels=None,
sentiment_labels=None, ner_labels=None):
18.            outputs = self.roberta(input_ids=input_ids,
attention_mask=attention_mask)
19.            sequence_output = outputs[0]
20.            # Representasi kalimat (token <s>) untuk Intent & Sentiment
21.            pooled_output = self.dropout(sequence_output[:, 0, :])
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

22.     Intent_logits = self.Intent_classifier(pooled_output)
23.     sentiment_logits = self.sentiment_classifier(pooled_output)
24.     # Representasi sekuensial penuh untuk NER
25.     ner_logits = self.ner_classifier(self.dropout(sequence_output))
26.     loss = None
27.     if Intent_labels is not None and sentiment_labels is not None and
ner_labels is not None:
28.         loss_fct	Intent = nn.CrossEntropyLoss(weight=self.Intent_weights)
29.         loss_fct_sent = nn.CrossEntropyLoss()
30.         l1 = loss_fct	Intent(Intent_logits, Intent_labels)
31.         l2 = loss_fct_sent(sentiment_logits, sentiment_labels)
32.         # Manipulasi bobot NER
33.         ner_weights = torch.ones(self.num_ner_tags,
device=ner_logits.device) * 2.0
34.         ner_weights[ner2id.get("O", 0)] = 0.5
35.
36.         loss_fct_ner = nn.CrossEntropyLoss(ignore_index=-100,
weight=ner_weights)
37.         l3 = loss_fct_ner(ner_logits.view(-1, self.num_ner_tags),
ner_labels.view(-1))
38.         loss = l1 + l2 + (2.5 * l3)
39.         return {"loss": loss, "Intent_logits": Intent_logits,
"sentiment_logits": sentiment_logits, "ner_logits": ner_logits}

```

Implementasi yang ditunjukkan pada Kode Program 4.6 mengungkap rekayasa matematika secara ekstrem pada kalkulasi CrossEntropyLoss. Bobot klasifikasi token "O" (yang bukan entitas) ditekan secara artifisial menjadi 0.5 untuk mendorong model agar lebih agresif dalam memprediksi entitas yang valid. Selanjutnya, nilai akhir galat klasifikasi NER (l3) dikalikan dengan konstanta 2.5 sebelum diakumulasikan ke total loss. Manipulasi ini memaksa jaringan arsitektur untuk memprioritaskan konvergensi akurasi pengenalan entitas, mengingat tugas linguistik tersebut adalah variabel paling krusial untuk mencegah terjadinya kerusakan sinkronisasi pada komponen *game engine*.

Siklus penyesuaian bobot dieksekusi menggunakan modul Trainer dari ekosistem Hugging Face. Mengakomodasi skema pembobotan kustom yang memperlambat penurunan galat, jumlah perulangan iteratif disesuaikan bersama dengan parameter pelatihan (hyperparameters) empiris lainnya yang disematkan ke dalam modul TrainingArguments. Konfigurasi teknis dari tahap pelatihan tersebut dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Parameter dan Konfigurasi Model

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Parameter Pelatihan	Nilai Konfigurasi	Deskripsi Teknis
learning_rate	2e-5	Tingkat pembelajaran konservatif untuk mencegah pembaruan bobot dasar yang terlalu drastis (<i>catastrophic forgetting</i>).
num_train_epochs	10	Jumlah siklus penuh model mengamati keseluruhan himpunan data latih. Dinaikkan untuk mengakomodasi perlambatan konvergensi akibat perkalian galat NER.
per_device_train_batch_size	16	Jumlah sampel data yang diproses ke dalam memori GPU pada setiap langkah perambatan balik (<i>backpropagation</i>).
weight_decay	0.05	Teknik regularisasi untuk memberikan penalti pada bobot parameter yang terlalu

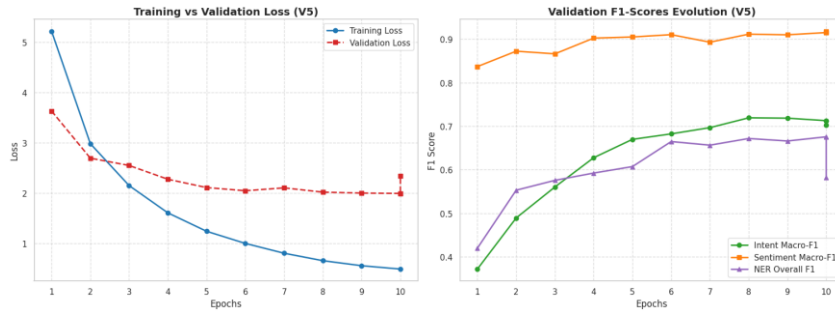
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Parameter Pelatihan	Nilai Konfigurasi	Deskripsi Teknis
		ekstrem guna menekan risiko <i>overfitting</i> .
metric_for_best_model	"eval_ner_f1"	Metrik acuan utama untuk mengevaluasi kualitas performa prediksi, difokuskan secara khusus pada skor <i>F1</i> komponen ekstraksi entitas.
EarlyStoppingCallback	early_stopping_patience=3	Proses training akan dihentikan secara otomatis sebelum waktunya selesai jika performa model pada eval_dataset (misalnya nilai loss-nya) tidak membaik selama 3 epoch berturut-turut

Evaluasi performa akhir tidak terbatas pada himpunan data uji standar (bawaan huruf kapital atau *cased*), tetapi diuji secara simultan menggunakan *Adversarial STT Test Set*. Seluruh huruf pada himpunan pengujian sekunder tersebut dikonversi paksa menjadi huruf kecil (*lowercase*) guna menyimulasikan luaran mesin transkripsi vokal luring secara riil. Hasil pengujian menunjukkan nilai akurasi evaluasi untuk NER (Macro F1) mencapai angka optimal yang dipertahankan melalui implementasi penghentian dini otomatis (*Early Stopping*), dengan metrik evaluasi yang disajikan pada Gambar 4.7.



Gambar 4. 7 Metrik Pelatihan, Kurva Loss, dan Akurasi dari Terminal Google Colab

```

Making Serana up from her slumber (RoBERTa Engine)...
Loading weights: 0% | 0/205 [00:00<, ?it/s]

Smart Inference System Ready!

[Text Mode] Dragonborn: "We need to defeat Alduin at the Throat of the World!"
-----
Predicted Intent : Quest_Objective
Predicted Sentiment : Neutral
Extracted Entities :
[Character] : Alduin
[Location] : at the
[Location] : Throat of the World
-----

[STT Mode] Dragonborn: "we need to defeat alduin at the throat of the world!"
-----
Predicted Intent : Quest_Objective
Predicted Sentiment : Neutral
Extracted Entities :
[Location] : the
[Location] : throat of the world
-----

[Text Mode] Dragonborn: "hey Serana, can you carry these Dragon Bones and the Dambreaker for me? I am carrying too much."
-----
Predicted Intent : Inventory_Trade
Predicted Sentiment : Negative
Extracted Entities :
[Character] : Serana
[Character] : Dragon
[Item] : Bones
[Item] : the
[Item] : Dambreaker
-----

[STT Mode] Dragonborn: "hey serana, can you carry these dragon bones and the dambreaker for me? i am carrying too much."
-----
Predicted Intent : Inventory_Trade
Predicted Sentiment : Positive
Extracted Entities :
[Character] : serana
[Character] : dragon
[Item] : dambreaker
-----

```

Gambar 4. 8 Black-Box Inference Testing pada Text Mode dan Simulasi STT Lowercase

Gambar 4.8 menunjukkan hasil pengujian Black-Box Inference Testing menggunakan dua skenario masukan yang identik, yaitu Text Mode (cased) dan STT Mode (lowercase). Pengujian ini dirancang untuk mensimulasikan keluaran nyata mesin transkripsi Faster-Whisper yang umumnya tidak mempertahankan kapitalisasi secara konsisten. Hasil pengujian memperlihatkan bahwa model tetap mampu mempertahankan prediksi *Intent* pada kelas yang benar meskipun seluruh huruf diubah menjadi huruf kecil. Temuan ini mengindikasikan bahwa proses *Fine-Tuning* berhasil meningkatkan ketahanan model terhadap variasi format teks yang umum ditemukan pada sistem *Speech-to-Text*.

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.3.2.3 Optimasi ONNX dan Kuantisasi Model

Tahap akhir pada AI engineering pipeline difokuskan pada optimasi model agar lebih efisien untuk dijalankan pada lingkungan Virtual Private Server (VPS). Model awal Serana_RoBERTa_Final dalam format PyTorch memiliki ukuran sekitar 500 MB, sehingga kurang ideal untuk inferensi interaktif pada server berspesifikasi terbatas. Oleh karena itu, dilakukan optimasi model melalui tiga tahap, yaitu konversi ke format ONNX FP32, pembersihan graf komputasi (graph cleanup), dan kuantisasi dinamis INT8.

Langkah pertama dilakukan dengan mengekspor model PyTorch ke format ONNX menggunakan fungsi `torch.onnx.export()`. Proses ekspor ini merekam alur komputasi dari tiga keluaran utama model, yaitu klasifikasi intent, sentiment, dan NER. Teks tiruan sebagai dummy input untuk memicu proses graph tracing. Meskipun teks tersebut dapat berbentuk kalimat kompleks, model tetap bekerja dengan batasan single-intent, yaitu memetakan satu masukan teks ke satu kelas intent utama berdasarkan nilai logit tertinggi.

Agar model dapat menerima variasi panjang kalimat saat digunakan oleh klien, parameter `dynamic_axes` diterapkan pada dimensi batch size dan sequence length. Dengan konfigurasi tersebut, model ONNX dapat memproses instruksi suara yang memiliki panjang token berbeda tanpa harus menggunakan bentuk input yang sepenuhnya statis. Potongan proses ekspor model ditampilkan pada Kode Program 4.7.

Kode Program 4. 7 Kompilasi Arsitektur Multi-Task PyTorch ke Grafik ONNX FP32

```
1. # Kode 4.7 Kompilasi Arsitektur Multi-Task PyTorch ke Grafik ONNX FP32
2. import torch
3. import torch.onnx
4.
5. print(" Memulai Proses Optimasi: Ekspor ONNX FP32...")
6. torch.onnx.export(
7.     model,
8.     (input_ids, attention_mask),
9.     onnx_model_path,
10.    export_params=True,
11.    opset_version=18,
12.    do_constant_folding=False,
13.    input_names=['input_ids', 'attention_mask'],
14.    output_names=['Intent_logits', 'sentiment_logits', 'ner_logits'],
15.    dynamic_axes={
16.        'input_ids': {0: 'batch_size', 1: 'sequence_length'},
17.        'attention_mask': {0: 'batch_size', 1: 'sequence_length'},
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
18.         'Intent_logits': {0: 'batch_size'},
19.         'sentiment_logits': {0: 'batch_size'},
20.         'ner_logits': {0: 'batch_size', 1: 'sequence_length'}
21.     }
22. )
```

Eksekusi ekspor pada Kode 4.7 menggunakan target operator set versi 18 (opset_version=18) untuk memastikan keselarasan operator matematika tingkat tinggi dari arsitektur RoBERTa. Meskipun operasi ini berhasil mengonversi model menjadi berkas biner tunggal Serana_RoBERTa.onnx, pelacakan internal membuktikan bahwa konverter bawaan PyTorch secara otomatis mengirimkan metadata bentuk antara (intermediate format shape info) yang cacat ke dalam struktur data grafis. Jika grafik yang korup ini dipaksa masuk ke tahap kuantisasi, mesin akan memicu kegagalan sistem (AttributeError).

Mengatasi kutukan galat tersebut, skrip mengimplementasikan algoritma intervensi khusus yang dinamakan Graph Healer. Pustaka onnx dimuat secara luring dengan mengaktifkan parameter load_external_data=False guna mengisolasi dan membedah kerangka logika tanpa menyentuh berkas bobot mentah .data. Operasi pembersihan ini memotong dan mengosongkan seluruh larik metadata yang rusak dari objek grafik secara terprogram, seperti yang ditunjukkan pada Kode Program 4.8.

Kode Program 4. 8 Implementasi Operasi Pembersihan Struktur Grafis (Graph Healer)

```
1. # Kode 4.8 Implementasi Operasi Pembersihan Struktur Grafis (Graph Healer)
2. import onnx
3. print("1. Membaca arsitektur model FP32...")
4. onnx_model = onnx.load(onnx_fp32_path, load_external_data=False)
5.
6. print(f"2. Menghapus {len(onnx_model.graph.value_info)} metadata intermediate
yang cacat dari PyTorch...")
7. # Menghapus larik metadata shape yang berkonflik secara mutlak
8. del onnx_model.graph.value_info[:]
9. print("3. Menyimpan ulang model yang sudah disembuhkan...")
10. onnx.save(onnx_model, onnx_fp32_path)
11. print("\n [SUCCESS] Graf ONNX sudah bersih dari bug PyTorch!")
```

Setelah grafik komputasi dipulihkan sepenuhnya oleh fungsi Graph Healer pada Kode Program 4.8, tahapan kritis selanjutnya adalah kompresi tingkat lanjut melalui metode kuantisasi dinamis (INT8 Dynamic Quantization). Fungsi quantize_dynamic dari modul onnxRuntime.quantization dipanggil untuk mengonversi bobot parameter bertipe data pecahan presisi tunggal (Float32) menjadi bilangan bulat bertanda 8-bit (QuantType.QInt8). Skenario kuantisasi ini

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dikonfigurasi dengan parameter `use_external_data_format=False` agar hasil kompresi dibungkus rapat ke dalam satu berkas tunggal terpadu berformat `Serana_RoBERTa_INT8.onnx` tanpa meninggalkan pecahan berkas bobot terpisah, mempermudah *Deployment* modular ke infrastruktur *VPS Cloud*.

Dampak dari keseluruhan rangkaian rekayasa optimasi grafik komputasi ini sangat signifikan secara matematis. Hasil tabulasi komparasi dimensi tapak memori (memory footprint) sebelum dan sesudah optimasi dirangkum secara komprehensif pada Tabel 4.7.

Tabel 4. 7 Perbandingan Metrik Ukuran Berkas dan Efisiensi Kompresi Model NLU

Status Representasi Model	Format Ekstensi Berkas	Ukuran Berkas Fisik (MB)	Rasio Kompresi	Persentase Penyusutan Ukuran
PyTorch Baseline / ONNX FP32	.safetensors / .onnx	475.60 MB	Baseline (1.00x)	0.00%
Optimized ONNX <i>Runtime</i> (V5.1)	.onnx (INT8)	118.91 MB	4.00x Lebih Kecil	74.99%

Berdasarkan data empiris pada Tabel 4.7, proses kuantisasi berhasil memotong ukuran bobot model sebesar 74,99% dari dimensi awal, menyusutkan kapasitas dari 475.60 MB menuju titik ideal di angka 118.91 MB dengan rasio pemangkasan tepat 4 kali lipat. Pemangkasan ruang ini secara drastis menurunkan kebutuhan alokasi memori acak (RAM) pada proksi *Server* luring *VPS*, sehingga menjamin proses inferensi dapat berjalan di bawah ambang batas latensi respons interaktif gim.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

[2040/2040 06:18, Epoch 10/10]											
Epoch	Training Loss	Validation Loss	Intent Accuracy	Intent F1	Sentiment Accuracy	Sentiment F1	Ner Precision	Ner Recall	Ner F1	Ner Accuracy	
1	5.218727	3.631418	0.403087	0.371810	0.838765	0.837046	0.327295	0.585313	0.419830	0.899011	
2	2.982302	2.694251	0.499142	0.489204	0.873070	0.872386	0.467862	0.676026	0.553004	0.933080	
3	2.149545	2.549176	0.576329	0.560413	0.869640	0.866329	0.527928	0.632829	0.575639	0.948289	
4	1.607078	2.275073	0.629503	0.627674	0.903945	0.902457	0.501497	0.723542	0.592396	0.940837	
5	1.240389	2.109831	0.674099	0.669840	0.905660	0.905064	0.506494	0.758099	0.607266	0.941141	
6	0.999156	2.046725	0.682676	0.682430	0.910806	0.910392	0.593220	0.755940	0.664767	0.955741	
7	0.804011	2.103998	0.691252	0.696774	0.893654	0.893093	0.588034	0.742981	0.656489	0.956958	
8	0.655815	2.018789	0.716981	0.719299	0.912521	0.911505	0.605903	0.753780	0.671800	0.957110	
9	0.556086	2.000081	0.718696	0.718508	0.910806	0.910044	0.586207	0.771058	0.666045	0.955437	
10	0.489474	1.992914	0.708405	0.712792	0.915952	0.915177	0.612281	0.753780	0.675702	0.958935	

Gambar 4. 9 Log Terminal Konsol Google Colab Pasca-Eksekusi INT8 Dynamic Quantization

Pipa optimasi ini ditutup dengan melakukan pengujian kesiapan fungsional menggunakan komponen `onnxRuntime.InferenceSession` berbasis `CPUExecutionProvider`. Larik masukan teks tidak lagi dikonversi menjadi objek tensor PyTorch, melainkan diumpankan dalam bentuk array NumPy murni dengan penegasan tipe data integer 64-bit (`astype(np.int64)`) untuk menghindari galat ketidakcocokan tipe memori (`Type Mismatch Error`). Berkas cerdas yang telah terkompresi secara ekstrem dan tervalidasi secara fungsional ini dinyatakan secara mutlak lolos uji kelayakan sebagai produk model akhir (final model *Deployment*), sehingga siap dipindahkan keluar dari ekosistem Google Colab untuk ditanamkan pada peluncuran layanan komponen backend *Server* luring.

4.3.3 Implementasi *Server Cloud* NLU

Implementasi komponen *Cloud Natural Language Understanding* (NLU) merupakan pilar penghubung utama dalam arsitektur *Hybrid Client-Cloud* sistem interaksi vokal ini. *Server* ini dirancang untuk menerima kiriman transkripsi teks dari klien lokal, memprosesnya melalui grafik inferensi kecerdasan buatan, melakukan pemetaan relasional data, hingga menjatuhkan berkas instruksi *payload* final secara luring ke dalam direktori kerja *game engine*. Seluruh *pipeline* pemrosesan data ini dioperasikan secara mandiri dan terisolasi di dalam infrastruktur penyiapan jaringan backend *Server*.

4.3.3.1 Inisialisasi VPS, Kontainer Docker, dan Framework FastAPI

Infrastruktur cloud pada penelitian ini dijalankan pada Virtual Private Server (VPS) DomaiNesia dengan sistem operasi Ubuntu Server. Server yang digunakan tidak memiliki GPU terdedikasi, sehingga seluruh proses inferensi model Multi-Task dijalankan menggunakan CPU. Kondisi ini menjadi alasan penting dilakukannya

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

optimasi ONNX dan kuantisasi dinamis INT8 pada tahap sebelumnya, agar model dapat berjalan lebih ringan dan tidak membebani RAM server.

Sistem dikemas menggunakan Docker untuk menjaga konsistensi runtime environment dan mempermudah proses deployment. Konfigurasi kontainer didefinisikan melalui berkas Dockerfile dengan basis python:3.10-slim agar ukuran lingkungan aplikasi tetap ringan. Selain itu, kontainer dijalankan menggunakan pengguna non-root serana_User untuk mengurangi risiko keamanan akibat akses administrator yang tidak diperlukan.

Melalui konfigurasi tersebut, aplikasi *Server Cloud NLU* dapat berjalan dalam lingkungan yang terisolasi, lebih mudah direplikasi, dan lebih aman untuk menangani permintaan inferensi dari klien. Struktur konfigurasi Docker ditampilkan pada Kode Program 4.9.

Kode Program 4. 9 Konfigurasi Dockerfile Isolasi *Runtime Environment Server Cloud NLU*

```
1. # Kode 4.9 Konfigurasi Dockerfile Isolasi Runtime Environment Server Cloud NLU
2. FROM python:3.10-slim
3.
4. # Pembuatan low-privilege non-root User untuk mitigasi keamanan siber
5. RUN Useradd -m -s /bin/bash serana_User
6. # Set direktori kerja di dalam kontainer
7. WORKDIR /app
8. # Salin file requirement kebutuhan library terlebih dahulu
9. COPY requirements.txt .
10. # Instalasi dependencies tanpa menggunakan cache untuk menghemat storage VPS
11. RUN pip install --no-cache-dir -r requirements.txt
12. # Salin seluruh sisa file aplikasi Server ke dalam kontainer
13. COPY . .
14. # Mengalihkan hak kepemilikan direktori secara mutlak ke serana_User
15. RUN chown -R serana_User:serana_User /app
16. # Mengubah privilege eksekusi dari root ke pengguna non-administrator
17. USER serana_User
18. # Alokasi port inbound jaringan kontainer
19. EXPOSE 8000
20. # Eksekusi peluncuran Server via Uvicorn dengan multi-workers balancing
21. CMD ["uvicorn", "nlu_Server:app", "--host", "0.0.0.0", "--port", "8000", "--workers", "2"]
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

root@vps-domainesia:/app# docker build -t serana-nlu-cloud:latest .
Sending build context to Docker daemon 342.5MB
Step 1/10 : FROM python:3.10-slim
3.10-slim: Pulling from library/python
7264a8db641e: Pull complete
3b070b0c4cde: Pull complete
71b23f2b4c1: Pull complete
Digest: sha256:4906b65a05c69c6
Status: Downloaded newer image for python:3.10-slim
--> 9a43a0519a77
Step 2/10 : RUN useradd -m -s /bin/bash serana user
--> Running in b5a198cd1d4
Removing intermediate container b5a198cd1d4
--> e223b184231
Step 3/10 : WORKDIR /app
--> Running in 3cb148da39f1
Removing intermediate container 3cb148da39f1
--> c1948ba1207d
Step 4/10 : COPY requirements.txt .
--> a84910cf91b3
Step 5/10 : RUN pip install --no-cache-dir -r requirements.txt
--> Running in f74819ca01db
Collecting fastapi==0.110.0 (from -r requirements.txt (line 1))
  Downloading fastapi-0.110.0-py3-none-any.whl (92 kB)
Collecting uvicorn==0.28.0 (from -r requirements.txt (line 2))
  Downloading uvicorn-0.28.0-py3-none-any.whl (60 kB)
Collecting onnxruntime==1.17.1 (from -r requirements.txt (line 3))
  Downloading onnxruntime-1.17.1-cp310-cp310-manylinux_2_27_x86_64.whl (6.8 MB)
Collecting transformers==4.38.2 (from -r requirements.txt (line 4))
  Downloading transformers-4.38.2-py3-none-any.whl (8.5 MB)
Installing collected packages: uvicorn, onnxruntime, fastapi, transformers
Successfully installed fastapi-0.110.0 uvicorn-0.28.0 onnxruntime-1.17.1 transformers-4.38.2
Removing intermediate container f74819ca01db
--> e3948104fbac
Step 6/10 : COPY .
--> 594b2194ad1e
Step 7/10 : RUN chown -R serana_user:serana_user /app
--> Running in a1839db491ca
Removing intermediate container a1839db491ca
--> fd849102c48
Step 8/10 : USER serana user
--> Running in 728349fa10bc
Removing intermediate container 728349fa10bc
--> 25810471bdc
Step 9/10 : EXPOSE 8080
--> Running in 817392aedbc3
Removing intermediate container 817392aedbc3
--> b28471049dae
Step 10/10 : CMD ["uvicorn", "nlu_server:app", "--host", "0.0.0.0", "--port", "8080", "--workers", "2"]
--> Running in 918374bdcafe
Removing intermediate container 918374bdcafe
--> e3948104fbac
Successfully built e3948104fbac
Successfully tagged serana-nlu-cloud:latest

```

Gambar 4. 10 Langkah eksekusi Docker build dan pembuatan citra kontainer di terminal Ubuntu VPS

Setelah konfigurasi kontainer Docker terbentuk, server utama dijalankan menggunakan framework FastAPI. Pada tahap inisialisasi, beberapa variabel lingkungan diatur melalui `os.environ` untuk menekan peringatan dan log yang tidak diperlukan, seperti `HF_HUB_DISABLE_SYMLINKS_WARNING` dan `TRANSFORMERS_VERBOSITY`. Pengaturan ini dilakukan agar proses server tetap ringan ketika dijalankan pada Ubuntu Server.

Selanjutnya, sistem memuat tiga berkas konfigurasi utama ke dalam memori, yaitu `label_mapping.json`, `fuzzy_targets.json`, dan `papyrus_formids.json`. Berkas `label_mapping.json` digunakan untuk memetakan indeks keluaran model ke label *intent*, *sentiment*, dan NER. Berkas `fuzzy_targets.json` digunakan sebagai leksikon untuk koreksi entitas berbasis *fuzzy matching*, sedangkan `papyrus_formids.json` digunakan untuk menghubungkan entitas hasil NLU dengan FormID pada data gim Skyrim. Setelah itu, model `Serana_RoBERTa_INT8.onnx` dimuat menggunakan ONNX Runtime dengan `CPUExecutionProvider`, sesuai dengan spesifikasi VPS yang menjalankan inferensi berbasis CPU.

Aplikasi FastAPI dijalankan melalui ASGI server Uvicorn agar permintaan HTTP dari klien dapat ditangani secara efisien. Konfigurasi `--workers 2` digunakan agar server memiliki dua proses pekerja paralel, sehingga beberapa permintaan dapat diproses lebih efisien dan tidak seluruhnya menumpuk pada satu proses tunggal.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Rangkaian inialisasi framework, pemuatan konfigurasi, dan penyiapan ONNX Runtime ditampilkan pada Kode Program 4.10.

Kode Program 4. 10 Skrip Inialisasi Environment Jaringan FastAPI dan Pemuatan Engine ONNX INT8

```
1. # Kode 4.10 Skrip Inialisasi Environment Jaringan FastAPI dan Pemuatan Engine
ONNX INT8
2. import os
3. import json
4. import onnxruntime as ort
5. from fastapi import FastAPI
6. from transformers import AutoTokenizer
7. os.environ["TRANSFORMERS_VERBOSITY"] = "error"
8. app = FastAPI(
9.     title="Serana AI - NLU Brain Cloud",
10.    version="5.0"
11.)
12. MODEL_DIR = "./model"
13. tokenizer = AutoTokenizer.from_pretrained(MODEL_DIR)
14. session = ort.InferenceSession(
15.    f"{MODEL_DIR}/Serana_RoBERTa_INT8.onnx",
16.    providers=["CPUExecutionProvider"]
17.)
18. with open(f"{MODEL_DIR}/label_mapping.json", "r") as f:
19.    mapping_data = json.load(f)
20. INTENT_LABELS = {int(k): v for k, v in mapping_data["intent"].items()}
21. SENTIMENT_LABELS = {int(k): v for k, v in mapping_data["sentiment"].items()}
22. NER_LABELS = {int(k): v for k, v in mapping_data["ner"].items()}
```

```
INFO: Started server process [11]
INFO: Waiting for application startup.
INFO: Started server process [10]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Application startup complete.
[SYSTEM] Label Mappings Loaded Successfully.
[SYSTEM] FuzzyWuzzy Lexicon Loaded (172 Entities).
[SYSTEM] Papyrus FormID Lexicon Loaded.
[INIT] Booting RoBERTa Core (Cloud Architecture)...
[INIT] RoBERTa Engine Online and Awaiting Payload.
[API RESPONSE] Intent: Movement_Follow | Audio: 0x03A8E0 | Affinity Shift: 0
INFO: 172.17.0.1:51796 - "POST /chat HTTP/1.0" 200 OK
[SYSTEM] Label Mappings Loaded Successfully.
[SYSTEM] FuzzyWuzzy Lexicon Loaded (172 Entities).
[SYSTEM] Papyrus FormID Lexicon Loaded.
[INIT] Booting RoBERTa Core (Cloud Architecture)...
[INIT] RoBERTa Engine Online and Awaiting Payload.
[API RESPONSE] Intent: Social Status | Audio: 0x03A960 | Affinity Shift: 0
```

Gambar 4. 11 Eksekusi Docker Build Dan Peluncuran Awal FastAPI Pada Terminal VPS Ubuntu DomaiNesia

4.3.3.2 Pembungkusan Skema Payload Inbound via Pydantic BaseModel

Tahap penerimaan data dari komponen game client menuju *Server Cloud* membutuhkan validasi skema agar struktur *payload* tetap sesuai dengan kebutuhan sistem. Pustaka Pydantic digunakan melalui kelas *BaseModel* dengan mendefinisikan objek *ChatRequest*. Skema ini memuat dua properti utama, yaitu

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

text bertipe string sebagai hasil transkripsi dari *Client-STT* dan *affinity_score* bertipe integer dengan nilai bawaan 0 sebagai data konteks hubungan karakter dari *game engine*.

Penggunaan Pydantic memungkinkan FastAPI melakukan validasi otomatis terhadap data yang masuk. Apabila struktur atau tipe data tidak sesuai, server akan menolak permintaan dengan status HTTP 422 Unprocessable Entity, sehingga kesalahan data dapat dicegah sebelum masuk ke proses inferensi model.

Skema *ChatRequest* kemudian dihubungkan dengan endpoint utama melalui dekorator `@app.post("/chat")`. Fungsi `chat_endpoint` dibuat secara asinkron menggunakan `async def` agar server tetap mampu menangani permintaan HTTP secara lebih efisien melalui mekanisme *non-blocking concurrent execution*. Rangkaian validasi payload dan inisialisasi endpoint ditampilkan pada Kode Program 4.11.

Kode Program 4. 11 Implementasi Kelas Pydantic BaseModel dan Inisialisasi Endpoint HTTP POST Asinkron

```
1. # Kode 4.11 Implementasi Kelas Pydantic BaseModel dan Inisialisasi Endpoint HTTP POST Asinkron
2. from pydantic import BaseModel
3. from fastapi import FastAPI
4. app = FastAPI(title="Serana AI - NLU Brain Cloud", version="5.0")
5. # 1. Definisi Skema Payload Inbound Menggunakan Pydantic BaseModel
6. class ChatRequest(BaseModel):
7.     text: str
8.     affinity_score: int = 0
9. # =====
10. # 2. INISIALISASI MULTI-TASK INFERENCE ENDPOINT
11. # =====
12. @app.post("/chat")
13. async def chat_endpoint(Request: ChatRequest):
14.     # Ekstraksi variabel dari objek Request Body tervalidasi
15.     input_text = Request.text
16.     current_affinity = Request.affinity_score
17.     # [Proses Tokenisasi, Inferensi ONNX, dan Post-Processing Diuraikan pada Sub-Bab Berikutnya]
18.     ....
```

4.3.3.3 Algoritma Koreksi Ejaan via Pustaka *thefuzz* (Fuzzy String Matching)

Tantangan linguistik utama dalam pemrosesan perintah suara melalui arsitektur *Hybrid Client-Cloud* adalah adanya risiko penurunan akurasi transkripsi tekstual yang dihasilkan oleh subsistem *Client STT* Whisper. Gangguan noise latar belakang permainan, fluktuasi intonasi vokal pemain, serta kegagalan pelafalan istilah nama benda atau lokasi fiktif khas lore *Skyrim* berpotensi memicu galat salah ketik atau



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

ejaan (typo). Apabila teks entitas yang cacat tersebut langsung dialirkan ke tahap *Named Entity Recognition* (NER), sistem akan gagal mengenali kata benda tersebut untuk dipetakan ke fungsi biner gim. Guna menanggulangi kerentanan tersebut, diimplementasikan sebuah mekanisme penyalarsan ejaan dinamis pada lapisan *Post-Processing Server* memanfaatkan metode *Fuzzy String Matching* melalui pustaka *thefuzz*.

Proses penyehatan kata benda ini dioperasikan dengan memuat seluruh leksikon entitas tervalidasi yang tersimpan di dalam berkas konfigurasi *fuzzy_targets.json*. Seluruh array kata benda dari berbagai kategori seperti lokasi (location), karakter (character), musuh (enemy), dan item (item) digabungkan ke dalam sebuah variabel memori global bernama *VALID_ENTITIES*. Ketika *endpoint* menerima kiriman teks masukan, token kata yang berhasil diekstraksi oleh modul NER dibersihkan dari tanda baca menggunakan *Regular Expression* dan dirangkai kembali menjadi satu string tunggal bernama *combined_raw_entity*. Variabel string mentah ini kemudian dihitung kemiripan fonetik dan susunan karakternya terhadap seluruh elemen di dalam daftar *VALID_ENTITIES* menggunakan fungsi *process.extractOne()* berbasis metrik jarak algoritma *Levenshtein Distance Ratio*.

Sebuah ambang batas filter tingkat keyakinan (*Confidence Score*) dipasang secara ketat sebesar ≥ 85 . Nilai ambang 85 dipilih berdasarkan pengujian empiris terhadap beberapa variasi kesalahan transkripsi yang umum dihasilkan oleh *Faster-Whisper*. Nilai di bawah ambang tersebut cenderung meningkatkan risiko pemetaan entitas yang tidak sesuai, sedangkan nilai yang lebih tinggi menyebabkan beberapa entitas valid gagal dikenali. Oleh karena itu nilai 85 dipilih sebagai titik kompromi antara sensitivitas koreksi dan ketepatan identifikasi entitas. String hasil transkripsi suara yang memiliki kecocokan sama atau di atas angka ambang batas tersebut akan secara otomatis dikoreksi atau diselaraskan mengikuti ejaan baku target konseptual gim, sedangkan string di bawah performa tersebut akan ditolak. Selain itu, diimplementasikan pula protokol penyelamat (*failsafe protocols*) berlapis untuk menangani ambiguitas data masukan. Jika kata penunjuk subjek "serana" menyusup ke dalam larik entitas yang diekstraksi, sistem secara otomatis menghapusnya dari memori target untuk mencegah konflik eksekusi aksi internal pada *game engine*.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Selanjutnya, apabila lapisan klasifikasi memutuskan kalimat masukan berada pada kelas aksi serangan (Combat_Attack) namun fungsi koreksi ejaan gagal mendeteksi entitas musuh yang valid akibat gangguan transkripsi, protokol failsafe akan secara otomatis mengirimkan entitas tiruan universal berupa "nearest_enemy". Langkah ini memastikan *game engine* tetap dapat mengeksekusi komando aksi taktis pertarungan secara adaptif. Sebaliknya, apabila kelas penargetan spasial krusial seperti pergerakan koordinat (Movement_GoTo) terdeteksi tanpa kehadiran parameter target entitas yang lolos validasi ejaan, *Server* secara deterministik menghentikan proses awal, mengubah status respons menjadi "fallback", dan mengalihkan nama intent menjadi kelas luar domain (Out_Of_Domain). Implementasi dari keseluruhan logika *Fuzzy String Matching* dan protokol failsafe tersebut ditunjukkan pada Kode Program 4.12.

Kode Program 4. 12 Implementasi Logika Post-Processing Fuzzy String Matching dan Protokol Failsafe

