



**ANALISIS IMPLEMENTASI K-DINAMIS DAN
REVALUASI IP PADA DETEKSI SERANGAN DDOS
BERBASIS *ENTROPY* PADA WEB *SERVER* UBUNTU**

SKRIPSI

**SAFINA AYU RASYA
2207421058**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026**



**ANALISIS IMPLEMENTASI K-DINAMIS DAN
REVALUASI IP PADA DETEKSI SERANGAN DDOS
BERBASIS *ENTROPY* PADA WEB *SERVER* UBUNTU**

SKRIPSI

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

**SAFINA AYU RASYA
2207421058**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026**



SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Safina Ayu Rasya

NIM : 2207421058

Jurusan/Program Studi: T.Informatika dan Komputer / T. Multimedia dan Jaringan

Judul skripsi : ANALISIS IMPLEMENTASI K-DINAMIS DAN
REVALUASI IP PADA DETEKSI SERANGAN DDOS BERBASIS ENTROPY
PADA WEB SERVER UBUNTU

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari 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, 9 Juni 2026

Yang membuat pernyataan

(Safina Ayu Rasya)

NIM. 2207421058

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta milik 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 mengaitkan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Safina Ayu Rasya

NIM : 2207421058

Program Studi : Teknik Multimedia dan Jaringan

Judul Skripsi : ANALISIS IMPLEMENTASI K-DINAMIS DAN
REVALUASI IP PADA DETEKSI SERANGAN DDOS BERBASIS ENTROPY
PADA WEB SERVER UBUNTU

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal 24,
Bulan Juni, Tahun 2026 dan dinyatakan LULUS.

Disahkan oleh:

Pembimbing I : Ayu Rosyida Zain, S.ST., M.T

Penguji I : Dr. Defiana Arnaldy, S.Tp., M.Si.

Penguji II : Asep Kurniawan S.Pd., M. Kom

Penguji III : Andika Riyadi Jasril, S.Pd., M.Pd.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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "ANALISIS IMPLEMENTASI K-DINAMIS DAN revaluasi IP PADA DETEKSI SERANGAN DDoS BERBASIS *ENTROPY* PADA WEB SERVER UBUNTU". Dalam proses penyusunan skripsi ini, penulis memperoleh banyak bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Allah Swt. atas segala rahmat, karunia, dan kemudahan yang diberikan sehingga penulis dapat menyelesaikan skripsi ini.
2. Kedua orang tua tercinta atas doa, kasih sayang, dukungan, dan motivasi yang tiada henti selama penulis menempuh pendidikan.
3. Ibu Ayu Rosyida Zain, S.ST., M.T, atas bimbingan, ilmu, arahan, dan motivasi yang diberikan selama perkuliahan dan penyusunan skripsi ini.
4. Pratama, Abian, Azzuri, Christian, Salsabilla, Reishafa, dan Izzaturrachmi atas dukungan, bantuan, serta kebersamaan yang diberikan selama masa perkuliahan hingga penyelesaian skripsi ini.
5. Kasci, kucing kesayangan penulis, yang telah pergi setelah menemani penulis selama kurang lebih dua tahun dan memberikan kenyamanan serta semangat selama proses perkuliahan dan penyusunan skripsi ini.

Depok, 9 Juni 2026

Safina Ayu Rasya



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 Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini :

Nama : Safina Ayu Rasya

NIM : 2207421058

Jurusan/Program Studi : T.Informatika dan Komputer / T. Multimedia dan Jaringan

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

ANALISIS IMPLEMENTASI K-DINAMIS DAN REVALUASI IP PADA DETEKSI SERANGAN DDOS BERBASIS ENTROPY PADA WEB SERVER UBUNTU

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, 9 Juni 2026

Yang menyatakan

Safina Ayu Rasya



ANALISIS IMPLEMENTASI K-DINAMIS DAN REVALUASI IP PADA DETEKSI SERANGAN DDOS BERBASIS ENTROPY PADA WEB SERVER UBUNTU

Abstrak

Serangan *Distributed Denial of Service (DDoS)* berbasis *HTTP Flood* menjadi ancaman serius bagi ketersediaan layanan web server karena karakteristiknya yang menyerupai trafik pengguna sah. Penelitian ini mengimplementasikan sistem deteksi serangan DDoS secara real-time pada web server Ubuntu menggunakan metode entropy Shannon dengan parameter *K-dinamis* dan mekanisme revaluasi IP. Sistem bekerja melalui mekanisme *sliding window* untuk menganalisis distribusi trafik, menghitung *normalized entropy*, menentukan *threshold adaptif* menggunakan *K-dinamis*, serta melakukan revaluasi terhadap alamat IP mencurigakan. Pengujian dilakukan dalam tiga skenario, yaitu kondisi normal, *flash crowd*, dan serangan *HTTP Flood* menggunakan *GoldenEye* dan *Locust* pada lingkungan jaringan lokal berbasis *VirtualBox*. Hasil pengujian menunjukkan bahwa pada kondisi serangan, sistem berhasil mencapai nilai Akurasi sebesar 90,36%, Presisi 100%, dan Recall sebesar 66,67%. Implementasi metode ini mampu menjaga stabilitas penggunaan memori (RAM) server pada rata-rata 48%, namun beban komputasi deteksi real-time dan tingginya volume request menyebabkan lonjakan penggunaan CPU hingga mencapai maksimum 100,00% pada puncak serangan.

Kata Kunci : DDoS, Shannon Entropy, K-dinamis, web server, sliding window

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME.....	ii
LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	v
Abstrak.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
BAB I	
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	3
1.4.1 Tujuan.....	3
1.4.2 Manfaat.....	4
1.5 Sistematika Penulisan.....	4
BAB II	
TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terdahulu.....	6
2.2 Serangan Distributed Denial of Service (DDoS).....	7
2.3 Ubuntu Server.....	8
2.4 Sliding Window.....	8
2.5 Metode Deteksi DDoS Berbasis Entropy.....	9
2.5.1 Konsep Shannon Entropy.....	9
2.5.2 Normalisasi Entropy.....	9
2.5.3 Mekanisme Deteksi dan Perubahan Pola Trafik.....	10
2.6 Metode Threshold Dinamis dengan Parameter K-Dinamis.....	10
2.6.1 Threshold Statis dan Dinamis.....	10
2.6.2 K-Dinamis.....	11
2.6.3 Perhitungan Threshold.....	11
2.7 Revaluasi IP Mencurigakan.....	12
2.8 Web Server dan Layanan HTTP.....	13
2.9 Confusion Matrix dan Metrik Evaluasi.....	13
2.10 Locust.....	15
2.11 GoldenEye.....	15



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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB III	
METODE PENELITIAN.....	16
3.1 Rancangan Penelitian.....	16
3.2 Tahapan Penelitian.....	16
3.3 Objek Penelitian.....	17
BAB IV	
HASIL DAN PEMBAHASAN.....	19
4.1. Analisis Kebutuhan.....	19
4.1.1 Kebutuhan Fungsional.....	19
4.1.2 Kebutuhan Non-Fungsional.....	20
4.1.3 Spesifikasi Perangkat.....	20
4.2 Perancangan Sistem.....	21
4.2.1 Blok Diagram.....	21
4.2.2 Flowchart Sistem.....	22
4.2.3 Topologi Jaringan.....	23
4.2.4 Pengalamatan IP Address.....	24
4.2.5 Spesifikasi Virtual Machine.....	25
4.3. Implementasi Sistem.....	26
4.3.1 Instalasi Virtual Machine dan Konfigurasi Ubuntu Server.....	26
4.3.2 Instalasi dan Konfigurasi Web Server Apache2.....	28
4.3.3 Instalasi MySQL dan Konfigurasi Database.....	30
4.3.4 Pembuatan Web Server.....	36
4.3.5 Implementasi Perhitungan Entropy.....	38
4.3.6 Implementasi Perhitungan K-Dinamis dan Threshold.....	44
4.3.7 Klasifikasi Hasil Deteksi Tahap Pertama.....	47
4.3.8 Implementasi Tahap Revaluasi IP.....	49
4.3.9 Implementasi Pengiriman Alert ke Telegram.....	54
4.3.10 Instalasi Alat Pengujian.....	57
4.4 Pengujian.....	58
4.4.1. Deskripsi Pengujian.....	58
4.4.2. Prosedur Pengujian.....	59
4.4.3 Data Hasil Pengujian.....	63
4.4.4 Analisa Data Pengujian Fungsionalitas Sistem Deteksi Serangan DDoS.....	72
4.4.5 Analisa Data Pengujian Performa Sistem Deteksi dengan Confusion Matrix.....	74
4.4.6 Analisa Data Pengujian Dampak Sistem Deteksi Terhadap Sumber Daya Server.....	75
BAB V	
PENUTUP.....	78
5.1 Simpulan.....	78



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 Politeknik Negeri Jakarta

5.2. Saran.....	79
DAFTAR PUSTAKA.....	80
DAFTAR RIWAYAT HIDUP PENULIS.....	83
LAMPIRAN.....	84





DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	16
Gambar 4.1 Blok Diagram Sistem.....	21
Gambar 4.2 Flowchart Sistem Deteksi Tahap 1.....	22
Gambar 4.3 Flowchart Sistem Deteksi Tahap 1.....	23
Gambar 4.4 Topologi Jaringan.....	24
Gambar 4.5 Instalasi OpenSSH.....	27
Gambar 4.6 Pengecekan Status OpenSSH.....	27
Gambar 4.7 Perintah Memperbarui Daftar Paket.....	28
Gambar 4.8 Instalasi Web Server Apache2.....	28
Gambar 4.9 Instalasi PHP, libapache2, dan mysql.....	28
Gambar 4.10 Konfigurasi Firewall UFW untuk Apache2.....	29
Gambar 4.11 Aktivasi Web Server Apache2.....	29
Gambar 4.12 Instalasi MySQL Server.....	30
Gambar 4.13 Pengecekan Status MySQL.....	30
Gambar 4.14 Pengaksesan MySQL dari Terminal.....	30
Gambar 4.15 Daftar Tabel Database.....	31
Gambar 4.16 Struktur Tabel raw_traffic.....	32
Gambar 4.17 Struktur Tabel entropy_log.....	32
Gambar 4.18 Struktur Tabel dynamic_k.....	33
Gambar 4.19 Struktur Tabel threshold.....	33
Gambar 4.20 Struktur Tabel suspicious_ip.....	34
Gambar 4.21 Struktur Tabel reevaluation_log.....	34
Gambar 4.22 Struktur Tabel classification.....	35
Gambar 4.23 Koneksi PHP ke MySQL.....	35
Gambar 4.24 Penambahan Include logger.php di dalam login.php.....	36
Gambar 4.25 Tipe Data yang Dicatat dalam Sebuah Request.....	37
Gambar 4.26 Skrip Pencatatan Data ke dalam Tabel raw_traffic.....	37
Gambar 4.27 Flowchart Pengambilan Data dalam Window.....	38
Gambar 4.28 Skrip Pengambilan dan Penyimpanan Data dalam Window.....	39
Gambar 4.29 Flowchart Pemanggilan Fungsi dan Validasi Window Kosong.....	40
Gambar 4.30 Skrip Pemanggilan Fungsi getSlidingWindow.....	40
Gambar 4.31 Skrip Pemanggilan Fungsi getSlidingWindow.....	41
Gambar 4.32 Flowchart Perhitungan Entropy dan Normalized Entropy.....	42
Gambar 4.33 Skrip Perhitungan Entropy dan Normalized Entropy.....	42
Gambar 4.34 Flowchart Perhitungan K-Dinamis.....	44
Gambar 4.35 Skrip Perhitungan K-Dinamis.....	44
Gambar 4.36 Flowchart Perhitungan Threshold.....	45



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.37 Skrip Perhitungan Threshold.....	46
Gambar 4.38 Flowchart Klasifikasi Tahap Pertama.....	47
Gambar 4.39 Skrip Klasifikasi Tahap Pertama.....	48
Gambar 4.40 Flowchart Perhitungan Entropy IP Mencurigakan.....	49
Gambar 4.41 Skrip Perhitungan Entropy IP Mencurigakan.....	50
Gambar 4.42 Flowchart Perhitungan Threshold IP Mencurigakan.....	51
Gambar 4.43 Skrip Perhitungan Threshold IP Mencurigakan.....	52
Gambar 4.44 Flowchart Klasifikasi Tahap Kedua.....	53
Gambar 4.45 Skrip Klasifikasi Tahap Kedua.....	53
Gambar 4.46 Skrip Penyimpanan Hasil revaluasi IP.....	54
Gambar 4.47 Pembuatan Bot Telegram.....	55
Gambar 4.48 Skrip Pengiriman Alert ke Telegram.....	56
Gambar 4.49 Penginstalan Python3 dan Modul Pip.....	57
Gambar 4.50 Penginstalan Python3 dan Modul Pip.....	57
Gambar 4.51 Pembuatan Virtual Environment.....	57
Gambar 4.52 Pengaktifan Virtual Environment.....	58
Gambar 4.53 Pengecekan Versi Locust.....	58
Gambar 4.54 Cloning Repositori GoldenEye dari Github.....	58
Gambar 4.55 Grafik Hasil Perhitungan Parameter Deteksi Kondisi Normal.....	64
Gambar 4.56 Grafik Hasil Perhitungan Parameter Deteksi Kondisi Flash Crowd.....	65
Gambar 4.57 Grafik Hasil Perhitungan Parameter Deteksi Kondisi Serangan.....	67
Gambar 4.58 Hasil Pengiriman Notifikasi ke Telegram.....	68
Gambar 4.59 Penggunaan Sumber Daya Pada Kondisi Normal.....	76
Gambar 4.60 Penggunaan Sumber Daya Pada Kondisi Flash Crowd.....	76
Gambar 4.61 Penggunaan Sumber Daya Pada Kondisi Serangan.....	77



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 Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	6
Tabel 4.1 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras.....	20
Tabel 4.2 Pengalamatan IP Address.....	24
Tabel 4.3 Spesifikasi Virtual Machine.....	25
Tabel 4.4 Jumlah VM Pada Kondisi Trafik Normal.....	60
Tabel 4.5 Jumlah VM Pada Kondisi Trafik Flash Crowd.....	60
Tabel 4.6 Jumlah VM Pada Kondisi Serangan.....	61
Tabel 4.7 Detail Simulasi Kondisi Serangan.....	61
Tabel 4.8 Mode Uji Sumber Daya Server.....	63
Tabel 4.9 Hasil Perhitungan Parameter Deteksi Pada Kondisi Normal.....	63
Tabel 4.10 Hasil Perhitungan Parameter Deteksi Pada Kondisi Flash Crowd.....	65
Tabel 4.11 Hasil Perhitungan Parameter Deteksi Pada Kondisi Serangan.....	66
Tabel 4.12 Hasil Perhitungan Tahap Revaluasi IP Pada Trafik Mencurigakan.....	66
Tabel 4.13 Hasil Confusion Matrix Kondisi Normal.....	68
Tabel 4.14 Hasil Confusion Matrix Kondisi Flash Crowd.....	69
Tabel 4.15 Hasil Confusion Matrix Kondisi Serangan.....	70
Tabel 4.16 Hasil Pemantauan Sumber Daya Kondisi Normal.....	71
Tabel 4.17 Hasil Pemantauan Sumber Daya Kondisi Flash Crowd.....	71
Tabel 4.18 Hasil Pemantauan Sumber Daya Kondisi Serangan.....	72

**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 Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Di era transformasi digital, ketersediaan layanan (*availability*) pada sektor-sektor krusial seperti perbankan, kesehatan, dan pemerintahan menjadi pilar utama dalam menjaga kepercayaan publik (Cheema et al., 2022). Namun, ketergantungan pada jaringan publik membuka celah keamanan yang signifikan (Kurniawan dan Matin, 2025), terutama terhadap serangan *Distributed Denial of Service* (DDoS). Pada kuartal ketiga tahun 2025, Cloudflare mencatat dan memitigasi 8,3 juta serangan DDoS dengan 29% atau 2,4 juta jenis serangannya berbasis HTTP. Serangan pada lapisan aplikasi (HTTP Flood) memiliki karakteristik yang menyerupai trafik sah pengguna sehingga sulit dibedakan oleh sistem keamanan konvensional (Solichin dan Nugroho, 2024).

Salah satu tantangan dalam pengelolaan *web server* berbasis Ubuntu adalah kerentanannya terhadap DDoS karena karakteristiknya yang melayani permintaan secara kontinu dan mengalokasikan sumber daya untuk setiap *request* (Catillo, Pecchia dan Villano, 2022). Penelitian yang dilakukan Irawan et al. (2025) menunjukkan serangan HTTP Flood pada *web server* Ubuntu tidak hanya menguras *bandwidth*, tetapi juga meningkatkan penggunaan CPU dan RAM. Hal ini menyebabkan penurunan performa hingga penolakan layanan total bagi pengguna sah. Berdasarkan kondisi ini, diperlukan mekanisme deteksi adaptif yang untuk mengatasi karakteristik serangan yang lebih dinamis.