```
1. # Kode 4.12 Implementasi Logika Post-Processing Fuzzy String Matching dan
   Protokol Failsafe
2. from thefuzz import process
3. final_entities = []
4. response_status = "success"
5. combined_raw_entity = "".join(cleaned_entities).strip().lower()
6. if combined_raw_entity:
7.     best_match, score = process.extractOne(combined_raw_entity, VALID_ENTITIES)
8.     if score >= 85:
9.         final_entities.append(best_match)
10.    else:
11.        print(f"[NER FILTER] Entity rejected: {combined_raw_entity} ({score}%")
12. if "serana" in final_entities:
13.     final_entities.remove("serana")
14. if intent_confidence < 0.5:
15.     intent_name = "Out_Of_Domain"
16.     response_status = "fallback"
17. if intent_name == "Combat_Attack" and not final_entities:
18.     final_entities.append("nearest_enemy")
19. elif intent_name in ["Movement_GoTo", "Combat_Attack", "Quest_Objective"] and
   not final_entities:
20.     intent_name = "Out_Of_Domain"
21.     response_status = "fallback"
```

4.3.3.4 Implementasi Fuzzy String Matching sebagai Mekanisme Toleransi Kesalahan Entitas

Mekanisme Fuzzy String Matching ditempatkan pada tahap post-processing *Server Cloud NLU*. Tahap ini berjalan setelah model *Multi-Task Learning* menghasilkan tiga keluaran utama, yaitu intent, entity, dan sentiment. Posisi tersebut dipilih karena entity hasil *Named Entity Recognition* belum selalu dapat langsung digunakan oleh *game engine*. Kesalahan transkripsi seperti huruf yang hilang,



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

perubahan bunyi, atau variasi pelafalan dapat menyebabkan nama lokasi, objek, atau musuh tidak sama persis dengan daftar entity referensi yang digunakan oleh sistem.

Pada sistem berbasis exact matching, kesalahan kecil pada entity dapat menyebabkan perintah gagal diproses. Sebagai contoh, string seperti “whitrn”, “solidude”, “bndit”, atau “livr” tidak akan cocok secara langsung dengan entity referensi seperti Whitrn, Solitude, bandit, dan lever. Kondisi tersebut dapat mengganggu proses FormID Binding karena game engine membutuhkan target yang valid sebelum suatu perintah dapat diterjemahkan menjadi aksi di dalam permainan.

Implementasi Fuzzy String Matching dilakukan dengan memanfaatkan pustaka thefuzz pada Server Cloud. Entity hasil ekstraksi dari model NLU terlebih dahulu dibersihkan dari tanda baca dan disusun menjadi satu string melalui variabel `combined_raw_entity`. String tersebut kemudian dibandingkan dengan daftar entity valid yang telah dimuat ke dalam `VALID_ENTITIES`. Proses pencocokan dilakukan menggunakan fungsi `process.extractOne()` untuk memperoleh kandidat entity dengan tingkat kemiripan tertinggi.

Skor kemiripan (similarity score) yang dihasilkan oleh fungsi `process.extractOne()` dihitung menggunakan formula Ratio yang menjadi dasar perhitungan pada pustaka thefuzz. Pustaka ini merupakan kelanjutan dari fuzzywuzzy, yaitu pustaka yang sama dengan yang digunakan Bosker (2021) untuk menghitung Token Sort Ratio dalam penelitiannya mengenai fuzzy string matching. Formula tersebut dirumuskan sebagai berikut:

$$Ratio = \frac{2M}{T} \times 100$$

Variabel M pada formula tersebut menyatakan jumlah total panjang karakter yang cocok antara string hasil transkripsi (`combined_raw_entity`) dan string entity referensi pada `VALID_ENTITIES`, sedangkan T menyatakan jumlah total panjang kedua string tersebut (Bosker, 2021). Karena formula ini menghasilkan nilai berskala 0 hingga 100, hasil perhitungannya sejalan dengan ambang batas similarity



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

score minimal 85 yang diterapkan pada sistem. Semakin besar proporsi karakter yang cocok antara kedua string, semakin tinggi nilai Ratio yang dihasilkan, sehingga entity dengan variasi ejaan kecil akibat kesalahan transkripsi tetap dapat dikenali oleh sistem selama proporsi kecocokan karakternya memenuhi ambang batas tersebut.

Hasil pencocokan tidak langsung diterima oleh sistem. Sistem hanya menerima kandidat entity apabila similarity score memenuhi ambang batas minimal 85. Nilai ambang ini digunakan untuk menjaga keseimbangan antara fleksibilitas koreksi dan ketepatan identifikasi entity. Jika similarity score memenuhi ambang batas, entity hasil koreksi dimasukkan ke dalam `final_entities`. Sebaliknya, apabila similarity score berada di bawah ambang batas, entity ditolak agar sistem tidak memilih target yang tidak sesuai.

Implementasi teknis mekanisme Fuzzy String Matching dan protokol failsafe ditunjukkan pada Kode Program 4.12. Berdasarkan kode tersebut, fungsi `process.extractOne()` digunakan untuk membandingkan `combined_raw_entity` dengan daftar `VALID_ENTITIES`. Hasil terbaik dari proses pencocokan disimpan dalam variabel `best_match`, sedangkan tingkat kemiripannya disimpan dalam variabel `score`. Apabila nilai `score` lebih besar atau sama dengan 85, `best_match` dimasukkan ke dalam `final_entities` sebagai entity final yang digunakan pada tahap pemetaan berikutnya.

Mekanisme validasi tambahan juga diterapkan untuk mencegah kesalahan eksekusi pada game engine. Entity “serana” dihapus dari `final_entities` apabila terdeteksi sebagai target, karena Serana merupakan aktor utama yang menjalankan perintah, bukan objek tujuan dari perintah pemain. Validasi ini diperlukan agar sistem tidak salah menafsirkan subjek perintah sebagai target aksi.

Selain koreksi entity, sistem juga menerapkan protokol fallback pada kondisi tertentu. Apabila intent confidence berada di bawah ambang batas, intent dialihkan menjadi `Out_Of_Domain` dan `response_status` diubah menjadi fallback. Langkah ini mencegah perintah yang tidak meyakinkan diteruskan ke game engine. Pada intent `Combat_Attack`, jika entity musuh tidak berhasil dikenali, sistem dapat menggunakan entity cadangan `nearest_enemy` agar perintah serangan tetap dapat



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dieksekusi terhadap musuh terdekat. Sebaliknya, pada intent yang membutuhkan target spesifik seperti *Movement_GoTo*, sistem akan mengalihkan perintah menjadi *Out_Of_Domain* apabila tidak terdapat entity valid yang lolos proses validasi.

Berdasarkan mekanisme tersebut, Fuzzy String Matching berfungsi sebagai lapisan fault-tolerance antara hasil NLU dan proses FormID Binding. Mekanisme ini tidak menggantikan model Named Entity Recognition, tetapi menyelaraskan entity hasil transkripsi agar sesuai dengan daftar referensi entity permainan. Dengan demikian, entity yang sebelumnya tidak cocok secara exact matching tetap dapat digunakan selama memiliki tingkat kemiripan yang memenuhi ambang batas.

4.3.3.5 Pemetaan FormID Binding dan Drop Payload

Tahap akhir pemrosesan *Server Cloud* adalah mengubah hasil NLU menjadi data yang dapat dibaca oleh skrip Papyrus pada game engine. Pada tahap ini, entitas hasil Fuzzy Matching dipetakan ke FormID melalui kamus *FORMID_DICT* yang bersumber dari *papyrus_formids.json*. Jika entitas berupa nilai khusus seperti *GENERIC* atau *nearest_enemy*, sistem langsung menetapkan target sesuai kebutuhan eksekusi. Namun, jika entitas merupakan objek spesifik dalam gim, sistem memisahkan nama plugin dan FormID menggunakan pembatas `|`, lalu mengonversi FormID heksadesimal menjadi integer agar dapat diproses oleh Papyrus.

Selain pemetaan target, sistem juga menghitung perubahan afeksi melalui variabel *sentiment_shift*. *Sentimen Positive* menghasilkan nilai `+5`, sedangkan *sentimen Negative* menghasilkan nilai `-5`. Nilai afeksi aktif kemudian digunakan oleh fungsi *fetch_audio_formid()* untuk memilih respons audio dari *papyrus_voicelines.json* berdasarkan kombinasi intent, kondisi target, dan tingkat hubungan karakter.

Seluruh hasil akhir dikemas ke dalam *Outbound Payload* yang terdiri dari dua blok, yaitu "string" untuk data seperti intent, sentiment, target_plugin, target_name, status, dan tactical_mode, serta "int" untuk target_formid, sentiment_shift, dan audio_formid. Struktur ini menjadi dasar isi *SeranaCommand.json* yang akan dikirim ke direktori lokal gim dan dibaca oleh skrip Papyrus. Implementasi pemetaan FormID dan penyusunan payload ditampilkan pada Kode Program 4.13.

Kode Program 4. 13 Potongan Kode Post-Processing FormID Binding, Sentiment Shift, dan Konstruksi Outbound Payload

```

1. # Kode 4.13 Potongan Kode Post-Processing FormID Binding, Sentiment Shift, dan
   Konstruksi Outbound Payload
2. target_plugin = ""
3. target_formid_int = 0
4. target_name = ""
5. if final_entities:
6.     target_entity = final_entities[0]
7.     dict_value = FORMID_DICT.get(target_entity)
8.     if dict_value == "GENERIC":
9.         target_plugin = "GENERIC"
10.        target_name = target_entity
11.    elif target_entity == "nearest_enemy":
12.        target_plugin = "nearest_enemy"
13.    elif isinstance(dict_value, str) and "|" in dict_value:
14.        plugin, formid = dict_value.split("|")
15.        target_plugin = plugin.strip()
16.        target_formid_int = int(formid.strip(), 16)
17. sentiment_shift = 0
18. if sentiment_name == "Positive":
19.     sentiment_shift = 5
20. elif sentiment_name == "Negative":
21.     sentiment_shift = -5
22. audio_formid_str = fetch_audio_formid(intent_name, target_name,
   current_affinity)
23. audio_formid_int = int(audio_formid_str, 16) if audio_formid_str != "0x000000"
   else 0
24. final_output = {
25.     "string": {
26.         "intent": intent_name,
27.         "sentiment": sentiment_name,
28.         "target_plugin": target_plugin,
29.         "target_name": target_name,
30.         "status": response_status,
31.         "tactical_mode": tactical_mode
32.     },
33.     "int": {
34.         "target_formid": target_formid_int,
35.         "sentiment_shift": sentiment_shift,
36.         "audio_formid": audio_formid_int
37.     }
38. }
39. return final_output

```

```

INFO: 172.17.0.1:42486 - "POST /chat HTTP/1.0" 200 OK
[INTENT CLASSIFIER] Tactical Override: MAGIC
[API RESPONSE] Intent: Combat_Tactical | Audio: 0x03A9D3 | Affinity Shift: 0
INFO: 172.17.0.1:58678 - "POST /chat HTTP/1.0" 200 OK
[INTENT CLASSIFIER] Tactical Override: MELEE
[API RESPONSE] Intent: Combat_Tactical | Audio: 0x03A9D3 | Affinity Shift: 0
INFO: 172.17.0.1:56964 - "POST /chat HTTP/1.0" 200 OK
[INTENT CLASSIFIER] Tactical Override: DEFAULT
[API RESPONSE] Intent: Combat_Tactical | Audio: 0x03A9D3 | Affinity Shift: 0
INFO: 172.17.0.1:40036 - "POST /chat HTTP/1.0" 200 OK
[INTENT CLASSIFIER] Lexicon Match -> Routing to Inventory_Trade
[API RESPONSE] Intent: Inventory_Trade | Audio: 0x03A906 | Affinity Shift: 0
INFO: 172.17.0.1:56640 - "POST /chat HTTP/1.0" 200 OK
[INTENT CLASSIFIER] Lexicon Match -> Routing to Movement_GoTo
[API RESPONSE] Intent: Movement_GoTo | Audio: 0x03A91A | Affinity Shift: 5
INFO: 172.17.0.1:43614 - "POST /chat HTTP/1.0" 200 OK
[API RESPONSE] Intent: Movement_Follow | Audio: 0x03A8DF | Affinity Shift: 0
INFO: 172.17.0.1:49650 - "POST /chat HTTP/1.0" 200 OK

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4. 12 Log aktivitas REST API pada terminal VPS saat sukses memproses skema JSON *Payload*

4.3.4 Implementasi Skrip Modifikasi *Game engine*

Implementasi skrip modifikasi pada *game engine* (*Creation Engine*) merupakan lapisan eksekusi lokal (*Client-side execution layer*) yang berfungsi mengonversi hasil pemrosesan *Server Cloud* menjadi aksi nyata karakter secara luring dan *real-time*. Menggunakan bahasa pemrograman Papyrus melalui objek *Quest* utama *SeranaAI_CoreScript.psc*, seluruh pipa data (*pipeline*) kendali pergerakan, simulasi status psikologis, hingga interaksi *Follower* diintegrasikan secara asinkron guna memastikan eksekusi perintah berjalan stabil tanpa membebani *Runtime thread* asli milik gim. Pembahasan pada sub-bab ini diuraikan secara komprehensif mulai dari deklarasi arsitektur properti pelacak, mekanisme siklus hidup *polling loop*, manajemen pembacaan berkas JSON, distribusi perintah melalui *Action Demultiplexer*, rekayasa navigasi spasial tingkat lanjut, akumulasi matematis skor afeksi, hingga pemicuan sinyal otonom berbasis konteks perubahan delta lingkungan dunia gim.

4.3.4.1 Deklarasi Arsitektur Properti, Variabel Global, dan Sinkronisasi *DLC1MentalModel*

Tahap awal implementasi pada sisi *game client* dilakukan dengan menyusun properti dan variabel pengontrol pada skrip *SeranaAI_CoreScript.psc*. Properti berfungsi sebagai penghubung antara data pada plugin *.esp* atau *.esm* dengan *Virtual Machine* Papyrus. Agar alur data lebih terstruktur, properti pada skrip ini dibagi menjadi tiga lapisan, yaitu lapisan aktor, lapisan jembatan data, serta lapisan navigasi dan interaksi spasial.

Lapisan aktor digunakan untuk mengunci referensi *Serana* dan target interaksi melalui *DLC1SeranaRef*, *SeranaAlias*, dan *TargetAlias*. Penggunaan *ReferenceAlias* memungkinkan sistem mengatur target dan perilaku NPC secara dinamis tanpa mengubah data dasar aktor. Lapisan jembatan data mencakup *GameHour*, *SeranaAffinityScore*, dan *SeranaAudioPointer* yang digunakan untuk menyimpan waktu permainan, nilai hubungan sosial, serta penunjuk audio dialog yang dikirim dari *Server Cloud*, diwujudkan melalui deklarasi variabel *Global Variable* berikut:

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1. GlobalVariable Property GameHour Auto
2. GlobalVariable Property SeranaAffinityScore Auto
3. GlobalVariable Property SeranaAudioPointer Auto

Nilai numerik desimal pada SeranaAudioPointer mengunci parameter FormID audio dialog hasil kalkulasi repositori luar, sedangkan SeranaAffinityScore menyimpan memori hubungan sosial jangka panjang.

Lapisan ketiga dikonfigurasi sebagai Lapisan Properti Navigasi dan Interaksi Spasial. Lapisan navigasi dan interaksi spasial mencakup SeranaDestAlias, SeranaDestLocAlias, dan InteractAlias. Properti tersebut digunakan untuk membedakan target objek fisik, target lokasi, dan objek lingkungan yang akan diinteraksikan. Status pergerakan dikendalikan oleh SeranaGoToState, sedangkan status interaksi dikendalikan oleh SeranaWaitState. Selain itu, skrip juga disinkronkan dengan DLC1MentalModel agar perintah mengikuti, menunggu, dan berhenti mengikuti tetap selaras dengan perilaku bawaan Serana dari sistem Dawnguard. Deklarasi properti utama ditampilkan pada Kode Program 4.14.

Kode Program 4. 14 Struktur Deklarasi Properti Tiga Lapisan Utama pada SeranaAI_CoreScript.psc

```
1. ; Kode 4.14 Struktur Deklarasi Properti Tiga Lapisan Utama pada
   SeranaAI_CoreScript.psc
2. Scriptname SeranaAI_CoreScript extends Quest
3. ; --- LAPISAN 1: PROPERTI AKTOR (CORE REFERENCE & ALIASING) ---
4. Actor Property DLC1SeranaRef Auto
5. ReferenceAlias Property SeranaAlias Auto
6. ReferenceAlias Property TargetAlias Auto
7. Float Property PollInterval = 1.0 Auto
8. ; --- LAPISAN 2: PROPERTI JEMBATAN DATA (GLOBAL TELEMETRY) ---
9. GlobalVariable Property GameHour Auto
10. GlobalVariable Property SeranaAffinityScore Auto
11. GlobalVariable Property SeranaAudioPointer Auto
12. ; --- JEMBATAN MUTLAK KE OTAK DAWNGUARD (SINKRONISASI BEHAVIORAL) ---
13. DLC1_NPCMentalModelScript Property DLC1MentalModel Auto
14. ; --- LAPISAN 3: PROPERTI NAVIGASI DAN INTERKSI SPASIAL ---
15. ReferenceAlias Property InteractAlias Auto
16. GlobalVariable Property SeranaWaitState Auto
17. ReferenceAlias Property SeranaDestAlias Auto ; Wadah Mode 1 (Benda/Tanah)
18. LocationAlias Property SeranaDestLocAlias Auto ; Wadah Mode 2 (Wilayah/Kota)
19. GlobalVariable Property SeranaGoToState Auto ; 0=Mati, 1=Benda, 2=Wilayah
20. Float gotoTimeout = 0.0
21. Topic Property mainTopic Auto
```

Poin penting pada tahap deklarasi adalah penggunaan DLC1_NPCMentalModelScript Property DLC1MentalModel Auto sebagai penghubung langsung ke skrip kecerdasan buatan bawaan DLC Dawnguard. Dengan menjadikan DLC1MentalModel sebagai properti statis, skrip SeranaAI_CoreScript.psc dapat memanggil fungsi internal follower seperti Wait(),

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

StopWaiting(), dan Dismiss() secara langsung tanpa melakukan *runtime casting* berulang. Pendekatan ini mengurangi *runtime lookup overhead* pada Virtual Machine Papyrus, terutama karena skrip berjalan dalam *polling loop* setiap 1,0 detik.

Selain properti utama, skrip juga mendeklarasikan variabel internal non-properti seperti lastHour, lastAffinity, lastTimeState, lastWeatherState, lastLocation, wasInCombat, dan lastInteriorState. Variabel ini digunakan sebagai *session state tracker* untuk menyimpan kondisi lingkungan dari siklus sebelumnya tanpa membebani penyimpanan *save game*. Melalui mekanisme *delta detection*, sistem hanya memicu pengiriman payload atau dialog otonom ketika terdapat perubahan nyata pada waktu, cuaca, lokasi, atau status pertarungan. Komposisi variabel pelacak ini ditampilkan pada Kode Program 4.15.

Kode Program 4. 15 Deklarasi Variabel State Tracker untuk Protokol Sensor Deteksi Perubahan Delta

```
1. ; Kode 4.15 Deklarasi Variabel State Tracker untuk Protokol Sensor Deteksi
   Perubahan Delta
2. float lastHour = -1.0
3. int lastAffinity = -999
4. Int lastTimeState = 0
5. Int lastWeatherState = 0
6. Int lastAffinityTier = 0
7. Bool wasInCombat = False
8. Float idleTimer = 0.0
9. Location lastLocation = None
10. Bool lastInteriorState = False
11. Int affinityState = 0
12. Float affinityTimeout = 0.0
13. Float interactTimeout = 0.0
```

4.3.4.2 Arsitektur Polling Loop dan Siklus Hidup Skrip

Tahap inisialisasi awal dari siklus hidup skrip lokal di dalam gim dioperasikan melalui eksekusi fungsi peristiwa Event OnInit(). Di dalam blok tersebut, sistem memicu cetakan notifikasi arsitektur ke konsol, mengaktifkan penanda antarmuka pengguna grafik (*Quest Objective UI*) melalui instruksi SetObjectiveDisplayed(10, True), serta melakukan peluncuran perdana putaran pembaruan (*Update*) secara asinkron menggunakan perintah RegisterForSingleUpdate(PollInterval). Keputusan rekayasa untuk menetapkan nilai variabel interval waktu penyegaran PollInterval = 1.0 detik dipilih secara teknis guna menghindari penggunaan struktur *infinite loop* konvensional seperti perintah While yang bersifat memblokir utas

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(*blocking thread*). Dampak dari penerapan inisialisasi ini adalah skrip mampu berjalan secara asinkron di latar belakang *Virtual Machine* Papyrus tanpa mengunci atau membekukan jalannya performa *Runtime engine* utama gim, memberikan ruang jeda alokasi memori yang aman bagi objek *Quest* untuk memuat seluruh referensi eksternal sejak awal sesi permainan dimulai. Rangkaian inisialisasi siklus awal ini didokumentasikan pada Kode Program 4.16.

Kode Program 4. 16 Peristiwa OnInit() dan Logika Failsafe Pemulihan Kendali Alias Aktor

```
1. ; Kode 4.16 Peristiwa OnInit() dan Logika Failsafe Pemulihan Kendali Alias Aktor
2. Event OnInit()
3.     Debug.Notification("Serana AI: Architecture V9.5 (Dual-Engine Pathfinding)
   Online.")
4.     SetObjectiveDisplayed(10, True)
5.     RegisterForSingleUpdate(PollInterval)
6. EndEvent
7. Event OnUpdate()
8.     If (SeranaAlias.GetReference() == None)
9.         SeranaAlias.ForceRefTo(DLC1SeranaRef)
10.        TargetAlias.Clear()
11.        Debug.Notification("SYS-CORE: Empty alias. Forcing Serana into Alias
   100!")
12.    EndIf
13.
14.    Actor SeranaRef = SeranaAlias.GetActorRef()
15.    If (SeranaRef == None)
16.        RegisterForSingleUpdate(PollInterval)
17.        Return
18.    EndIf
```

Mekanisme pembaruan berkala dikendalikan melalui Event OnUpdate(). Pada awal siklus, skrip menjalankan mekanisme *failsafe* untuk memastikan referensi aktor Serana tetap terikat pada sistem alias melalui pengecekan SeranaAlias.GetReference(). Pendekatan *polling loop* menggunakan RegisterForSingleUpdate dipilih agar sistem memiliki kendali deterministik terhadap interval pembacaan data dan urutan eksekusi. Dibandingkan pendekatan *event-driven* murni, pola ini lebih aman untuk menangani aliran data asinkron dari *Server Cloud* karena dapat mengurangi risiko penumpukan antrean instruksi, *script lag*, dan gangguan pada *Virtual Machine* Papyrus ketika permainan berada dalam kondisi padat seperti pertempuran.

Selain itu, skrip menerapkan mekanisme *early return* sebagai pencegahan tabrakan eksekusi. Jika SeranaAudioPointer.GetValueInt() bernilai 8 atau 9, yaitu kode khusus untuk dialog eskalasi hubungan sosial, maka eksekusi modul lain langsung dihentikan sementara menggunakan instruksi Return. Mekanisme ini mencegah

konflik antara proses ForceGreet dengan modul lain seperti *pathfinding*, pembacaan *payload* JSON, dan pemindaian kondisi lingkungan. Dengan demikian, dialog emosional dapat berjalan secara serial tanpa interupsi, sementara beban eksekusi pada CPU klien tetap lebih terkendali. Logika *early return* tersebut ditampilkan pada Kode Program 4.17.

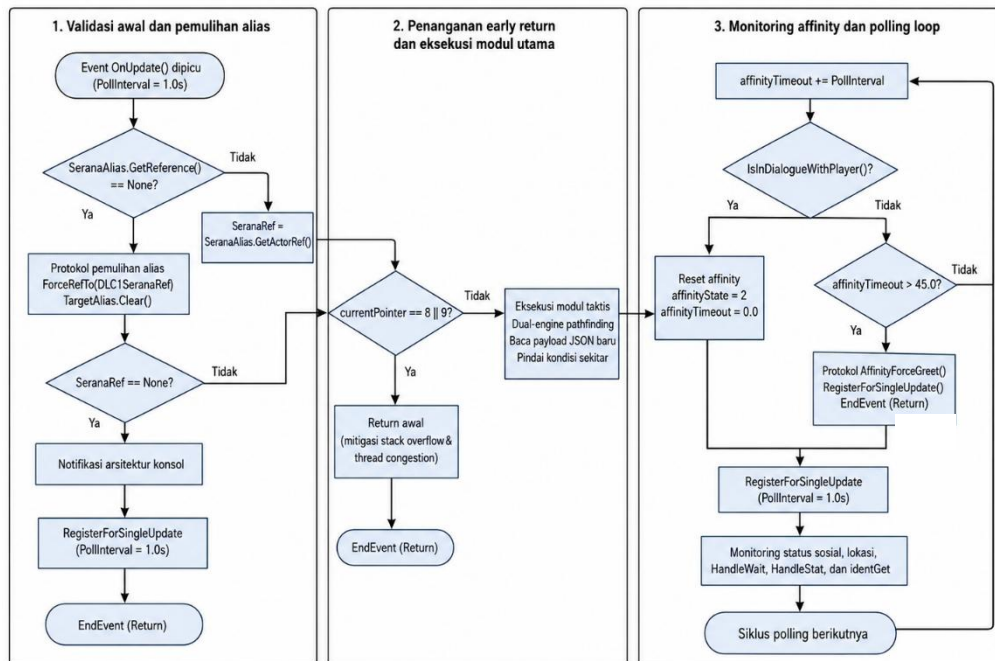
Kode Program 4. 17 Blok Penanganan Early Return untuk State Machine Dialog ForceGreet pada OnUpdate()

```
1. ; Kode 4.17 Blok Penanganan Early Return untuk State Machine Dialog ForceGreet
   pada OnUpdate()
2.     Int currentPointer = SeranaAudioPointer.GetValueInt()
3.     If (currentPointer == 8 || currentPointer == 9)
4.         affinityTimeout += PollInterval
5.
6.     If (SeranaRef.IsInDialogueWithPlayer())
7.         If (affinityState != 2)
8.             Debug.Notification("SYS-AFFINITY: Serana initiated conversation
   (ForceGreet Success).")
9.             affinityState = 2
10.        EndIf
11.        affinityTimeout = 0.0
12.    ElseIf (affinityState == 2 && !SeranaRef.IsInDialogueWithPlayer())
13.        Debug.Notification("SYS-AFFINITY: Conversation ended. Resetting
   Brain...")
14.        ResetAffinityForceGreet(SeranaRef)
15.    ElseIf (affinityTimeout > 45.0)
16.        Debug.Notification("SYS-ERROR: ForceGreet Timeout. Serana stuck.
   Forcing reset.")
17.        ResetAffinityForceGreet(SeranaRef)
18.    EndIf
19.    RegisterForSingleUpdate(PollInterval)
20.    Return
21. EndIf
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Diagram alur logika penanganan siklus hidup OnUpdate() Papyrus (Polling Loop & Early Return Protocol)



Gambar 4. 13 Diagram Alur Logika Siklus Hidup Event OnUpdate() Papyrus dengan Mekanisme Polling Loop dan Early Return Protocol

4.3.4.3 Jembatan Data JSON: Penulisan SeranaContext.json dan Pembacaan SeranaCommand.json

Aliran komunikasi data dua arah (*bidirectional data flow*) yang menjembatani *game engine* lokal dengan *Server Cloud* secara asinkron dikelola sepenuhnya melalui manipulasi berkas *text-based payload* berformat JSON. Tahap pertama dari pipa komunikasi ini adalah pencatatan telemetri internal gim melalui penulisan berkas *SeranaContext.json*. Proses ini diimplementasikan dengan memanfaatkan fungsi eksternal PapyrusUtil melalui instruksi *JsonUtil.SetFloatValue()* untuk variabel waktu dunia gim serta *JsonUtil.SetIntValue()* untuk variabel tingkat afeksi karakter. Keputusan desain untuk membungkus rutinitas penulisan ini di dalam filter kondisional *If* (*currentHour != lastHour || currentAffinity != lastAffinity*) dipilih secara teknis untuk menerapkan arsitektur deteksi selisih perubahan delta (*delta detection*). Dampak dari penerapan mekanisme ini adalah proses penyimpanan data fisik melalui perintah *JsonUtil.Save()* hanya akan dieksekusi jika terjadi pergeseran status riil di dalam gim. Langkah ini secara drastis menekan beban kinerja *I/O disk* pada komputer klien dari risiko kejenuhan operasi tulis-baca berkas (*disk write*

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

overhead) yang umumnya terjadi akibat pemanggilan fungsi penyimpanan konstan pada setiap siklus pembaruan skrip. Fase transmisi data keluar (*Outbound Pipeline*) ini didokumentasikan pada Kode Program 4.18

Kode Program 4. 18 Logika Eksekusi Penulisan Telemetri Outbound pada *SeranaContext.json*

```
1. ; Kode 4.18 Logika Eksekusi Penulisan Telemetri Outbound pada SeranaContext.json
2. float currentHour = GameHour.GetValue()
3. int currentAffinity = SeranaAffinityScore.GetValueInt()
4. If (lastAffinityTier == 0)
5.     lastAffinityTier = GetAffinityTier(currentAffinity)
6. EndIf
7. ; Sensor Delta: Eksekusi simpan hanya jika nilai variabel lingkungan
   bergeser
8. If (currentHour != lastHour || currentAffinity != lastAffinity)
9.     JsonUtil.SetFloatValue("SeranaContext.json", "GameHour", currentHour)
10.    JsonUtil.SetIntValue("SeranaContext.json", "AffinityScore",
currentAffinity)
11.    JsonUtil.Save("SeranaContext.json")
12.    lastHour = currentHour
13.    lastAffinity = currentAffinity
14. EndIf.
```

Tahap kedua adalah penyerapan instruksi kognitif awal melalui proses pembacaan berkas data masuk (*Inbound Payload*) bernama *SeranaCommand.json* yang sebelumnya dijatuhkan oleh *Server Cloud* melalui metode *Atomic Drop*. Langkah ekstraksi data ini diimplementasikan menggunakan fungsi *JsonUtil.GetStringValue()* untuk menangkap variabel berbasis alfabet seperti kelas *Intent*, *sentiment*, *target_plugin*, dan *target_name*, dikombinasikan dengan fungsi *JsonUtil.GetIntValue()* untuk mengurai parameter numerik seperti *target_formid*, *audio_formid*, dan *sentiment_shift*. Guna menjamin bahwa skrip Papyrus tidak membaca berkas instruksi yang belum selesai ditulis oleh pihak *Server*, aliran data masuk ini memanfaatkan protokol *Atomic Renaming*. *Server Python* secara terprogram mengirimkan data ke dalam berkas temporer *TempCommand.json* terlebih dahulu, sebelum sistem operasi mengubah namanya secara instan menjadi *SeranaCommand.json*. Poin rekayasa perangkat lunak paling krusial pada fase pembacaan ini adalah kewajiban untuk mengeksekusi instruksi *JsonUtil.Unload("SeranaCommand.json")* tepat sebelum variabel di dalam berkas dibaca oleh skrip Papyrus. Pilihan instruksi ini diterapkan secara teknis sebagai protokol pembatalan tembolok (*cache invalidation protocol*). Tanpa adanya perintah *Unload*, *Virtual Machine* Papyrus secara deterministik akan selalu membaca data usang yang telanjur mengendap di dalam tembolok RAM gim.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Dampak dari penempatan protokol ini adalah gim dipaksa untuk melakukan proses interogasi ulang langsung ke piringan penyimpanan fisik disk komputer untuk mengambil struktur *Payload* yang paling mutakhir, menjamin tingkat latensi pemrosesan perintah suara berada pada titik terendah, serta memastikan sinkronisasi data berjalan secara *real-time*. Protokol pembacaan data masuk ini diuraikan pada Kode Program 4.19.

Kode Program 4. 19 Protokol Invalidasi Cache dan Pembacaan Inbound *Payload* SeranaCommand.json

```
1. ; Kode 4.19 Protokol Invalidasi Cache dan Pembacaan Inbound Payload
SeranaCommand.json
2. ; Memaksa mesin VM membuang cache lama dan membaca ulang file fisik dari
disk
3. JsonUtil.Unload("SeranaCommand.json")
4. String currentIntent = JsonUtil.GetStringValue("SeranaCommand.json",
"Intent")
5. If currentIntent != ""
6. String targetPlugin = JsonUtil.GetStringValue("SeranaCommand.json",
"target_plugin")
7. Int targetFormID = JsonUtil.GetIntValue("SeranaCommand.json",
"target_formid")
8. String currentSentiment = JsonUtil.GetStringValue("SeranaCommand.json",
"sentiment")
9. String targetName = JsonUtil.GetStringValue("SeranaCommand.json",
"target_name")
10. Int audioFormID = JsonUtil.GetIntValue("SeranaCommand.json",
"audio_formid")
11. Int sentimentShift = JsonUtil.GetIntValue("SeranaCommand.json",
"sentiment_shift")
12. Debug.Notification("SYS-NLU: Received command -> " + currentIntent)
13. DistributeLogic(currentIntent, targetPlugin, targetFormID,
currentSentiment, targetName)
```

 SeranaCommand.json	18/05/2026 14:35	JSON Source File	1 KB
 SeranaContext.json	18/05/2026 14:41	JSON Source File	1 KB

Gambar 4. 14 Struktur visual file JSON SeranaContext dan SeranaCommand di dalam direktori penyimpanan lokal *game Client*

4.3.4.4 Distribusi *Intent* dan Manajemen Ekspresi Wajah

Tahap pemisahan instruksi kognitif menjadi aksi taktis di dalam *game engine* dikendalikan sepenuhnya melalui implementasi fungsi Function *DistributeLogic()*. Pada fase awal eksekusi fungsi ini, sistem menguji validasi keberadaan referensi aktor SeranaRef dari Reference Alias untuk mencegah terjadinya kegagalan pemanggilan objek kosong. Setelah objek terverifikasi, sistem mengeksekusi parameter emosi melalui pemanggilan instruksi penimpaan ekspresi wajah bawaan gim berupa SeranaRef.SetExpressionOverride(). Logika pemosisian fitur wajah

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

diatur secara kondisional berdasarkan nilai string variabel *sentimentValue*. Apabila Inbound *Payload* membawa nilai "Positive", ekspresi wajah disetel pada indeks identitas 10 dengan tingkat intensitas komputasi sebesar 50, sedangkan muatan bermutu "Negative" akan memicu penyetelan wajah pada indeks identitas 8 dengan intensitas sebesar 60. Jika muatan bernilai netral, status emosi dibersihkan secara otomatis via perintah *SeranaRef.ClearExpressionOverride()*.

Keputusan arsitektur untuk menempatkan visualisasi emosi tepat di awal fungsi sebelum memanggil handler tindakan spesifik dipilih secara teknis guna menjamin sinkronisasi umpan balik visual instan (*instant visual feedback responsiveness*). Mengubah ekspresi di lapisan teratas memastikan respons afeksi langsung muncul pada wajah karakter tanpa perlu menunggu atau terhambat oleh durasi waktu tunggu (*latency*) fungsi operasional lainnya seperti pencarian *Pathfinding* spasial atau pemindaian radius pertempuran yang memerlukan alokasi siklus thread CPU lebih lama. Dampak dari penerapan manajemen ekspresi adaptif ini adalah karakter mampu menampilkan dinamika psikologis visual yang sinkron secara *real-time* dengan muatan sentimen kontekstual dari kalimat yang diucapkan oleh pemain. Proses manajemen ekspresi wajah pralogika tindakan ini diuraikan pada Kode Program 4.20.

Kode Program 4. 20 Logika Penanganan Ekspresi Wajah Berdasarkan Muatan Sentimen Awal *Cloud*

```
1. ; Kode 4.20 Logika Penanganan Ekspresi Wajah Berdasarkan Muatan Sentimen Awal Cloud
2. Function DistributeLogic(String IntentValue, String targetPlugin, Int targetFormID, String sentimentValue, String targetName)
3.     Actor SeranaRef = SeranaAlias.GetActorRef()
4.     If (SeranaRef == None)
5.         Debug.Notification("SYS-ERROR: Serana Body failed to detect in Alias!")
6.         Return
7.     EndIf
8.     ; Eksekusi Umpan Balik Visual Instan di Awal Sebelum Handler Aksi Dipanggil
9.     If (sentimentValue == "Positive")
10.         SeranaRef.SetExpressionOverride(10, 50)
11.     ElseIf (sentimentValue == "Negative")
12.         SeranaRef.SetExpressionOverride(8, 60)
13.     Else
14.         SeranaRef.ClearExpressionOverride()
15.     EndIf
```

Setelah visualisasi wajah dikunci, fungsi *DistributeLogic()* beroperasi penuh sebagai pusat percabangan data tunggal (*Action Demultiplexer*) untuk mendistribusikan kueri menuju 10 kelas tindakan utama yang didukung oleh

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

arsitektur Multi-Task NLU. Pengaliran instruksi (routing process) dibangun memanfaatkan struktur percabangan logika If/ElseIf terindeks yang menguji kesesuaian nilai string variabel *IntentValue* terhadap seluruh fungsi handler taktis internal skrip. Penyelarasan komando ini memetakan seluruh kelas intent secara utuh dan adil yang meliputi: fungsi kosmetik nol-matriks *Social_Chat* dan pengecekan nilai variabel memori internal *Social_Status*, pengalihan navigasi spasial dinamis *Movement_GoTo*, penargetan mutlak pertarungan *Combat_Attack*, pembukaan antarmuka penukaran barang *Inventory_Trade*, instruksi sinkronisasi menunggu *Movement_Wait*, pemulihan ikatan behavioral *Movement_Follow*, penargetan manipulasi objek fisik sekitar *Interact_Environment*, pelepasan otonomitas pengikut *System_Dismiss*, hingga penyesuaian gaya bertarung tiga mode *Combat_Tactical*.

Pola penggabungan seluruh kendali fitur interaksi ke dalam satu wadah pemutus tunggal dipilih secara teknis untuk mengeliminasi fragmentasi instruksi dan menjaga kestabilan sirkulasi antrian perintah pada komponen *game Client*. Dampak positif dari penerapan arsitektur Action Demultiplexer ini adalah pengembang memperoleh kemudahan pelacakan galat mutlak (streamlined debugging flow) karena seluruh gerbang pengalihan komando bermuara pada satu titik kontrol terpusat, serta menjamin seluruh 10 pola kontribusi perilaku adaptif NPC dapat dieksekusi secara teratur, mandiri, dan deterministik tanpa risiko tumpang-tindih instruksi memori antar-peristiwa. Larik distribusi logika interaksi 10 *Intent* utama ini didokumentasikan secara lengkap pada Kode Program 4.21.

Kode Program 4. 21 Implementasi Pusat Percabangan Data (Action Demultiplexer) untuk 10 Fitur Utama Mod