Pendekatan yang umum digunakan dalam mendeteksi serangan DDoS adalah metode *entropy* (Ding et al., 2021). Prinsip utamanya adalah membandingkan penurunan nilai *entropy* terhadap suatu ambang batas (*threshold*) saat trafik menjadi terlalu seragam akibat serangan DDoS (Fan et al., 2022). Namun, efektivitas metode ini masih memiliki kekurangan karena penggunaan *threshold* statis. Menurut Hemmati et al. (2021), nilai *threshold* yang statis sering kali gagal menangani fenomena *flash crowd* (lonjakan trafik sah yang mendadak), yang mengakibatkan tingginya tingkat *false positive* karena pengguna sah yang teridentifikasi mencurigakan sehingga tidak dapat mengakses server.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Penelitian yang dilakukan oleh Pebriyanto dan Suryani (2025) mengusulkan pengembangan metode deteksi DDoS berbasis *entropy* dengan dua kontribusi utama, yaitu penggunaan parameter sensitivitas dinamis (K-dinamis) dan mekanisme revaluasi IP. Penentuan nilai K secara dinamis memungkinkan sistem menyesuaikan sensitivitas *threshold* terhadap perubahan pola trafik jaringan, sehingga lebih adaptif dibandingkan penggunaan nilai statis. Selain itu, mekanisme revaluasi IP memungkinkan sistem melakukan verifikasi ulang terhadap alamat IP yang terdeteksi mencurigakan untuk membedakan antara penyerang dan pengguna sah. Hasil pengujian menunjukkan bahwa kombinasi kedua mekanisme tersebut mampu memberikan kinerja deteksi yang optimal dalam mendeteksi serangan DDoS pada dataset yang digunakan.

Berdasarkan latar belakang di atas, penelitian ini berfokus pada implementasi mekanisme deteksi serangan DDoS secara *near real-time* berbasis *entropy* dengan parameter K dinamis dan revaluasi IP pada *web server* Ubuntu lokal. Pendekatan ini mengadaptasi metode yang diusulkan pada penelitian sebelumnya ke dalam lingkungan server nyata. Dengan demikian, penelitian ini diharapkan mampu memberikan gambaran awal mengenai efektivitas deteksi DDoS pada lingkungan *web server* berbasis Ubuntu.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah disusun, maka rumusan masalah yang menjadi fokus dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan metode deteksi DDoS berbasis *entropy* dengan K-dinamis dan revaluasi IP dalam mendeteksi anomali trafik pada *web server* Ubuntu?
2. Bagaimana efektivitas kombinasi K-dinamis dan revaluasi IP dalam mendeteksi serangan DDoS berdasarkan hasil *confusion matrix*?
3. Bagaimana dampak penerapan sistem deteksi DDoS berbasis *entropy* dengan K-dinamis dan revaluasi IP terhadap penggunaan CPU dan RAM *web server*?



1.3 Batasan Masalah

Penelitian yang dilakukan untuk menganalisis implementasi sistem deteksi serangan DDoS memiliki beberapa batasan masalah sebagai berikut:

1. Jenis serangan DDoS yang dianalisis adalah HTTP Flood (layer 7) dengan target endpoint POST pada *web server*.
2. Lingkungan pengujian dilakukan pada *web server* Ubuntu yang dijalankan secara lokal (*private LAN*) menggunakan platform virtualbox dengan simulasi serangan dilakukan dari VM *attacker* ke VM target.
3. Metode deteksi dirancang menggunakan perhitungan Shannon Entropy berdasarkan distribusi alamat IP dan jumlah permintaan (*request*).
4. Mekanisme deteksi serangan DDoS menggunakan metode berbasis *entropy* dengan parameter K-dinamis serta mekanisme revaluasi alamat IP untuk klasifikasi trafik mencurigakan.
5. Penelitian tidak menggunakan pendekatan *machine learning* ataupun *intrusion detection system* (IDS) konvensional sebagai pembandingan.
6. Parameter evaluasi dibatasi dalam tingkat *accuracy*, *precision*, *recall*, *F1-score* serta penggunaan CPU dan RAM *web server*.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun tujuan penelitian yang dilakukan adalah sebagai berikut:

1. Mengimplementasikan metode deteksi serangan DDoS berbasis *entropy* dengan parameter K-dinamis dan mekanisme revaluasi IP untuk mendeteksi anomali trafik pada *web server* Ubuntu.
2. Mengevaluasi efektivitas kombinasi parameter K-dinamis dan revaluasi IP dalam mendeteksi serangan DDoS berdasarkan hasil *confusion matrix*.
3. Menganalisis dampak penerapan sistem deteksi DDoS berbasis *entropy* dengan K-dinamis dan revaluasi IP terhadap penggunaan CPU dan RAM pada lingkungan server.

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 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 Politeknik Negeri Jakarta

1.4.2 Manfaat

Adapun manfaat penelitian yang dilakukan adalah sebagai berikut:

1. Memberikan gambaran implementatif mengenai sistem deteksi serangan DDoS berbasis *entropy* dengan parameter K dinamis dan mekanisme revaluasi IP pada *web server* Ubuntu.
2. Memberikan kontribusi terhadap pengembangan studi di bidang keamanan jaringan dan sistem informasi dalam pendekatan deteksi serangan DDoS berbasis *web layer* (Layer 7).
3. Menjadi referensi bagi administrator server dalam memilih mekanisme deteksi DDoS yang adaptif dan mampu mengurangi *false positive*.
4. Menjadi referensi dalam pengembangan penelitian lanjutan terkait deteksi anomali maupun pengecekan ulang IP yang terindikasi mencurigakan.

1.5 Sistematika Penulisan

Berikut sistematika penulisan yang digunakan dalam penyusunan proposal penelitian:

1. BAB I PENDAHULUAN
Bab ini berisi uraian mengenai latar belakang penelitian, perumusan masalah yang disusun berdasarkan latar belakang tersebut, batasan masalah yang digunakan untuk memperjelas ruang lingkup penelitian, serta tujuan dan manfaat yang diharapkan dari pelaksanaan penelitian.
2. BAB II TINJAUAN PUSTAKA
Bab ini membahas landasan teori dan konsep-konsep yang berkaitan dengan permasalahan penelitian, serta kajian terhadap penelitian-penelitian terdahulu yang relevan sebagai dasar dan pembanding dalam penelitian ini.
3. BAB III METODE PENELITIAN
Bab ini menjelaskan metode penelitian yang digunakan, meliputi perancangan penelitian, tahapan pelaksanaan penelitian, objek dan model penelitian, teknik pengumpulan data, metode analisis data, serta jadwal pelaksanaan penelitian.
4. BAB IV HASIL DAN PEMBAHASAN



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 Politeknik Negeri Jakarta

Bab ini memuat hasil penelitian yang mencakup analisis kebutuhan sistem, perancangan dan implementasi sistem, pengujian yang dilakukan, serta pembahasan dan analisis hasil deteksi serangan terhadap sistem yang dikembangkan.

5. BAB V PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran-saran yang dapat digunakan sebagai acuan untuk pengembangan dan penelitian selanjutnya.





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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Tabel berikut menjelaskan penelitian terdahulu untuk memperlihatkan posisi keterbaruan penelitian ini dibandingkan literatur yang ada.

Tabel 2.1 Penelitian Terdahulu

Nama Penulis	Judul	Hasil	Kekurangan	Perbedaan
Pebrianto dan Suryani (2025)	Adaptive DDoS Attack Detection: Entropy-based Model with Dynamic Threshold and Suspicious IP Reevaluation	Deteksi DDoS menggunakan metode entropy dengan dynamic threshold (K dynamic) dan suspicion IP re-evaluation berdasarkan dataset CIC-DDoS2019 mencapai akurasi hingga 100%.	Penelitian keakuratan deteksi dilakukan berdasarkan dataset yang sudah ada (bukan real-time).	Penelitian yang dilakukan secara <i>real-time</i> berdasarkan hasil deteksi pada web server Ubuntu.
Tsobdjou et al. (2022)	An Online Entropy-Based DDoS Flooding Attack Detection System With Dynamic Threshold	Sistem deteksi DDoS berbasis entropy yang terdiri dari 5 modul dan threshold dinamis menunjukkan tingkat akurasi, detection rate, dan precision yang tinggi.	Nilai K tidak berubah selama pengujian (bersifat statis). Tidak punya mekanisme untuk antisipasi kondisi lonjakan trafik sah.	Penelitian yang dilakukan secara <i>real-time</i> berdasarkan hasil deteksi pada web server Ubuntu, parameter K dinamis.
Park et al. (2021)	HTTP DDoS Flooding	Sistem deteksi dan	Fokus pada lingkungan	Penelitian 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

	Attack Mitigation in Software-Defined Networking	mitigasi serangan HTTP Flood berbasis Software-Defined Networking (SDN) dengan mekanisme mitigasi CAPTCHA berhasil memitigasi serangan dengan mengalihkan dan memblokir klien mencurigakan	SDN (Software Defined Networking), threshold perbandingan bersifat statis.	dilakukan diimplementasikan dalam web server Ubuntu menggunakan deteksi berbasis <i>entropy</i> dan <i>threshold</i> adaptif.
--	--	--	--	---

2.2 Serangan Distributed Denial of Service (DDoS)

Serangan *Distributed Denial of Service* (DDoS) dilakukan dengan membanjiri *bandwidth* suatu server sehingga mengganggu ketersediaan layanan, menyebabkan penolakan layanan bagi pengguna, serta menurunkan sumber daya jaringan seperti CPU dan kapasitas memori (Solichin dan Nugroho, 2024). Berdasarkan lapisan pada model Open Systems Interconnection (OSI), Vishwakarma dan Jain (2020) mengklasifikasikan serangan DDoS pada *app layer* dan *infrastructure layer*. Serangan pada *app layer* menargetkan layanan aplikasi dengan mengirimkan permintaan yang terlihat valid, seperti HTTP Flood dan DNS Amplification, sedangkan serangan pada *infrastructure layer* mengeksploitasi kerentanan pada lapisan transport dan jaringan, contohnya UDP Flood dan SYN Flood.

Serangan HTTP Flood dilakukan dengan cara mengirimkan permintaan HTTP dalam jumlah besar secara simultan, sering kali menggunakan botnet, sehingga server dipaksa untuk terus memproses permintaan yang akhirnya mengakibatkan turunnya ketersediaan layanan (Park et al., 2021).



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 mengumpukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.3 Ubuntu Server

Ubuntu Server merupakan sistem operasi berbasis Linux yang dikembangkan untuk memenuhi kebutuhan lingkungan server. Sistem operasi ini bersifat *open-source* dan dapat digunakan untuk berbagai layanan, seperti layanan web, *database*, cloud, hingga pengelolaan keamanan jaringan. Ubuntu Server memiliki reputasi yang baik dalam hal stabilitas, penggunaan sumber daya yang efisien, serta dukungan pembaruan jangka panjang, sehingga banyak dimanfaatkan dalam kegiatan penelitian maupun pengembangan sistem (Nasrulah et al., 2025).

Pada penelitian ini, Ubuntu Server dipilih sebagai platform utama karena mampu menjalankan berbagai layanan secara efisien. Selain itu, sistem ini mendukung instalasi perangkat lunak tambahan, seperti *web server* Apache, serta memberikan kemudahan dalam proses konfigurasi keamanan dan pemantauan jaringan. Keunggulan lainnya terletak pada kompatibilitasnya dengan beragam perangkat keras dan lingkungan virtual, seperti VirtualBox maupun layanan cloud server.

2.4 Sliding Window

Mekanisme *sliding window* bekerja dengan cara memproses data trafik jaringan dalam rentang waktu tertentu yang terus bergeser (*sliding*) mengikuti aliran trafik. Pada mekanisme ini, setiap *window* mengumpulkan dan memproses sejumlah data atau durasi data tertentu, kemudian *window* akan bergeser maju dengan interval yang telah ditentukan untuk melakukan proses analisis berikutnya. Metode *sliding window* memungkinkan terjadinya tumpang tindih (*overlap*) antar *window*, sehingga satu kejadian atau data dapat termasuk ke dalam beberapa *window* sekaligus. Dengan mekanisme tersebut, sistem dapat menangkap serta mengevaluasi pola trafik pada periode waktu yang berurutan secara lebih kontinu (Baldini dan Amerini, 2022).

Dalam proses deteksi anomali pada data jaringan, *sliding window* berperan penting sebagai metode segmentasi temporal. Mekanisme ini memungkinkan sistem memantau perubahan data dalam interval waktu tertentu sehingga pola trafik dapat dipisahkan dan dianalisis secara lebih terstruktur. Duan et al. (2022)



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 Politeknik Negeri Jakarta

menjelaskan bahwa penggunaan *sliding window* pada deteksi multi-skala mampu meningkatkan kemampuan deteksi anomali jaringan secara signifikan karena dapat mengungkap informasi yang sebelumnya tersembunyi dalam aliran trafik yang panjang.

2.5 Metode Deteksi DDoS Berbasis Entropy

2.5.1 Konsep Shannon Entropy

Dalam teori informasi, *entropy* adalah besaran yang mengukur ketidakpastian variabel acak (Khumaidi, 2021). Konsep *entropy* yang dikembangkan lebih lanjut oleh Shannon dikenal sebagai Shannon Entropy. Shannon Entropy merupakan ukuran statistik yang digunakan untuk merepresentasikan tingkat ketidakteraturan atau kompleksitas suatu sistem berdasarkan distribusi probabilitas dari variabel yang diamati. Dalam keamanan jaringan, *entropy* digunakan sebagai indikator untuk mendeteksi anomali pada trafik jaringan. Rumus dari Shannon Entropy adalah sebagai berikut:

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2(p(x_i)) \quad (1)$$

Dalam persamaan ini, $H(X)$ adalah *entropy* dari variabel acak X , x_i merupakan nilai ke- i dari variabel acak X , dan $p(x_i)$ menyatakan hasil probabilitas dari kejadian X .

2.5.2 Normalisasi Entropy

Nilai *entropy* murni sangat bergantung pada jumlah variabel unik dalam satu jendela waktu, sehingga menyulitkan penetapan ambang batas yang konsisten jika jumlah IP unik berfluktuasi secara ekstrem. Oleh karena itu, diperlukan proses normalisasi untuk mentransformasikan nilai *entropy* ke dalam rentang skala 0 hingga 1 (Pebrianto dan Suryani, 2025). Entropy yang dinormalisasi memberikan gambaran derajat ketidakteraturan trafik yang relatif terhadap kapasitas maksimumnya. Rumus normalisasi *entropy* adalah sebagai berikut:

$$H_{norm} = \frac{H(X)}{\log_2(n)} \quad (2)$$



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

$H(X)$ adalah nilai Shannon Entropy dan n adalah jumlah alamat IP unik yang muncul dalam jendela waktu pengamatan.

2.5.3 Mekanisme Deteksi dan Perubahan Pola Trafik

Deteksi DDoS berbasis *entropy* bekerja dengan mengidentifikasi perubahan pola pada trafik jaringan melalui pengukuran tingkat ketidakteraturan dari aliran data. Pendekatan ini didasarkan pada asumsi bahwa pola trafik saat terjadi serangan DDoS cenderung mengalami penurunan nilai *entropy* secara signifikan, yang menandakan pergeseran mendadak dari trafik yang bersifat relatif acak menjadi lebih seragam dan terpusat. Penelitian yang dilakukan oleh Fan *et al.* (2022) menunjukkan bahwa metode berbasis *entropy* mampu mendeteksi serangan dengan cepat, khususnya melalui metode *combined entropy*, dimana nilai *entropy* yang dihasilkan pada kondisi serangan terbukti lebih rendah dibandingkan dengan kondisi trafik normal.

2.6 Metode Threshold Dinamis dengan Parameter K-Dinamis

2.6.1 Threshold Statis dan Dinamis

Dalam metode deteksi DDoS berbasis *entropy*, penentuan nilai *threshold* bersifat penting karena berkaitan langsung dengan tingkat sensitivitas sistem terhadap perubahan pola trafik jaringan. *Threshold* statis menetapkan nilai yang bersifat tetap, sehingga kurang mampu menyesuaikan diri terhadap fluktuasi trafik yang terjadi secara tiba-tiba. Hemmati *et al.* (2021) menunjukkan bahwa pendekatan *threshold* statis memiliki efektivitas yang rendah dalam menghadapi perubahan trafik yang sering terjadi pada jaringan yang bersifat dinamis, kondisi ini berpotensi meningkatkan tingkat *false positive* akibat kesalahan dalam membedakan trafik normal dan trafik serangan.

Sebagai alternatif, penerapan *threshold* dinamis memungkinkan sistem deteksi untuk beradaptasi secara langsung terhadap kondisi trafik yang sedang berlangsung. Dengan menyesuaikan nilai ambang berdasarkan karakteristik trafik terkini, sistem dapat meningkatkan akurasi dalam mengidentifikasi anomali. Tsobdjou *et al.* (2022) mengembangkan mekanisme *threshold* dinamis yang memantau fluktuasi trafik secara real-time, sehingga mampu menekan



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 Politeknik Negeri Jakarta

kemungkinan terjadinya false positive dan mempertahankan sensitivitas deteksi terhadap variasi trafik yang bersifat dinamis.