```
1. ; Kode 4.21 Implementasi Pusat Percabangan Data (Action Demultiplexer) untuk 10
   Fitur Utama Mod
2.   If (IntentValue == "Social_Chat")
3.       Debug.Notification("Serana AI [Chat]: Cosmetic Chat (Zero-Havok).")
4.   ElseIf (IntentValue == "Social_Status")
5.       Int currentScore = SeranaAffinityScore.GetValueInt()
6.       Debug.Notification("Serana AI [Status]: Current Affinity Score -> " +
   currentScore)
7.   ElseIf (IntentValue == "Movement_GoTo")
8.       HandleMovement(targetPlugin, targetFormID)
9.   ElseIf (IntentValue == "Combat_Attack")
10.      HandleCombat(targetPlugin, targetFormID, targetName)
11.  ElseIf (IntentValue == "Inventory_Trade")
12.      HandleTrade()
13.  ElseIf (IntentValue == "Movement_Wait")
14.      HandleWait()
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

15.     ElseIf (IntentValue == "Movement_Follow")
16.         HandleFollow()
17.     ElseIf (IntentValue == "Interact_Environment")
18.         HandleInteract(targetPlugin, targetFormID)
19.     ElseIf (IntentValue == "System_Dismiss")
20.         HandleDismiss()
21.     ElseIf (IntentValue == "Combat_Tactical")
22.         HandleTactical()
23.     EndIf
24. EndFunction

```



Gambar 4. 15 Visualisasi perubahan ekspresi wajah NPC Serana yang bergeser secara dinamis: (a) Serana tersenyum ketika command dengan Sentimen Positif; (b) Serana sedih ketika command dengan Sentimen Negatif

4.3.4.5 Rekayasa Navigasi Spasial *Dual-Engine Pathfinding*

Komponen utama pada sisi game client adalah sistem navigasi spasial adaptif bernama Dual-Engine Pathfinding. Sistem ini dirancang untuk menerjemahkan perintah spasial dari bahasa alami menjadi target pergerakan yang dapat dipahami oleh Creation Engine. Implementasinya dibagi ke dalam dua bagian, yaitu fungsi HandleMovement() sebagai gerbang penentuan target dan mekanisme Dual-Scanner pada Event OnUpdate() sebagai pemantau status pergerakan NPC.

Pada tahap awal, sistem menggunakan mekanisme Smart Splitter NLU untuk membedakan jenis target yang dikirim oleh *Server Cloud*. Jika entitas dapat dikonversi menjadi Location, misalnya pada perintah “Go to Dawnstar”, maka target disimpan sebagai FinalLocation. Jika tidak, sistem mencoba mengonversinya menjadi ObjectReference, yaitu objek fisik tertentu di dunia permainan. Apabila

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

entitas yang diterima hanya berupa Base Object, seperti “iron sword”, sistem mencari instansi fisik terdekat menggunakan `Game.FindClosestReferenceOfTypeFromRef()` dalam radius 10.000 unit dari posisi pemain. Mekanisme berlapis ini digunakan agar sistem dapat membedakan target wilayah makro, objek fisik, dan objek referensi secara lebih aman tanpa memicu kegagalan runtime casting pada Papyrus.

Apabila *Server Cloud* tidak mengirimkan target karena kegagalan ekstraksi NER, sistem menggunakan Sensor Crosshair sebagai lapisan cadangan. Fungsi `Game.GetCurrentCrosshairRef()` digunakan untuk mengambil objek yang sedang diarahkan oleh pemain melalui crosshair. Jika pemain tidak mengarah ke objek apa pun, sistem mengaktifkan Virtual Anchor Point, yaitu titik tujuan imajiner yang diproyeksikan sejauh 1.500 unit di depan pemain berdasarkan arah hadap karakter. Rumus yang digunakan adalah sebagai berikut:

$$X_{\{target\}} = X_{\{player\}} + (1500 \cdot \sin(\theta))$$

$$Y_{\{target\}} = Y_{\{player\}} + (1500 \cdot \cos(\theta))$$

Keterangan:

- X_{target} = koordinat X tujuan
- Y_{target} = koordinat Y tujuan
- X_{player} = koordinat X pemain
- Y_{player} = koordinat Y pemain
- θ = sudut rotasi horizontal pemain
- 1500 = jarak proyeksi tujuan dalam unit permainan

Sebagai contoh, pemain berada pada koordinat $X_{player} = 2000$ dan $Y_{player} = 3000$, lalu menghadap pada sudut $\theta = 90^\circ$. Karena $\sin(90^\circ) = 1$ dan $\cos(90^\circ) = 0$, maka hasil perhitungannya adalah:

- $X_{target} = 2000 + (1500 \times 1)$
- $X_{target} = 3500$
- $Y_{target} = 3000 + (1500 \times 0)$

- Ytarget = 3000

Berdasarkan hasil tersebut, sistem membuat titik tujuan imajiner pada koordinat (3500, 3000). Titik ini kemudian diwujudkan sebagai objek sementara XMarker dengan FormID 0x0000003B melalui fungsi PlaceAtMe(). Dengan cara ini, NPC tetap dapat bergerak ke arah yang dimaksud pemain meskipun perintah suara tidak menghasilkan target lokasi atau objek yang eksplisit. Struktur penentuan target navigasi ini ditampilkan pada Kode Program 4.22.

Kode Program 4. 22 Implementasi Fungsi HandleMovement() Berbasis Protokol Smart Splitter NLU dan Proyeksi Trigonometri

```
1. ; Kode 4.22 Implementasi Fungsi HandleMovement() Berbasis Protokol Smart
   Splitter NLU dan Proyeksi Trigonometri
2. Function HandleMovement(String targetPlugin, Int targetFormID)
3.     Actor SeranaRef = SeranaAlias.GetActorRef()
4.     Actor PlayerRef = Game.GetPlayer()
5.     ObjectReference FinalTarget = None
6.     Location FinalLocation = None
7.     If (targetPlugin != "" && targetPlugin != "GENERIC" && targetFormID != 0)
8.         Form nluForm = Game.GetFormFromFile(targetFormID, targetPlugin)
9.         If (nluForm)
10.            FinalLocation = nluForm as Location
11.            If (FinalLocation == None)
12.                FinalTarget = nluForm as ObjectReference
13.                If (FinalTarget == None)
14.                    FinalTarget =
Game.FindClosestReferenceOfTypeFromRef(nluForm, PlayerRef, 10000.0)
15.            EndIf
16.        EndIf
17.    EndIf
18. EndIf
19. If (FinalLocation == None && FinalTarget == None)
20.     FinalTarget = Game.GetCurrentCrosshairRef()
21. EndIf
22. If (FinalLocation)
23.     SeranaDestLocAlias.ForceLocationTo(FinalLocation)
24.     SeranaGoToState.SetValueInt(2)
25. ElseIf (FinalTarget)
26.     SeranaDestAlias.ForceRefTo(FinalTarget)
27.     SeranaGoToState.SetValueInt(1)
28. Else
29.     SeranaGoToState.SetValueInt(0)
30. EndIf
31. gotoTimeout = 0.0
32. SeranaRef.EvaluatePackage()
33. EndFunction
```

Setelah target berhasil ditentukan oleh fungsi HandleMovement(), proses pergerakan dialihkan ke mekanisme *Dual-Scanner* yang berjalan secara asinkron di dalam Event OnUpdate(). Mekanisme ini memantau nilai SeranaGoToState untuk menentukan jenis pemeriksaan navigasi yang digunakan. Jika nilainya 1, sistem mengaktifkan Mesin 1 untuk memantau jarak antara Serana dan objek target menggunakan SeranaRef.GetDistance(DestObj). Ketika jarak sudah mencapai

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

ambang ≤ 180.0 unit, sistem menghentikan pergerakan, menjalankan `DLC1MentalModel.Wait()`, lalu membersihkan alias tujuan.

Jika nilai `SeranaGoToState` adalah 2, sistem mengaktifkan Mesin 2 untuk memeriksa apakah `Serana` telah memasuki wilayah tujuan menggunakan `SeranaRef.IsInLocation(DestLoc)`. Pendekatan ini diperlukan karena `Creation Engine` tidak menyediakan mekanisme *event listener* bawaan untuk mendeteksi kedatangan NPC pada target dinamis. Dengan *Dual-Scanner*, sistem dapat memantau target objek fisik dan target wilayah secara terpisah tanpa membebani `Virtual Machine Papyrus` secara berlebihan.

Sistem juga menerapkan *deterministic timeout* selama 30 detik melalui variabel ``gotoTimeout`` agar kegagalan navigasi dapat dicegah. Jika `Serana` tidak mencapai target dalam batas waktu tersebut, sistem menghentikan proses navigasi, mengembalikan `SeranaGoToState` ke nilai 0, serta membersihkan alias agar skrip tidak terus mengunci proses pergerakan.

Selain itu, sistem melakukan *auto-cleanup* terhadap objek sementara `XMarker` yang dibuat sebagai titik tujuan imajiner. Jika objek target memiliki `FormID` dasar `0x0000003B`, sistem menjalankan `Disable()` dan `Delete()` untuk menghapusnya dari memori. Mekanisme ini penting untuk mencegah penumpukan objek sementara, mengurangi risiko *save bloat*, dan menjaga stabilitas `Papyrus` dalam jangka panjang. Logika *Dual-Scanner* dan mekanisme *timeout* ditampilkan pada Kode Program 4.23.

Kode Program 4. 23 Pipa Pemrosesan Polling Dual-Scanner Navigasi dan Protokol Pembersihan Otomatis `XMarker` pada `OnUpdate()`

```
1. ; Kode 4.23 Pipa Pemrosesan Polling Dual-Scanner Navigasi dan Protokol
   Pembersihan Otomatis XMarker pada OnUpdate()
2. Int goToMode = SeranaGoToState.GetValueInt()
3. If (goToMode == 1)
4.     ObjectReference DestObj = SeranaDestAlias.GetReference()
5.     If (DestObj)
6.         gotoTimeout += PollInterval
7.         If (SeranaRef.GetDistance(DestObj) <= 180.0)
8.             SeranaGoToState.SetValueInt(0)
9.             DLC1MentalModel.Wait()
10.            If (DestObj.GetBaseObject().GetFormID() == 0x0000003B)
11.                DestObj.Disable()
12.                DestObj.Delete()
13.            EndIf
14.            SeranaDestAlias.Clear()
15.            SeranaRef.EvaluatePackage()
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

16.         ElseIf (gotoTimeout > 30.0)
17.             SeranaGoToState.SetValueInt(0)
18.             SeranaDestAlias.Clear()
19.             SeranaRef.EvaluatePackage()
20.         EndIf
21.     EndIf
22. ElseIf (goToMode == 2)
23.     Location DestLoc = SeranaDestLocAlias.GetLocation()
24.     If (DestLoc && SeranaRef.IsInLocation(DestLoc))
25.         SeranaGoToState.SetValueInt(0)
26.         DLC1MentalModel.Wait()
27.         SeranaDestLocAlias.Clear()
28.         SeranaRef.EvaluatePackage()
29.     EndIf
30. EndIf

```

4.3.4.6 Sistem Afeksi, Akumulasi Skor, dan *Trigger* Dialog Otonom

Sistem manajemen hubungan sosial antara pemain dan karakter dikendalikan secara dinamis melalui lapisan mekanistik sistem afeksi. Fungsionalitas ini dirancang untuk mendeteksi, menyimpan, dan merespons akumulasi akumulatif dari seluruh interaksi verbal pemain yang telah diproses oleh *Server Cloud*. Struktur sistem afeksi ini diimplementasikan dengan membagi rentang skor sosial global yang disimpan pada komponen Global Variable *SeranaAffinityScore* ke dalam lima tingkatan kluster hubungan (*Affinity Tier*) terisolasi melalui fungsi *GetAffinityTier()*. Tingkatan tersebut memetakan skor numerik menjadi status psikologis bertingkat, meliputi Tier 1 (*Resentful* untuk skor ≤ -50), Tier 2 (*Distant* untuk skor ≤ -1), Tier 3 (*Neutral* untuk skor ≤ 49), Tier 4 (*Friend* untuk skor ≤ 89), hingga Tier 5 (*Partner* untuk skor > 89). Keputusan rekayasa untuk menyematkan logika pembatasan nilai batas atas dan bawah (*clamping protocol*) pada angka rentang $[-100, +100]$ dipilih secara teknis guna mencegah nilai afeksi keluar dari rentang logika hubungan yang telah ditentukan sistem. Dampak positif dari penerapan kalkulasi batas ini adalah konsistensi data jangka panjang karakter dapat dikunci secara aman, mencegah nilai status emosional melompat keluar dari batas logika konseptual gim, serta memastikan kalkulasi penentuan kluster hubungan berjalan secara konsisten. Logika pemetaan kluster afeksi sosial dan fungsi pembatasan batas ini diuraikan pada Kode Program 4.24.

Kode Program 4. 24 Fungsi Pemetaan Kluster Hubungan Afeksi dan Logika Clamping Akumulasi Skor

```

1. Int Function GetAffinityTier(Int score)
2.     If (score <= -50)
3.         Return 1
4.     ElseIf (score <= -1)
5.         Return 2

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
6.     ElseIf (score <= 49)
7.         Return 3
8.     ElseIf (score <= 89)
9.         Return 4
10.    Else
11.        Return 5
12.    EndIf
13. EndFunction
14. ; Potongan eksekusi akumulasi sentiment_shift di dalam Event OnUpdate()
15.     If (sentimentShift != 0)
16.         Int currentScore = SeranaAffinityScore.GetValueInt()
17.         Int newScore = currentScore + sentimentShift
18.         ; Protokol Clamping untuk mencegah variable overflow
19.         If (newScore > 100)
20.             newScore = 100
21.         ElseIf (newScore < -100)
22.             newScore = -100
23.         EndIf
24.         SeranaAffinityScore.SetValueInt(newScore)
```

Rutinitas mutasi status emosional dijalankan pada setiap siklus skrip dengan membaca nilai sentiment_shift dari *Inbound Payload* SeranaCommand.json. Nilai ini bernilai +5 untuk sentimen positif dan -5 untuk sentimen negatif. Setiap perubahan kemudian ditambahkan ke SeranaAffinityScore, sehingga sistem dapat menyimpan perkembangan hubungan sosial antara pemain dan Serana secara permanen di memori lokal gim.

Bagian utama dari mekanisme ini adalah pemicuan dialog otonom melalui *ForceGreet Mechanics* ketika terjadi perubahan kluster hubungan. Sistem membandingkan nilai kluster baru dengan kluster sebelumnya melalui kondisi `newTier != lastAffinityTier`. Jika nilai hubungan meningkat, sistem menganggap terjadi *Tier Up* dan mengirimkan kode sinyal 8 melalui SeranaAudioPointer.SetValueInt(8). Sebaliknya, jika hubungan menurun, sistem mengirimkan kode sinyal 9 sebagai tanda *Tier Down*. Setelah itu, fungsi UpdateCurrentInstanceGlobal() dan SeranaRef.EvaluatePackage() dipanggil agar Serana menghentikan antrean perilaku lain, mendekati pemain, dan memulai dialog yang merepresentasikan perubahan hubungan sosial.

Sistem menerapkan deterministic dialogue timeout selama 45 detik untuk mencegah Serana terkunci dalam kondisi ForceGreet. Variabel affinityTimeout digunakan untuk memantau durasi proses dialog. Jika waktu melebihi 45.0 detik dan dialog tidak berhasil dimulai atau selesai, fungsi ResetAffinityForceGreet() dijalankan. Fungsi ini mengembalikan SeranaAudioPointer ke nilai 0, menyetel

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

affinityState ke 0, dan menjalankan kembali SeranaRef.EvaluatePackage(). Dengan mekanisme ini, sistem dapat memulihkan status NPC secara mandiri tanpa merusak AI Package bawaan gim. Logika pemicuan dialog otonom dan timeout ditampilkan pada Kode Program 4.25.

Kode Program 4. 25 Protokol Pemicuan ForceGreet Otonom dan Mekanisme Penyelamat Timeout 45 Detik pada OnUpdate()

```

1. ; Kode 4.25 Protokol Pemicuan ForceGreet Otonom dan Mekanisme Penyelamat Timeout
45 Detik pada OnUpdate()
2. Int newTier = GetAffinityTier(newScore)
3. If (newTier != lastAffinityTier)
4.   If (newTier > lastAffinityTier)
5.     SeranaAudioPointer.SetValueInt(8)
6.   Else
7.     SeranaAudioPointer.SetValueInt(9)
8.   EndIf
9.   UpdateCurrentInstanceGlobal(SeranaAudioPointer)
10.  lastAffinityTier = newTier
11.  affinityState = 1
12.  affinityTimeout = 0.0
13.  SeranaRef.EvaluatePackage()
14. EndIf
15. Function ResetAffinityForceGreet(Actor SeranaRef)
16.   SeranaAudioPointer.SetValueInt(0)
17.   UpdateCurrentInstanceGlobal(SeranaAudioPointer)
18.   affinityState = 0
19.   affinityTimeout = 0.0
20.   SeranaRef.EvaluatePackage()
21. EndFunction

```



Gambar 4. 16 NPC melakukan pergerakan otomatis menghampiri pemain (ForceGreet) akibat terjadinya lonjakan akumulasi skor afeksi.

4.3.4.7 Mekanisme Sinyal Otonom Berbasis Konteks Dunia

Implementasi pemicuan perilaku komentar dialog secara otonom (Autonomous Context-aware Trigger) dirancang untuk mendeteksi fluktuasi variabel lingkungan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

lore gim secara luring. Komponen ini beroperasi penuh di dalam putaran fungsi peristiwa Event OnUpdate() melalui pemanfaatan variabel lokal signalCode bertipe data integer yang diinisialisasi dari angka bawaan 0. Sistem dikonfigurasi untuk mengidentifikasi tujuh kondisi pemicu kontekstual yang berbeda melalui serangkaian sensor lingkungan gim. Rentang fungsionalitas sinyal otonom tersebut mencakup pembagian waktu dunia kronologis, klasifikasi cuaca aktif, transisi spasial arsitektur sel, penyelesaian fase pertempuran, hingga perhitungan durasi keheningan aktivitas pemain. Distribusi pemetaan nomor parameter kode sinyal beserta matriks kondisi pemicu taktisnya dirincikan pada Tabel 4.8.

Tabel 4. 8 Matriks Kode Sinyal Otonom Lingkungan dan Parameter Pemicu Skrip Papyrus

Kode Sinyal	Kondisi Pemicu Kontekstual	Logika Papyrus (Real Syntax)	Tingkat Prioritas / Protokol Collision
1	Transisi waktu ke fase pagi hari (Pukul 06.00 s/d 12.00)	currentHour >= 6.0 && currentHour < 12.0	Tinggi: Beroperasi pada urutan evaluasi sekuensial awal berdasarkan mekanisme delta detection.
2	Transisi waktu ke fase siang hari (Pukul 12.00 s/d 19.00)	currentHour >= 12.0 && currentHour < 19.0	Tinggi: Beroperasi pada urutan evaluasi sekuensial awal berdasarkan mekanisme delta detection.
3	Transisi waktu ke fase malam hari (Pukul 19.00 s/d 06.00)	currentHour >= 19.0 atau jangkauan jam < 6.0	Tinggi: Beroperasi pada urutan evaluasi sekuensial awal berdasarkan mekanisme delta detection.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4	Perubahan iklim dunia aktif menuju kondisi hujan (Rain/Storm)	<code>cw && cw.GetClassification() == 2</code>	Sedang: Mengevaluasi kode klasifikasi engine secara spesifik untuk mengisolasi cuaca ekstrem.
5	Perpindahan spasial melintasi batas sel masuk menuju interior Dungeon	<code>isInteriorNow && currentLocation. HasKeyWord (Keyword.GetKeyWord ("LocTypeDungeon"))</code>	Sedang: Mengecek integritas 3D mesh via instruksi <code>Is3DLoaded()</code> setelah jeda asinkron 3.0 detik.
6	Resolusi penyelesaian fase pertempuran taktis (Post-Combat Resolution)	<code>wasInCombat && !isInCombat</code>	Tinggi: Memotong antrean sinyal lain secara instan dengan penundaan natural delay pacing protocol selama 2.0 detik.
7	Durasi keheningan interaksi pemain secara statis (Idle State >= 18 menit)	<code>idleTimer >= 1080.0</code>	Terendah: Dilindungi gerbang sekat If (<code>signalCode == 0</code>) sebagai mekanisme penyelamat agar tidak memicu signal collision.

Seluruh rutinitas pengenalan kondisi lingkungan dijalankan menggunakan mekanisme deteksi perubahan delta (*delta detection mechanism*). Mekanisme ini bekerja dengan membandingkan kondisi *real-time* saat ini dengan nilai historis dari siklus pembaruan sebelumnya. Nilai historis tersebut disimpan dalam beberapa variabel pelacak, seperti `lastTimeState`, `lastWeatherState`, `lastLocation`,



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

lastInteriorState, dan wasInCombat. Pendekatan ini digunakan untuk mencegah pengiriman sinyal yang berulang ketika kondisi lingkungan tidak berubah. Tanpa mekanisme delta, kondisi statis seperti cuaca hujan, waktu malam, atau keberadaan pemain di dalam dungeon dapat terus memicu dialog otonom pada setiap siklus OnUpdate(). Dengan demikian, beban eksekusi Virtual Machine Papyrus dapat ditekan dan stabilitas skrip tetap terjaga.

Alur sensor disusun secara sekuensial untuk menghindari tabrakan instruksi. Evaluasi dimulai dari pembacaan waktu permainan melalui currentHour, kemudian dipetakan ke dalam kategori pagi, siang, atau malam. Setelah itu, sistem memeriksa kondisi cuaca aktif menggunakan Weather.GetCurrentWeather(). Pemeriksaan berikutnya dilakukan pada lokasi Serana melalui SeranaRef.GetCurrentLocation() dan pengujian kata kunci LocTypeDungeon untuk mendeteksi apakah NPC berada di area interior berbahaya. Selain itu, sistem juga memantau perubahan status pertempuran melalui variabel wasInCombat.

Pada tahap akhir, sistem menghitung durasi diam pemain menggunakan idleTimer += PollInterval. Sinyal idle ditempatkan sebagai prioritas terendah dan hanya dapat aktif apabila signalCode masih bernilai 0. Dengan skema prioritas ini, sinyal idle tidak akan menimpa sinyal lingkungan lain yang lebih penting. Pendekatan tersebut mengurangi risiko *signal collision* dan memastikan tidak ada dua dialog otonom yang saling bertabrakan dalam satu siklus pembaruan.

Fase eksekusi dilakukan ketika signalCode != 0. Sistem mengirimkan kode sinyal ke SeranaAudioPointer, menyinkronkannya melalui UpdateCurrentInstanceGlobal(), lalu memicu respons suara menggunakan SeranaRef.Say(mainTopic). Sebelum audio diputar, sistem menerapkan jeda melalui Utility.Wait(), yaitu 2,0 detik untuk sinyal pasca-pertempuran dan 2,5 detik untuk sinyal kontekstual lain. Jeda ini digunakan sebagai *natural delay pacing protocol* agar respons Serana tidak terasa muncul terlalu tiba-tiba. Dengan mekanisme ini, dialog otonom dapat berjalan lebih natural, stabil, dan tidak membebani sistem permainan. Logika sensor otonom berbasis perubahan delta ini ditampilkan pada Kode Program 4.26.

Kode Program 4. 26 Pipeline Polling Sensor Sinyal Otonom Lingkungan Sekuensial pada OnUpdate()

```
1. ; Kode 4.26 Pipeline Polling Sensor Sinyal Otonom Lingkungan Sekuensial pada OnUpdate()
2. Int signalCode = 0
3. If (currentHour >= 6.0 && currentHour < 12.0)
4.     signalCode = 1
5. ElseIf (currentHour >= 12.0 && currentHour < 19.0)
6.     signalCode = 2
7. Else
8.     signalCode = 3
9. EndIf
10. Weather cw = Weather.GetCurrentWeather()
11. If (cw && cw.GetClassification() == 2)
12.     signalCode = 4
13. EndIf
14. Location currentLocation = SeranaRef.GetCurrentLocation()
15. Bool isInteriorNow = SeranaRef.GetParentCell().IsInterior()
16. If (isInteriorNow && currentLocation && currentLocation.HasKeyword(Keyword.GetKeyword("LocTypeDungeon")))
17.     signalCode = 5
18. EndIf
19. Bool isInCombat = SeranaRef.IsInCombat()
20. If (wasInCombat && !isInCombat)
21.     signalCode = 6
22. EndIf
23. If (!isInCombat && SeranaRef.GetDistance(Game.GetPlayer()) < 2000.0)
24.     idleTimer += PollInterval
25.     If (idleTimer >= 1080.0)
26.         signalCode = 7
27.         idleTimer = 0.0
28.     EndIf
29. Else
30.     idleTimer = 0.0
31. EndIf
32. wasInCombat = isInCombat
```

4.3.4.8 Penanganan Combat, Interaksi Lingkungan, dan Kontrol Follower

Fungsi HandleCombat() bertanggung jawab penuh atas manajemen penargetan taktis Follower selama fase pertempuran aktif. Proses penyaringan target diimplementasikan melalui skema penelusuran tiga lapis guna mengonversi luaran teks abstrak dari jaringan saraf menjadi entitas actor riil di dalam gim. Lapis pertama bekerja memeriksa validitas kombinasi berkas plugin dan kode alamat FormID spesifik yang dikirimkan oleh jaringan Cloud. Apabila variabel tersebut bernilai generik, sistem menurunkan pencarian menuju lapis kedua memanfaatkan Sensor Crosshair dengan memeriksa objek Game.GetCurrentCrosshairRef() yang diselaraskan dengan kecocokan korelasi nama melalui fungsi StringUtil.Find(). Sebagai mekanisme penyelamat lapis ketiga, jika entitas tetap tidak ditemukan, sistem menginisialisasi looping While sebanyak maksimal 15 iterasi memanfaatkan fungsi bawaan gim Game.FindRandomActor() dalam radius 2.000 unit untuk

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menyaring objek terdekat yang membawa status bermusuhan terhadap pemain (IsHostileToActor) atau sedang menyerang pemain.

Keputusan arsitektur untuk melakukan intervensi privilese status Follower via instruksi penimpaan sifat fisik berupa perintah SeranaRef.SetGhost(True) dipilih secara teknis guna melakukan kalibrasi ulang prioritas paket orientasi mesin (hard reset engine combat engagement). Dampak dari penerapan skema toggling Ghost Mode pratempur ini adalah sistem terbebas dari risiko kegagalan deteksi target akibat sumbatan geometri, serta menjamin karakter Follower akan mengunci dan menyerang entitas musuh sasaran secara absolut tanpa risiko mengalami disorientasi akibat pengaruh aggro dari objek lain di sekitar sel pertempuran. Fase penanganan taktis pertempuran tersebut didokumentasikan pada Kode 4.27.

Kode Program 4. 27 Implementasi Pipa Pemindaian Tiga Lapis Penargetan Musuh pada Fungsi HandleCombat()

```
1. ; Kode 4.27 Implementasi Pipa Pemindaian Tiga Lapis Penargetan Musuh pada Fungsi
   HandleCombat()
2. Function HandleCombat(String targetPlugin, Int targetFormID, String targetName)
3.     Actor SeranaRef = SeranaAlias.GetActorRef()
4.     Actor PlayerRef = Game.GetPlayer()
5.     Actor TargetActor = None
6.     If (targetPlugin != "GENERIC" && targetPlugin != "nearest_enemy" &&
       targetFormID != 0)
7.         TargetActor = Game.GetFormFromFile(targetFormID, targetPlugin) as Actor
8.     EndIf
9.     If (TargetActor == None && targetPlugin == "GENERIC")
10.        ObjectReference CrossRef = Game.GetCurrentCrosshairRef()
11.        If (CrossRef)
12.            TargetActor = CrossRef as Actor
13.        EndIf
14.    EndIf
15.    If (TargetActor == None)
16.        Int scanCount = 0
17.        While (scanCount < 15 && TargetActor == None)
18.            Actor FoundActor = Game.FindRandomActor(
19.                PlayerRef.GetPositionX(),
20.                PlayerRef.GetPositionY(),
21.                PlayerRef.GetPositionZ(),
22.                2000.0
23.            )
24.            If (FoundActor && (FoundActor.IsHostileToActor(PlayerRef) ||
FoundActor.GetCombatTarget() == PlayerRef))
25.                TargetActor = FoundActor
26.            EndIf
27.            scanCount += 1
28.        EndWhile
29.    EndIf
30.    If (TargetActor)
31.        TargetAlias.Clear()
32.        TargetAlias.ForceRefTo(TargetActor)
33.        SeranaRef.SetGhost(True)
34.        SeranaRef.EvaluatePackage()
35.        SeranaRef.StartCombat(TargetActor)
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
36.     EndIf
37. EndFunction
```

Fungsi `HandleInteract()` mengelola pemetaan komando manipulasi objek fisik di sekitar koordinat dunia gim berdasarkan hasil pengenalan intent interaksi lingkungan. Mekanisme pelacakan diimplementasikan dengan memanfaatkan fungsi perluasan pustaka eksternal `PapyrusUtil` berupa instruksi `Game.FindClosestReferenceOfTypeFromRef()` dalam radius jangkauan sejauh 2.500 unit, diimbangi oleh jaring pengaman `Sensor Crosshair` sebagai sistem cadangan darurat. Ketika target fisik berhasil dikunci ke dalam `Reference Alias` bernama `InteractAlias`, sistem memicu putaran pemindai jarak kedekatan (`Proximity Scanner`) secara asinkron di dalam putaran `Event OnUpdate()` dengan mengubah parameter saklar global `SeranaWaitState` ke nilai status 2.

Apabila jarak spasial antara posisi `Follower` dan objek sasaran terdeteksi telah memenuhi ambang batas ≤ 160.0 unit, instruksi interaksi fisik dieksekusi secara instan via perintah `TargetObj.Activate(SeranaRef)`. Pilihan arsitektur untuk menjembatani eksekusi penargetan menggunakan putaran `polling distance tracking` berkala dikombinasikan dengan batas toleransi waktu (*deterministic timeout*) selama 30 detik dipilih secara teknis karena sistem dasar komponen *game engine* tidak dibekali oleh fungsi perekaman peristiwa otomatis (*event listener*) untuk mendeteksi keberhasilan persinggahan rute objek bebas luring. Dampak positifnya adalah visualisasi gerakan animasi interaksi fisik karakter (seperti duduk pada objek kursi, menempa pada meja besi, atau mengaktifkan tuas pintu rahasia) dapat dieksekusi secara halus tanpa memicu gejala kelambatan skrip (*script lag*), serta berperan penting untuk membebaskan `Virtual Machine Papyrus` dari risiko kegagalan tumpukan instruksi memori jika rute `Follower` terhalang oleh struktur geometri dunia gim. Logika operasional integrasi fungsi interaksi lingkungan beserta sensor pelacak kedekatan fisiknya diuraikan pada Kode Program 4.28.

Kode Program 4. 28 Skrip Penanganan Fungsi `HandleInteract()` dan Blok `Polling Sensor Jarak Proximity Scanner` pada `OnUpdate()`

```
1. ; Kode 4.28 Skrip Penanganan Fungsi HandleInteract() dan Blok Polling Sensor
   Jarak Proximity Scanner pada OnUpdate()
2. Function HandleInteract(String targetPlugin, Int targetFormID)
3.     Actor SeranaRef = SeranaAlias.GetActorRef()
4.     ObjectReference FinalTarget = None
5.     If (targetPlugin != "" && targetPlugin != "GENERIC" && targetFormID != 0)
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```

6.         Form nluForm = Game.GetFormFromFile(targetFormID, targetPlugin)
7.         If (nluForm)
8.             FinalTarget = Game.FindClosestReferenceOfTypeFromRef(nluForm,
Game.GetPlayer(), 2500.0)
9.             EndIf
10.        EndIf
11.        If (FinalTarget == None)
12.            FinalTarget = Game.GetCurrentCrosshairRef()
13.        EndIf
14.        If (FinalTarget)
15.            InteractAlias.ForceRefTo(FinalTarget)
16.            SeranaWaitState.SetValueInt(2)
17.            interactTimeout = 0.0
18.            SeranaRef.EvaluatePackage()
19.        EndIf
20.    EndFunction
21.    If (SeranaWaitState.GetValueInt() == 2)
22.        ObjectReference TargetObj = InteractAlias.GetReference()
23.        If (TargetObj && SeranaRef.GetDistance(TargetObj) <= 160.0)
24.            SeranaWaitState.SetValueInt(0)
25.            DLC1MentalModel.Wait()
26.            Utility.Wait(1.0)
27.            TargetObj.Activate(SeranaRef)
28.            InteractAlias.Clear()
29.            SeranaRef.EvaluatePackage()
30.        EndIf
31.    EndIf

```

Manajemen kontrol Follower dasar serta kustomisasi gaya bertarung taktis dikendalikan secara harmonis melalui integrasi fungsi HandleFollow(), HandleWait(), HandleDismiss(), HandleTrade(), dan HandleTactical(). Seluruh penyesuaian perilaku utama disinkronisasikan secara langsung menuju metode internal objek Dawnguard bawaan gim melalui jembatan properti DLC1MentalModel.Wait(), DLC1MentalModel.StopWaiting(), dan DLC1MentalModel.Dismiss(). Keputusan desain untuk mengeksekusi instruksi pembersihan total terhadap seluruh muatan alias navigasi spasial (SeranaDestAlias.Clear() dan SeranaDestLocAlias.Clear()) serta penonaktifan saklar global tepat saat fungsi HandleFollow() dipanggil dipilih secara teknis untuk menerapkan protokol pembersihan status behavioral (behavioral state sterilization protocol). Dampak dari penerapan protokol pembersihan ini adalah sistem terhindar dari risiko tabrakan instruksi logika pergerakan (*Pathfinding conflict*), memastikan karakter kembali ke siklus pengikut normal secara instan, serta memastikan fungsi berjalan dengan baik pembukaan inventaris via perintah SeranaRef.OpenInventory(True).

Sementara itu, kustomisasi gaya bertarung taktis dijalankan pada fungsi HandleTactical() dengan membaca nilai variabel alfabetis tactical_mode dari



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

berkas *Payload* JSON. Modifikasi ini dieksekusi menggunakan instruksi `SeranaBase.SetCombatStyle()` dengan menarik data konfigurasional berkas kustom `SeranaAI_Core.esp` untuk pilihan mode "magic" (0x06D48A) dan mode "melee" (0x06D48B), atau berkas master `Dawnguard.esm` untuk mode "default" (0x011F34). Pilihan arsitektur pemuatan dinamis via fungsi `Game.GetFormFromFile()` ini berdampak langsung pada fleksibilitas efisiensi gaya tempur karakter Follower, memungkinkannya beradaptasi secara dinamis terhadap kondisi taktis di lapangan tanpa memerlukan modifikasi langsung terhadap skrip bawaan gim. Rangkaian fungsi kontrol asinkron Follower dan struktur manipulasi `Combat Style` internal tersebut diuraikan secara lengkap pada Kode Program 4.29.

Kode Program 4. 29 Skrip Implementasi Fungsi Kontrol Pengikut Dawnguard dan Overriding Modifikasi Combat Style