2.6.2 K-Dinamis

Penentuan nilai parameter sensitivitas k memiliki peran penting dalam proses pembentukan *threshold* pada metode deteksi berbasis *entropy*. Semakin besar nilai k , maka rentang *threshold* akan semakin longgar, sedangkan nilai k yang lebih kecil akan membuat sistem lebih sensitif terhadap perubahan trafik.

Dalam penelitian yang dilakukan oleh Tsobdjou et al. (2022), pendekatan deteksi berbasis entropi menggunakan *threshold* yang membandingkan nilai entropi dengan rata-rata dan standar deviasi. Namun, dalam pendekatan tersebut, nilai k ditetapkan secara statis. Hal ini menyebabkan sistem deteksi menjadi kurang responsif terhadap perubahan pola trafik jaringan yang fluktuatif sehingga dapat menurunkan akurasi deteksi.

Untuk mengatasi batasan tersebut, penelitian ini mengadopsi rumus K-Dinamis yang diusulkan oleh Pebrianto dan Suryani (2025). Secara matematis, rumus K-Dinamis yang digunakan adalah sebagai berikut:

$$k_{dynamic} = \frac{(\ln(1+E_{max}))^2}{(E_{max} + P_{total}) \cdot \ln(1+P_{total})} \quad (3)$$

Dimana E_{max} merupakan nilai *entropy* maksimum, sedangkan P_{total} adalah jumlah total trafik pada jendela waktu pengamatan. Penggunaan logaritma natural ($\ln$) dalam rumus bertujuan untuk menghasilkan skala pertumbuhan yang lebih lambat sehingga dampak dari lonjakan nilai ekstrem dapat dikurangi. Pendekatan ini memberikan beberapa keuntungan, seperti kemampuan untuk menyesuaikan *threshold* secara adaptif terhadap perubahan pola trafik dan meningkatkan akurasi deteksi dengan mempertimbangkan variasi dinamis pada data.

2.6.3 Perhitungan Threshold

Pada penelitian Pebrianto dan Suryani (2025), perhitungan *threshold* dilakukan secara dinamis dengan mempertimbangkan rata-rata *entropy*, standar deviasi, dan parameter K-Dinamis dari trafik yang berada dalam pengamatan. Pendekatan 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

memungkinkan *threshold* menyesuaikan diri terhadap perubahan karakteristik trafik pada setiap jendela waktu pengamatan. Perhitungan *threshold* dinamis dirumuskan sebagai berikut:

$$\delta = M_{entropy} - k \cdot \sigma_{entropy} \quad (4)$$

Dimana $M_{entropy}$ merupakan nilai rata-rata *entropy* yang diamati, $\sigma_{entropy}$ adalah standar deviasi *entropy*, dan k adalah parameter sensitivitas yang mengatur tingkat sensitivitas *threshold* terhadap perubahan nilai *entropy*.

2.7 Revaluasi IP Mencurigakan

Pebriyanto dan Suryani (2025) mengembangkan pendekatan revaluasi IP mencurigakan sebagai keterbaruan dari metode deteksi berbasis *entropy* sebelumnya yang mengacu pada algoritma dalam jurnal Tsobdjou et al. (2022). Pada pendekatan awal, proses deteksi dilakukan melalui perhitungan nilai *entropy* dan *threshold* untuk mengidentifikasi anomali trafik jaringan. Namun, pendekatan tersebut masih memiliki keterbatasan karena beberapa pola serangan dengan intensitas rendah atau trafik yang tidak konsisten dapat lolos dari proses deteksi awal. Kondisi ini menyebabkan beberapa alamat IP penyerang tidak teridentifikasi sebagai penyerang pada tahap pertama pendeteksian.

Untuk mengatasi permasalahan tersebut, Pebriyanto dan Suryani (2025) menambahkan tahapan revaluasi IP. Pada tahap ini, alamat IP yang dianggap mencurigakan akan disimpan sementara dalam sebuah himpunan data, kemudian dilakukan perhitungan ulang nilai *entropy* dan *threshold* pada interval pengamatan berikutnya. Sebuah alamat IP dikategorikan sebagai IP mencurigakan ketika nilai *entropy* $E(X)$ yang dihasilkan memenuhi kondisi berikut:

$$E(X) < \delta \quad (5)$$

Apabila kondisi tersebut terpenuhi, maka nilai *entropy* dari IP tersebut akan dimasukkan ke dalam Entropy Vector (EV) untuk digunakan pada proses selanjutnya. Secara matematis, mekanisme tersebut dirumuskan sebagai berikut:

$$E(V) = \{E(X) \mid E(X) < \delta\} \quad (6)$$



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 Politeknik Negeri Jakarta

Entropy Vector digunakan untuk menyimpan kumpulan nilai *entropy* dari alamat IP mencurigakan yang akan dianalisis kembali pada proses deteksi berikutnya. Untuk menjaga efisiensi memori dan mencegah akumulasi data berlebih, jumlah elemen pada EV dibatasi hingga maksimum N data terbaru. Dengan pendekatan ini, sistem dapat lebih berfokus pada pola trafik anomali terkini tanpa terbebani oleh data historis yang kurang relevan.

Pendekatan revaluasi IP memungkinkan sistem untuk melakukan verifikasi ulang terhadap pola serangan yang memiliki intensitas rendah. Selain itu, mekanisme ini membantu mengurangi kemungkinan alamat IP penyerang yang lolos saat proses deteksi awal. Apabila nilai *entropy* hasil revaluasi masih berada dibawah *threshold*, maka IP tersebut dikonfirmasi sebagai sumber serangan. Sebaliknya, jika nilai *entropy* berada di atas *threshold*, IP tersebut diklasifikasikan ulang sebagai trafik normal.

2.8 Web Server dan Layanan HTTP

Web server merupakan perangkat lunak yang berfungsi untuk menerima, memproses, dan merespons permintaan dari klien melalui protokol HTTP atau HTTPS (Chandra, 2019). Dalam menjalankan fungsinya, *web server* mendukung berbagai metode HTTP, seperti GET untuk mengambil data dan POST untuk mengirimkan data ke server. Salah satu implementasi *web server* yang paling umum digunakan adalah Apache HTTP Server karena memiliki fleksibilitas yang tinggi, dukungan komunitas yang luas, dan tersedia untuk berbagai sistem operasi (Januhandini dan Oktawati, 2025).

2.9 Confusion Matrix dan Metrik Evaluasi

Confusion matrix merupakan metode evaluasi yang umum digunakan untuk mengukur performa sistem klasifikasi (Erbani et al., 2024). Confusion matrix menggunakan tabel perbandingan antara kondisi sebenarnya dengan hasil prediksi atau klasifikasi yang dihasilkan oleh suatu metode.

Secara umum, *confusion matrix* terdiri dari empat komponen utama, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). *True Positive* (TP) merupakan kondisi ketika data positif berhasil diklasifikasikan



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 Politeknik Negeri Jakarta

sebagai positif. *False Positive* (FP) adalah kondisi ketika data negatif salah diklasifikasikan sebagai positif. *False Negative* (FN) merupakan kondisi ketika data positif salah diklasifikasikan sebagai negatif. Sementara itu, *True Negative* (TN) adalah kondisi ketika data negatif berhasil diklasifikasikan sebagai negatif. Keempat komponen tersebut digunakan sebagai dasar dalam menghitung berbagai metrik evaluasi performa klasifikasi.

Berdasarkan nilai pada *confusion matrix*, performa suatu metode klasifikasi dapat diukur menggunakan beberapa metrik evaluasi matematis berikut (Swaminathan dan Tantri, 2024):

- Accuracy merupakan metrik untuk mengukur tingkat ketepatan prediksi secara keseluruhan.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

- Precision digunakan untuk mengukur tingkat ketepatan prediksi terhadap data yang diklasifikasikan sebagai positif. Nilai precision yang tinggi menunjukkan bahwa kejadian *false positive* relatif rendah.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

- Recall digunakan untuk mengukur kemampuan metode dalam mengidentifikasi seluruh data positif secara benar dari keseluruhan data positif yang tersedia.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

- F1-Score merupakan nilai rata-rata harmonis antara precision dan recall yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut.

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (10)$$

**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 Politeknik Negeri Jakarta

2.10 Locust

Locust merupakan tools pengujian beban (*load testing*) berbasis Python yang bersifat open-source yang digunakan untuk mensimulasikan sejumlah pengguna yang mengakses suatu aplikasi atau layanan secara bersamaan (Shrivastava dan SB, 2020). Selain bersifat ringan dan relatif mudah digunakan, Locust memiliki keunggulan dalam fleksibilitas dalam mengatur jumlah pengguna, laju permintaan, serta pola akses (Yenugula et al., 2019).