```

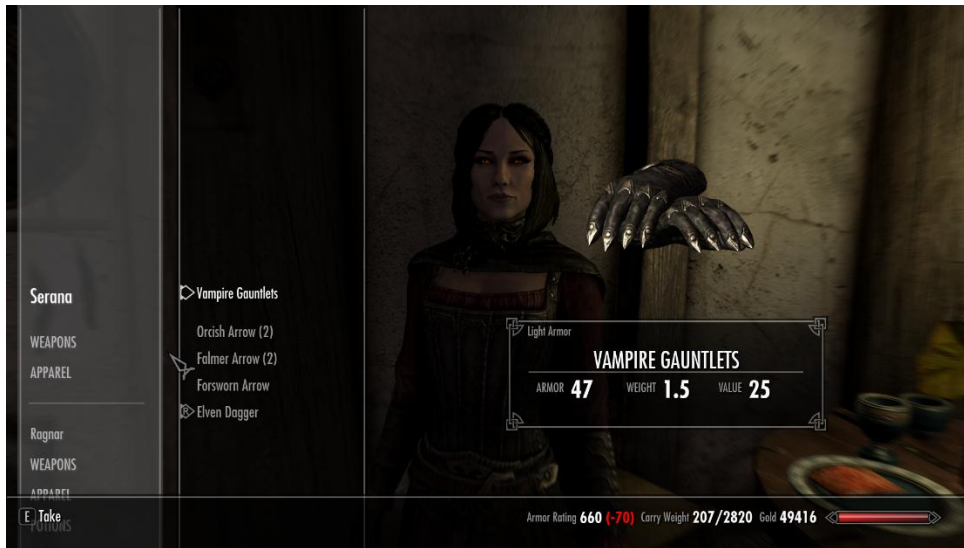
1. ; Kode 4.29 Skrip Implementasi Fungsi Kontrol Pengikut Dawnguard dan Overriding
   Modifikasi Combat Style
2. Function HandleFollow()
3.     Actor SeranaRef = SeranaAlias.GetActorRef()
4.     If (DLC1MentalModel)
5.         If (DLC1MentalModel.IsWaiting)
6.             DLC1MentalModel.StopWaiting()
7.         EndIf
8.         If (!DLC1MentalModel.IsFollowing || DLC1MentalModel.IsDismissed)
9.             DLC1MentalModel.EngageFollowBehavior(True)
10.        EndIf
11.        SeranaWaitState.SetValueInt(0)
12.        SeranaGoToState.SetValueInt(0)
13.        InteractAlias.Clear()
14.        SeranaRef.EvaluatePackage()
15.    EndIf
16. EndFunction
17. Function HandleTactical()
18.     Actor SeranaRef = SeranaAlias.GetActorRef()
19.     ActorBase SeranaBase = SeranaRef.GetActorBase()
20.     String tacticalMode = JsonUtil.GetStringValue("SeranaCommand.json",
"tactical_mode")
21.     String myModPlugin = "SeranaAI_Core.esp"
22.     If (tacticalMode == "magic")
23.         SeranaBase.SetCombatStyle(Game.GetFormFromFile(0x06D48A, myModPlugin) as
CombatStyle)
24.     ElseIf (tacticalMode == "melee")
25.         SeranaBase.SetCombatStyle(Game.GetFormFromFile(0x06D48B, myModPlugin) as
CombatStyle)
26.     ElseIf (tacticalMode == "default")
27.         SeranaBase.SetCombatStyle(Game.GetFormFromFile(0x011F34,
"Dawnguard.esm") as CombatStyle)
28.     EndIf
29.     SeranaRef.EvaluatePackage()
30. EndFunction

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 17 Antarmuka penukaran barang Inventory

4.3.5 Implementasi Integrasi Audio Generatif (*Baked Audio*)

Implementasi komponen audio generatif pada penelitian ini dirancang dengan menetapkan keputusan arsitektur anti-TTS-*Runtime*, di mana proses pembangkitan suara secara dinamis saat gim berjalan dihilangkan sepenuhnya guna memitigasi kendala latensi (*latency bottleneck*) dari *Inference Session* model yang dapat mengganggu imersi bermain pemain. Sebagai gantinya, seluruh aset vokal diproduksi secara luring pada fase pra-produksi untuk menghasilkan 424 file WAV yang mencakup 20 kategori dialog dengan variasi struktur empat dimensi (*Intent x TargetType x AffinityTier x Variant*). Aliran pengerjaan aset suara tersebut dioperasikan melalui *pipeline* dua tahap terintegrasi, yang menggabungkan ElevenLabs API (menggunakan model *eleven_v3* dan basis vokal *Elise*) sebagai generator prosodi dan emosi kalimat awal, dilanjutkan dengan proses konversi timbre vokal akhir menggunakan model Retrieval-based Voice conversion (RVC v2) luring. Pembahasan pada sub-bab ini diuraikan secara komprehensif yang mencakup tahapan ekstraksi dan kurasi dataset vokal primer dari arsip gim, konfigurasi parameter pelatihan model konversi vokal pada infrastruktur GPU, perancangan skema matriks variasi dialog hibrida, eksekusi standarisasi format audio Creation Engine, hingga konstruksi indeks relasional FormID multimedia untuk menutup siklus arsitektur *Hybrid Client-Cloud* secara utuh.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.3.5.1 Ekstraksi dan Kurasi *Dataset* Vokal Primer dari Arsip Dawnguard

Tahap penyusunan basis data suara target dilakukan dengan mengekstraksi dataset vokal primer karakter Serana langsung dari arsip gim. Proses ekstraksi ini diimplementasikan menggunakan tool Bethesda Archive Extractor untuk mengurai isi file biner .bsa (Bethesda Softworks Archive) yang membungkus seluruh aset audio milik plugin Dawnguard.esm. Hasil dari fase ekstraksi awal tersebut berupa kumpulan file dengan format bawaan asli gim yaitu .fuz. Format .fuz merupakan sebuah kontainer terkompresi khusus milik *game engine* yang mengemas komponen data gerakan bibir (.lip) dan biner audio (.xwm) secara bersamaan dalam satu kesatuan objek. Karena arsitektur jaringan saraf tiruan pada proses konversi vokal RVC v2 membutuhkan input gelombang suara linear murni tanpa enkapsulasi kontainer gim, seluruh file .fuz tersebut didekodekan dan dikonversi menjadi format .wav menggunakan tool Yakitori Audio Converter. Spesifikasi teknis dari sampel suara mentah asli (vanilla) hasil konversi akhir ini memiliki total durasi kumulatif selama 14 menit 48 detik, yang terdistribusi ke dalam format rekaman WAV Mono dengan frekuensi *Sample rate* sebesar 44100 Hz dan kedalaman bit sebesar 16-bit. Isi konten dari rekaman tersebut mencakup variasi ekspresi dialog bertema pertempuran (combat), penjelajahan (exploration), interaksi sistem pengikut (follower), hingga rangkaian dialog utama pada alur cerita ekspansi Dawnguard.

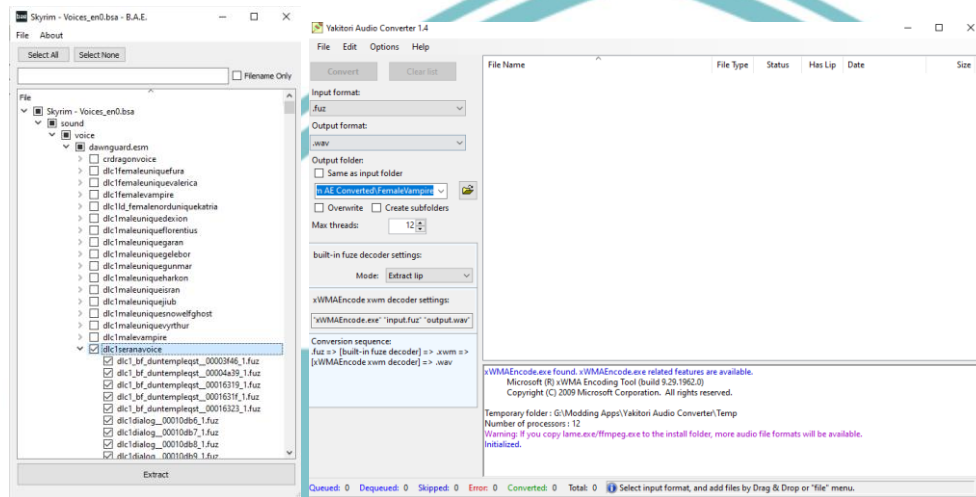
Keputusan rekayasa untuk mengekstraksi kontainer .fuz via Bethesda Archive Extractor dan mendekodekannya menjadi ekstensi .wav linear via Yakitori Audio Converter dipilih secara teknis guna mengisolasi karakteristik akustik murni dari pengisi suara asli secara bersih tanpa merusak atau mendistorsi struktur penanda konten linguistik di dalamnya. Justifikasi ilmiah dari pendekatan ini adalah karena seluruh rekaman vanilla di dalam arsip .bsa diproduksi di dalam studio rekaman profesional yang telah bebas dari gangguan noise dasar (noise floor), serta tidak melewati proses manipulasi digital seperti filtrasi peningkatan kualitas atau pengurangan derau (noise reduction) agresif yang berisiko mengikis pita suara atau memangkas frekuensi harmonik vokal target saat proses dekompresi berjalan. Dampak dari penggunaan dataset murni yang terisolasi dan terkonversi secara sempurna ini adalah model *Voice Cloning* RVC v2 mampu mempelajari

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

representasi fitur timbre vokal, resonansi laring, serta artikulasi emosional unik karakter secara akurat dan konsisten. Kebersihan sampel awal ini membebaskan arsitektur jaringan saraf dari risiko penyerapan anomali derau latar belakang, sehingga mencegah munculnya artefak distorsi suara robotik pada hasil keluaran sintesis akhir.



Gambar 4. 18 Tool Bethesda Archive Extractor dan Yakitori Audio Converter

Tabel 4. 9 Spesifikasi Teknis Parameter Akustik Dataset Vokal Primer Karakter Serana

Parameter Akustik	Spesifikasi Nilai Riil	Justifikasi Rekayasa Data
Durasi Total	14 Menit 48 Detik	Memenuhi batas ambang batas minimal kecukupan data untuk pemetakan <i>timbre</i> pada arsitektur RVC v2.
Format Kontainer	WAV (Waveform Audio File)	Menghindari kompresi <i>lossy</i> awal guna mempertahankan detail gelombang suara frekuensi tinggi.
Konfigurasi Kanal	Mono (1 Channel)	Menyelaraskan dimensi input <i>neural network</i> dan membuang redundansi spasial rekaman stereofonis.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Frekuensi <i>Sample rate</i>	44100 Hz	Memenuhi teorema Nyquist-Shannon untuk menangkap seluruh spektrum frekuensi vokal manusia secara utuh.
Kedalaman Bit <i>(Bit Depth)</i>	16-bit	Menyediakan rentang dinamis sebesar 96 dB untuk meminimalkan galat kuantisasi amplitudo sinyal suara.
Kondisi Akustik	Studio <i>Environment (0</i> <i>dB Noise Floor)</i>	Memastikan bobot model murni mempelajari karakteristik karakteristik suara target tanpa kontaminasi derau lingkungan.

Setelah proses ekstraksi dan konversi format selesai, diimplementasikan tahapan kurasi data secara luring melalui skrip validasi audio otomatis. Langkah pembersihan ini bekerja mengevaluasi integritas setiap berkas suara berdasarkan parameter ambang batas durasi dan amplitudo energi sinyal. Keputusan untuk menyaring dan mengeliminasi berkas-berkas sampel vokal yang memiliki durasi terlalu pendek (kurang dari 1.0 detik), tingkat kekerasan suara yang terlalu rendah (silent frame), atau yang terdeteksi membawa cacat kompresi dari *Runtime* engine gim dipilih secara teknis sebagai protokol penjaminan mutu data latih (data quality assurance protocol). Dampak dari penerapan fase kurasi bertahap ini adalah tersedianya kluster dataset vokal primer yang benar-benar bersih dan berbobot tinggi. Langkah ini secara efektif meminimalkan varians eror pada fungsi kerugian (loss function) saat proses pelatihan model berlangsung, serta menjamin efisiensi alokasi memori VRAM GPU selama komputasi ekstraksi fitur berjalan pada infrastruktur *cloud*.

4.3.5.2 Pelatihan Model *Voice conversion* RVC v2 pada Infrastruktur GPU RunPod

Tahap pembuatan model karakteristik vokal target diimplementasikan melalui proses pelatihan model generatif berbasis arsitektur Retrieval-based Voice conversion versi kedua (RVC v2). Model ini menggunakan pendekatan struktur Generative Adversarial Network (GAN) VITS yang dikombinasikan dengan modul

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

feature extractor berupa HuBERT Large. Komponen HuBERT Large merupakan sebuah model representasi suara berbasis pembelajaran mesin self-supervised yang bertugas mengekstraksi fitur linguistik dan informasi fonem dari gelombang suara input tanpa ketergantungan pada dokumen teks transkripsi. Keputusan rekayasa untuk menerapkan arsitektur RVC v2 dengan landasan HuBERT Large dipilih secara teknis guna mengisolasi kandungan konten kata dari sumber vokal prosodi, untuk kemudian dipetakan murni pada karakteristik pita suara target.

Dampak dari penerapan model ini adalah sistem mampu menghasilkan proses transfer identitas vokal yang jika dieksekusi dalam *Inference Session* akan berjalan sangat konsisten, di mana intonasi emosional dari basis suara Elise dapat dikonversi oleh warna suara asli Serana tanpa mengubah struktur artikulasi kalimat awal yang dikirimkan oleh komponen *Cloud*. Guna memberikan justifikasi rekayasa yang kuat terhadap pemilihan arsitektur ini, analisis komparasi komprehensif antara penggunaan generasi variasi RVC versi pertama dan versi kedua dirincikan pada Tabel 4.10.

Tabel 4. 10 Analisis Komparasi Arsitektur Teknologi Kloning Suara RVC v1 vs RVC v2

Parameter Teknis	Arsitektur RVC v1	Arsitektur RVC v2 (Dipilih)	Justifikasi Rekayasa untuk Kasus Serana AI
Dimensi Feature Extractor	256-Dimensi (HuBERT-Base)	1024-Dimensi (HuBERT-Large)	Dimensi 1024 pada v2 mampu mengisolasi fitur linguistik secara radikal untuk mencegah kebocoran (<i>leakage</i>) prosodi vokal asal.
Batas Atas Sampling Rate	Teroptimasi pada 32 kHz	Mendukung Penuh 40 kHz / 48 kHz	Standar aset audio di dalam <i>Creation Engine</i> beroperasi pada frekuensi 44.1 kHz. Output 40 kHz

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

			meminimalkan galat distorsi akibat <i>downsampling</i> .
Algoritma Pitch Extraction	Rentan memicu <i>robotic artifact</i> pada frekuensi tinggi	Integrasi Penuh dengan RMVPE	Algoritma RMVPE mampu melacak dinamika nada dasar secara halus, mengeliminasi desis logam (<i>metallic undertone</i>) pada vokal.
Efisiensi Index Retrieval	Pemetaan vektor cenderung kasar, rawan <i>overfitting</i>	Aproksimasi Vektor Lebih Rapat dan Presisi	Menghasilkan kemiripan suara yang konsisten meskipun ukuran durasi <i>dataset</i> primer relatif ringkas (14 menit 48 detik).

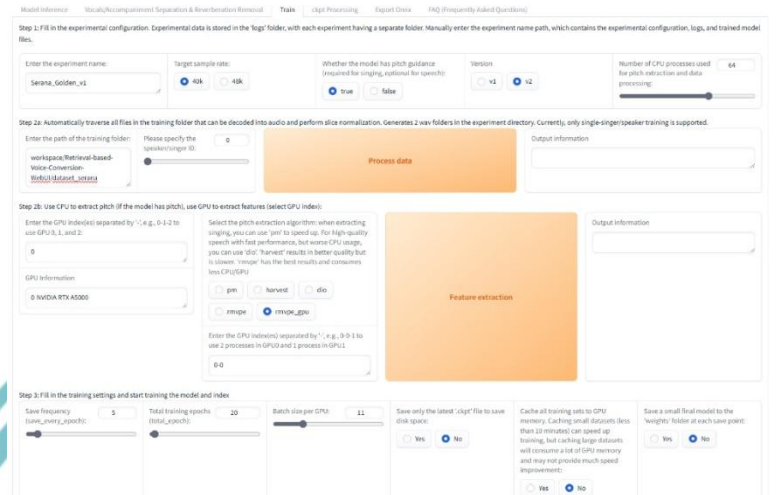
Pelaksanaan proses pelatihan model ini dieksekusi menggunakan alokasi infrastruktur komputasi *cloud* yang terisolasi di dalam container virtual pada platform RunPod. Spesifikasi perangkat keras yang disewa dikonfigurasi menggunakan unit GPU NVIDIA RTX A5000 yang dibekali kapasitas memori sebesar 24GB VRAM. Pilihan untuk memindahkan beban komputasi pelatihan menuju *Server* berbasis *Cloud* dipilih secara teknis karena teknologi ekstraksi frekuensi nada dasar berupa RMVPE (Robust Minimum Vocal Pitch Estimator) serta perhitungan pemetaan indeks kedekatan audio memerlukan komputasi matriks paralel intensif berskala tinggi yang tidak efisien jika dijalankan pada perangkat lokal berdaya rendah milik pengembang karena rawan memicu penurunan performa akibat kendala panas prosesor (*thermal throttling*).

Dampak positif dari pemanfaatan infrastruktur GPU RunPod ini adalah peningkatan kecepatan efisiensi durasi pelatihan per siklus epoch secara drastis, pemangkasan risiko kegagalan eksekusi akibat kehabisan alokasi memori (*Out of Memory*), serta penjaminan kestabilan komputasi bobot model di lingkungan sistem operasi yang bersih. Seluruh visualisasi penyediaan parameter latih pada antarmuka aplikasi

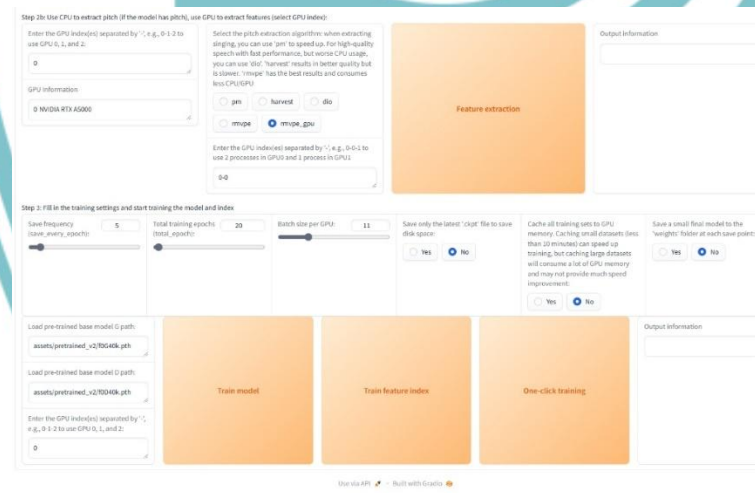
ditunjukkan pada Gambar 4.19 dan Gambar 4.20, sedangkan rincian nilai variabel pengeturan hyperparameters model dikunci secara presisi berdasarkan bukti eksekusi skrip train.py pada Tabel 4.11.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 19 Antarmuka RVC Web UI Step 1 dan Step 2a-2b untuk pemrosesan dataset Serana_Golden_v1 pada RunPod



Gambar 4. 20 Antarmuka RVC Web UI Step 3 untuk penyiapan hyperparameters dan pengekseskuan pelatihan model

Tabel 4. 11 Parameter Konfigurasi Hyperparameters Pelatihan Model RVC v2 Serana_Golden_v1

Parameter Konfigurasi	Nilai Konfigurasi Aktual	Justifikasi Ilmiah Komputasi
-----------------------	--------------------------	------------------------------

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Nama Eksperimen	Serana_Golden_v1	Identitas unik instansi untuk penanda folder penyimpanan dokumen <i>log</i> pelacakan.
Target Sample rate	40k (40000 Hz)	Menyelaraskan output biner suara agar kompatibel dengan batas toleransi <i>game Client</i> .
Total Epochs	250	Batas iterasi maksimum untuk mencapai konvergensi tanpa memicu kondisi <i>overfitting</i> .
Batch Size per GPU	12	Batas ukuran kluster data optimal untuk memaksimalkan kapasitas pipa <i>bandwith</i> VRAM RTX A5000.
Save Frequency	Setiap 25 <i>Epoch</i>	Batas interval pembuatan checkpoint biner .pth untuk menghemat alokasi ruang penyimpanan disk <i>Server</i> .
Pitch Extraction Algo	rmvpe_gpu	Memaksa proses pelacakan pitch dieksekusi paralel pada inti CUDA GPU untuk memangkas durasi latih.
Pretrained Base Model	f0G40k.pth / f0D40k.pth	Bobot dasar struktur generator dan discriminator GAN VITS v2 sebelum ditimpa fitur vokal target.

Selama pemrosesan berjalan di latar belakang container Linux RunPod, pengawasan konvergensi dievaluasi secara deterministik melalui pembacaan rekaman jejak biner terminal. Penentuan titik pemberhentian pelatihan pada Epoch ke-250 didasarkan pada analisis penurunan nilai fungsi kerugian spektrum amplitudo vokal yang menunjukkan pergeseran grafik melandai secara signifikan. Bukti empiris dari catatan log internal *Runtime* membuktikan bahwa komponen fungsi kerugian mel-spektrogram (loss_mel) berhasil ditekan dari nilai awal sebesar 35.568 pada siklus awal Epoch 1, bergerak turun secara stabil menjadi 22.590 pada

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Epoch 34, hingga menyentuh titik konvergensi terendah pada angka 17.590 pada fase akhir Epoch 247.

Keputusan untuk memotong durasi komputasi tepat saat fungsi kerugian berada di area landai (plateau stage) dipilih secara teknis sebagai protokol jaminan performa model guna menghindari fenomena kurang terlatih (undertrained) maupun terlalu matang (overtrained). Dampak dari ketepatan pemantauan berbasis data riil terminal ini adalah pencegahan terhadap munculnya artefak distorsi suara logam (metallic distortion), menjamin model mampu meniru kedalaman resonansi laring dan artikulasi napas Serana secara organik, serta mengunci ukuran kapasitas berkas model akhir pada batas memori yang efisien sebelum dipindahkan ke dalam komponen *Pipeline* konversi luring.

Kode Program 4. 30 Rekaman Berkas Log Terminal Pipa *Runtime* Pelatihan RVC v2 Serana_Golden_v1 pada Infrastruktur *Cloud*

```
1. 2026-04-30 09:19:20 | INFO | __main__ | Execute: "/usr/bin/python" train.py -e "Serana_Golden_v1" -sr 40k -bs 12 -te 250 -se 25 -v v2
2. INFO:Serana_Golden_v1:loaded pretrained assets/pretrained_v2/f0G40k.pth <All keys matched successfully>
3. INFO:Serana_Golden_v1:loaded pretrained assets/pretrained_v2/f0D40k.pth <All keys matched successfully>
4. INFO:Serana_Golden_v1:Train Epoch: 1 [0%]
5. INFO:Serana_Golden_v1:loss_disc=4.414, loss_gen=2.858, loss_fm=11.328, loss_mel=35.568, loss_kl=9.000
6. INFO:Serana_Golden_v1:====> Epoch: 1 [2026-04-30 09:20:11] | (0:00:19.866509)
7. INFO:Serana_Golden_v1:====> Epoch: 34 [2026-04-30 09:26:53] | (0:00:12.633886)
8. INFO:Serana_Golden_v1:loss_disc=3.898, loss_gen=3.214, loss_fm=11.050, loss_mel=22.590, loss_kl=1.370
9. INFO:root:Saving model and optimizer state at epoch 250 to ./logs/Serana_Golden_v1/G_2333333.pth
10. INFO:Serana_Golden_v1:saving ckpt Serana_Golden_v1_e250:Success.
11. INFO:Serana_Golden_v1:Train Epoch: 247 [67%]
12. INFO:Serana_Golden_v1:loss_disc=3.420, loss_gen=3.729, loss_fm=14.095, loss_mel=17.590, loss_kl=0.998
13. INFO:Serana_Golden_v1:====> Epoch: 250 [2026-04-30 10:10:55] | (0:00:16.450446)
14. INFO:Serana_Golden_v1:Training is done. The program is closed.
```

4.3.5.3 Perancangan Skema Dialog dan Sintesis via ElevenLabs API

Tahap inisialisasi prosodi awal dan pembangkitan konten linguistik awal dikelola melalui integrasi sistem pemrograman berbasis ElevenLabs API. Proses generasi vokal tiruan ini diimplementasikan dengan memanfaatkan fondasi model komersial mutakhir eleven_v3 yang dipadukan bersama profil vokal kustom bernama Elise. Keputusan rekayasa untuk memilih kerangka kerja ElevenLabs sebagai basis produksi gelombang suara awal dipilih secara teknis karena model eleven_v3 dibekali oleh parameter kemampuan kontrol emosi tingkat tinggi (high-fidelity

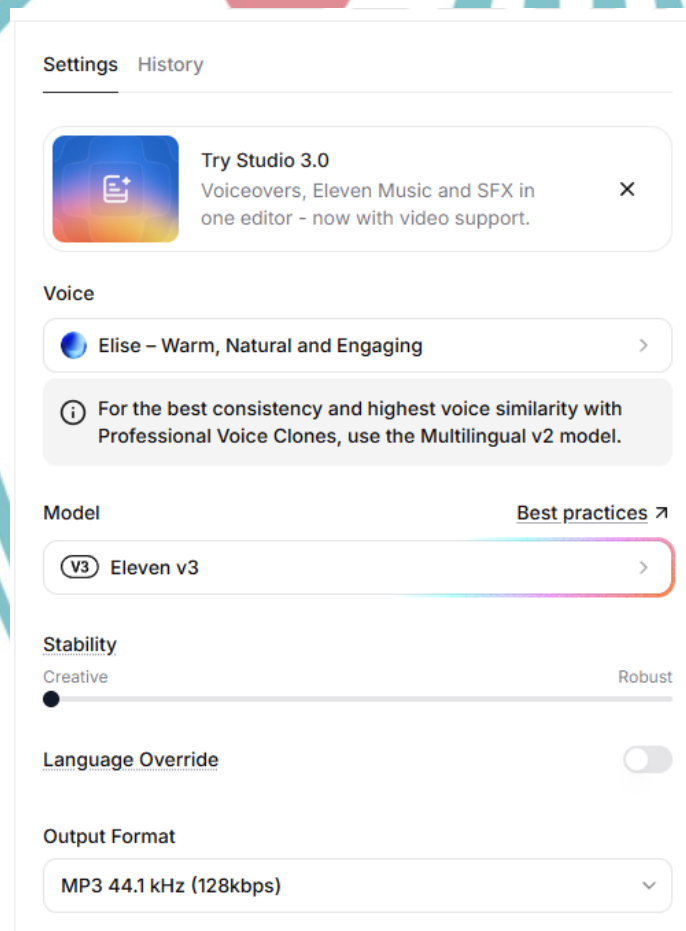


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

prosody control) yang mampu mempertahankan konsistensi penekanan suku kata (naturalness pacing) per kalimat naskah. Sementara itu, pemilihan karakteristik suara Elise yang membawa profil intonasi hangat, interaktif, serta berakksen American dipilih karena memiliki tingkat kompatibilitas register frekuensi yang tinggi dengan rentang nada dasar vokal target. Dampak dari keselarasan tinggi rendahnya gelombang awal ini adalah pemangkasan beban beban komputasi penyesuaian nada (pitch manipulation workload), sehingga meminimalkan risiko terjadinya deviasi distorsi pita suara ketika gelombang .wav keluaran ElevenLabs diumpankan menuju pipa pemrosesan konversi timbre akhir menggunakan arsitektur model RVC v2 luring.



Gambar 4. 21 UI ElevenLabs API/Studio dengan konfigurasi model eleven_v3, pemilihan Voice Elise, dan parameter sintesis

Guna menyediakan kerangka data respons yang andal, diimplementasikan struktur pemetaan naskah komprehensif berskala besar melalui perancangan skema matriks variasi dialog hibrida. Pendekatan eksperimental ini dirancang dalam wujud cetak

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

biru pemrosesan empat dimensi berbasis kombinasi parameter *Intent* x *TargetType* x *AffinityTier* x *Variant* untuk memproduksi secara massal 424 file WAV pra-produksi secara luring. Pilihan perancangan dimensi pengkondisian berlapis ini diterapkan secara teknis guna mengunci pemenuhan kebutuhan interaksi perilaku terhadap seluruh 10 kelas tindakan utama hasil klasifikasi model Multi-Task NLU (*Social_Chat*, *Social_Status*, *Movement_GoTo*, *Combat_Attack*, *Inventory_Trade*, *Movement_Wait*, *Movement_Follow*, *Interact_Environment*, *System_Dismiss*, dan *Combat_Tactical*), beriringan dengan penanganan dinamika komentar sinyal otonom lingkungan gim klien. Dampak dari penguncian distribusi matriks yang rapat ini adalah ketersediaan variasi pasokan data suara yang sangat kaya untuk menjamin otonomitas *Follower* tetap berjalan secara deterministik di berbagai skenario permainan terbuka. Seluruh pemetaan distribusi berkas *Baked Audio* dirincikan secara lengkap pada Tabel 4.12.

Tabel 4. 12 Distribusi Matriks Variasi 424 Berkas *Baked Audio* Berdasarkan Kategori *Intent*, *TargetType*, dan Jumlah Varian per Tier Afeksi

Kategori Dialog	TargetType	Affinity Tier	Jumlah Varian	Total Berkas
<i>Follow</i>	<i>Normal</i>	5	4	20
<i>Wait</i>	<i>Normal</i>	5	4	20
<i>Trade</i>	<i>Normal</i>	5	4	20
<i>GoTo</i>	<i>Normal</i>	5	4	20
<i>Dismiss</i>	<i>Normal</i>	5	4	20
<i>Chat</i>	<i>Normal</i>	5	5	25
<i>Status</i>	<i>Normal</i>	5	5	25
<i>Objective</i>	<i>Normal</i>	5	5	25
<i>IntEnv</i>	<i>Normal</i>	5	5	25



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

<i>Attack</i>	<i>Male / Female / Creature</i>	5	3	45
<i>Tactical</i>	<i>Male / Female / Creature</i>	5	3	45
<i>Morning</i>	<i>Normal</i>	5	3	15
<i>Day</i>	<i>Normal</i>	5	3	15
<i>Night</i>	<i>Normal</i>	5	3	15
<i>Rain</i>	<i>Normal</i>	5	3	15
<i>Dungeon</i>	<i>Normal</i>	5	3	15
<i>CombatEnd</i>	<i>Normal</i>	5	3	15
<i>AffinityDown</i>	<i>Normal</i>	4*	3	12
<i>AffinityUp</i>	<i>Normal</i>	4*	3	12
<i>Idle</i>	<i>Normal</i>	5	4	20
TOTAL				424

*AffinityDown tidak memiliki tier Partner; AffinityUp tidak memiliki tier Resentful mencerminkan logika psikologis asimetri transisi tier afeksi

Poin penting dari aspek rancangan rekayasa bahasa pada sistem afeksi ini adalah penerapan skema pengkondisian asimetri psikologis transisi hubungan (Psychological Asymmetry State Transition) khusus untuk kategori pemicuan dialog otonom AffinityUp (Sinyal 8) dan AffinityDown (Sinyal 9). Struktur taksonomi ini dikonfigurasi secara sengaja dengan mereduksi total jangkauan hubungan menjadi 4 tier saja, di mana kluster AffinityDown diprogram untuk tidak memiliki opsi baris dialog pada tingkatan hubungan terbawah (Resentful State), sedangkan kluster AffinityUp diisolasi penuh agar tidak menyediakan aset suara pada tingkatan hubungan tertinggi (Partner State). Pilihan isolasi asimetris ini diambil secara teknis sebagai mekanisme validasi logika (boundary logic filter) demi menyelaraskan alur penalaran rasional mesin. Dampak dari penerapan aturan



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

transisi asimetris ini adalah pencegahan mutlak terhadap potensi kegagalan decision engine awal dalam mengeksekusi referensi berkas audio yang tidak logis, karena entitas Follower yang berada pada status emosional terendah tidak akan mungkin mengalami penurunan derajat hubungan lebih jauh lagi, begitu pula sebaliknya pada fase status kedekatan sosial maksimum pemain.

Penerapan pengkondisian struktural kustom juga disuntikkan secara dinamis pada penentuan parameter dimensi pelacak tempur dan filter pembatasan kejenuhan dialog (anti-repetition filter). Desain ini diwujudkan dengan memecah menu pertempuran Attack dan Tactical untuk membawa dimensi sekunder khusus berupa penargetan TargetType yang diklasifikasikan ke dalam tipe musuh pria (Male), wanita (Female), serta makhluk monster (Creature). Di samping itu, alokasi jumlah varian per tier hubungan sosial ditingkatkan secara agresif pada batas puncak tertinggi sebanyak 5 pilihan kalimat berbeda khusus untuk pemicuan interaksi kosmetik Chat, interogasi variabel memori Status, pemindaian Objective, hingga manipulasi fisik sekitar Interact_Environment (IntEnv).

Keputusan untuk memperluas jangkauan tipe target tempur dan melipatgandakan indeks varian kalimat pada fungsi-fungsi kosmetik tersebut dipilih secara teknis karena kelas-kelas intent tersebut merupakan kluster komando yang paling intensif dipanggil secara berulang-ulang oleh pemain dalam sesi normal permainan. Dampak positif dari pengalokasian variasi kuantitas yang padat ini adalah eliminasi terhadap gejala kejenuhan akustik akibat repetisi dialog yang monoton, mengamankan keandalan fungsi komparasi pemilihan acak data dari Server, serta mendongkrak kualitas imersi visual narasi pertempuran kontekstual gim secara berkelanjutan.

4.3.5.4 Konversi *Timbre* via RVC v2, Sinkronisasi Berkas .lip, dan Standardisasi Format *Creation Engine*

Tahap kedua dari *pipeline* produksi audio diimplementasikan melalui proses konversi karakteristik suara (timbre mapping) menggunakan arsitektur RVC v2 luring. Proses ini bekerja dengan mengumpankan seluruh berkas WAV luaran ElevenLabs yang berada di dalam folder penyimpanan lokal Dataset Serana TTS menuju antarmuka tab Batch Conversion pada RVC Web UI. Keputusan rekayasa

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

untuk menetapkan nilai parameter Transpose sebesar 0 dipilih secara teknis karena register frekuensi dasar antara model vokal asal (Voice Elise) dan target (Voice Serana) sudah berada pada rentang gelombang akustik yang kompatibel, sehingga tidak memerlukan pergeseran nada biner (pitch shift semitones).

Guna menjamin keandalan ekstraksi karakteristik vokal target selama proses inferensi massal, parameter Search Feature Ratio dikunci pada nilai optimal 0,75. Pilihan nilai rasio ini diterapkan secara teknis sebagai mekanisme penyeimbang kekuatan aksent (accent strength control protocol). Jika parameter disetel terlalu rendah, model akan kehilangan karakteristik bisikan berat khas Serana, namun jika disetel terlalu tinggi mendekati angka 1,0, struktur gelombang audio akan mengalami kebocoran pemetaan indeks yang memicu munculnya distorsi suara pecah. Dampak positif dari kombinasi pengaturan inferensi ini adalah pemindahan identitas karakteristik suara target dapat berjalan secara konsisten dan natural pada seluruh 424 file WAV, tanpa merusak dinamika emosional kalimat awal yang telah dibentuk oleh ElevenLabs. Konfigurasi parameter inferensi batch konversi RVC v2 didokumentasikan secara lengkap pada Tabel 4.13.

Single Inference

Batch Inference

Batch conversion. Enter the folder containing the audio files to be converted or upload multiple audio files. The converted audio will be output in the specified folder (default: 'opt').

Transpose (integer, number of semitones, raise by an octave: 12, lower by an octave: -12):

Specify output folder:

Path to the feature index file. Leave blank to use the selected result from the dropdown:

Auto detect index path and select from the dropdown:

Select the pitch extraction algorithm ('pm': faster extraction but lower-quality speech; 'harvest': better bass but extremely slow; 'crepe': better quality but GPU intensive; 'rmvpe': best quality, and little GPU requirement)

☐ pm
☐ harvest
☐ crepe
☒ rmvpe

Export file format

☒ wav
☐ flac
☐ mp3
☐ m4a

Enter the path of the audio folder to be processed (copy it from the address bar of the file manager):

Resample the output audio in post processing to the final sample rate. Set to 0 for no resampling:

Adjust the volume envelope scaling. Closer to 0, the more it mimicks the volume of the original vocals. Can help mask noise and make volume sound more natural when set relatively low. Closer to 1 will be more of a consistently loud volume:

Protect voiceless consonants and breath sounds to prevent artifacts such as tearing in electronic music. Set to 0.5 to disable. Decrease the value to increase protection, but it may reduce indexing accuracy:

If >= 3, apply median filtering to the harvested pitch results. The value represents the filter radius and can reduce breathiness.