2.11 GoldenEye

GoldenEye merupakan *tools* berbasis Python yang bersifat ringan dan mudah digunakan untuk melakukan pengujian yang mensimulasikan serangan Distributed Denial of Service (DDoS) pada lapisan aplikasi, khususnya serangan HTTP Flood (Deressu et al., 2026). Tools ini bekerja dengan menghasilkan sejumlah besar permintaan HTTP secara terus-menerus ke server target sehingga sumber daya server dapat mengalami peningkatan beban yang signifikan (Goel et al., 2023).

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 Politeknik Negeri Jakarta

BAB III

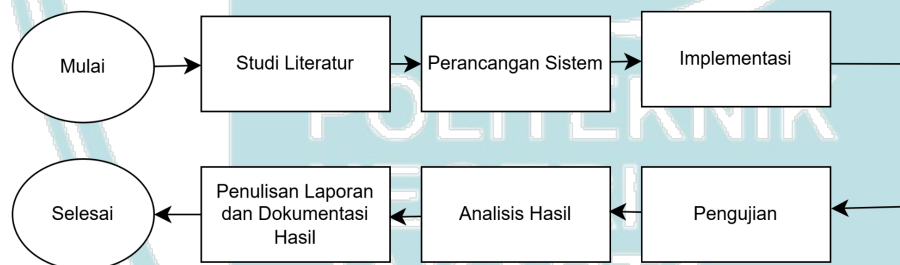
METODE PENELITIAN

3.1 Rancangan Penelitian

Rancangan pengerjaan penelitian ini menggunakan metode kuantitatif dan rekayasa sistem. Fokus utama penelitian ini adalah perancangan dan implementasi sistem deteksi serangan DDoS berbasis *entropy* pada *web server* Ubuntu. Pendekatan kuantitatif dipilih karena evaluasi efektivitas sistem deteksi dilakukan berdasarkan pengukuran data numerik dan analisis statistik. Pendekatan rekayasa sistem diterapkan karena penelitian ini melibatkan proses perancangan, pengintegrasian, dan implementasi berbagai komponen deteksi, meliputi perhitungan *entropy*, parameter K-dinamis, serta mekanisme revaluasi IP, ke dalam sebuah sistem deteksi yang berjalan secara *near real-time*.

3.2 Tahapan Penelitian

Tahapan penelitian digunakan guna memastikan proses riset berjalan secara sistematis dan terstruktur. Berikut adalah tahapan penelitian yang dilakukan:



Gambar 3.1 Tahapan Penelitian

1. Studi Literatur

Analisis kebutuhan berupa pengumpulan jurnal dan buku terkait penelitian terdahulu sebagai referensi untuk penelitian yang akan dilakukan.

2. Perancangan Sistem

Perancangan alur logika sistem deteksi serangan DDoS dilakukan berdasarkan hasil studi literatur, dengan mengacu pada algoritma A3 yang dikemukakan oleh Pebriyanto dan Suryani (2025). Tahap ini mencakup perancangan mekanisme pencatatan trafik HTTP, perhitungan nilai *entropy*



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 Politeknik Negeri Jakarta

- secara real-time, penentuan *threshold* dinamis menggunakan parameter K-dinamis, serta mekanisme revaluasi alamat IP mencurigakan.
3. Implementasi
Sistem deteksi serangan diimplementasikan pada *web server* Ubuntu sesuai dengan rancangan yang telah dibuat.
4. Pengujian
Pengujian dilakukan dengan menjalankan serangan HTTP Flood secara terkontrol bersamaan dengan trafik sah untuk mengamati kemampuan sistem dalam mendeteksi dan memitigasi serangan.
5. Analisis Hasil
Tahap ini menganalisis hasil pengujian berdasarkan waktu deteksi, *confusion matrix*, serta penggunaan sumber daya CPU dan RAM untuk menilai kinerja dan efektivitas sistem deteksi.
6. Penulisan Laporan dan Dokumentasi Hasil
Seluruh hasil penelitian didokumentasikan dalam bentuk laporan, meliputi pembahasan hasil pengujian, penarikan kesimpulan, serta pemberian saran untuk pengembangan penelitian selanjutnya.

3.3 Objek Penelitian

Objek penelitian ini adalah sebuah *web server* berbasis Ubuntu yang menerima permintaan HTTP *request* berupa data teks sebagai simulasi layanan web dan dirancang menjadi tempat pengujian utama bagi sistem deteksi serangan DDoS yang dikembangkan. Pada *web server* ini terdapat mekanisme deteksi anomali trafik berbasis *entropy*, yang bekerja menghitung tingkat keberagaman distribusi alamat IP sumber dalam interval waktu tertentu guna menangkap pola ketidaknormalan pada arus lalu lintas jaringan. Nilai *entropy* yang diperoleh dari proses perhitungan tersebut kemudian dibandingkan dengan nilai *threshold* dinamis yang ditentukan oleh parameter K-dinamis, sehingga sistem dapat mengidentifikasi indikasi awal serangan DDoS secara adaptif sesuai kondisi trafik yang sedang berlangsung. Alamat IP yang terdeteksi mencurigakan berdasarkan perbandingan tersebut kemudian diproses lebih lanjut melalui mekanisme revaluasi IP, yang bertujuan untuk membedakan secara lebih akurat antara pengguna sah yang mengalami lonjakan permintaan wajar dengan alamat IP yang



© Hak Cipta milik 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 Politeknik Negeri Jakarta

memang bersifat terpusat dan terindikasi sebagai penyerang. Seluruh data yang dihasilkan selama proses deteksi, mencakup alamat IP, waktu permintaan, nilai *entropy*, K-dinamis, *threshold* dan hasil klasifikasi akan tersimpan ke dalam database.





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 Politeknik Negeri Jakarta

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Kebutuhan

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kemampuan dasar yang harus dapat dieksekusi oleh sistem. Berdasarkan rumusan masalah penelitian, kebutuhan fungsional perangkat dirumuskan sebagai berikut:

1. Analisis Pola Trafik

Sistem harus mampu menganalisis pola distribusi trafik menggunakan metode berbasis *entropy*, termasuk proses normalisasi dan penyesuaian parameter dinamis, guna mengidentifikasi perubahan signifikan pada pola lalu lintas.

2. Penentuan Ambang Batas (Threshold)

Sistem harus mampu menghitung nilai ambang batas secara dinamis berdasarkan histori data sebelumnya, yang digunakan sebagai dasar dalam proses klasifikasi kondisi trafik.

3. Klasifikasi Kondisi Trafik

Sistem harus mampu melakukan klasifikasi kondisi trafik ke dalam kategori normal atau mencurigakan berdasarkan hasil perbandingan antara nilai analisis dan *threshold* yang telah ditentukan.

4. Identifikasi dan Evaluasi IP Mencurigakan

Sistem harus mampu mengidentifikasi alamat IP yang memiliki perilaku mencurigakan serta melakukan evaluasi lanjutan untuk menentukan apakah kondisi tersebut merupakan serangan DDoS atau bukan.

5. Penyimpanan Hasil Analisis

Sistem harus mampu menyimpan seluruh hasil proses analisis, termasuk nilai perhitungan, parameter yang digunakan, serta hasil klasifikasi, ke dalam basis data untuk keperluan evaluasi.

6. Eksekusi Deteksi Secara Berkala

Sistem harus mampu menjalankan proses deteksi secara otomatis dan berulang dalam interval waktu tertentu guna mendukung deteksi secara *near real-time*.



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 Politeknik Negeri Jakarta

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional menentukan kriteria operasional, performa, dan kualitas dari sistem yang dibangun. Berdasarkan rumusan masalah penelitian, kebutuhan non-fungsional perangkat dirumuskan sebagai berikut:

1. Performa Sistem

Sistem deteksi harus mampu berjalan pada Ubuntu Server tanpa mengganggu ketersediaan layanan *web server* utama. Proses analisis trafik yang mencakup perhitungan *entropy*, K-dinamis, *threshold*, serta revaluasi IP harus dapat dijalankan secara bersamaan dengan layanan Apache2 dan MySQL dalam kondisi pembebanan trafik yang bervariasi, mulai dari trafik normal hingga lonjakan trafik intensif (*flash crowd* dan serangan).

2. Kecepatan Respons Deteksi

Sistem harus mampu menghasilkan keputusan klasifikasi trafik dalam setiap interval *window* pengamatan secara *near real-time*. Proses deteksi yang mencakup pengambilan data dari tabel *raw_traffic* menggunakan mekanisme *sliding window*, perhitungan *normalized entropy*, penentuan *threshold* K-dinamis, hingga penyelesaian tahap revaluasi IP apabila diperlukan, harus selesai dieksekusi dalam satu siklus cronjob tanpa menyebabkan antrian proses yang menumpuk pada sistem operasi.