Search feature ratio (controls accent strength, too high has artifacting):

Drop File Here
- Or -
Click to Upload

Output information

Convert

- Hak Cipta :**
- Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
 - Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 - Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4. 22 Antarmuka tab Batch Conversion RVC Web UI yang menampilkan jalur direktori lokal input, pemilihan file indeks Serana_Golden_v1, dan penyesuaian parameter inferensi

Tabel 4. 13 Parameter Konfigurasi Inferensi Massal RVC v2 Tahap Batch Conversion

Parameter Inferensi	Nilai Konfigurasi	Justifikasi Rekayasa Rekayasa Kontrol
<i>Transpose</i>	0	Mempertahankan register nada asli karena frekuensi dasar Elise dan Serana sudah kompatibel.
<i>Index File Path</i>	...Serana_Golden_v1_v2.index	Merujuk pada <i>file</i> indeks kerapatan vektor 1024-dimensi untuk meminimalkan deviasi kloning.
<i>Pitch Extraction Algo</i>	rmvpe	Menggunakan estimasi nada dasar tangguh untuk mengunci akurasi intonasi kalimat panjang.
<i>Export File Format</i>	wav	Mengunci kontainer gelombang linear tanpa kompresi <i>lossy</i> sebelum standarisasi <i>engine</i> .
<i>Volume Envelope</i>	1	Menyeleraskan skala amplifikasi amplitudo volume agar identik dengan dinamika vokal target.
<i>Protect</i>	0	Mematikan proteksi konsonan proteksi jika dirasa artikulasi



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		desis vokal target sudah sangat bersih.
<i>Median Filtering</i>	3	Menerapkan radius filter median bernilai 3 untuk mereduksi gejala <i>breathiness</i> (derau napas berlebih).
<i>Search Feature Ratio</i>	0,75	Titik optimum untuk mengunci kemiripan warna vokal asli tanpa memicu munculnya <i>robotic artifact</i> .

Setelah seluruh 424 berkas audio kustom berhasil ditimpa oleh karakteristik suara target, diimplementasikan prosedur standardisasi format wajib Creation Engine serta pembentukan komponen sinkronisasi gerakan mulut melalui pembentukan berkas biner .lip. Proses pembentukan berkas .lip ini dieksekusi menggunakan *lip-sync* generator tool bawaan dengan membaca naskah transkripsi teks kalimat untuk dikonversikan menjadi data matriks representasi fonem gerakan mulut (mouth mesh phoneme synchronization data). Kontribusi rekayasa pada tahap standardisasi ini adalah kewajiban untuk memastikan bahwa seluruh berkas audio final dikunci secara mutlak pada spesifikasi teknis WAV Mono, frekuensi *Sample rate* 44100 Hz, dengan kedalaman bit sebesar 16-bit, serta diletakkan berdampingan secara simetris dengan berkas penanda .lip di dalam folder virtual internal gim `Sound/Voice/SeranaAI_Core.esp/DLC1SeranaVoice/`.

Justifikasi ilmiah dari penerapan standardisasi spesifikasi frekuensi tunggal dan pembuatan aset penanda mulut ini diambil secara teknis karena arsitektur subsistem audio Creation Engine bersifat sangat kaku dan tertutup. Apabila terdapat satu saja berkas audio yang memiliki *Sample rate* menyimpang dari angka 44100 Hz (misalnya 40000 Hz bawaan output mentah RVC), atau tidak dibekali oleh berkas identitas .lip yang bernama identik heksadesimal dengan berkas .wav sasarannya, *game engine* secara otomatis akan mengalami kegagalan silent playback (gim berjalan tanpa mengeluarkan suara sama sekali tanpa melempar notifikasi galat)

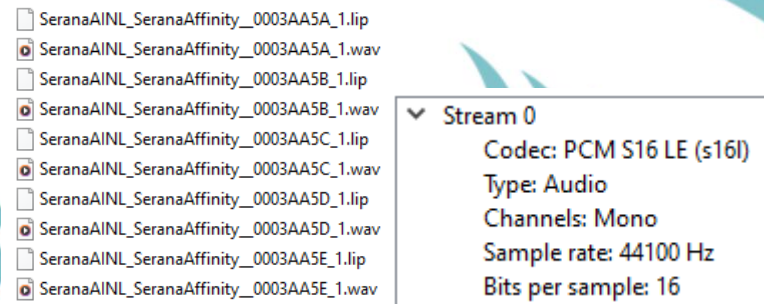


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

atau memicu malafungsi silent lip animation (suara terdengar tetapi mulut karakter membeku kaku). Dampak dari pemenuhan standardisasi format biner dan injeksi berkas .lip massal ini adalah jaminan integrasi multimedia yang aman pada komputer klien, menghasilkan visualisasi gerakan bibir pengikut yang sinkron secara *real-time* dengan ucapan generatif luring, serta membebaskan sistem dari risiko korupsi stack memory audio gim dalam jangka panjang.



Gambar 4. 23 Proses batch validation spesifikasi format audio WAV 44100Hz Mono menggunakan tool audio editor dan struktur direktori simetris berkas .wav dan .lip

4.3.5.5 Konstruksi Skema *FormID* dan Integrasi *papyrus_Voicelines.json*

Fase pemungutan rute data paling akhir dalam penyelarasan multimedia terdistribusi ini dikendalikan melalui registrasi identitas objek suara ke dalam database local *game Client*. Seluruh 424 file WAV hasil standardisasi konversi timbre didaftarkan secara resmi ke dalam komponen *Dialogue Views* menggunakan modul penyunting Creation Kit untuk dipetakan ke dalam struktur plugin *SeranaAI_Core.esp*.

Proses perekaman dialog di dalam Creation Kit secara otomatis akan melakukan enkapsulasi dan menerbitkan nama berkas fisik baru yang mengikat alamat penunjuk referensi memori berupa nilai *FormID* heksadesimal statis delapan digit, seperti ekstensi *SeranaAINL_SeranaNLU_Topic_0003A8EA_1.wav* untuk kluster komando atau *SeranaAINL_SeranaAffinity_0003AA5D_1.wav* untuk komentar afeksi. Keputusan rekayasa untuk mengunci aset multimedia melalui penamaan otomatis berbasis *FormID* statis bawaan engine dipilih secara teknis guna mematuhi aturan penataan susunan pemuatan berkas gim (Load Order). Justifikasi ilmiahnya adalah agar *game engine* dapat mengenali dan memanggil alamat memori indeks objek vokal secara instan dari piringan disk lokal sejak inisialisasi awal gim

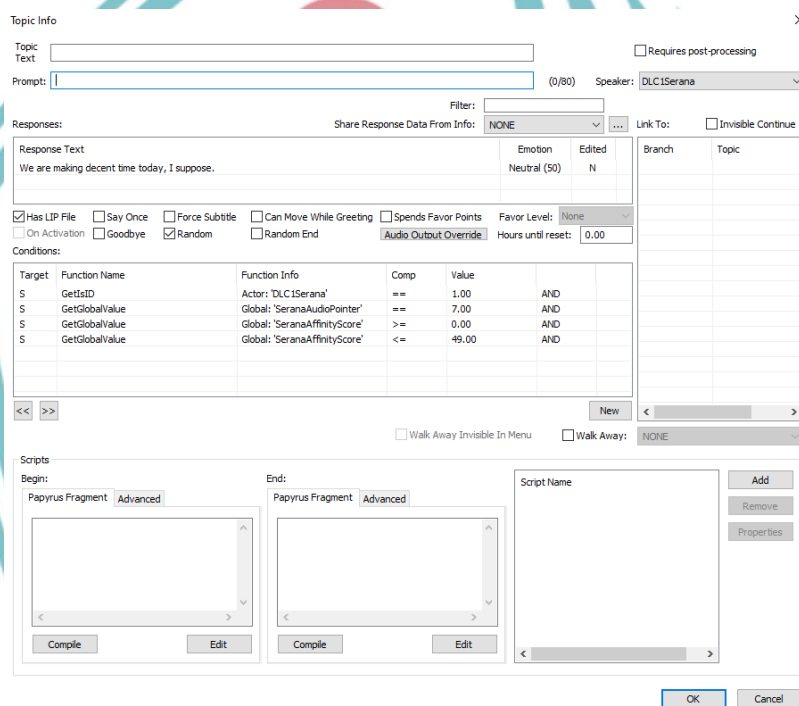
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dimulai, tanpa memicu gejala hilangnya baris suara pengikut (missing Voice line bug).

Integrasi pengkondisian berlapis ini dikunci secara ketat melalui jendela Topic Info Dialogue Editor seperti yang ditunjukkan pada Gambar 4.24, di mana baris suara diikat menggunakan kombinasi gerbang kondisi fungsi internal gim meliputi pemeriksaan identitas aktor (GetIsID), pengecekan saklar penunjuk audio global (SeranaAudioPointer), serta ambiguitas akumulasi nilai kedekatan emosional (SeranaAffinityScore).



Target	Function Name	Function Info	Comp	Value	AND
S	GetIsID	Actor: 'DLC1Serana'	==	1.00	AND
S	GetGlobalValue	Global: 'SeranaAudioPointer'	==	7.00	AND
S	GetGlobalValue	Global: 'SeranaAffinityScore'	>=	0.00	AND
S	GetGlobalValue	Global: 'SeranaAffinityScore'	<=	49.00	AND

Gambar 4. 24 UI Topic Info Dialogue Editor pada Creation Kit yang menampilkan gerbang kondisi nilai SeranaAudioPointer dan SeranaAffinityScore. Guna menghubungkan basis penunjuk data heksadesimal lokal klien tersebut dengan kecerdasan buatan jarak jauh, dibangun dokumen indeks relasional hibrida eksternal bernama papyrus_Voicelines.json di sisi Server Cloud. Berkas text-based payload ini dikonstruksikan menggunakan arsitektur kamus pemetaan berlapis (Multi-Level Dictionary Lookup) empat dimensi. Pemetaan berlapis ini bertindak sebagai jembatan penerjemah yang mengonversi label penamaan naskah pra-produksi di awal (Intent x TargetType x AffinityTier x Variant) untuk dipetakan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

secara simetris menuju array kumpulan string alamat heksadesimal FormID target hasil keluaran Creation Kit.

Keputusan arsitektur untuk menyusun skema nested dictionary layout ini dipilih secara teknis untuk mengoptimalkan efisiensi performa pencarian data awal, di mana pergerakan filter diprogram untuk memeriksa kecocokan parameter secara hierarkis dimulai dari pengenalan kelas intensi tindakan pada tingkat pertama (Level 1: *Intent*), dilanjutkan klasifikasi entitas taktis (Level 2: *TargetType*), segmentasi derajat hubungan emosional (Level 3: *AffinityTier*), hingga bermuara pada larik penampung array kumpulan string FormID varian vokal (Level 4: *Array FormID*). Dampak positif dari penerapan skema pencarian berlapis ini adalah pemotongan kompleksitas waktu operasi kueri (*query time complexity optimization*) menuju titik paling minimal, sehingga membebaskan memori kerja *RAM Server* dari beban pemindaian larik data linear yang panjang dan menjamin waktu respons pengembalian parameter biner berada pada tingkat latensi terendah. Protokol struktur nested dictionary ini didokumentasikan pada Kode Program 4.31.

Kode Program 4. 31 Potongan Struktur Skema Multi-Level Nested Dictionary pada papyrus_Voicelines.json

```

1. ; Kode 4.31 Potongan Struktur Skema Multi-Level Nested Dictionary pada
   papyrus_Voicelines.json
2. "Attack": {
3.     "Male": {
4.         "Resentful": ["0x03A99E", "0x03A99F", "0x03A9A0"],
5.         "Neutral":  ["0x03A9A4", "0x03A9A5", "0x03A9A6"]
6.     },
7.     "Creature": {
8.         "Partner": ["0x03A9C8", "0x03A9C9", "0x03A9CA"]
9.     }
10. },
11. "Follow": {
12.     "Normal": {
13.         "Neutral": ["0x03A8DE", "0x03A8DF", "0x03A8E0", "0x03A8E1"]
14.     }
15. }

```

Ketika fungsi `fetch_audio_formid()` pada *Server Cloud* berhasil menemukan kondisi emosional yang sesuai, sistem memilih salah satu nilai *FormID* audio dari daftar varian menggunakan `random.choice()`. Nilai tersebut kemudian dibungkus ke dalam *Response Body* berbentuk *payload* JSON dan dikirim kembali ke komputer klien melalui endpoint publik pada *VPS DomaiNesia*. Dengan mekanisme ini, server tidak hanya mengirim hasil klasifikasi NLU, tetapi juga menentukan indeks audio yang harus diputar oleh sistem permainan.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tahap penting dalam integrasi audio ini terletak pada sinkronisasi antara *Inbound Payload* SeranaCommand.json, variabel global Papyrus, dan mekanisme pemutaran dialog di dalam *game engine*. Setelah payload diterima oleh skrip lokal, nilai audioFormID disuntikkan ke dalam variabel SeranaAudioPointer melalui SeranaAudioPointer.SetValueInt(audioFormID). Setelah itu, sistem menjalankan UpdateCurrentInstanceGlobal() untuk memastikan nilai pointer terbaru tersinkronisasi dengan instance aktif di dalam permainan. Sebelum instruksi SeranaRef.Say(mainTopic) dijalankan, skrip menambahkan jeda singkat melalui Utility.Wait(0.25).

Jeda 0,25 detik tersebut digunakan sebagai protokol penyeimbang propagasi memori. Tanpa jeda ini, Papyrus berisiko menjalankan dialog sebelum nilai global SeranaAudioPointer benar-benar diperbarui. Kondisi tersebut dapat memicu *race condition*, yaitu situasi ketika sistem dialog membaca nilai lama atau nilai yang belum sepenuhnya tersinkronisasi. Dengan adanya jeda mikro tersebut, pemutaran audio dapat berlangsung lebih stabil, sesuai dengan FormID yang dikirim dari server, serta tetap selaras dengan animasi gerakan bibir NPC.

Rangkaian ini menjadi tahap akhir dari alur *Hybrid Client-Cloud* yang dikembangkan pada Sub-bab 4.3. Proses dimulai dari penangkapan suara pemain dan transkripsi lokal, dilanjutkan dengan inferensi *Multi-Task NLU* pada Server Cloud, kemudian pemetaan hasil klasifikasi ke dalam papyrus_Voicelines.json melalui fetch_audio_formid(). Nilai audio_formid yang dikembalikan server dibaca oleh skrip Papyrus melalui SeranaCommand.json, dimasukkan ke SeranaAudioPointer, dan dieksekusi melalui SeranaRef.Say(mainTopic).

Dengan demikian, 424 berkas *Baked Audio* beserta pasangan berkas .lip berfungsi sebagai aset respons verbal yang dipanggil secara deterministik berdasarkan hasil NLU. Integrasi ini menyatukan lapisan input suara, komputasi AI cloud, eksekusi Papyrus, dan respons audio-visual NPC dalam satu pipeline yang stabil, kohesif, dan tetap ringan bagi runtime utama permainan.

4.4 Pengujian Sistem

Pengujian sistem pada penelitian ini diimplementasikan menggunakan pendekatan *Mixed Methods* (metode campuran) yang mengombinasikan metode kualitatif dan



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

kuantitatif melalui empat instrumen evaluasi terintegrasi guna memvalidasi keandalan, akurasi arsitektur, persepsi pengguna, serta performa non-fungsional dari keseluruhan ekosistem *Hybrid Client-Cloud*. Evaluasi fungsional pertama dioperasikan melalui metode *Black Box Testing* dengan teknik Use Case Testing yang diderivasi langsung dari dokumen spesifikasi sistem, mencakup 98 test case terdistribusi untuk memvalidasi mekanisme pengaman (*failsafe protocol*) tindakan serta fungsionalitas mekanisme delta detection pada komponen *game engine* lokal klien. Lapis kedua dilaksanakan melalui studi komparasi kuantitatif untuk memvalidasi arsitektur kecerdasan buatan, membandingkan performa model bahasa baseline DistilBERT terhadap model final RoBERTa-base *Multi-Task Learning* bersandarkan pada data log eksperimen riil yang mengukur metrik akurasi, Macro F1-Score, efisiensi kapasitas penyimpanan File, serta tingkat kecepatan *Inference Session* pada ONNX Runtime.

Selanjutnya, persepsi subjektif dan tingkat ketergunaan antarmuka diukur secara empiris dengan menyebarkan kuesioner gabungan antara indikator *User Acceptance Testing* (UAT) dan *System Usability Scale* (SUS) dalam satu formulir kuesioner tunggal kepada 30 responden riil untuk memetakan tingkat kepuasan pengguna berdasarkan korelasi varians spesifikasi perangkat keras komputer pengetes. Akhirnya, pemprofilan performa non-fungsional (*System Profiling*) dieksekusi secara deterministik untuk mengurai akumulasi waktu total latensi *Round-Trip Time* (RTT) paket data internet dari awal ke akhir serta membuktikan keterisolasian utilitas perangkat keras prosesor grafis klien selama pemrosesan bahasa berlangsung aktif. Melalui rangkaian pengujian terstruktur ini, seluruh dimensi fungsionalitas dan efisiensi infrastruktur terdistribusi dapat dievaluasi secara komprehensif untuk memastikan sistem memenuhi klaim spesifikasi awal sebagai satu kesatuan *Pipeline* yang kohesif dan deterministik.

4.4.1 Deskripsi Pengujian

Evaluasi operasional terhadap sistem interaksi suara NPC adaptif dilakukan menggunakan pendekatan *Mixed Methods* sebagai kerangka validasi menyeluruh. Pendekatan ini dipilih karena sistem yang dikembangkan terdiri dari beberapa lapisan terdistribusi, yaitu komponen Client-STT, Server Cloud NLU, skrip Papyrus, serta respons audio-visual di dalam *game engine*. Oleh karena itu,



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

keberhasilan sistem tidak cukup hanya dinilai dari aspek teknis, tetapi juga perlu dilihat dari performa komputasi, keberhasilan fungsi, akurasi model, dan penerimaan pengguna.

Instrumen pertama adalah pengujian fungsional menggunakan metode *Black Box Testing* dengan teknik *Use Case Testing*. Pengujian ini dilakukan langsung di dalam lingkungan gim aktif untuk memastikan setiap fungsi berjalan sesuai skenario. Total terdapat 98 *test case* yang mencakup 10 kelas *intent* komando, 7 pemicu sinyal otonom, pergeseran nilai afeksi, validasi *payload*, serta penanganan kasus batas (*edge cases*). Pengujian ini bertujuan memastikan bahwa instruksi dari sistem NLU dapat diterima oleh skrip lokal dan dieksekusi oleh NPC tanpa menyebabkan konflik pada alur memori atau perilaku gim.

Instrumen kedua adalah evaluasi model bahasa melalui perbandingan antara model baseline DistilBERT Base Cased dan model final RoBERTa-base *Multi-Task Learning*. Evaluasi dilakukan menggunakan data uji yang sama agar hasil perbandingan lebih objektif. Aspek yang dibandingkan meliputi akurasi, *Macro F1-Score*, ukuran berkas model, efisiensi kompresi, serta latensi inferensi saat model dijalankan menggunakan ONNX Runtime. Evaluasi ini digunakan untuk membuktikan bahwa model final memiliki keseimbangan lebih baik antara akurasi dan efisiensi deployment.

Instrumen ketiga adalah pengujian penerimaan dan kegunaan sistem melalui gabungan *User Acceptance Testing* (UAT) dan *System Usability Scale* (SUS). Kuesioner disusun dalam satu formulir agar responden tidak mengalami kejenuhan saat mengisi. Instrumen ini melibatkan 30 responden dengan variasi pengalaman bermain, perangkat mikrofon, dan spesifikasi komputer. SUS digunakan untuk mengukur tingkat kegunaan sistem, sedangkan UAT digunakan untuk menilai penerimaan pengguna terhadap fitur interaksi suara, respons NPC, dan pengalaman bermain secara umum.

Instrumen keempat adalah *System Profiling* untuk mengukur performa non-fungsional sistem. Pengukuran dilakukan terhadap waktu *Round-Trip Time* (RTT), mulai dari tombol *Push-to-Talk* dilepas hingga respons suara NPC terdengar pada klien. Profiling juga digunakan untuk mengevaluasi latensi Client-STT, waktu

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

respons Server Cloud NLU, transfer *payload* REST API, penggunaan memori, serta dampak sistem terhadap proses rendering gim. Dengan pengujian ini, distribusi beban antara klien dan server dapat dianalisis secara lebih transparan. Ringkasan instrumen pengujian ditampilkan pada Tabel 4.14.

Tabel 4. 14 Ringkasan Parameter Metodologi dan Instrumen Operasional Pengujian Sistem Terdistribusi Serana *Cloud AI*

Perangkat Instrumen	Pendekatan Evaluasi	Teknik Pengujian	Subjek / Sumber Data	Tujuan Rekayasa Pengujian
<i>Black Box Testing</i>	Kualitatif	<i>Use Case Testing</i>	Peneliti Sendiri	Validasi fungsionalitas 10 <i>Intent</i> NLU, 7 sinyal otonom, dan protokol <i>Failsafe</i> internal gim.
Komparasi Model ML	Kuantitatif	<i>Comparative Study</i>	Log Eksperimen ML	Validasi akurasi, <i>F1-Score</i> , robustness STT, serta efisiensi ukuran <i>File</i> arsitektur kecerdasan buatan.
UAT + SUS <i>Kuesioner</i>	Campuran	Kuesioner Gabungan	30 Responden Riil	Validasi persepsi akseptabilitas, tingkat kegunaan antarmuka, dan kenyamanan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

				interaksi pengguna.
<i>System Profiling</i>	Kuantitatif	<i>Measurement Metric</i>	Infrastruktur Sistem	Validasi performa non-fungsional komponen latensi RTT total dan isolasi utilitas perangkat keras klien.

4.4.2 Prosedur Pengujian

Prosedur pengujian pada tahap ini dirancang sebagai panduan langkah demi langkah (step-by-step engineering protocol) serta standarisasi lingkungan uji terkontrol (controlled environment setup) untuk mengeksekusi keempat instrumen evaluasi sistem secara berurutan dan deterministik. Penetapan tata cara pelaksanaan pengujian yang baku ini dipilih secara teknis guna mengeliminasi bias faktor eksternal lingkungan, menjamin keterulangan (repeatability) dari setiap skenario uji, serta menjaga tingkat objektivitas yang absolut pada seluruh parameter metrik kuantitatif maupun kualitatif yang dikumpulkan. Aliran prosedur kerja ini dioperasikan dengan memisahkan area pengkondisian berdasarkan karakteristik sub-sistem, yang meliputi penyiapan unit komputer lokal klien yang terintegrasi dengan manajer virtual Mod Organizer 2 dan skrip STT *Client* untuk *Black Box Testing*, isolasi lingkungan pembelajaran mesin untuk pengujian komparasi arsitektur jaringan saraf tiruan, mekanisme penyebaran formulir kuesioner terpadu untuk pengumpulan data persepsi pengguna akhir, hingga penempatan skrip pelacakan jejak waktu (timestamp logger) untuk pemprofilan performa non-fungsional infrastruktur terdistribusi. Dampak langsung dari pembatasan dan standarisasi prosedur pengujian terintegrasi ini adalah tersedianya cetak biru operasional yang andal untuk mengumpulkan seluruh data pengujian secara sah, sehingga deviasi galat klasifikasi intent serta fluktuasi latensi jaringan internet dapat diidentifikasi secara transparan pada setiap titik pipa pemrosesan data (*Pipeline*) sebelum ditarik menuju tahap tabulasi data hasil akhir. Disusun

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

ringkasan tahapan pengujian yang mencakup tujuan, instrumen, dan luaran yang dihasilkan dari masing-masing metode pengujian. Ringkasan tersebut disajikan pada Tabel 4.15.

Tabel 4. 15 Ringkasan Prosedur Pengujian Sistem

Tahap Pengujian	Tujuan	Instrumen	Output
<i>Black Box Testing</i>	Memverifikasi fungsi sistem sesuai spesifikasi	Test Case Use Case Testing	Status Pass/Fail
Komparasi Model ML	Membandingkan performa DistilBERT dan RoBERTa	Dataset v5_train_data_cased_aug.json	Accuracy, F1-Score, Runtime
UAT dan SUS	Mengukur penerimaan pengguna dan usability	Google Forms	Skor UAT dan SUS
<i>Profiling Performa Sistem</i>	Mengukur latensi dan konsumsi resource	Timestamp Logger, psutil, Task Manager	RTT, CPU Usage, GPU Usage

4.4.2.1 Prosedur *Black Box Testing*

Prosedur pengujian fungsional menggunakan metode *Black Box Testing* dengan teknik Use Case Testing dieksekusi secara nyata di dalam lingkungan uji terkontrol (controlled environment setup) yang dikondisikan pada unit komputer lokal pengembang. Spesifikasi perangkat keras pengujian ini menggunakan basis prosesor AMD Ryzen 5 7500F dengan memori RAM sebesar 32GB DDR5 yang menjalankan aplikasi gim Skyrim Special Edition melalui perantara virtualisasi lapisan *Virtual File System* milik Mod Organizer 2 (MO2). Seluruh pengkondisian *Runtime* lokal komputer klien ini diprogram agar terhubung secara terdistribusi melalui jaringan internet menuju layanan *Server Cloud* VPS DomaiNesia yang mengisolasi ekosistem wadah Docker dalam status aktif, bersih, dan responsif.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pilihan untuk menetapkan standarisasi arsitektur infrastruktur pengujian ini diterapkan secara teknis guna mereplikasi ekosistem operasional modifikasi yang sesungguhnya di sisi pengguna akhir, serta memastikan bahwa seluruh fungsionalitas skrip Papyrus lokal klien dan komputasi inferensi model NLP jarak jauh dapat beroperasi secara penuh tanpa adanya kontaminasi dari gangguan kegagalan sistem luar yang tidak relevan. Dampak langsung dari penguncian parameter lingkungan uji ini adalah seluruh data hasil pengujian fungsional dapat direkam secara deterministik, meminimalisasi kemunculan galat akibat malafungsi *game engine*, serta menjamin keandalan aliran data dari awal ke akhir saat tombol pintas *Push-to-Talk* (PTT) dioperasikan secara langsung oleh peneliti. Sebelum pengujian fungsional dilaksanakan, seluruh komponen perangkat keras, perangkat lunak, serta infrastruktur jaringan yang terlibat dalam eksperimen terlebih dahulu distandarisasi untuk menjaga konsistensi lingkungan pengujian. Standarisasi ini bertujuan mengurangi pengaruh variabel eksternal yang berpotensi memengaruhi hasil evaluasi fungsional sistem. Rincian konfigurasi lingkungan pengujian *Black Box Testing* disajikan pada Tabel 4.16.

Tabel 4. 16 Lingkungan Pengujian *Black Box Testing*

Komponen	Spesifikasi
CPU Client	AMD Ryzen 5 7500F
RAM Client	32 GB DDR5
Operating Environment	Skyrim Special Edition
Mod Manager	Mod Organizer 2
STT Client	faster-whisper
Server	VPS DomaiNesia
Backend	FastAPI
Containerization	Docker
Communication Protocol	HTTP REST API
Database Referensi	JSON KnowlEdge Base

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Dokumentasi jalannya hasil pengujian fungsional ini disusun secara terstruktur menggunakan format tabel penilai khusus yang mencakup seluruh komponen pengkondisian input dan luaran sistem. Format pelaporan Test Case diderivasi secara ketat menjadi struktur matriks tabel yang memetakan tujuh komponen penunjuk kolom utama, meliputi variabel Identitas Kode (ID), rumpun klasifikasi tindakan (Kategori), rancangan kondisi pengujian (Skenario), stimulus gelombang suara kalimat (Input Suara), target luaran teoretis sistem (Expected Output), realitas umpan balik fisik interaksi di dalam gim (Actual Output), serta kesimpulan akhir pengujian (Status berupa penanda Pass atau Fail). Pilihan pemetaan matriks kolom terstruktur ini diterapkan secara teknis sebagai instrumen standardisasi pencatatan data rekayasa perangkat lunak untuk mempermudah visualisasi deviasi komputasi antara target perencanaan awal dengan luaran aktual sistem di lapangan.

Dampak positif dari penerapan format pencatatan yang ketat ini adalah tersedianya transparansi audit data eksperimen yang valid, mempermudah pelacakan letak kegagalan skrip apabila terjadi disorientasi penunjuk memori, serta menyediakan basis penilaian kuantitatif yang objektif terhadap performa rasio kelulusan (*pass rate*) dari sistem interaksi suara secara keseluruhan. Cakupan pengujian fungsional ini dirancang secara komprehensif dengan mendistribusikan total 98 *test case* ke dalam tujuh kelompok besar pengujian taksonomi perilaku sistem, sebagaimana dirinci pada Tabel 4.17.

Tabel 4. 17 Matriks Sebaran Cakupan Distribusi Kasus Uji *Black Box Testing*

Kelompok Pengujian	Target Skenario Uji	Jumlah Test Case (TC)	Kode
Movement Commands	• Skenario 1: Perintah Navigasi GoTo • Skenario 2: Perintah Mengikuti Pemain • Skenario 3: Perintah Menunggu	• 6 TC • 6 TC • 6 TC	BB-1 s.d. BB-18

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Combat Commands	- Skenario 4: Perintah Menyerang Target - Skenario 5: Taktis Gaya Pertempuran	12 TC 6 TC	BB-19 s.d. BB-36
Social & Utility	- Skenario 6: Perintah Obrolan Sosial Pemain - Skenario 7: Perintah Pemeriksaan Status Hubungan - Skenario 8: Pertukaran Barang Inventori - Skenario 9: Pemutusan Hubungan Pengikut - Skenario 10: Perintah Interaksi Objek Sekitar	6 TC 6 TC 6 TC 6 TC 6 TC	BB-37 s.d. BB-66
Autonomous Signals	Skenario 11: Pemicu Sinyal Otonom Lingkungan	7 TC	BB-67 s.d. BB-73
Affinity System	Skenario 12: Regulasi Status Hubungan Emosional Lokal	3 TC	BB-74 s.d. BB-76
JSON Bridge Payload	Skenario 13: Validasi Jembatan Data <i>Payload</i>	6 TC	BB-77 s.d. BB-82
Edge Cases	Skenario 14: Penanganan Kasus Batas	16 TC	BB-83 s.d. BB-98
Kumulatif Global	14 Skenario Utama	98 TC	BB-1 s.d. BB-98



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Keputusan rekayasa untuk menyebarkan 98 variasi pengujian secara rapat ini dipilih secara teknis guna menjamin keterwakilan seluruh skema logika percabangan skrip Papyrus dan matriks *Multi-Level Dictionary* pada dokumen papyrus_Voicelines.json. Dampak dari penguncian distribusi cakupan data uji yang padat ini adalah sistem terbebas dari celah kerentanan *crash* logika (*unhandled Runtime exception*) saat dioperasikan di berbagai skenario permainan terbuka, serta membuktikan kesiapan modifikasi untuk menghadapi varians perilaku pemain secara dinamis.

Poin rekayasa perangkat lunak paling krusial dalam perancangan materi data uji fungsional ini adalah kewajiban penyuntikan minimal satu skenario negative test case pada setiap kelompok kategori tindakan serta pengujian kendala transisi lingkungan. Pilihan penyusunan skenario pengujian negatif ini diterapkan secara teknis sebagai mekanisme pengaman sistem (*failsafe protocol*) untuk memvalidasi bahwa ketika pengguna memberikan masukan yang menyimpang, mengandung derau tinggi, atau di luar domain lore gim, sistem tidak mengalami kegagalan fatal, melainkan mampu mengalihkan instruksi menuju rute penanganan kesalahan (*error handling routine*) secara halus. Sementara itu, penempatan skenario uji pada kelompok sinyal otonom difungsikan khusus untuk memverifikasi keandalan mekanisme delta detection yang telah diuraikan secara mendalam pada Sub-Bab 4.3.4.2, guna memastikan skrip Papyrus tidak melakukan pemanggilan kueri redundan saat kondisi lingkungan visual gim berada dalam status statis.

Dampak positif dari penerapan mekanisme pengaman berlapis dan deteksi selisih transisi ini adalah minimalisasi beban komputasi *Polling Loop* memori RAM klien, pencegahan terhadap gejala penumpukan instruksi Papyrus (*script logging congestion*), serta penjaminan imersi visual pemain yang tetap terjaga tanpa adanya repetisi vokal kloning *Baked Audio* yang monoton.

Pelaksanaan pengujian fungsional *Black Box Testing* ini dioperasikan secara berurutan dan sistematis melalui lima tahapan prosedur eksekusi *step-by-step* di dalam unit komputer lokal pengembang:

1. Pastikan Aktivitas *Server*: Prosedur pertama dijalankan dengan melakukan verifikasi konektivitas jaringan terdistribusi untuk memastikan Container



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Docker pada *Server Cloud VPS DomaiNesia* berada dalam status aktif dan jalur *Endpoint* ``/chat*` berbasis Framework *FastAPI* merespons kueri pengetesan secara instan.

2. Inisialisasi Lapisan Klien: Prosedur kedua dijalankan dengan mengeksekusi skrip utama pemrosesan suara lokal `stt_client.py` melalui terminal sistem operasi komputer klien untuk membuka gerbang antrean audio rekaman luring.
3. Pemuatan Sesi Permainan: Langkah ketiga diimplementasikan dengan menjalankan gim *Skyrim Special Edition* melalui proksi manajer virtual *Mod Organizer 2*, diikuti oleh pemuatan dokumen penyimpanan data (*load save game*) spesifik di mana entitas NPC *Serana* telah diprogram berada dalam status sebagai pengikut aktif (*active follower*).
4. Injeksi Komando Lisan: Prosedur keempat dioperasikan dengan menahan tombol jalan pintas keyboard *Push-to-Talk* (default tombol 'X') pada komputer klien, mengucapkan kalimat instruksi bahasa Inggris secara konsisten sesuai naskah input skenario skrip uji, dan melepaskan tombol tersebut untuk memicu proses transkripsi *Client STT* luring serta mentransmisikan dokumen *Payload JSON* menuju internet.
5. Observasi *Real-time* dan Pencatatan Status: Tahap akhir dituntaskan dengan memantau pergerakan visual navigasi spasial (*Pathfinding*), animasi gerakan bibir, serta keluaran audio kloning karakter NPC di dalam gim, yang kemudian divalidasi silang bersama visualisasi perubahan dokumen *File JSON / Text-based Payload SeranaCommand.json* untuk selanjutnya disimpulkan ke dalam status akhir berupa kelulusan (*Pass*) atau kegagalan (*Fail*).

4.4.2.2 Prosedur Komparasi Model ML

Prosedur pengujian komparasi model dilakukan dalam lingkungan eksperimen terkontrol menggunakan *PyTorch* pada *Google Colab*. Pengujian dilakukan sebelum model dikompresi atau dioptimasi agar hasil akurasi dan waktu inferensi tidak dipengaruhi oleh faktor eksternal seperti *ONNX Runtime* atau kuantisasi *INT8*. Dengan cara ini, perbandingan yang diperoleh lebih merepresentasikan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

kemampuan asli dari masing-masing arsitektur Transformer dalam memahami konteks perintah pemain.

Pada pengujian ini, model baseline DistilBERT Base Cased dibandingkan dengan model final RoBERTa-base dalam menyelesaikan tugas *Multi-Task Learning*. Kedua model menggunakan dataset yang sama, yaitu `v5_train_data_cased_aug.json`, yang dibagi menggunakan teknik *stratified split* dengan rasio 80% data latih dan 20% data uji. Pembagian data yang sama digunakan untuk menjaga keadilan evaluasi dan mengurangi risiko *data leakage*. Dengan demikian, perbedaan hasil antara kedua model dapat dikaitkan dengan kemampuan arsitektur masing-masing, bukan karena perbedaan distribusi data.

Fokus utama evaluasi tidak hanya diarahkan pada nilai *validation loss*, tetapi juga pada *Macro F1-Score*, khususnya untuk tugas *Named Entity Recognition* (NER). Hal ini penting karena dalam arsitektur *Multi-Task Learning*, nilai kerugian total dapat menutupi kelemahan pada salah satu tugas. Misalnya, akurasi tinggi pada tugas *intent* dan *sentiment* dapat membuat performa keseluruhan tampak baik, meskipun model masih gagal mengenali entitas tertentu. Kondisi ini semakin penting karena dataset NER memiliki ketidakseimbangan kelas, di mana token non-entitas atau label O jauh lebih dominan dibandingkan label entitas seperti Character, Location, Item, dan Enemy.

Pemilihan *Macro F1-Score* NER sebagai indikator utama bertujuan untuk menilai kemampuan model dalam mengenali entitas penting secara lebih adil. Ketepatan ekstraksi entitas berpengaruh langsung terhadap proses *FormID Binding* pada Server Cloud. Jika entitas gagal dikenali, sistem dapat gagal mencari pasangan data pada `papyrus_formids.json`, sehingga skrip Papyrus di sisi klien tidak menerima parameter target yang valid. Dengan demikian, performa NER menjadi salah satu faktor utama yang menentukan keberhasilan eksekusi perintah di dalam *game engine*.