4.1.3 Spesifikasi Perangkat

Dalam proses pengerjaannya, diperlukan beberapa perangkat lunak dan tools pendukung seperti yang ditulis pada tabel berikut:

Tabel 4.1 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

No	Nama Perangkat	Spesifikasi
Hardware		
1.	Laptop	CPU: Intel Core i3 @1.80Ghz RAM: 8GB Storage: 943GB
Software		
1.	VirtualBox	Version 7.2.6



Hak Cipta :

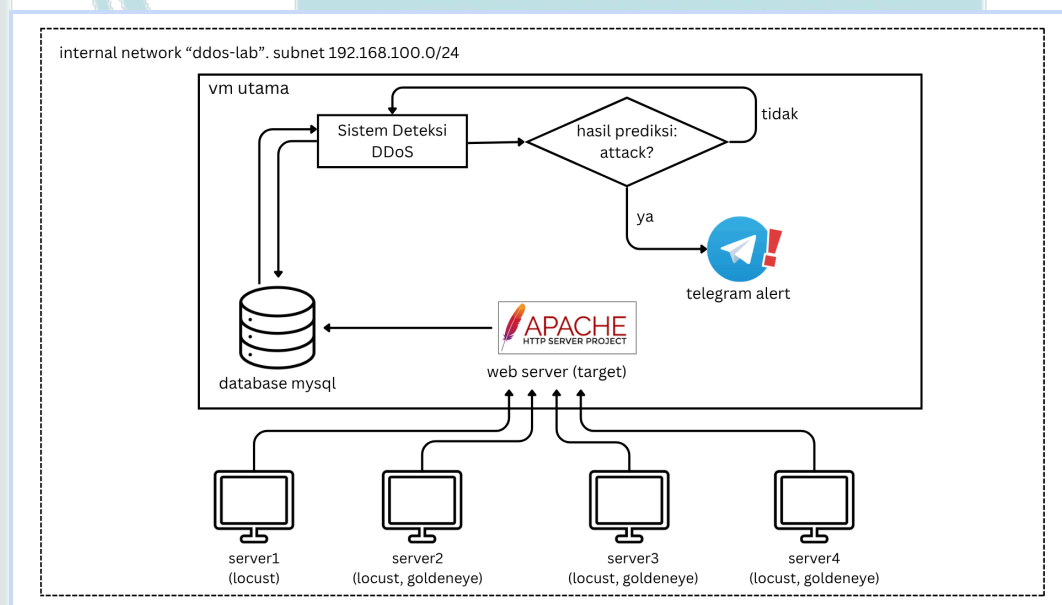
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

No	Nama Perangkat	Spesifikasi
2.	Ubuntu Server	Description: Ubuntu 22.04.04 LTS Release: 22.04.04
3.	Apache2	Server version: Apache/2.4.52 (Ubuntu) Server built: 2026-03-05T18:04:29
4.	MySQL	Ver 8.0.45-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
5.	PHP	Ver 8.1.2-1ubuntu2.23
6.	Locust	Locust 1.4.3
7.	GoldenEye	GoldenEye v2.1 by Jan Seidl
8.	OpenSSH	OpenSSH_8.9p1 Ubuntu-3ubuntu0.15, OpenSSL 3.0.2 15 Mar 2022

4.2 Perancangan Sistem

4.2.1 Blok Diagram

Blok diagram digunakan untuk membantu mengidentifikasi hubungan antar komponen dalam penelitian secara visual.



Gambar 4.1 Blok Diagram Sistem

Gambar 4.1 menunjukkan blok diagram sistem deteksi serangan DDoS yang dibangun pada jaringan internal ddos-lab dengan subnet 192.168.100.0/24. Sistem



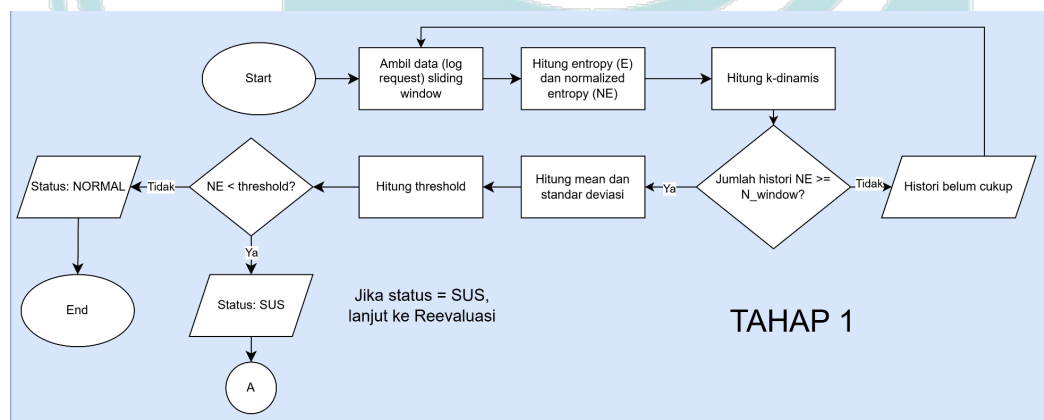
Hak Cipta :

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dijalankan pada sebuah virtual machine utama (*main*) yang berfungsi sebagai pusat layanan dan deteksi. Di dalam VM tersebut terdapat *web server* berbasis Apache sebagai target pengujian, database MySQL untuk menyimpan data log dan hasil analisis, serta sistem deteksi DDoS yang melakukan pemantauan trafik secara berkala menggunakan metode *entropy*, *sliding window*, dan revaluasi IP. Trafik jaringan berasal dari empat VM klien, yang terdiri dari server1, server2, server3, dan server4. Keempat VM klien akan mengirim trafik sesuai dengan skenario yang dibutuhkan. Sistem deteksi akan menganalisis data trafik yang masuk dan menentukan hasil prediksi apakah terjadi serangan atau tidak. Jika hasil analisis menunjukkan status “ATTACK”, sistem akan tetap berjalan sekaligus mengirimkan notifikasi melalui *channel* telegram.

4.2.2 Flowchart Sistem

Flowchart sistem menunjukkan alur kerja sistem deteksi serangan DDoS yang dikembangkan menjadi dua tahapan utama, yaitu tahap deteksi awal (tahap 1) dan tahap revaluasi IP (tahap 2).

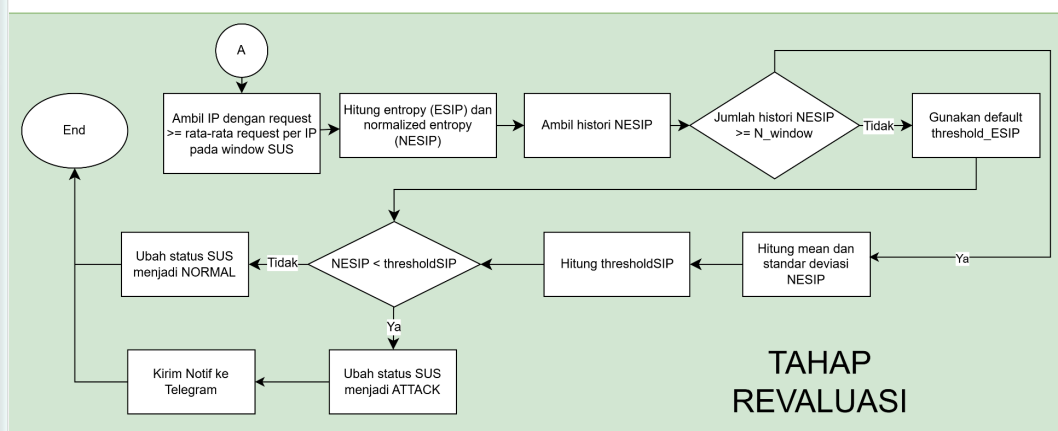


Gambar 4.2 Flowchart Sistem Deteksi Tahap 1

Pada gambar 4.2, proses dimulai dengan pengambilan data *log request* yang dikelompokkan menggunakan metode *sliding window*. Data tersebut digunakan untuk menghitung nilai *entropy* dan kemudian dinormalisasi menjadi *normalized entropy* (NE). Selanjutnya, sistem menghitung nilai K-dinamis dan menentukan *threshold* berdasarkan rata-rata dan standar deviasi nilai NE dari beberapa *window* sebelumnya. Nilai NE kemudian dibandingkan dengan *threshold* untuk melakukan klasifikasi awal. Jika NE lebih kecil dari *threshold*, maka trafik dikategorikan



sebagai *suspicious* (SUS), sedangkan jika nilainya di atas *threshold*, maka trafik dianggap sebagai normal.