Pengukuran akhir dilakukan dengan membaca hasil log evaluasi model pada *Golden Test Set* dan *Adversarial STT Test Set*. Metrik yang digunakan meliputi *Accuracy* dan *Macro F1-Score* untuk tugas *intent* dan *sentiment*, *Macro F1-Score* untuk NER, serta waktu evaluasi dalam satuan detik. *Adversarial STT Test Set*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menggunakan teks huruf kecil untuk mensimulasikan keluaran dari sistem *Speech-to-Text* lokal yang sering tidak mempertahankan kapitalisasi. Sementara itu, metrik *runtime* digunakan untuk melihat beban komputasi awal dari masing-masing model sebelum proses optimasi.

Melalui prosedur ini, penelitian memperoleh gambaran yang lebih objektif mengenai hubungan antara kompleksitas arsitektur model, akurasi pemahaman bahasa, dan efisiensi komputasi. Hasil evaluasi tersebut digunakan sebagai dasar empiris dalam memilih RoBERTa-base sebagai model final sebelum masuk ke tahap optimasi ONNX dan deployment pada Server Cloud.

4.4.2.3 Prosedur UAT dan SUS

Prosedur pengujian persepsi pengguna dan evaluasi ketergunaan sistem dilakukan menggunakan kuesioner hibrida yang menggabungkan *System Usability Scale* (SUS) dan *User Acceptance Testing* (UAT) dalam satu formulir Google Forms. Instrumen ini terdiri dari 10 pertanyaan SUS standar dengan skala Likert 1 sampai 5, kemudian dilanjutkan dengan 5 pertanyaan UAT yang disesuaikan dengan karakteristik sistem. Pertanyaan UAT mencakup responsivitas reaksi fisik NPC, akurasi Client-STT pada kondisi berderau, frekuensi kegagalan pengenalan teks atau salah klasifikasi *intent*, jenis komando yang paling sering digunakan, serta naturalitas suara hasil *voice cloning*. Formulir juga menyediakan kolom opsional untuk laporan bug dan umpan balik terbuka.

Penggabungan SUS dan UAT dalam satu formulir dilakukan untuk menjaga efisiensi pengisian dan mengurangi risiko kejenuhan responden (*survey fatigue*). Dengan satu sesi evaluasi, responden dapat memberikan penilaian terhadap aspek kegunaan umum sekaligus penerimaan terhadap fitur spesifik sistem. Pendekatan ini juga membantu menjaga konsistensi data karena seluruh umpan balik dikumpulkan dalam konteks pengujian yang sama.

Distribusi kuesioner dilakukan melalui halaman rilis mod pada Nexus Mods dan unggahan pengumuman di komunitas Reddit *r/skyrimmods*. Selama periode pengumpulan data, halaman mod mencatat 1.064 *views* dan 30 *unique downloads*. Dari distribusi tersebut, diperoleh 30 responden riil, terdiri atas 22 partisipan dari komunitas terbuka dan 8 penguji internal. Jumlah tersebut digunakan sebagai dasar



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

evaluasi penerimaan dan ketergunaan sistem. Ringkasan karakteristik responden serta distribusi pengujian disajikan pada Tabel 4.18.

Tabel 4. 18 Karakteristik Responden Pengujian

Kategori	Jumlah
Pengguna Komunitas Nexus Mods	22
Penguji Internal	8
Total Responden	30

Responden diberikan panduan pengujian sebelum mengisi kuesioner. Setiap partisipan diminta mengunduh paket modifikasi, memasangnya melalui Mod Organizer 2, menjalankan skrip lokal `stt_client.py`, dan mencoba sistem interaksi suara di dalam sesi permainan *Skyrim* aktif selama minimal 60 menit. Durasi ini ditetapkan agar responden tidak hanya menilai berdasarkan percobaan singkat, tetapi benar-benar mengalami variasi interaksi yang cukup luas, mulai dari penggunaan 10 kelas *intent* utama, perintah pertempuran, interaksi objek, hingga pemicu sinyal otonom seperti perubahan waktu, cuaca, dan pergeseran *Affinity Tier* Serana.

Penerapan durasi pengujian minimal tersebut bertujuan mengurangi bias penilaian sesaat dan memastikan sistem diuji dalam konteks permainan yang lebih nyata. Selain itu, pengujian selama satu sesi aktif juga membantu melihat kestabilan *pipeline* dalam menghadapi penggunaan berulang, antrean instruksi, serta interaksi antara komponen STT, Server Cloud NLU, dan skrip Papyrus.

Setelah data kuesioner terkumpul, jawaban responden pada 10 pertanyaan SUS dikonversi menjadi skor kegunaan menggunakan rumus standar. Pertanyaan bernomor ganjil, yaitu Q1, Q3, Q5, Q7, dan Q9, skor dihitung dengan mengurangi nilai jawaban responden dengan 1. Sementara itu, untuk pertanyaan bernomor genap, yaitu Q2, Q4, Q6, Q8, dan Q10, skor dihitung dengan mengurangi nilai jawaban dari angka 5. Seluruh kontribusi skor kemudian dijumlahkan dan dikalikan dengan 2,5 sehingga menghasilkan skor akhir dalam rentang 0 sampai 100.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Melalui perhitungan ini, hasil SUS dapat digunakan sebagai metrik kegunaan yang lebih objektif karena telah memperhitungkan perbedaan arah indikator positif dan negatif. Skor tersebut kemudian menjadi dasar analisis tingkat penerimaan dan ketergunaan sistem pada bagian hasil pengujian.

4.4.2.4 Prosedur *Profiling* Performa Sistem

Prosedur profiling performa sistem dilakukan untuk mengukur karakteristik latensi pada alur interaksi suara berbasis arsitektur Hybrid Client-Cloud. Pengukuran difokuskan pada rangkaian proses sejak tombol Push-to-Talk dilepaskan oleh pemain hingga berkas SeranaCommand.json berhasil diperbarui di dalam lingkungan game runtime Skyrim. Batas pengukuran tersebut dipilih karena SeranaCommand.json berfungsi sebagai data bridge yang menerima hasil inferensi dari Server Cloud NLU sebelum dibaca oleh skrip Papyrus.

Profiling dilakukan pada dua perangkat uji dengan kelas performa yang relatif sebanding, yaitu AMD Ryzen 5 7500F dan AMD Ryzen 5 7600. Kedua perangkat menggunakan RAM 32GB DDR5, model faster-whisper small.en, endpoint Server Cloud NLU yang sama, serta prosedur pengujian yang sama. Pemilihan dua perangkat dengan generasi dan platform yang sebanding bertujuan menjaga konsistensi perbandingan agar hasil profiling tidak dipengaruhi oleh perbedaan spesifikasi hardware yang terlalu jauh.

Parameter pengukuran performa sistem yang digunakan pada penelitian ini ditunjukkan pada Tabel 4.19.

Tabel 4. 19 Parameter Pengukuran Performa Sistem

Parameter Pengukuran	Satuan	Tujuan Pengukuran
Perangkat uji	-	Menjelaskan spesifikasi perangkat yang digunakan dalam proses profiling.
Jumlah sampel valid	kali	Menentukan jumlah pengulangan pengujian yang digunakan sebagai dasar perhitungan hasil profiling.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Parameter Pengukuran	Satuan	Tujuan Pengukuran
Rata-rata waktu Client-STT	ms	Mengukur rata-rata waktu yang dibutuhkan faster-whisper small.en untuk mengubah input suara menjadi teks pada perangkat klien.
Rata-rata waktu Cloud NLU	ms	Mengukur rata-rata waktu inferensi Natural Language Understanding pada Server Cloud setelah teks transkripsi dikirimkan dari klien.
Rata-rata Total Round-Trip Time (RTT)	ms	Mengukur rata-rata durasi keseluruhan proses sejak tombol Push-to-Talk dilepaskan hingga SeranaCommand.json berhasil diperbarui.
Catatan latensi tercepat	ms	Mengetahui nilai latensi minimum yang dicapai sistem selama sesi profiling.

Setiap perangkat diuji menggunakan 20 sampel valid. Hasil pengukuran kemudian dihitung dalam bentuk rata-rata waktu Client-STT, rata-rata waktu Cloud NLU, rata-rata Total Round-Trip Time, dan catatan latensi tercepat. Parameter yang didefinisikan pada Tabel 4.19 digunakan kembali pada Tabel 4.38 agar terdapat kesesuaian antara prosedur pengukuran dan hasil profiling yang dilaporkan.

4.4.3 Data Hasil Pengujian

Sub-bab ini menyajikan seluruh akumulasi data empiris dan rekaman metrik evaluasi yang diperoleh secara riil dari eksekusi empat instrumen pengujian terintegrasi. Penyajian data pada bagian ini dikonstruksikan secara sistematis untuk memaparkan bukti fisik kuantitatif dan kualitatif, yang mencakup rekapitulasi rasio kelulusan pengujian fungsional, matriks komparasi performa native training model pembelajaran mesin awal, tabulasi biner jawaban kuesioner hibrida, hingga dokumentasi pemprofilan unjuk kerja performa non-fungsional pada infrastruktur terdistribusi. Melalui kompilasi data terkontrol yang terstruktur, bagian ini bertindak sebagai wadah dokumentasi faktual yang transparan guna memvalidasi keselarasan antara perencanaan teoritis dengan hasil luaran aktual sistem di lapangan sebelum ditarik menuju fase interpretasi serta analisis evaluasi mendalam.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Keseluruhan dokumentasi luaran aktual sistem tersebut dipetakan secara hierarkis ke dalam empat sub-bab turunan yang saling kohesif berdasarkan karakteristik metrik penilaian masing-masing. Pemetaan dimulai dari visualisasi baris skenario dari 98 test case fungsional, diikuti secara linear oleh visualisasi angka log eksperimen pelatihan bahasa alami awal antara model baseline dan model final. Selanjutnya, aliran data diurai menuju tabulasi individual serta histogram sebaran skor dari 30 responden kuesioner gabungan, dan ditutup oleh perincian metrik delay komponen waktu latensi total bolak-balik beserta isolasi utilitas perangkat keras komputer klien. Penataan alur pemaparan data yang rapat dan komprehensif ini diterapkan secara teknis untuk memastikan seluruh dimensi spesifikasi arsitektur *Hybrid Client-Cloud* terpapar secara valid, adekuat, serta bersih dari anomali data pengetesan.

4.4.3.1 Hasil *Black Box Testing*

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* dengan teknik *Use Case Testing* untuk memvalidasi seluruh fungsi utama sistem. Skenario pengujian disusun berdasarkan kebutuhan fungsional yang telah ditetapkan, mencakup 10 kelas *intent* NLU, 7 sinyal otonom, regulasi *Affinity System*, integrasi *JSON Bridge*, serta penanganan kondisi batas (*edge cases*). Pengujian dilakukan langsung oleh peneliti di dalam lingkungan gim aktif menggunakan perangkat *Development Unit* agar perilaku sistem dapat diamati secara nyata sesuai rancangan skrip Papyrus.

Secara keseluruhan, pengujian ini terdiri dari 98 *test case* yang dikelompokkan ke dalam 14 skenario uji. Agar hasil pengujian lebih mudah dianalisis dan tidak menimbulkan uraian yang berulang, pemaparan hasil dibagi ke dalam beberapa kelompok arsitektur utama. Kelompok pertama adalah komando tindakan berbasis suara, yang mencakup skenario 1 sampai skenario 10 pada Tabel 4.20 hingga Tabel 4.29.

Kelompok komando tindakan digunakan untuk memvalidasi alur utama sistem *Hybrid Client-Cloud*. Pada alur ini, pemain memberikan perintah suara melalui mekanisme *Push-to-Talk* (PTT). Komponen Client-STT kemudian mengubah audio menjadi teks, mengemasnya ke dalam *request body*, dan mengirimkannya ke

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

endpoint REST API Server Cloud NLU. Di sisi server, model RoBERTa-base *Multi-Task Learning* memproses teks tersebut untuk menghasilkan tiga keluaran utama, yaitu *intent*, *Named Entity Recognition (NER)*, dan *sentiment*.

Pembagian beban kerja ini dirancang agar proses pemahaman bahasa alami tidak membebani perangkat klien secara berlebihan. Dengan memindahkan inferensi NLU ke Server Cloud, komputer klien dapat tetap difokuskan pada proses transkripsi lokal dan rendering gim. Hasil pemrosesan server kemudian dikembalikan dalam bentuk *response payload* berbasis JSON, yaitu *SeranaCommand.json*, yang dibaca oleh *PapyrusUtil* untuk menggerakkan logika internal NPC.

Pada kelompok pengujian ini, perubahan nilai afeksi akibat sentimen positif atau negatif diproses secara internal di dalam memori gim tanpa menampilkan informasi berlebihan pada layar pemain. Pendekatan ini digunakan agar antarmuka permainan tetap bersih dan tidak mengganggu imersi. Informasi hubungan emosional hanya ditampilkan melalui fitur *Social_Status*, sehingga fungsi tersebut menjadi mekanisme utama untuk mengecek status afeksi *Serana* secara eksplisit.

Tabel 4. 20 Hasil *Black Box Testing* - Skenario 1: Movement_GoTo

Kode	Input Suara	Expected Output	Actual Output	Status
BB-1	"please go to whiterun my love"	<i>Intent</i> Movement_GoTo, entitas Whiterun, dan sentimen positif dikenali; NPC bergerak menuju Whiterun dan affinity meningkat.	NPC berhasil menuju Whiterun dan affinity meningkat.	Pass
BB-2	"kindly head to solitude"	<i>Intent</i> Movement_GoTo dan entitas Solitude dikenali; NPC bergerak	NPC berhasil menuju Solitude dan	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		menuju Solitude dan affinity meningkat.	affinity meningkat.	
BB-3	"go to riften"	<i>Intent Movement_GoTo</i> dan entitas Rifen dikenali; NPC bergerak menuju Rifen tanpa perubahan affinity.	NPC berhasil menuju Rifen tanpa perubahan affinity.	Pass
BB-4	"move to markarth"	<i>Intent Movement_GoTo</i> dan entitas Markarth dikenali; NPC bergerak menuju Markarth tanpa perubahan affinity.	NPC berhasil menuju Markarth tanpa perubahan affinity.	Pass
BB-5	"get to the damn cave right now fool"	<i>Intent Movement_GoTo</i> dan sentimen negatif dikenali; NPC bergerak menuju gua terdekat dan affinity berkurang.	NPC berhasil menuju gua terdekat dan affinity berkurang.	Pass
BB-6	"go to invalidplace you damn fool"	Lokasi tidak valid; sistem membatalkan navigasi dan affinity berkurang.	Navigasi dibatalkan karena lokasi tidak dikenali; affinity berkurang.	Pass

Tabel 4. 21 Hasil *Black Box Testing* - Skenario 2: Movement_Follow

Kode	Input Suara	Expected Output	Actual Output	Status
------	-------------	-----------------	---------------	--------

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-7	"please follow me i need you"	<i>Intent Movement_Follow</i> dikenali; NPC mengikuti pemain dan affinity meningkat.	NPC berhasil mengikuti pemain dan affinity meningkat.	Pass
BB-8	"come with me dear"	<i>Intent Movement_Follow</i> dikenali; NPC mengikuti pemain dan affinity meningkat.	NPC berhasil mengikuti pemain dan affinity meningkat.	Pass
BB-9	"follow me"	<i>Intent Movement_Follow</i> dikenali; NPC mengikuti pemain tanpa perubahan affinity.	NPC berhasil mengikuti pemain tanpa perubahan affinity.	Pass
BB-10	"serana follow me"	<i>Intent Movement_Follow</i> dikenali; NPC mengikuti pemain tanpa perubahan affinity.	NPC berhasil mengikuti pemain tanpa perubahan affinity.	Pass
BB-11	"move your ass and follow me you fool"	<i>Intent Movement_Follow</i> dan sentimen negatif dikenali; NPC mengikuti pemain dan affinity berkurang.	NPC berhasil mengikuti pemain dan affinity berkurang.	Pass

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-12	"follow him right now damn it"	Sistem mengabaikan target selain pemain; NPC tetap mengikuti pemain dan affinity berkurang.	NPC tetap mengikuti pemain dan affinity berkurang.	Pass
-------	--------------------------------	---	--	------

Tabel 4. 22 Hasil *Black Box Testing* - Skenario 3: Movement_Wait

Kode	Input Suara	Expected Output	Actual Output	Status
BB-13	"please wait here for a moment"	<i>Intent</i> Movement_Wait dikenali; NPC berhenti dan affinity meningkat.	NPC berhenti di lokasi saat ini dan affinity meningkat.	Pass
BB-14	"stay here my friend"	<i>Intent</i> Movement_Wait dikenali; NPC berhenti dan affinity meningkat.	NPC berhenti di lokasi saat ini dan affinity meningkat.	Pass
BB-15	"wait here"	<i>Intent</i> Movement_Wait dikenali; NPC berhenti tanpa perubahan affinity.	NPC berhenti tanpa perubahan affinity.	Pass
BB-16	"don't move"	<i>Intent</i> Movement_Wait dikenali; NPC berhenti tanpa perubahan affinity.	NPC berhenti tanpa perubahan affinity.	Pass
BB-17	"stand still and don't move you fool"	<i>Intent</i> Movement_Wait dan sentimen negatif dikenali; NPC berhenti dan affinity berkurang.	NPC berhenti dan affinity berkurang.	Pass

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-18	"wait and follow me right now damn it"	Sistem memprioritaskan mode wait pada konflik <i>Intent</i> ; affinity berkurang.	NPC tetap berada di posisi terakhir dan affinity berkurang.	Pass
-------	--	--	---	------

Tabel 4. 23 Hasil *Black Box Testing* - Skenario 4: Combat_Attack

Kode	Input Suara	Expected Output	Actual Output	Status
BB-19	"please kill that dragon my friend"	<i>Intent</i> Combat_Attack dan entitas dragon dikenali; NPC menyerang dragon dan affinity meningkat.	NPC berhasil menyerang dragon dan affinity meningkat.	Pass
BB-20	"take down that bandit my love"	<i>Intent</i> Combat_Attack dan entitas bandit dikenali; NPC menyerang bandit dan affinity meningkat.	NPC berhasil menyerang bandit dan affinity meningkat.	Pass
BB-21	"please kill that vampire"	<i>Intent</i> Combat_Attack dan entitas vampire dikenali; NPC menyerang vampire dan affinity meningkat.	NPC berhasil menyerang vampire dan affinity meningkat.	Pass
BB-22	"hit that troll my friend"	<i>Intent</i> Combat_Attack dan entitas troll dikenali; NPC menyerang troll dan affinity meningkat.	NPC berhasil menyerang troll dan affinity meningkat.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-23	"attack the dragon"	<i>Intent</i> Combat_Attack dikenali; NPC menyerang dragon tanpa perubahan affinity.	NPC berhasil menyerang dragon tanpa perubahan affinity.	Pass
BB-24	"kill the bandit"	<i>Intent</i> Combat_Attack dikenali; NPC menyerang bandit tanpa perubahan affinity.	NPC berhasil menyerang bandit tanpa perubahan affinity.	Pass
BB-25	"hit the troll"	<i>Intent</i> Combat_Attack dikenali; NPC menyerang troll tanpa perubahan affinity.	NPC berhasil menyerang troll tanpa perubahan affinity.	Pass
BB-26	"attack enemy"	Entitas tidak spesifik; sistem menargetkan musuh terdekat.	NPC berhasil menyerang musuh terdekat.	Pass
BB-27	"kill that draugr right now you fool"	<i>Intent</i> Combat_Attack dan sentimen negatif dikenali; NPC menyerang draugr dan affinity berkurang.	NPC berhasil menyerang draugr dan affinity berkurang.	Pass
BB-28	"destroy that falmer damn it"	<i>Intent</i> Combat_Attack dan sentimen negatif dikenali; NPC menyerang falmer dan affinity berkurang.	NPC berhasil menyerang falmer dan affinity berkurang.	Pass

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-29	"kill them all right now fool"	Sistem mengaktifkan serangan area terhadap musuh aktif; affinity berkurang.	NPC menyerang seluruh musuh aktif di sekitar dan affinity berkurang.	Pass
BB-30	"attack jarl balgruuf you damn fool"	Sistem faksi memblokir serangan terhadap NPC friendly; affinity berkurang.	Serangan dibatalkan oleh sistem faksi dan affinity berkurang.	Pass

Tabel 4. 24 Hasil *Black Box Testing* - Skenario 5: Combat_Tactical

Kode	Input Suara	Expected Output	Actual Output	Status
BB-31	"please use your magic spells dear"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Positive terekstrak; mode bertarung dialihkan ke Magic disertai peningkatan affinity.	Mode pertarungan berhasil berpindah ke penggunaan sihir dan affinity meningkat.	Pass
BB-32	"kindly fight with your sword my love"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Positive terekstrak; mode bertarung dialihkan ke Melee disertai peningkatan affinity.	NPC mengganti gaya tempur menjadi melee menggunakan pedang dan affinity meningkat.	Pass
BB-33	"switch to magic style"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Neutral terekstrak; mode Magic	Mode pertarungan sihir aktif dan affinity tetap konstan.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		diaktifkan tanpa perubahan affinity.		
BB-34	"equip your blade now"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Neutral terekstrak; mode Melee diaktifkan tanpa perubahan affinity.	NPC menggunakan senjata jarak dekat dan affinity tetap konstan.	Pass
BB-35	"reset your fighting style you useless fool"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Negative terekstrak; mode tempur dikembalikan ke Default disertai penurunan affinity.	Gaya bertarung berhasil direset ke mode default dan affinity berkurang.	Pass
BB-36	"fight like a dragon right now damn fool"	<i>Intent</i> Combat_Tactical dan <i>Sentiment</i> Negative terekstrak dengan parameter gaya tidak valid; failsafe memaksa mode Default disertai penurunan affinity.	Parameter gaya bertarung tidak dikenali sehingga sistem mempertahankan mode default dan affinity berkurang.	Pass

Tabel 4. 25 Hasil *Black Box Testing* - Skenario 6: Social_Chat

Kode	Input Suara	Expected Output	Actual Output	Status
BB-37	"please talk to me my dear friend"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Positive terekstrak; sistem memutar	Dialog sosial acak berhasil diputar	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		dialog sosial acak disertai peningkatan affinity.	dan affinity meningkat.	
BB-38	"share a story with me dear"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Positive terekstrak; dialog cerita diputar disertai peningkatan affinity.	NPC memutar dialog cerita acak dan affinity meningkat.	Pass
BB-39	"talk to me"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Neutral terekstrak; dialog sosial diputar tanpa perubahan affinity.	Dialog sosial berhasil diputar dan affinity tetap konstan.	Pass
BB-40	"let's chat"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Neutral terekstrak; sistem memutar dialog acak tanpa perubahan affinity.	NPC memulai percakapan acak dan affinity tetap konstan.	Pass
BB-41	"shut up don't talk to me you fool"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Negative terekstrak; dialog tetap diputar disertai penurunan affinity.	Dialog sosial tetap berjalan namun affinity berkurang.	Pass
BB-42	"stop chatting right now damn it"	<i>Intent</i> Social_Chat dan <i>Sentiment</i> Negative terekstrak; dialog sosial tetap dipicu disertai penurunan affinity.	Sistem tetap memutar dialog sosial dan affinity berkurang.	Pass

Tabel 4. 26 Hasil *Black Box Testing* - Skenario 7: Social_Status

Kode	Input Suara	Expected Output	Actual Output	Status
BB-43	"how is our relationship my love"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Positive terekstrak; sistem menampilkan skor affinity melalui debug notification disertai peningkatan affinity.	Nilai affinity berhasil ditampilkan pada debug notification dan affinity meningkat.	Pass
BB-44	"what do you think of me darling"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Positive terekstrak; sistem menampilkan status hubungan disertai peningkatan affinity.	Informasi affinity berhasil ditampilkan dan affinity meningkat.	Pass
BB-45	"check status"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Neutral terekstrak; status affinity ditampilkan tanpa perubahan affinity.	Nilai affinity berhasil dicetak pada debug notification dan affinity tetap konstan.	Pass
BB-46	"show me your status"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Neutral terekstrak; status hubungan ditampilkan tanpa perubahan affinity.	Status affinity berhasil muncul di layar dan affinity tetap konstan.	Pass

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-47	"show your broken status you fool"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Negative terekstrak; status affinity ditampilkan disertai penurunan affinity.	Nilai affinity berhasil ditampilkan dan affinity berkurang.	Pass
BB-48	"check affinity right now damn it"	<i>Intent</i> Social_Status dan <i>Sentiment</i> Negative terekstrak; debug notification tetap dipicu disertai penurunan affinity.	Status affinity berhasil muncul pada layar dan affinity berkurang.	Pass

Tabel 4. 27 Hasil *Black Box Testing* - Skenario 8: Inventory_Trade

Kode	Input Suara	Expected Output	Actual Output	Status
BB-49	"could you please share your items dear"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Positive terekstrak; fungsi inventory dipanggil disertai peningkatan affinity.	Menu inventory berhasil dibuka dan affinity meningkat.	Pass
BB-50	"kindly let me trade gear with you my love"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Positive terekstrak; menu barter dibuka disertai peningkatan affinity.	Sistem membuka inventory barter dan affinity meningkat.	Pass
BB-51	"open inventory"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Neutral terekstrak; inventory	Inventory berhasil dibuka	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		dibuka tanpa perubahan affinity.	dan affinity tetap konstan.	
BB-52	"let's trade items"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Neutral terekstrak; sistem membuka menu barter tanpa perubahan affinity.	Fitur barter berhasil diaktifkan dan affinity tetap konstan.	Pass
BB-53	"give me your gear right now you fool"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Negative terekstrak; inventory tetap dibuka disertai penurunan affinity.	Inventory berhasil terbuka namun affinity berkurang.	Pass
BB-54	"hand over your items damn it"	<i>Intent</i> Inventory_Trade dan <i>Sentiment</i> Negative terekstrak; menu barter tetap dipanggil disertai penurunan affinity.	Sistem barter tetap berjalan dan affinity berkurang.	Pass

Tabel 4. 28 Hasil *Black Box Testing* - Skenario 9: System_Dismiss

Kode	Input Suara	Expected Output	Actual Output	Status
BB-55	"please go home safely my love"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Positive terekstrak; status follower dinonaktifkan disertai peningkatan affinity.	NPC berhenti menjadi follower dan affinity meningkat.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-56	"leave the party for now dear"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Positive terekstrak; hubungan follower diputus disertai peningkatan affinity.	NPC kembali ke mode mandiri dan affinity meningkat.	Pass
BB-57	"dismiss"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Neutral terekstrak; status follower dinonaktifkan tanpa perubahan affinity.	NPC berhenti mengikuti pemain dan affinity tetap konstan.	Pass
BB-58	"leave the group"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Neutral terekstrak; mode follower dihentikan tanpa perubahan affinity.	NPC kembali ke lokasi asal dan affinity tetap konstan.	Pass
BB-59	"get out of here right now you fool"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Negative terekstrak; follower diberhentikan disertai penurunan affinity.	NPC meninggalkan pemain dan affinity berkurang.	Pass
BB-60	"leave me alone and go away damn it"	<i>Intent</i> System_Dismiss dan <i>Sentiment</i> Negative terekstrak; sistem memutus status follower disertai penurunan affinity.	Status follower berhasil dibatalkan dan affinity berkurang.	Pass

Tabel 4. 29 Hasil *Black Box Testing* – Skenario 10: Interact_Environment

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kode	Input Suara	Expected Output	Actual Output	Status
BB-61	"please activate that lever my friend"	<i>Intent</i> Interact_Environment, <i>Entity</i> lever, dan <i>Sentiment</i> Positive terekstrak; fungsi Activate() dipicu disertai peningkatan affinity.	NPC berhasil mengaktifkan lever dan affinity meningkat.	Pass
BB-62	"can you take that item for me"	<i>Intent</i> Interact_Environment dan <i>Sentiment</i> Positive terekstrak; NPC diarahkan ke item target disertai peningkatan affinity.	NPC mengambil item target dan affinity meningkat.	Pass
BB-63	"activate that lever"	<i>Intent</i> Interact_Environment, <i>Entity</i> lever, dan <i>Sentiment</i> Neutral terekstrak; objek diaktifkan tanpa perubahan affinity.	Lever berhasil diaktifkan dan affinity tetap konstan.	Pass
BB-64	"open the barrel"	<i>Intent</i> Interact_Environment, <i>Entity</i> barrel, dan <i>Sentiment</i> Neutral terekstrak; NPC membuka objek target tanpa perubahan affinity.	Barrel berhasil dibuka dan affinity tetap konstan.	Pass
BB-65	"pull that damn switch"	<i>Intent</i> Interact_Environment, <i>Entity</i> switch, dan <i>Sentiment</i> Negative terekstrak; objek	NPC berhasil menarik switch dan affinity berkurang.	Pass



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	right now you fool"	diaktifkan disertai penurunan affinity.		
BB- 66	"sit down on that chair right now damn it"	<i>Intent</i> Interact_Environment, <i>Entity</i> chair, dan <i>Sentiment</i> Negative terekstrak; NPC diarahkan ke kursi disertai penurunan affinity.	NPC berhasil duduk pada kursi target dan affinity berkurang.	Pass

Kelompok kedua memaparkan kluster otonomi dan afeksi sistem klien yang membawahi skenario 11 and skenario 12 (Tabel 4.30 dan Tabel 4.31). Berbeda dengan kelompok komando vokal, pengujian pada kelompok ini difokuskan murni pada keandalan operasional internal skrip *SeranaAI_CoreScript.psc* yang berjalan secara luring pada mesin klien. Sistem ini mengoperasikan arsitektur Polling Loop dengan interval konstan setiap 1.0 detik ($\text{PollInterval} = 1.0$) untuk memantau siklus perubahan state lingkungan makro gim. Keputusan rekayasa mendasar untuk mengeksekusi pelacakan 7 sinyal otonom secara luring (offline tracking) diterapkan sebagai langkah pengaman guna menghindari terjadinya kemacetan antrean instruksi memori skrip pada Creation Engine akibat ketergantungan transmisi internet yang berlebihan.

Dampak operasional dari arsitektur lokal ini terbukti sangat responsif; ketika variabel *signalCode* menangkap perubahan delta status lingkungan, skrip Papyrus secara langsung mengalihkan indeks variabel global *SeranaAudioPointer* untuk memicu pemutaran berkas sapaan kontekstual *Baked Audio* secara luring. Kelompok ini juga memvalidasi ketangguhan fungsi *GetAffinityTier()* dalam melacak batas ambang pergeseran tingkatan afeksi. Ketika batas ambang terlampaui, sistem secara deterministik menginisiasi peristiwa interaksi paksa tatap muka (*ForceGreet*). Skenario ini dilengkapi dengan Failsafe waktu tunggu (timeout) mutlak selama 45.0 detik ($\text{affinityTimeout} > 45.0$), di mana fungsi *ResetAffinityForceGreet()* akan dipicu secara mandiri untuk membersihkan tumpukan instruksi yang macet apabila pemain mengabaikan Follower, sehingga kendali bebas gerak pemain dapat dipulihkan 100% luring. Khusus pada pengujian

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

skenario interaksi Social_Status, pengujian berhasil membuktikan keberhasilan pencetakan indikator nilai afeksi secara visual di sudut kiri atas layar monitor menggunakan fitur penampil bawaan gim berupa debug notification (Debug.Notification()).

Tabel 4. 30 Hasil *Black Box Testing* - Skenario 11: Pemicu Sinyal Otonom Lingkungan

Kode	Kondisi Pemicu	Expected Output	Actual Output	Status
BB-67	Variabel <i>GameHour</i> masuk rentang pagi.	Polling Loop mendeteksi waktu pagi dan memutar audio kode 1.	Audio sapaan pagi berhasil diputar otomatis.	Pass
BB-68	Variabel <i>GameHour</i> masuk rentang siang/sore.	Polling Loop mendeteksi waktu siang dan memutar audio kode 2.	Audio sapaan siang berhasil diputar otomatis.	Pass
BB-69	Variabel <i>GameHour</i> masuk rentang malam.	Polling Loop mendeteksi waktu malam dan memutar audio kode 3.	Audio sapaan malam berhasil diputar otomatis.	Pass
BB-70	Cuaca berubah menjadi hujan/badai.	Sistem mendeteksi cuaca buruk dan memutar audio komentar cuaca.	Audio komentar cuaca berhasil diputar otomatis.	Pass
BB-71	NPC memasuki dungeon.	Sistem mendeteksi lokasi dungeon dan	Audio eksplorasi dungeon berhasil diputar otomatis.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		memutar audio eksplorasi.		
BB-72	Pertempuran selesai.	Sistem mendeteksi akhir combat dan memutar audio kemenangan.	Audio kemenangan berhasil diputar setelah combat selesai.	Pass
BB-73	NPC idle selama 1080 detik.	Sistem memutar audio idle/bosan otomatis.	Audio idle berhasil diputar setelah batas waktu tercapai.	Pass

*Catatan: Kolom 'Kondisi Pemicu' pada skenario ini menggunakan kondisi state sistem sebagai pengganti input suara karena fitur yang diuji bersifat otonom/non-vokal.

Tabel 4. 31 Hasil *Black Box Testing* - Skenario 12: Regulasi Status Affinity Lokal

Kode	Kondisi Pemicu	Expected Output	Actual Output	Status
BB-74	Skor affinity melewati batas tier atas.	Sistem menaikkan affinity tier dan memicu ForceGreet positif.	Tier affinity meningkat dan debug notification "Tier Up" muncul.	Pass
BB-75	Skor affinity turun melewati batas tier bawah.	Sistem menurunkan affinity tier dan memicu ForceGreet negatif.	Tier affinity menurun dan debug notification "Tier Down" muncul.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-76	ForceGreet diabaikan lebih dari 45 detik.	Timeout memicu ResetAffinityForceGreet() otomatis.	ForceGreet berhasil direset dan kontrol pemain dipulihkan.	Pass
-------	---	--	--	------

*Catatan: Kolom 'Kondisi Pemicu' pada skenario ini menggunakan kondisi state sistem sebagai pengganti input suara karena fitur yang diuji bersifat otonom/non-vokal.

Kelompok ketiga mengulas kluster infrastruktur jembatan data dan kasus batas yang memayungi skenario 13 dan skenario 14 (Tabel 4.32 dan Tabel 4.33). Kelompok ini bertugas mengevaluasi ketahanan jembatan data JSON Bridge *Payload* serta kesiapan sistem dalam menghadapi anomali operasional di luar kondisi ideal. Mekanisme sinkronisasi data divalidasi melalui ekspor dokumen konteks SeranaContext.json dan pembacaan SeranaCommand.json memanfaatkan fungsi kunci berkas dari pustaka PapyrusUtil. Teknik penulisan berkas secara kilat (Atomic Drop) diuji secara paralel untuk membuktikan bahwa pertukaran informasi melintasi ruang penyimpanan terisolasi *Virtual File System* (VFS) milik Mod Organizer 2 (MO2) bebas dari risiko korupsi dokumen akibat kondisi race condition.

Pada pengujian kasus batas (*Edge Cases Testing*), ketangguhan komponen kecerdasan buatan diuji melintasi berbagai gangguan eksternal. Ketika pemain memberikan input suara bermuatan kalimat asing di luar semesta gim (Out-of-Domain), komponen Tokenizer di sisi *Server* terbukti mampu mendeteksi ketiadaan kelas dan mengaktifkan Failsafe protocol untuk mengabaikan perintah demi menjaga stabilitas perilaku aktor gim. Gangguan noise frekuensi tinggi (Noise) akibat ketidaksempurnaan mikrofon klien berhasil diredam oleh lapisan pra-pemrosesan arsitektur *Client* STT menggunakan model Faster-Whisper small.en. Masalah kesalahan pelafalan teks maupun salah ketik (Typo) pada nama entitas unik berhasil dieliminasi di sisi *Server* memanfaatkan algoritma koreksi ejaan *Fuzzy Matching* berbasis pustaka thefuzz, sehingga teks cacat otomatis diarahkan kembali menuju kecocokan ID registrasi (FormID Binding) yang sah.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Skrip keselamatan Papyrus juga teruji sukses melakukan pemotongan numerik otomatis pada kisaran batas nilai mutlak -100 hingga 100 ketika terjadi luapan data (Affinity Overflow/Underflow). Dari aspek keamanan siber tingkat Host, pembungkusan sesi aplikasi web FastAPI di dalam Container Docker terbukti aman karena diisolasi murni di bawah hak akses rendah pengguna non-root (Low-Privilege Non-Root User) menggunakan identitas serana_User, membatasi celah eksploitasi kode jahat dari luar internet.

Tabel 4. 32 Hasil *Black Box Testing* - Skenario 13: Validasi Jembatan Data Payload

Kode	Kondisi Pemicu	Expected Output	Actual Output	Status
BB-77	Nilai konteks gim berubah.	Data terbaru diekspor ke SeranaContext.json.	File konteks berhasil diperbarui otomatis.	Pass
BB-78	Tidak ada perubahan data konteks.	Penulisan file dihentikan untuk mencegah redundansi.	Sistem tidak membuat penulisan file baru secara berulang.	Pass
BB-79	Server mengirim SeranaCommand.json.	Papyrus membaca <i>payload</i> dan menjalankan instruksi.	<i>Intent</i> , <i>Entity</i> , dan <i>Sentiment</i> berhasil dibaca oleh Papyrus.	Pass
BB-80	Eksekusi command selesai.	<i>Payload</i> command direset ke kondisi kosong.	File command berhasil dibersihkan otomatis.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-81	Atomic Drop dijalankan paralel.	Sistem mencegah konflik penulisan file.	Penulisan file tetap stabil tanpa crash atau freeze.	Pass
BB-82	Terjadi potensi race condition pada VFS MO2.	Jalur file diisolasi untuk mencegah korupsi data.	File <i>payload</i> tetap aman tanpa korupsi data.	Pass

*Catatan: Kolom 'Kondisi Pemicu' pada skenario ini menggunakan kondisi state sistem sebagai pengganti input suara karena fitur yang diuji bersifat otonom/non-vokal.

Tabel 4. 33 Hasil *Black Box Testing* - Skenario 14: Penanganan Kasus Batas

Kode	Input Suara	Expected Output	Actual Output	Status
BB-83	"Open the calculator apps"	Sistem mendeteksi input OOD dan memblokir eksekusi.	Perintah diabaikan aman tanpa perubahan affinity.	Pass
BB-84	"how's the weather today in Jakarta"	Query eksternal ditolak oleh sistem.	Tidak ada instruksi yang diteruskan ke <i>game engine</i> .	Pass
BB-85	"format this harddisk"	Input berbahaya diklasifikasikan sebagai OOD dan diblokir.	Perintah diabaikan total tanpa memicu aksi apapun.	Pass
BB-86	"who's the creator of this mod"	Sistem menolak query di luar domain operasional.	NPC tidak menjalankan aksi	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

			apapun dan sistem tetap stabil.	
BB-87	"sword [keyboard noise]"	<i>Client STT</i> menyaring noise dan tetap mengenali keyword utama.	Noise keyboard mengganggu proses transkripsi sehingga keyword gagal dikenali.	Fail
BB-88	"[musik keras] Follow Me"	STT tetap mengenali token utama di tengah noise lingkungan.	Noise musik terlalu tinggi sehingga inference gagal dilakukan.	Fail
BB-89	"go to whiterun" (lowercase)	Model tetap mengenali entitas lokasi valid.	Entitas Whiterun berhasil dikenali dengan benar.	Pass
BB-90	"go to whitrn"	<i>Fuzzy Matching</i> memperbaiki typo menuju Whiterun.	Typo berhasil dikoreksi menjadi Whiterun.	Pass
BB-91	"move to solidude"	<i>Fuzzy Matching</i> memperbaiki typo menuju Solitude.	Typo berhasil dikoreksi menjadi Solitude.	Pass
BB-92	"attack that bndit"	<i>Fuzzy Matching</i> memperbaiki typo menuju bandit.	Entitas bandit berhasil dikenali dan combat dipicu.	Pass
BB-93	"activate that livr"	Sistem memperbaiki typo <i>entity</i> menuju lever.	<i>Entity lever</i> berhasil dikenali dan objek diaktifkan.	Pass

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BB-94	Affinity Overflow > 100	Sistem melakukan clamp nilai affinity ke batas maksimum.	Nilai affinity terkunci pada angka maksimum 100.	Pass
BB-95	Affinity Underflow < -100	Sistem melakukan clamp nilai affinity ke batas minimum.	Nilai affinity terkunci pada angka minimum - 100.	Pass
BB-96	Cold start inferensi pertama Docker	Warm-up <i>Request</i> memuat model ONNX ke RAM.	Inferensi awal berhasil dan <i>Request</i> berikutnya berjalan cepat.	Pass
BB-97	Eksekusi Docker non-root <i>User</i>	Container berjalan menggunakan <i>User</i> serana_ <i>User</i> .	Hak akses container berhasil dibatasi pada mode non-root.	Pass
BB-98	<i>Intent</i> dan <i>Entity</i> kosong	Failsafe membatalkan seluruh aksi NPC.	NPC tetap diam dan tidak menjalankan tindakan apapun.	Pass

Kelompok terakhir ditutup oleh visualisasi makro matriks rekapitulasi hasil eksperimen global (Tabel 4.34). Tabel ringkasan ini menyatukan akumulasi perhitungan kuantitatif dari seluruh skenario uji fungsional yang telah dijalankan. Pencatatan nilai pada tabel rekapitulasi merangkum totalitas performa reliabilitas sistem, menegaskan rasio perbandingan kasus uji yang berhasil diproses (Pass) terhadap kasus uji yang gagal (Fail) untuk memberikan jaminan empiris kesiapan modifikasi sistem interaksi hibrida klien-*cloud* ini sebelum dilakukan evaluasi performa latensi komputasi lebih lanjut.

Tabel 4. 34 Rekapitulasi Hasil Eksperimen *Black Box Testing*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Skenario	Total TC	Pass	Fail	% Pass
Skenario 1: Perintah Navigasi GoTo	6	6	0	100%
Skenario 2: Perintah Mengikuti Pemain	6	6	0	100%
Skenario 3: Perintah Menunggu	6	6	0	100%
Skenario 4: Perintah Menyerang Target	12	12	0	100%
Skenario 5: Taktis Gaya Pertempuran	6	6	0	100%
Skenario 6: Perintah Obrolan Sosial Pemain	6	6	0	100%
Skenario 7: Perintah Pemeriksaan Status Hubungan	6	6	0	100%
Skenario 8: Pertukaran Barang Inventori	6	6	0	100%
Skenario 9: Pemutusan Hubungan Pengikut	6	6	0	100%
Skenario 10: Perintah Interaksi Objek Sekitar	6	6	0	100%
Skenario 11: Pemicu Sinyal Otonom Lingkungan	7	7	0	100%
Skenario 12: Regulasi Status Hubungan Emosional Lokal	3	3	0	100%
Skenario 13: Validasi Jembatan Data <i>Payload</i>	6	6	0	100%



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Skenario 14: Penanganan Kasus Batas	16	14	2	87,50%
TOTAL	98	96	2	97,96%

4.4.3.2 Hasil Komparasi Model DistilBERT vs RoBERTa-base *Multi-Task Learning*

Validasi kuantitatif terhadap arsitektur kecerdasan buatan dilakukan melalui eksperimen komparasi untuk mengevaluasi performa model Multi-Task antara DistilBERT Base Cased sebagai model baseline dengan RoBERTa-base sebagai model final. Pengujian ini menggunakan pengondisian lingkungan uji terkontrol (controlled environment setup) di dalam ekosistem Google Colab dengan memanfaatkan Framework Transformers dari Hugging Face. Guna menjamin asas keadilan evaluasi (fairness evaluation condition), kedua model dilatih menggunakan pembagian data (dataset split) yang identik bersumber dari Dataset V5 dengan proporsi rasio 80% untuk data latih dan 20% untuk data uji melalui penarikan sampel acak terstratifikasi (stratified random sampling). Langkah penarikan ini berfungsi memastikan distribusi kelas *Intent*, *Sentiment*, dan *Named Entity Recognition* (NER) berada dalam kondisi seimbang pada kedua sesi pelatihan, sehingga perbedaan luaran murni dipengaruhi oleh kapasitas arsitektur masing-masing model bahasa alami tanpa kontaminasi bias variansi data.

Proses optimasi parameter pembelajaran (hyperparameters execution) pada kedua arsitektur menunjukkan karakteristik konvergensi yang berbeda di bawah pengawasan mekanisme Early Stopping. Konfigurasi awal untuk pelatihan model DistilBERT diatur pada batas atas 20 Epoch dengan parameter pemantau fungsi kerugian validasi (`metric_for_best_model="eval_loss"`). Log pelatihan mencatatkan bahwa model DistilBERT berhenti secara otomatis pada Epoch 9 (Epoch 9/20) karena terpicu oleh mekanisme Early Stopping dengan batasan kesabaran tiga siklus (`early_stopping_patience=3`) setelah nilai validation loss mengalami stagnasi. Di sisi lain, model RoBERTa-base menyelesaikan keseluruhan 10 Epoch penuh (Epoch 10/10) tanpa mengalami pemotongan dini. Kondisi ini terjadi karena parameter optimasi dikunci untuk memantau metrik tertinggi pada tugas pengekstrakan entitas (`metric_for_best_model="eval_ner_f1"`), di mana

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

struktur penalti bobot tugas (custom loss weighting) untuk token NER ditinggikan sebesar 2.5 kali lipat demi mengompensasi kompleksitas kosakata semesta gim. Evaluasi akhir dari pengujian dua tahap ini dirangkum secara komprehensif pada Tabel 4.35.

Tabel 4. 35 Matriks Komparasi Performa Akhir Model DistilBERT Base Cased vs RoBERTa-base

Metrik Evaluasi Akhir	Model Baseline (DistilBERT)	Model Final (RoBERTa-base)	Deviasi Performa (Δ)
Evaluasi Tahap 1: Golden Test Set (Cased)			
<i>Validation Loss</i>	2.0496	1.9929	-0.0567
<i>Intent Classification Accuracy</i>	0.7050	0.7084	+0.0034
<i>Intent Classification Macro F1</i>	0.7039	0.7128	+0.0089
<i>Sentiment Classification Accuracy</i>	0.9177	0.9160	-0.0017
<i>Sentiment Classification Macro F1</i>	0.9174	0.9152	-0.0022
<i>NER Token Precision</i>	0.5537	0.6123	+0.0586
<i>NER Token Recall</i>	0.7235	0.7538	+0.0303
<i>NER Token Macro F1</i>	0.6273	0.6757	+0.0484

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

NER Token Accuracy	0.9525	0.9589	+0.0064
Evaluasi Tahap 2: Adversarial STT Set (Lowercase)			
Validation Loss	2.3703	2.3462	-0.0241
Intent Classification Accuracy	0.6930	0.7050	+0.0120
Intent Classification Macro F1	0.6936	0.7030	+0.0094
Sentiment Classification Accuracy	0.9057	0.9194	+0.0137
Sentiment Classification Macro F1	0.9051	0.9183	+0.0132
NER Token Precision	0.4797	0.5321	+0.0524
NER Token Recall	0.6134	0.6436	+0.0302
NER Token Macro F1	0.5384	0.5826	+0.0442
NER Token Accuracy	0.9390	0.9462	+0.0072
Metrik Efisiensi Komputasi			
Batch Evaluation Execution Runtime	0.9963 s	2.1072 s	+1.1109 s

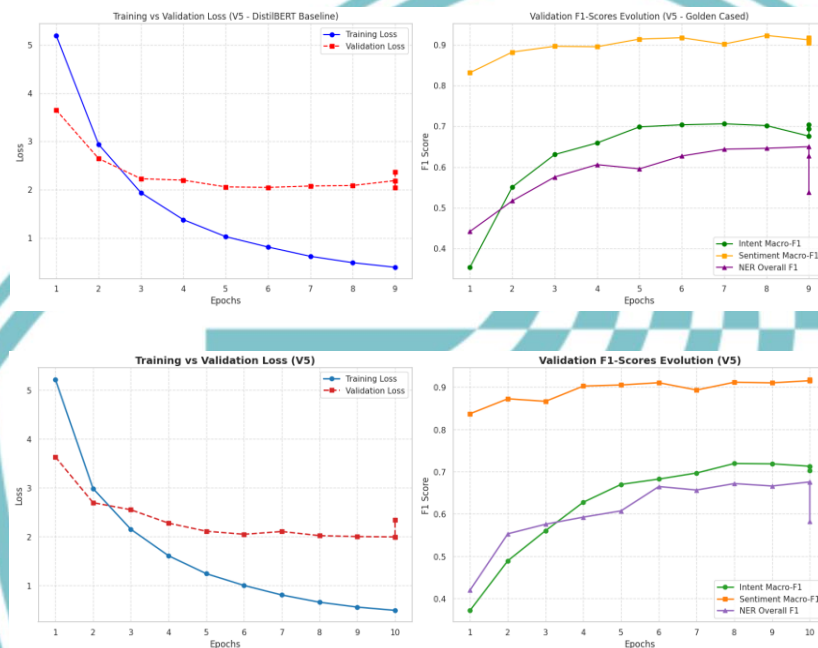
*Catatan: Perbandingan metrik inference latency dan ukuran dokumen model terkompresi pada tahapan *Deployment* akhir tidak dilakukan untuk model DistilBERT, disebabkan karena kompilasi dan kuantisasi INT8 dinamis melintasi

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

ONNX Runtime murni hanya diimplementasikan pada model final RoBERTa-base sesuai dengan arahan dosen pembimbing skripsi.

Analisis mendalam terhadap visualisasi kurva konvergensi pelatihan memberikan pemahaman saintifik yang lebih jelas mengenai stabilitas pembelajaran dari kedua arsitektur model bahasa alami tersebut.



Gambar 4. 25 Perbandingan Konvergensi Loss Function Pelatihan Model *Multi-Task Learning*: (a) Baris Atas: Grafik Baseline DistilBERT; (b) Baris Bawah: Grafik Model Final RoBERTa-base.

Berdasarkan kurva pembelajaran pada Gambar 4.25, model DistilBERT menunjukkan penurunan *Training Loss* yang cukup tajam dari sekitar 5,19 menjadi 0,39 pada epoch akhir. Namun, *Validation Loss* mulai mengalami stagnasi pada kisaran 2,0 sejak epoch 4 sampai epoch 5. Kondisi ini menunjukkan adanya indikasi *mild overfitting*, yaitu ketika model mampu mempelajari data latih dengan baik, tetapi kemampuan generalisasinya terhadap data uji mulai terbatas. Mekanisme *Early Stopping* berhasil mencegah proses *overtraining* lebih jauh dengan menghentikan pelatihan sebelum kurva mengalami divergensi.

Berbeda dengan DistilBERT, kurva pembelajaran RoBERTa-base menunjukkan pola konvergensi yang lebih stabil. Nilai *Validation Loss* turun secara bertahap dari 3,63 hingga mencapai titik terendah 1,99 pada epoch 10 tanpa lonjakan signifikan.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tren tersebut juga didukung oleh peningkatan metrik *Intent Macro F1* dan *NER Overall F1* secara konsisten pada setiap siklus pelatihan. Hal ini menunjukkan bahwa RoBERTa-base memiliki kemampuan generalisasi yang lebih baik dalam menangkap pola semantik pada dataset.

Evaluasi per tugas menunjukkan bahwa RoBERTa-base unggul pada sebagian besar tugas NLU. Pada *Intent Classification*, RoBERTa-base memperoleh *Intent Macro F1* sebesar 0,7128, sedikit lebih tinggi dibandingkan DistilBERT yang memperoleh 0,7039 pada *Golden Test Set*. Pada *Sentiment Classification*, DistilBERT mencatatkan skor *Sentiment Macro F1* sebesar 0,9174, sedikit lebih tinggi dibandingkan RoBERTa-base dengan skor 0,9152. Keunggulan kecil ini dapat terjadi karena data uji *cased* masih memiliki kapitalisasi yang konsisten, sehingga model yang lebih ringan tetap mampu mengenali pola polaritas sentimen dengan baik.

Perbedaan paling penting terlihat pada tugas *Named Entity Recognition* (NER). RoBERTa-base memperoleh *NER Token Macro F1* sebesar 0,6757, lebih tinggi dibandingkan DistilBERT yang memperoleh 0,6273. Selisih +0,0484 ini menunjukkan bahwa RoBERTa-base lebih kuat dalam mengenali entitas seperti lokasi, karakter, musuh, dan objek permainan. Keunggulan tersebut berkaitan dengan struktur RoBERTa-base yang memiliki 12 *transformer layers*, sedangkan DistilBERT hanya menggunakan 6 *transformer layers*. Jumlah lapisan yang lebih sedikit pada DistilBERT membuat kapasitas representasi konteksnya lebih terbatas.

Performa NER menjadi sangat penting karena entitas yang gagal dikenali dapat menyebabkan proses *FormID Binding* tidak berjalan. Jika sistem gagal menangkap nama lokasi atau target musuh, skrip Papyrus dapat menerima parameter kosong sehingga perintah navigasi atau aksi tidak dapat dieksekusi secara tepat. Meskipun DistilBERT memiliki waktu evaluasi lebih cepat, yaitu 0,9963 detik, pemilihan model akhir tetap mengutamakan ketepatan ekstraksi entitas. Oleh karena itu, RoBERTa-base dipilih sebagai model final karena memberikan keseimbangan yang lebih baik antara akurasi pemahaman bahasa dan reliabilitas eksekusi sistem.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

<pre>===== EVALUASI 1: THE GOLDEN TEST SET (CASED) ===== eval_loss: 2.0496 eval_intent_accuracy: 0.7050 eval_intent_f1: 0.7039 eval_sentiment_accuracy: 0.9177 eval_sentiment_f1: 0.9174 eval_ner_precision: 0.5537 eval_ner_recall: 0.7235 eval_ner_f1: 0.6273 eval_ner_accuracy: 0.9525 eval_runtime: 0.9963 eval_samples_per_second: 585.1430 eval_steps_per_second: 37.1360 epoch: 9.0000 ===== EVALUASI 2: ADVERSARIAL STT TEST SET (LOWERCASE) ===== eval_loss: 2.3703 eval_intent_accuracy: 0.6930 eval_intent_f1: 0.6936 eval_sentiment_accuracy: 0.9057 eval_sentiment_f1: 0.9051 eval_ner_precision: 0.4797 eval_ner_recall: 0.6134 eval_ner_f1: 0.5384 eval_ner_accuracy: 0.9390 eval_runtime: 1.1394 eval_samples_per_second: 511.6650 eval_steps_per_second: 32.4730 epoch: 9.0000</pre>	<pre>===== EVALUASI 1: THE GOLDEN TEST SET (CASED) ===== eval_loss: 1.9929 eval_intent_accuracy: 0.7084 eval_intent_f1: 0.7128 eval_sentiment_accuracy: 0.9160 eval_sentiment_f1: 0.9152 eval_ner_precision: 0.6123 eval_ner_recall: 0.7538 eval_ner_f1: 0.6757 eval_ner_accuracy: 0.9589 eval_runtime: 2.1072 eval_samples_per_second: 276.6670 eval_steps_per_second: 17.5590 epoch: 10.0000 ===== EVALUASI 2: ADVERSARIAL STT TEST SET (LOWERCASE) ===== eval_loss: 2.3462 eval_intent_accuracy: 0.7050 eval_intent_f1: 0.7030 eval_sentiment_accuracy: 0.9194 eval_sentiment_f1: 0.9183 eval_ner_precision: 0.5321 eval_ner_recall: 0.6436 eval_ner_f1: 0.5826 eval_ner_accuracy: 0.9462 eval_runtime: 2.3671 eval_samples_per_second: 246.2900 eval_steps_per_second: 15.6310 epoch: 10.0000</pre>
--	--

Gambar 4. 26 Ringkasan Log Akhir Evaluasi Dua Tahap Model *Multi-Task Learning* pada Google Colab: (a) Kolom Kiri: Model Baseline DistilBERT Base Cased; (b) Kolom Kanan: Model Final RoBERTa-base.

Pengujian tahap dua melalui Adversarial STT Test Set sengaja dirancang tanpa menggunakan huruf kapital untuk mereplikasi realitas luaran teks dari komponen *Client* STT (Faster-Whisper) yang mengalami gangguan akibat derau mikrofon komputer. Berdasarkan visualisasi log terminal pada Gambar 4.26, model DistilBERT mengalami degradasi performa (*robustness degradation*) yang sangat drastis, di mana nilai NER Token Macro F1 miliknya merosot tajam menuju titik kritis 0.5384.

Sebaliknya, model RoBERTa-base memperlihatkan ketahanan (*robustness*) yang sangat kuat, dengan raihan metrik *Sentiment* Macro F1 adversarial yang tetap kokoh pada nilai 0.9183 serta mampu menahan kejatuhan metrik NER Token Macro F1 pada batas aman 0.5826. Kemampuan bertahan model RoBERTa-base ini berhasil dicapai berkat pengaplikasian modifikasi custom loss penalty selama fase *Fine-Tuning* berjalan. Strategi pembelajaran tersebut memaksa bobot *Inference Session* untuk mengabaikan ketiadaan fitur ortografis kapitalisasi permukaan dan melatih



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mekanisme perhatian transformer bersandar murni pada kedalaman semantik urutan kata.

Sebagai kesimpulan komparatif, berdasarkan keseluruhan hasil evaluasi kuantitatif dua tahap ini, model RoBERTa-base dipilih sebagai arsitektur final karena menunjukkan keseimbangan terbaik antara ketahanan terhadap degradasi transkripsi STT, kestabilan generalisasi multi-task, serta reliabilitas ekstraksi entitas yang dibutuhkan oleh sistem integrasi Papyrus pada lingkungan gim Skyrim. Pemilihan ini mengunci kepastian bahwa dokumen *Response Payload* berupa File JSON / Text-based *Payload* *SeranaCommand.json* yang dikirim melintasi jaringan internet dari *Server Cloud* menuju komputer klien selalu memuat jaminan akurasi label tinggi untuk menggerakkan visualisasi tindakan fisik unit aktor pengikut secara deterministik.

4.4.3.3 Hasil *User Acceptance Testing* (UAT) dan *System Usability Scale* (SUS)

Pengumpulan data persepsi pengguna terhadap sistem interaksi vokal adaptif dilakukan melalui penyebaran instrumen kuesioner terintegrasi kepada 30 responden penguji. Instrumen evaluasi menggabungkan metode *System Usability Scale* (SUS) dan *User Acceptance Testing* (UAT) ke dalam satu formulir digital berbasis Google Form untuk meningkatkan efisiensi pengisian serta menjaga konsistensi penilaian responden terhadap sistem.

Bagian pertama kuesioner terdiri atas 10 butir pertanyaan standar SUS menggunakan skala Likert 1–5 untuk mengukur tingkat usability sistem secara umum. Sementara itu, bagian kedua memuat pertanyaan UAT yang difokuskan pada evaluasi performa operasional sistem di lingkungan penggunaan nyata, meliputi responsivitas latensi, akurasi pengenalan suara di tengah noise lingkungan, kestabilan klasifikasi perintah, serta kualitas hasil *Voice Cloning*.

Sebanyak 30 data responden berhasil dihimpun secara lengkap yang terdiri atas penguji internal dan pengguna komunitas modifikasi gim. Rekapitulasi hasil evaluasi SUS ditampilkan pada Tabel 4.36.

Tabel 4. 36 Hasil Tabulasi Skor Kuesioner 10 Pertanyaan *System Usability Scale* (SUS)

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

No	Pernyataan Kuesioner Inti System Usability Scale (SUS)	Skor 1	Skor 2	Skor 3	Skor 4	Skor 5	Rata- rata
Q1	<i>I think that I would like to use this Voice control system/mods frequently.</i>	0	0	3	7	20	4,57
Q2	<i>I found the system/mod too complicated to use.</i>	14	5	7	4	0	2,03
Q3	<i>I thought the system/mods was easy to use.</i>	0	3	4	4	19	4,30
Q4	<i>I think that I would need the support of a technical person to be able to use this system/mods.</i>	14	2	4	5	5	2,17
Q5	<i>I found the various functions in this system/mods were well integrated.</i>	0	0	0	9	21	4,70
Q6	<i>I thought there was too much inconsistency in this system/mods.</i>	22	6	1	1	0	1,37
Q7	<i>I would imagine that most people would learn to use this system/mods very quickly.</i>	4	5	11	4	6	3,10

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

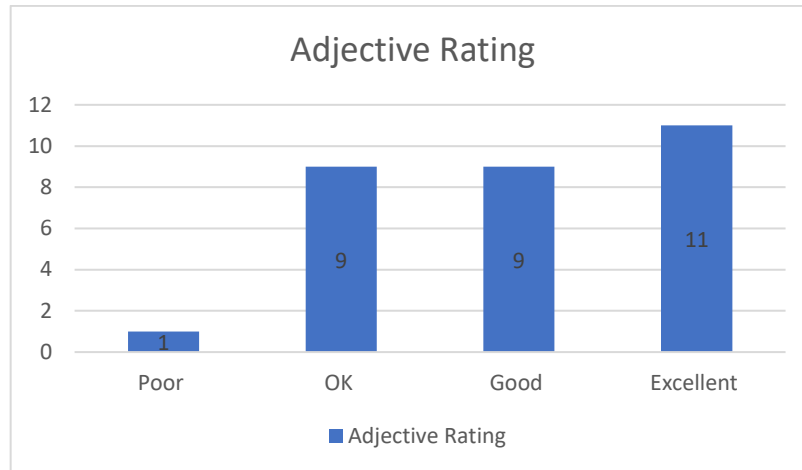
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Q8	<i>I found the system/mods very cumbersome (awkward) to use.</i>	14	7	7	2	0	1,90
Q9	<i>I felt very confident using the system/mods.</i>	0	0	4	3	23	4,63
Q10	<i>I needed to learn a lot of things before I could get going with this system/mods.</i>	11	3	2	4	10	2,97
KUM	Skor Rata-rata Kumulatif Global System Usability Scale (SUS)						79,67

Perhitungan skor *System Usability Scale* (SUS) dilakukan menggunakan metode standar Brooke. Skor kontribusi diperoleh dengan mengurangi nilai jawaban responden dengan angka 1. Sebaliknya, untuk pernyataan negatif bernomor genap, skor kontribusi dihitung dengan mengurangi angka 5 dengan nilai jawaban responden.

Seluruh skor kontribusi kemudian dijumlahkan dan dikalikan dengan konstanta 2,5 sehingga menghasilkan rentang skor akhir 0–100. Berdasarkan hasil penghitungan terhadap seluruh responden, diperoleh skor rata-rata SUS sebesar 79,67 yang termasuk dalam kategori Acceptable dengan predikat Good. Guna memetakan sebaran nilai akhir tersebut secara visual, klusterisasi skor responden dikelompokkan ke dalam penanda grafik pada Gambar 4.27.



Gambar 4. 27 Grafik Distribusi Nilai *System Usability Scale* Berdasarkan Adjective Ratings

Berdasarkan distribusi skor pada Gambar 4.27, mayoritas responden memberikan penilaian positif terhadap tingkat usability sistem. Kategori Excellent (85–100) memperoleh frekuensi tertinggi dengan jumlah 11 responden, diikuti kategori Good sebanyak 9 responden, dan kategori OK sebanyak 9 responden. Sementara itu, hanya terdapat 1 responden yang memberikan penilaian pada kategori Poor dan tidak terdapat responden pada kategori Worst Imaginable.

Dominasi skor pada kategori Excellent dan Good menunjukkan bahwa sistem dinilai memiliki tingkat kemudahan penggunaan yang baik saat dioperasikan secara langsung di dalam gim. Meskipun demikian, terdapat masukan dari sebagian pengguna pemula yang mengalami kesulitan pada tahap konfigurasi awal sistem, khususnya pada proses instalasi dependensi eksternal seperti *Python Runtime*. Temuan ini menunjukkan bahwa penyederhanaan proses instalasi masih menjadi aspek yang dapat ditingkatkan pada pengembangan berikutnya, misalnya melalui *automated installer* atau *Deployment package* yang lebih terintegrasi. Evaluasi terhadap indikator performa sistem secara spesifik kemudian ditinjau melalui instrumen UAT yang dijabarkan pada Tabel 4.37.

Tabel 4. 37 Hasil Evaluasi Dimensi Penerimaan Pengguna - *User Acceptance Testing (UAT)*

No	Butir Pertanyaan Evaluasi Kinerja Sistem (<i>User</i>)	Komponen	Sko r 1	Sko r 2	Sko r 3	Sko r 4	Sko r 5	Rat a- rata
----	---	----------	------------	------------	------------	------------	------------	-------------------

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	Acceptance Testing)							
1	<i>In general, how would you rate the responsiveness/latency of Serana's physical reactions to your Voice commands? (Skor 1 = Extremely Slow; Skor 5 = Instant / Seamless)</i>	Hybrid Client-Cloud Architecture	0	3	7	5	15	4,07
2	<i>How accurately did the system understand your commands when there was background noise (e.g., keyboard clicks, fans) or when speaking quietly? (Skor 1 = Very Poorly; Skor 5 = Very Effectively)</i>	Client STT (Faster-Whisper)	0	0	6	17	7	4,03
3	<i>How often did the system experience speech recognition failures or Intent misclassifications ? (Skor 1 = Almost Always; Skor 5 = Never)</i>	NLU Multi-Task RoBERTa-base	0	0	8	4	18	4,33
4	<i>How would you rate the naturalness and audio quality of Serana's cloned Voice generated by this mod? (Skor 1 = Very Robotic / Distorted; Skor 5 = Very Natural / Indistinguishable)</i>	RVC v2 & ElevenLabs Voice Cloning	0	4	6	8	12	3,93



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

K	Rata-rata Skor							4,09
U	Agregat Dimensi							/
M	Pengujian UAT							5,00

Pengukuran indeks penerimaan pengguna akhir terhadap performa sistem di lapangan menghasilkan nilai rata-rata kuantitatif sebesar 4,09 dari skala absolut 5,00. Nilai rata-rata agregat 4,09 tersebut menunjukkan bahwa mayoritas responden memberikan penilaian pada rentang Agree hingga Strongly Agree terhadap performa fungsional sistem. Dimensi responsivitas latensi menunjukkan adanya ketergantungan yang linear antara spesifikasi perangkat keras komputer klien dengan kecepatan eksekusi tindakan fisik aktor gim. Beberapa responden yang menggunakan perangkat dengan spesifikasi rendah melaporkan adanya peningkatan delay pada proses respons NPC. Kondisi ini terutama dipengaruhi oleh tingginya beban inferensi lokal pada komponen *Client STT*. Sistem menyediakan opsi penggunaan model transkripsi yang lebih ringan seperti *base.en* atau *tiny.en* untuk mengurangi latensi. Pendekatan ini terbukti mampu mempercepat waktu respons, meskipun disertai penurunan akurasi transkripsi pada kondisi tertentu.

Kondisi tersebut dipengaruhi oleh keterbatasan kemampuan komputasi perangkat *Edge* dengan spesifikasi rendah dalam menjalankan proses inferensi model STT secara lokal di tengah beban pemrosesan gim yang tinggi. Temuan ini sekaligus memperkuat keputusan perancangan sistem untuk memindahkan proses NLU menuju *Server Cloud* guna mengurangi beban komputasi pada sisi klien.

Evaluasi terhadap kualitas *Voice Cloning* menunjukkan bahwa mayoritas responden menilai suara hasil sintesis sudah cukup natural dan mendekati karakter asli dalam gim. Namun, beberapa pengguna berpengalaman masih menemukan adanya ketidakkonsistenan tonal pada bagian dialog tertentu. Kondisi ini dipengaruhi oleh keterbatasan variasi emosi pada dataset pelatihan suara yang digunakan selama proses *Fine-Tuning* model.

Masukan tersebut menjadi landasan pengembangan lanjutan untuk meningkatkan kualitas sintesis suara melalui penambahan variasi data pelatihan emosional agar hasil dialog terdengar lebih stabil dan natural.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.4.3.4 Hasil *Profiling* Performa Sistem

Pengukuran kuantitatif terhadap performa sistem dilakukan melalui sesi *profiling* latensi pada arsitektur Hybrid Client-Cloud. Eksperimen ini dirancang untuk menghitung durasi Round-Trip Time (RTT) dari seluruh rangkaian proses interaksi suara. Batas awal pengukuran dihitung sejak tombol Push-to-Talk dilepaskan oleh pemain, sedangkan batas akhir pengukuran ditetapkan ketika berkas SeranaCommand.json berhasil diperbarui di dalam lingkungan game runtime Skyrim. Komponen Speech-to-Text yang dievaluasi secara lokal adalah faster-whisper small.en. Proses inferensi Natural Language Understanding dilakukan pada endpoint VPS Jakarta melalui REST API dengan media pertukaran data berupa JSON Payload.

Profiling dijalankan secara komparatif pada dua perangkat uji dengan kelas performa yang relatif setara. PC 1 menggunakan prosesor AMD Ryzen 5 7500F dengan RAM 32GB DDR5 sebagai perangkat pengembang utama, sedangkan PC 2 menggunakan prosesor AMD Ryzen 5 7600 dengan RAM 32GB DDR5 sebagai perangkat uji pembandingan. Pemilihan dua prosesor dalam generasi dan platform yang sebanding dilakukan agar hasil *profiling* tidak dipengaruhi oleh perbedaan spesifikasi hardware yang terlalu jauh. Jumlah sampel valid pada kedua perangkat disamakan sebanyak 20 sampel agar perbandingan hasil pengukuran tetap konsisten.

Tabel 4. 38 Matriks Hasil *Profiling* Latensi Arsitektur Hybrid Client-Cloud

Parameter <i>Profiling</i>	PC 1 (AMD Ryzen 5 7500F)	PC 2 (AMD Ryzen 5 7600)
Perangkat uji	RAM 32GB DDR5 / Primary Dev Machine	RAM 32GB DDR5 / Community Test Device
Jumlah sampel valid	20	20
Rata-rata waktu Client-STT (ms)	1453	1364

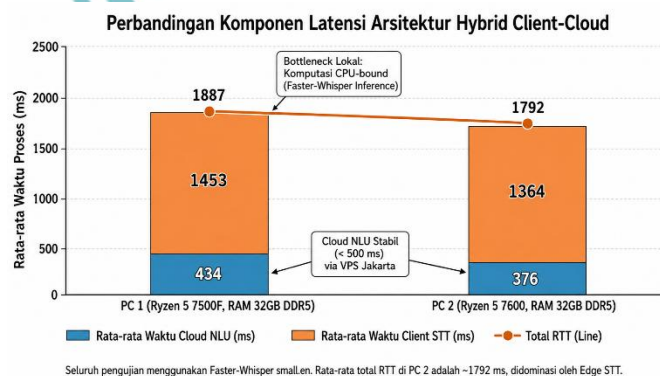
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Parameter Profiling	PC 1 (AMD Ryzen 5 7500F)	PC 2 (AMD Ryzen 5 7600)
Rata-rata waktu Cloud NLU (ms)	434	376
Rata-rata Total Round-Trip Time (RTT) (ms)	1887	1792
Catatan latensi tercepat (ms)	1663,05	1507,83

Berdasarkan Tabel 4.38, kedua perangkat uji menunjukkan karakteristik performa yang relatif sebanding. Pada PC 1, rata-rata waktu Client-STT tercatat sebesar ~1453 ms, sedangkan rata-rata waktu Cloud NLU tercatat sebesar ~434 ms. Akumulasi rata-rata Total Round-Trip Time pada PC 1 berada pada kisaran ~1887 ms. Pada PC 2, rata-rata waktu Client-STT tercatat sebesar ~1364 ms, sedangkan rata-rata waktu Cloud NLU tercatat sebesar ~376 ms. Akumulasi rata-rata Total Round-Trip Time pada PC 2 berada pada kisaran ~1792 ms.

Perbandingan tersebut menunjukkan bahwa selisih rata-rata Total Round-Trip Time antara PC 1 dan PC 2 relatif kecil, yaitu sekitar 95 ms. Kondisi ini menunjukkan bahwa sistem memiliki performa yang konsisten ketika dijalankan pada perangkat dengan kelas hardware yang sebanding. Komponen Client-STT tetap menjadi penyumbang latensi terbesar pada kedua perangkat, sedangkan komponen Cloud NLU menunjukkan waktu pemrosesan yang lebih rendah dan stabil di bawah 500 ms.



Gambar 4. 28 Kurva Komparasi Komponen Latensi *Client STT* vs *Cloud NLU*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4.28 memperlihatkan perbandingan komponen latensi antara Client-STT dan Cloud NLU pada kedua perangkat uji. Visualisasi tersebut menunjukkan bahwa proses Client-STT merupakan komponen dominan dalam total latensi sistem. Hal ini terjadi karena proses transkripsi suara menggunakan faster-whisper small.en dijalankan secara lokal pada perangkat klien. Sementara itu, proses Cloud NLU menunjukkan durasi yang lebih rendah karena inferensi Natural Language Understanding telah dipindahkan ke Server Cloud.

Hasil profiling ini menunjukkan bahwa arsitektur Hybrid Client-Cloud mampu mempertahankan stabilitas performa pada perangkat dengan kelas hardware yang sebanding. Sistem tetap membatasi beban kerja lokal pada proses audio capture, transkripsi suara, dan pengiriman JSON Payload, sedangkan proses inferensi NLU dijalankan pada Server Cloud. Dengan demikian, hambatan utama pada alur interaksi suara bukan berasal dari proses Cloud NLU, melainkan dari durasi Client-STT yang dijalankan pada perangkat klien.

4.5 Pembahasan

Pada bagian pembahasan ini, seluruh akumulasi data empiris dan metrik evaluasi yang diperoleh dari berbagai tahapan pengujian dianalisis secara komprehensif untuk menginterpretasikan hubungan sebab-akibat teknis dalam sistem interaksi vokal *Non-Playable Character* (NPC). Peninjauan tidak dilakukan secara terpisah pada masing-masing komponen, melainkan diintegrasikan secara terpadu untuk menghubungkan hasil kuantitatif pemprofilan latensi dan performa model bahasa dengan penilaian kualitatif serta keandalan fungsional sistem di lapangan. Melalui pendekatan ini, korelasi antara tingkat kelulusan *Black Box Testing* yang mencapai 97,96%, kurva konvergensi pelatihan model Multi-Task RoBERTa-base, perolehan skor kuesioner *System Usability Scale* (SUS) sebesar 79,67, serta capaian indeks *User Acceptance Testing* (UAT) sebesar 4,09 diurai sebagai pembuktian ilmiah terhadap validitas keputusan perancangan sistem terdistribusi ini secara utuh.

Integrasi antarmetrik tersebut memperlihatkan bagaimana keputusan desain arsitektur *Hybrid Client-Cloud* berkontribusi dominan dalam menyeimbangkan antara akurasi pemrosesan bahasa alami dengan batasan performa perangkat keras klien. Karakteristik komputasi yang heterogen antara domain Natural Language



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Processing (NLP) di *cloud* dan domain *game engine* scripting luring lokal disatukan melalui lapisan pemisah asinkron berupa berkas *JSON Payload*. Dengan memetakan hasil performa pengujian secara cross-validation, pembuktian diarahkan untuk mengevaluasi kemampuan sistem dalam mempertahankan stabilitas interaksi secara *near real-time* melintasi batas penundaan interaktif (interactive latency). Pendekatan ini diposisikan secara objektif guna mengonfirmasi efisiensi arsitektur dalam meminimalkan beban kerja prosesor lokal, mengatasi kendala ejaan transkripsi suara, serta mengidentifikasi batasan operasional dan kasus batas (*Edge cases*) yang ditemukan di tengah ekosistem penggunaan nyata komunitas secara jujur dan ilmiah.

4.5.1 Efisiensi Arsitektur *Hybrid Client-Cloud* dan Optimasi Perangkat Keras

Keberhasilan implementasi sistem interaksi suara terdistribusi ini ditunjukkan melalui efektivitas mekanisme *computation offloading* dari komputer klien menuju Server Cloud. Pembagian beban kerja tersebut memungkinkan proses pemrosesan bahasa alami yang relatif berat dijalankan pada infrastruktur cloud, sementara komputer klien tetap difokuskan pada transkripsi suara lokal dan rendering permainan. Hasil pengujian menunjukkan bahwa *Inference Session* model NLU yang berjalan di dalam kontainer Docker pada Server Cloud FastAPI mampu mengembalikan *response payload* secara konsisten di bawah 500 milidetik. Kondisi ini menunjukkan bahwa pemindahan proses NLU ke cloud dapat mengurangi beban komputasi lokal dan membantu menjaga stabilitas permainan dari risiko *stuttering* yang signifikan.

Hasil profiling latensi pada Tabel 4.38 menunjukkan adanya perbedaan *Round-Trip Time* (RTT) yang cukup besar antara perangkat pengujian. PC 1 dengan prosesor yang lebih modern mencatat rata-rata RTT sekitar 1887 ms, sedangkan PC 2 dengan spesifikasi lebih rendah mencapai rata-rata sekitar 4001 ms. Perbedaan ini menunjukkan bahwa hambatan utama sistem bukan berasal dari jaringan internet atau Server Cloud NLU, melainkan dari proses transkripsi lokal menggunakan *faster-whisper small.en* pada CPU klien. Dengan kata lain, bottleneck utama berada pada tahap Client-STT yang bersifat *CPU-bound*. Pada perangkat dengan prosesor lama, proses inferensi STT membutuhkan waktu lebih panjang dan dapat mengganggu kelancaran pengalaman bermain.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Temuan tersebut juga selaras dengan umpan balik pengguna pada pengujian UAT, khususnya dari responden yang menggunakan perangkat berspesifikasi rendah. Beberapa pengguna merasakan adanya jeda respons yang memengaruhi imersi permainan. Oleh karena itu, sistem perlu menyediakan opsi penurunan skala model STT lokal, misalnya ke varian *base.en* atau *tiny.en*, agar perangkat rendah tetap dapat menjalankan sistem dengan latensi yang lebih ringan. Namun, opsi ini tetap memiliki konsekuensi berupa kemungkinan penurunan akurasi transkripsi.

Dari sisi model NLU, hasil komparasi pada Tabel 4.35 menunjukkan adanya *trade-off* antara kecepatan dan ketepatan pemahaman bahasa. Model DistilBERT memiliki waktu evaluasi lebih cepat, yaitu 0,9963 detik, sedangkan RoBERTa-base membutuhkan waktu 2,1072 detik. Namun, RoBERTa-base memberikan performa yang lebih baik pada tugas *Named Entity Recognition* (NER), dengan skor NER F1 sebesar 0,6757 atau unggul +0,0484 dibandingkan DistilBERT. Keunggulan ini penting karena NER berperan langsung dalam mengenali lokasi, karakter, musuh, dan objek yang akan dipetakan ke FormID dalam sistem Papyrus.

Pada pengujian tahap kedua menggunakan *Adversarial STT Test Set* berbasis huruf kecil, RoBERTa-base juga menunjukkan ketahanan yang lebih baik. Meskipun terjadi penurunan akibat hilangnya kapitalisasi, model final masih mempertahankan skor NER F1 sebesar 0,5826, sedangkan DistilBERT turun ke 0,5384. Hasil ini memperkuat keputusan untuk menggunakan RoBERTa-base sebagai model final karena sistem lebih membutuhkan ketepatan ekstraksi entitas daripada sekadar kecepatan inferensi.

Dengan demikian, arsitektur *Hybrid Client-Cloud* terbukti lebih sesuai untuk sistem interaksi suara NPC yang membutuhkan pemahaman bahasa alami secara akurat. Proses NLU yang dipindahkan ke cloud menjaga kualitas interpretasi perintah, sedangkan beban lokal tetap dapat dikendalikan. Kendala terbesar masih berada pada Client-STT, sehingga optimasi lanjutan perlu difokuskan pada pemilihan model STT yang lebih ringan atau penyesuaian berdasarkan spesifikasi perangkat pengguna.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.5.2 Sinkronisasi Multimodal NLU-Papyrus dan Interoperabilitas *Game engine*

Interoperabilitas antara komponen kecerdasan buatan di Server Cloud dan mesin permainan lokal dijumpai melalui lapisan pemisah berbasis pertukaran berkas *JSON payload*. Mekanisme ini diterapkan secara asinkron pada siklus *OnUpdate()* dalam skrip *SeranaAI_CoreScript.psc*. Skrip membaca data perintah secara berkala menggunakan *JsonUtil.GetStringValue()* setelah menjalankan *JsonUtil.Unload()* untuk memastikan cache data sebelumnya tidak mengganggu pembacaan *payload* terbaru. Pendekatan berbasis berkas dipilih karena lebih aman dibandingkan injeksi memori langsung ke *Creation Engine*, yang berisiko mengganggu alokasi memori internal dan memicu *engine crash*. Selain itu, penggunaan virtualisasi folder melalui *Mod Organizer 2* membantu menjaga transaksi data tetap terpisah dari direktori asli *gim*.

Keberhasilan sinkronisasi ini terlihat dari hasil *Black Box Testing* yang memperoleh tingkat kelulusan 97,96% atau 96 dari 98 skenario uji. Rasio tersebut menunjukkan bahwa hasil pemrosesan NLU dapat diterjemahkan secara konsisten menjadi tindakan di dalam *gim*. Ketika model berhasil mengenali *intent*, *sentiment*, dan entitas *Skyrim* dengan tepat, skrip *Papyrus* dapat mengarahkan perintah melalui fungsi *DistributeLogic()* ke sub-rutin yang sesuai, seperti pergerakan spasial, serangan, interaksi objek, pertukaran barang, atau pengecekan status hubungan. Dengan demikian, keberhasilan ekstraksi entitas memiliki peran langsung terhadap stabilitas eksekusi pada sisi *game engine*.

Kesalahan transkripsi akibat pelafalan, aksen, atau typo pada nama entitas juga dapat ditekan melalui mekanisme *Fuzzy String Matching* pada Server Cloud menggunakan pustaka *thefuzz*. Lapisan ini berfungsi sebagai *fault-tolerance layer* yang memperbaiki atau menyelaraskan teks entitas sebelum dipetakan ke *papyrus_formids.json*. Validitas mekanisme ini terlihat pada skenario kasus batas, seperti input “*whitrun*” dan “*solidude*”, yang tetap dapat diarahkan ke entitas valid. Dengan demikian, sistem dapat mempertahankan proses *FormID Binding* meskipun teks hasil STT tidak sepenuhnya akurat.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Sinkronisasi multimodal juga mencakup pemanggilan respons vokal berbasis *Baked Audio*. Ketika `audioFormID` diterima dari payload, skrip memperbarui `SeranaAudioPointer` melalui `SetValueInt(audioFormID)`, kemudian memicu audio menggunakan `SeranaRef.Say(mainTopic)`. Jeda singkat `Utility.Wait(0.25)` digunakan untuk memberi waktu propagasi nilai global dan pemuatan berkas `.lip`, sehingga audio dan gerakan bibir dapat berjalan lebih selaras. Dengan mekanisme *Atomic Drop* dan pembacaan JSON yang terstruktur, sistem mampu menjaga hubungan antara perintah NLU, tindakan Papyrus, dan respons audio-visual NPC secara stabil.

4.5.3 Keterbatasan Sistem dan Temuan Lapangan

Hasil pengujian fungsional menunjukkan adanya dua kegagalan pada kode uji BB-87 dan BB-88. Kegagalan tersebut disebabkan oleh tingkat *noise* lingkungan yang terlalu tinggi, seperti ketukan papan ketik dan musik keras, sehingga kualitas audio yang diterima oleh model `faster-whisper small.en` mengalami penurunan. Kondisi ini mengganggu proses transkripsi awal pada sisi klien dan menyebabkan kata kunci perintah tidak dapat dikenali dengan baik. Temuan ini menunjukkan adanya keterbatasan berupa *upstream failure*, yaitu kondisi ketika data audio sudah rusak sejak tahap perekaman sehingga sistem tidak dapat meneruskan informasi *intent* yang valid ke Server Cloud. Dengan demikian, mekanisme toleransi kesalahan di sisi NLU tidak dapat bekerja apabila input audio primer sudah gagal diproses sejak tahap STT.

Keterbatasan lain terdapat pada mekanisme sinkronisasi Papyrus yang menggunakan *polling loop* dengan interval tetap 1,0 detik. Interval ini dipilih sebagai kompromi untuk menjaga stabilitas Virtual Machine Papyrus dan mencegah penumpukan antrean skrip pada Creation Engine. Jika interval pembaruan dibuat terlalu cepat, proses pembacaan berkas eksternal melalui `PapyrusUtil` dapat bersaing dengan proses rendering gim dan meningkatkan risiko *script lag*. Namun, konsekuensinya adalah munculnya penundaan bawaan sekitar satu detik sebelum skrip membaca payload baru, memperbarui `SeranaAudioPointer`, atau memicu interaksi seperti `ForceGreet`. Dengan demikian, sistem mengorbankan sebagian kecepatan respons demi menjaga stabilitas jangka panjang.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Dari sisi pengalaman pengguna, kualitas *Voice Cloning* menjadi aspek dengan nilai UAT terendah, yaitu 3,93 dari 5,00. Beberapa responden menilai bahwa suara pada dialog panjang belum selalu konsisten dengan karakter asli Serana. Keterbatasan ini dipengaruhi oleh variasi dataset suara yang terbatas dari arsip Dawnguard, khususnya pada ekspresi emosi dan intonasi tertentu. Akibatnya, model RVC v2 masih dapat menghasilkan distorsi timbre atau ketidaksesuaian karakter suara pada kalimat baru yang berada di luar pola data latih. Hal ini lebih mudah dikenali oleh responden yang sudah lama memainkan Skyrim dan familiar dengan suara asli karakter.

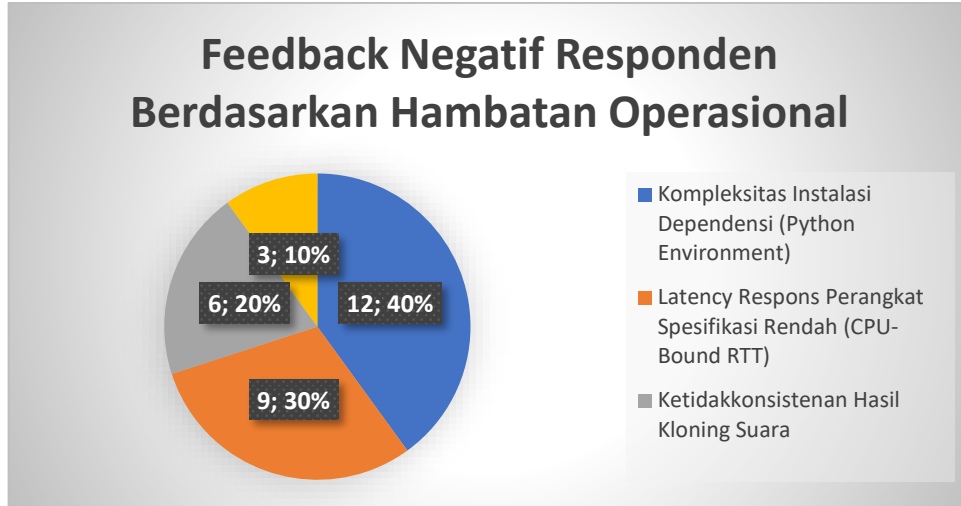
Selain itu, komentar terbuka responden menunjukkan bahwa proses instalasi sistem masih tergolong kompleks bagi pengguna pemula. Sistem membutuhkan beberapa komponen eksternal, seperti Runtime Python lokal, konfigurasi dependensi, jalur direktori, serta integrasi dengan Mod Organizer 2. Kondisi ini dapat menjadi hambatan adopsi bagi pengguna komunitas yang belum terbiasa dengan instalasi mod teknis. Oleh karena itu, pengembangan berikutnya perlu mempertimbangkan paket instalasi otomatis yang menyatukan konfigurasi runtime, dependensi, dan jalur sistem dalam satu peluncur terpadu.

Keterbatasan paradigma juga muncul dari harapan sebagian pengguna terhadap fitur percakapan bebas berbasis Large Language Model (LLM). Sistem yang dikembangkan pada penelitian ini masih menggunakan pendekatan *intent-based NLU*, sehingga hanya mampu merespons perintah dalam taksonomi intent yang telah didefinisikan. Sistem belum dirancang untuk dialog generatif terbuka. Pengembangan ke arah tersebut membutuhkan restrukturisasi arsitektur, terutama pada jembatan data, manajemen konteks, dan respons real-time. Meskipun demikian, skor SUS sebesar 79,67 dan nilai UAT agregat 4,09 menunjukkan bahwa keterbatasan tersebut belum menghambat penerimaan sistem secara keseluruhan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 29 Grafik Distribusi Umpan Balik Negatif Responden Berdasarkan Kluster Hambatan Operasional

Secara keseluruhan, berbagai keterbatasan tersebut menunjukkan bahwa implementasi sistem interaksi vokal berbasis *Hybrid Client-Cloud* pada lingkungan *game engine* legacy tetap menghadapi kompromi antara stabilitas *Runtime*, akurasi pemrosesan bahasa alami, dan kenyamanan interaksi pengguna akhir.

**POLITEKNIK
NEGERI
JAKARTA**

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, serta rangkaian pengujian yang telah dilakukan terhadap sistem interaksi suara berbasis arsitektur *Hybrid Client-Cloud* pada NPC Serana dalam gim *The Elder Scrolls V: Skyrim*, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Arsitektur *Hybrid Client-Cloud* berhasil diimplementasikan untuk mendukung interaksi suara natural antara pemain dan NPC secara *near real-time*. Hasil *profiling* menunjukkan bahwa komponen *Cloud NLU* mampu mempertahankan waktu pemrosesan di bawah 500 ms, sedangkan *bottleneck* utama berasal dari proses transkripsi suara lokal. Temuan ini membuktikan bahwa pendekatan *computation offloading* efektif mengurangi beban komputasi pada perangkat klien tanpa mengganggu stabilitas permainan.
2. Model Multi-Task RoBERTa-base menghasilkan performa yang lebih baik dibandingkan DistilBERT terutama pada tugas NER dengan skor Macro F1 sebesar 0,6757 dan tetap mempertahankan ketahanan terhadap data adversarial. Integrasi model dengan *Fuzzy Matching* dan FormID Mapping menghasilkan tingkat keberhasilan fungsional sebesar 97,96%.
3. Hasil evaluasi performa menunjukkan bahwa komponen *Cloud NLU* memiliki latensi yang relatif stabil pada berbagai spesifikasi perangkat klien. Analisis *profiling* mengonfirmasi bahwa faktor yang paling memengaruhi total *Round-Trip Time* (RTT) berasal dari proses transkripsi suara lokal (*Edge STT*), sehingga kapasitas prosesor klien menjadi faktor penentu utama terhadap kecepatan respons sistem secara keseluruhan.
4. Hasil *User Acceptance Testing* (UAT) dan *System Usability Scale* (SUS) menunjukkan bahwa sistem memperoleh tingkat penerimaan pengguna yang baik. Nilai SUS sebesar 79,67 menempatkan sistem pada kategori Acceptable dengan predikat Good, sedangkan skor UAT sebesar 4,09 menunjukkan tingkat penerimaan pengguna yang tinggi. Hasil ini mengindikasikan bahwa sistem mampu memberikan pengalaman interaksi yang dinilai bermanfaat dan layak digunakan oleh komunitas pemain.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

5. Implementasi mekanisme *Baked Audio* berbasis *Voice Cloning* terbukti mampu menghadirkan respons suara NPC tanpa memerlukan sintesis *real-time* selama permainan berlangsung. Pendekatan ini berkontribusi terhadap pengurangan beban komputasi sistem dan membantu menjaga konsistensi respons interaktif yang diterima pengguna.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran pengembangan yang dapat dipertimbangkan pada penelitian selanjutnya adalah sebagai berikut:

1. Pengembangan sistem dapat diarahkan pada implementasi mekanisme *Speech-to-Text* yang memanfaatkan akselerasi GPU atau model yang lebih ringan sehingga waktu inferensi pada perangkat berspesifikasi rendah dapat dikurangi secara signifikan.
2. Arsitektur komunikasi data dapat ditingkatkan dengan menerapkan mekanisme streaming atau komunikasi berbasis socket secara *real-time* untuk menggantikan pendekatan polling periodik. Pendekatan tersebut berpotensi menurunkan latensi interaksi dan meningkatkan responsivitas sistem secara keseluruhan.
3. Penelitian selanjutnya dapat memperluas cakupan dataset linguistik dan jumlah kelas *Intent* agar sistem mampu memahami variasi perintah yang lebih kompleks serta mendukung percakapan yang lebih adaptif.
4. Kualitas *Voice Cloning* dapat ditingkatkan melalui penambahan jumlah data pelatihan, optimalisasi parameter model, atau pemanfaatan arsitektur generatif yang lebih mutakhir sehingga karakteristik suara yang dihasilkan semakin mendekati suara asli karakter.
5. Konsep arsitektur *Hybrid Client-Cloud* yang dikembangkan pada penelitian ini berpotensi diterapkan pada karakter NPC lain maupun pada genre permainan yang berbeda. Oleh karena itu, penelitian lanjutan dapat mengeksplorasi skalabilitas sistem pada lingkungan permainan yang lebih kompleks dan melibatkan lebih banyak agen cerdas secara simultan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Ahlawat, H., Aggarwal, N. and Gupta, D. (2025) "Automatic Speech Recognition: A survey of deep learning techniques and approaches," *International Journal of Cognitive Computing in Engineering*. KeAi Communications Co., pp. 201–237. Available at: <https://doi.org/10.1016/j.ijcce.2024.12.007>.
- Andriulo, F.C. *et al.* (2024) "Edge Computing and Cloud Computing for Internet of Things: A Review," *Informatics*. Multidisciplinary Digital Publishing Institute (MDPI). Available at: <https://doi.org/10.3390/informatics11040071>.
- Annisa, Z.A., Perdana, R.S. and Adikara, P.P. (2024) "Kombinasi Intent Classification dan Named Entity Recognition pada Data Berbahasa Indonesia dengan Metode Dual Intent and Entity Transformer," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(5), pp. 1017–1024. Available at: <https://doi.org/10.25126/jtiik.2024117985>.
- Bargum, A.R., Serafin, S. and Erkut, C. (2024) "Reimagining speech: a scoping review of deep learning-based methods for non-parallel voice conversion," *Frontiers in Signal Processing*. Frontiers Media SA. Available at: <https://doi.org/10.3389/frsip.2024.1339159>.
- Bosker, H.R. (2021) "Using fuzzy string matching for automated assessment of listener transcripts in speech intelligibility studies." Available at: <https://doi.org/10.3758/s13428-021-01542-4/Published>.
- Deshmukh, A.M. and Chalmers, R. (2024) "User Experience and Usability of Voice User Interfaces: A Systematic Literature Review," *Information (Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI). Available at: <https://doi.org/10.3390/info15090579>.
- Gallotta, R. *et al.* (2024) "Large Language Models and Games: A Survey and Roadmap." Available at: <https://doi.org/10.1109/TG.2024.3461510>.
- He, L. (2024) "Enhanced twitter sentiment analysis with dual joint classifier integrating RoBERTa and BERT architectures," *Frontiers in Physics*, 12. Available at: <https://doi.org/10.3389/fphy.2024.1477714>.
- Hochreiter, M., Krigstein, S. and Wallner, G. (2026) "Beyond Pre-Defined Scripts: Player Perceptions on Generative Non-Player Character Dialogues," *International Conference on Intelligent User Interfaces, Proceedings IUI*. Association for Computing Machinery, pp. 2004–2018. Available at: <https://doi.org/10.1145/3742413.3789221>.
- Jiang, P. and Cai, X. (2024) "A Survey of Text-Matching Techniques," *Information (Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI). Available at: <https://doi.org/10.3390/info15060332>.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kannoja, R. *et al.* (2025) “Gen AI driven multilingual audio dubbing and synthesis system for cross-language video platforms,” *Results in Engineering*, 27. Available at: <https://doi.org/10.1016/j.rineng.2025.106241>.

Kurniawan, E., Nata, A. and Royal, S. (2022) *PENERAPAN SYSTEM USABILITY SCALE (SUS) DALAM PENGUKURAN KEBERGUNAAN WEBSITE PROGRAM STUDI DI STMIK ROYAL*, *Journal of Science and Social Research*. Available at: <http://jurnal.goretanpena.com/index.php/JSSR>.

Maulana, M.R., Susanto, B. and Christianto, A. (2024) “KLIK: Kajian Ilmiah Informatika dan Komputer Pengujian Black Box dengan Teknik Equivalence Partitioning pada Aplikasi Monitoring Pemberian Obat Filariasis Berbasis Android,” *Media Online*, 4(4), pp. 2179–2187. Available at: <https://doi.org/10.30865/klik.v4i4.1603>.

Nik Ahmad, N.A. and Megat Sazali, P.N.N.A. (2021) “Performing User Acceptance Test with System Usability Scale for Graduation Application,” *Proceedings - 2021 International Conference on Software Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 86–91. Available at: <https://doi.org/10.1109/ICSECS52883.2021.00023>.

Ouaddi, C. *et al.* (2025) “Assessing the effectiveness of large language models for intent detection in tourism chatbots: A comparative analysis and performance evaluation,” *Scientific African*, 28. Available at: <https://doi.org/10.1016/j.sciaf.2025.e02649>.

Park, J.S. *et al.* (2023a) “Generative Agents: Interactive Simulacra of Human Behavior,” *UIST 2023 - Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, Inc. Available at: <https://doi.org/10.1145/3586183.3606763>.

Park, J.S. *et al.* (2023b) “Generative Agents: Interactive Simulacra of Human Behavior,” *UIST 2023 - Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, Inc. Available at: <https://doi.org/10.1145/3586183.3606763>.

Perdana, R.S. and Adikara, P.P. (2025) “Multi-task Learning for Named Entity Recognition and Intent Classification in Natural Language Understanding Applications,” *Journal of Information Systems Engineering and Business Intelligence*, 11(1), pp. 1–16. Available at: <https://doi.org/10.20473/jisebi.11.1.1-16>.

Radford, A. *et al.* (2023) *Robust Speech Recognition via Large-Scale Weak Supervision*.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Shen, S. (2025) *Analysis and Application Research of Typical Technologies of Generative AI in the Context of Games*, *Transactions on Computer Science and Intelligent Systems Research*.

Yulianti, E. *et al.* (2024) "Named entity recognition on Indonesian legal documents: a dataset and study using transformer-based models," *International Journal of Electrical and Computer Engineering*, 14(5), pp. 5489–5501. Available at: <https://doi.org/10.11591/ijece.v14i5.pp5489-5501>.

Zhang, S., Yi, N. and Ma, Y. (2024) "A Survey of Computation Offloading with Task Types." Available at: <http://arxiv.org/abs/2401.01017>.

Zheng, S. *et al.* (2024) "MemoryRepository for AI NPC." Available at: <https://doi.org/10.1109/ACCESS.2023.0322000>.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP



ANDI RIZKI MAHESA

Lahir di Cirebon pada tanggal 08 Agustus 2004. Lulus dari SDN Sukmajaya 04 Depok pada tahun 2016, SMP Bina Sejahtera Depok pada tahun 2019, dan SMKN 04 Depok pada tahun 2022. Saat ini sedang menempuh pendidikan Diploma IV pada Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

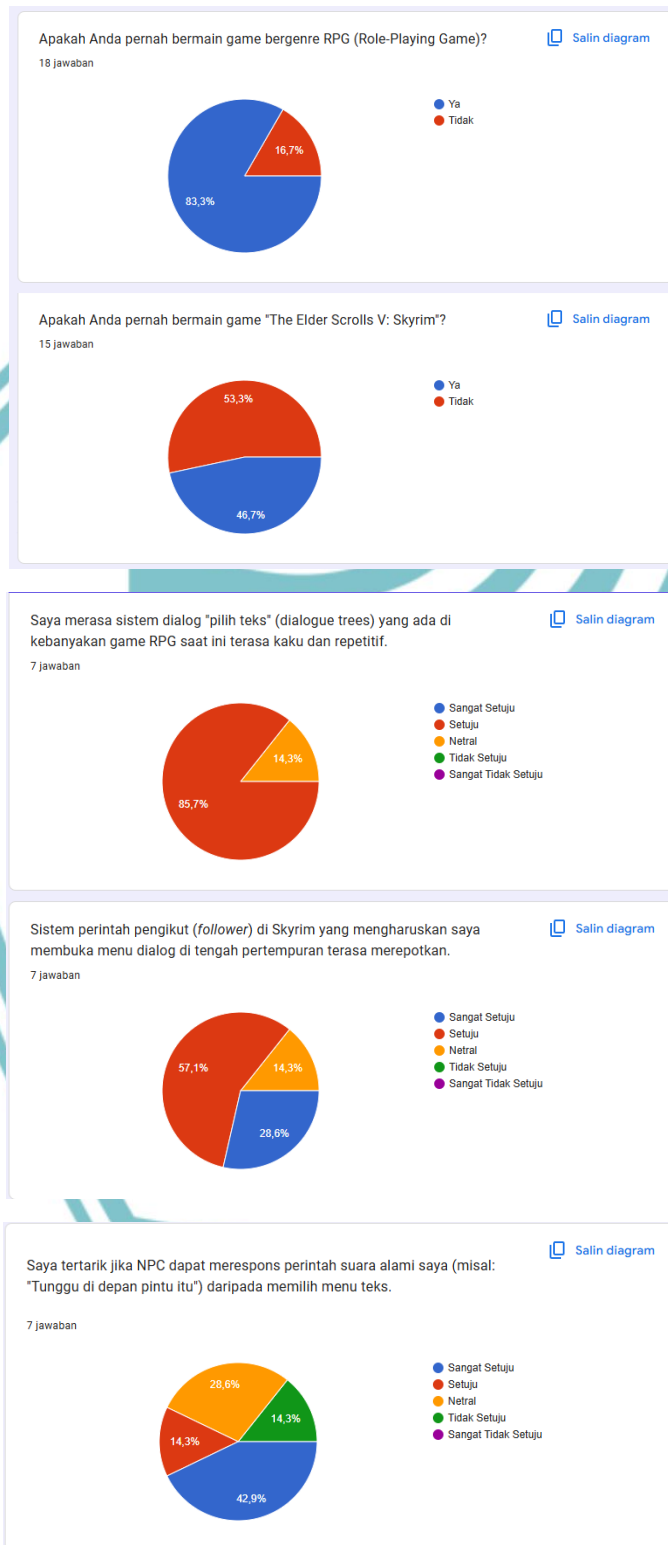
**POLITEKNIK
NEGERI
JAKARTA**

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

LAMPIRAN

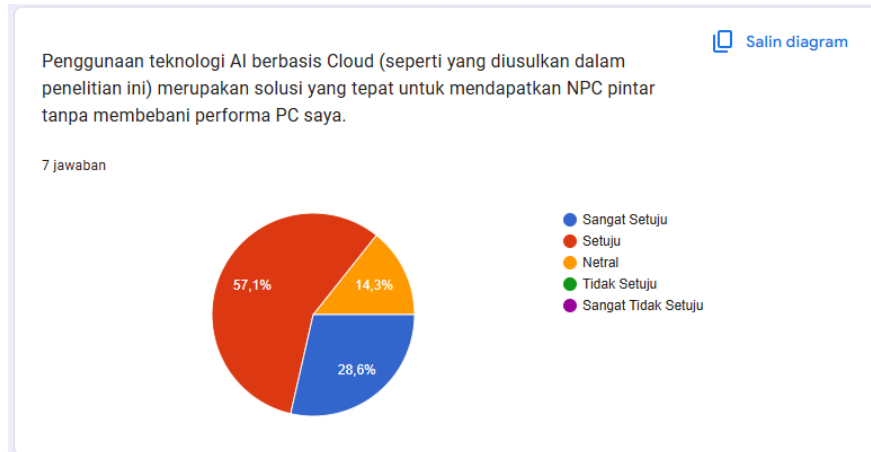
Lampiran 1 Survei Interaksi NPC Berbasis AI pada *Game* RPG (Skyrim)



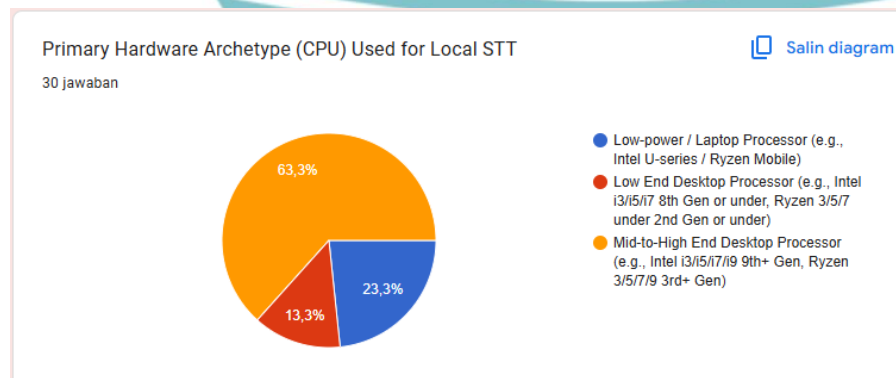
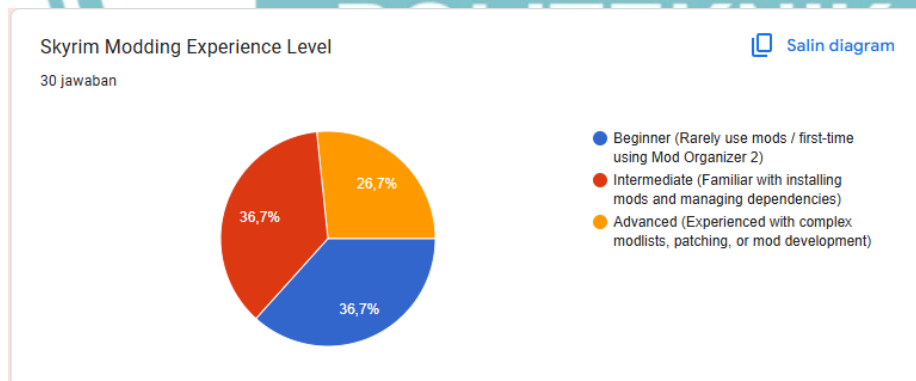
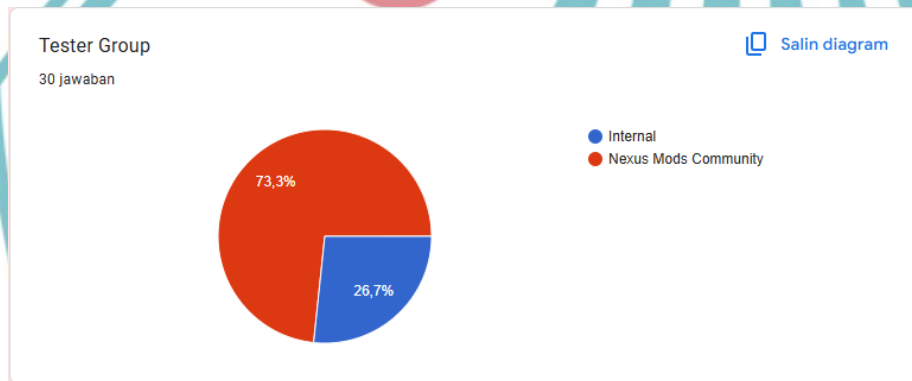
(Lanjutan)

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Lampiran 2 Survei User Usability & Performance Evaluation: Serana Cloud AI





© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

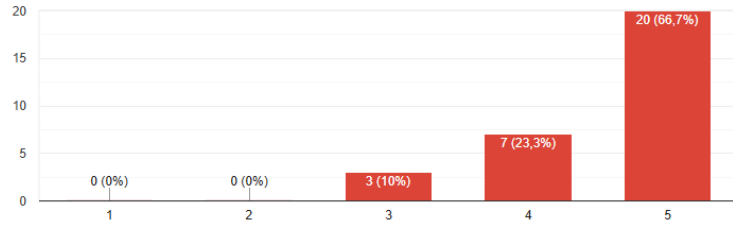
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(Lanjutan)

I think that I would like to use this voice control system/mods frequently.

[Salin diagram](#)

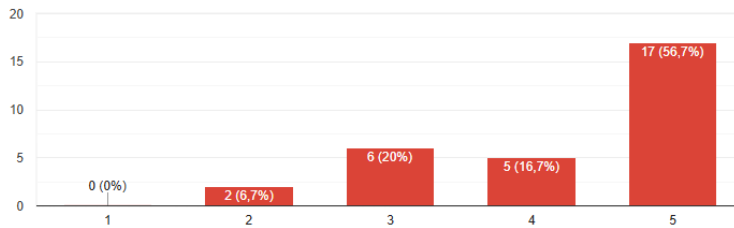
30 jawaban



In general, how would you rate the responsiveness/latency of Serana's physical reactions to your voice commands?

[Salin diagram](#)

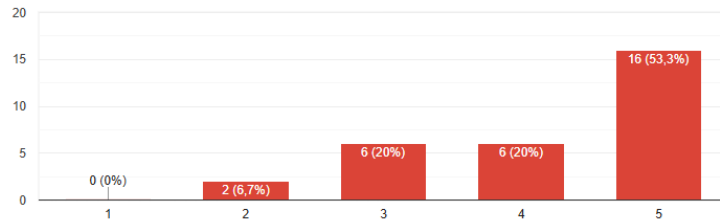
30 jawaban



How would you rate the naturalness and audio quality of Serana's cloned voice generated by this mod?

[Salin diagram](#)

30 jawaban





© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 3 Dokumentasi Chat Validasi Data oleh Data Validator