Gambar 4.3 Flowchart Sistem Deteksi Tahap 1

Gambar 4.3 menjelaskan alur sistem ketika trafik terdeteksi sebagai mencurigakan di tahap klasifikasi pertama. Proses dimulai dengan memilih IP yang memiliki jumlah *request* di atas rata-rata dari keseluruhan IP di dalam *window* yang dianggap mencurigakan tersebut. Berdasarkan data yang didapat, dihitung nilai *entropy* baru yang dinormalisasi menjadi *normalized ESIP* (NESIP). Nilai NESIP kemudian dibandingkan dengan *threshold* ESIP yang diperoleh dari histori sebelumnya. Jika NESIP lebih kecil dari *threshold* ESIP, maka trafik diklasifikasikan sebagai serangan. Setelahnya, notifikasi adanya trafik serangan dikirimkan lewat bot telegram.

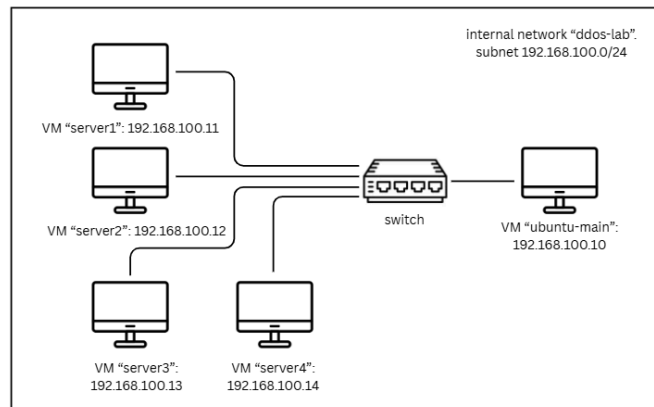
4.2.3 Topologi Jaringan

Topologi jaringan digunakan untuk merancang dan memetakan bagaimana perangkat komputer saling terhubung di dalam suatu jaringan. Topologi jaringan dapat memberikan panduan visual yang jelas dalam menentukan arah komunikasi perangkat seperti komputer, *router*, dan *switch*.



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 kritis 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 Politeknik Negeri Jakarta



Gambar 4.4 Topologi Jaringan

Gambar 4.4 menunjukkan topologi jaringan kabel (*wired*) lokal yang digunakan dalam lingkungan simulasi penelitian ini. Seluruh perangkat virtual terhubung secara tersentralisasi melalui sebuah perangkat penengah berupa Switch di dalam satu jaringan internal (Internal Network) terisolasi bernama ddos-lab dengan lingkup subnet 192.168.100.0/24.

Infrastruktur terdiri dari satu unit target server (ubuntu-main) serta empat mesin klien yang bertindak sebagai pengirim trafik. Sisi klien mencakup satu pengguna sah dan tiga mesin penyerang. Melalui perantara *switch*, lalu lintas data dari seluruh klien akan dialirkan menuju server target untuk dianalisis menggunakan sistem deteksi DDoS berbasis *entropy*.

4.2.4 Pengalamatan IP Address

Berikut adalah informasi alamat IP dari *virtual machine* di lingkungan simulasi penelitian yang dibangun:

Tabel 4.2 Pengalamatan IP Address

Nama Hostname	Interface	IP address
ubuntu-main	enp0s3	192.168.100.10
server1	enp0s3	192.168.100.11
server2	enp0s3	192.168.100.12
server3	enp0s3	192.168.100.13
server4	enp0s3	192.168.100.14



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada tahap perancangan jaringan, setiap virtual machine diberikan alamat IP statis pada interface enp0s3. Seluruh interface terhubung Switch sebagai perangkat penengah yang membentuk internal network ddos-lab. Adapun pengalamatan IP pada jaringan internal tersebut tertera pada tabel 4.2. Selain itu, virtual machine juga dikonfigurasi dengan adapter NAT untuk akses internet serta adapter host-only yang digunakan untuk kebutuhan SSH dari *host machine*.

4.2.5 Spesifikasi Virtual Machine

Berikut adalah informasi spesifikasi *virtual machine* di lingkungan simulasi penelitian yang dibangun:

Tabel 4.3 Spesifikasi Virtual Machine

Nama VM	OS	CPU	RAM	Storage	Role
ubuntu-main	Ubuntu 22.04.04	2 core, Intel Core i3-3217U CPU @ 1.80GHz	4096 MB	25 GB	Web Server, Sistem Deteksi
server1, server2, server3, server4	Ubuntu 22.04.04	2 core, Intel Core i3-3217U CPU @ 1.80GHz	768 MB	25 GB	Simulasi Pengujian

Spesifikasi perangkat keras dan sistem operasi yang dialokasikan untuk masing-masing virtual machine disajikan pada tabel 4.3. Berdasarkan tabel tersebut, mesin ubuntu-main dikonfigurasi dengan kapasitas sumber daya yang lebih besar karena berfungsi sebagai tempat *web server* dan sistem deteksi berjalan. Sementara itu, untuk mesin klien (server1 hingga server4), alokasi sumber daya diatur secara seragam dengan kapasitas yang disesuaikan untuk memenuhi kebutuhan simulasi pengujian. Seluruh konfigurasi ini diimplementasikan menggunakan platform virtualisasi VirtualBox yang berjalan di atas *host machine*.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.3. Implementasi Sistem

Pada tahap ini, rancangan sistem yang telah disusun sebelumnya direalisasikan ke dalam lingkungan simulasi virtual machine. Secara garis besar, tahapan implementasi pada penelitian ini dibagi menjadi tiga bagian utama, yaitu:

1. Implementasi Infrastruktur Dasar dan Lingkungan Web Server: Meliputi instalasi Ubuntu Server, konfigurasi Apache2, penyusunan basis data, dan sistem pencatatan aktivitas web.
2. Implementasi Sistem Deteksi DDoS Berbasis Entropy: Meliputi seluruh proses pemrograman utama sistem deteksi seperti pembacaan paket, perhitungan nilai *entropy* dan *threshold*, klasifikasi hasil deteksi, tahap revaluasi, hingga pengiriman notifikasi ke Telegram.
3. Persiapan Instrumen Pengujian: Meliputi instalasi tools serangan seperti Locust dan Goldeneye.

4.3.1 Instalasi Virtual Machine dan Konfigurasi Ubuntu Server

Pada implementasi sistem, platform virtualisasi VirtualBox digunakan sebagai media untuk membangun lingkungan simulasi jaringan. VirtualBox dipilih karena bersifat open-source, ringan, serta mampu menjalankan beberapa mesin virtual secara bersamaan dalam satu perangkat fisik. Melalui virtualisasi ini, proses simulasi sistem dapat dilakukan secara terisolasi serta memberikan fleksibilitas dalam pengaturan skenario serangan dan pengujian sistem deteksi DDoS.

Sistem operasi yang digunakan pada seluruh mesin virtual adalah Ubuntu Server 22.04.4 LTS. Versi tersebut dipilih karena memiliki dukungan jangka panjang (Long Term Support/LTS), stabil untuk kebutuhan server, serta kompatibel dengan berbagai layanan seperti Apache2, MySQL, PHP, dan tools pengujian jaringan.

Topologi implementasi terdiri dari lima virtual machine (VM) yang saling terhubung melalui jaringan Internal Network bernama “ddos-lab”. Jaringan ini berfungsi sebagai Local Area Network (LAN) virtual sehingga komunikasi antar mesin dapat dilakukan tanpa akses langsung ke jaringan luar. Konfigurasi alamat IP dilakukan secara statis menggunakan Netplan pada direktori `/etc/netplan/*.yaml` dengan subnet jaringan 192.168.100.0/24. Penggunaan IP



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

statis bertujuan agar setiap mesin memiliki alamat yang tetap sehingga memudahkan proses konfigurasi layanan, pengujian trafik, serta monitoring sistem deteksi.

Dalam implementasinya, digunakan total lima virtual machine dengan rincian sebagai berikut:

- 1 VM digunakan sebagai server target yang menjalankan *web server*, database, aplikasi login berbasis PHP, serta sistem deteksi DDoS berbasis *entropy*.
- 4 VM digunakan sebagai klien untuk menciptakan trafik yang menargetkan *web server* target.

Setelah sistem operasi utama berhasil dikonfigurasi, selanjutnya dilakukan instalasi OpenSSH Server pada mesin ubuntu-main dan seluruh mesin klien.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.15).
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
```

Gambar 4.5 Instalasi OpenSSH

Layanan OpenSSH diinstal menggunakan perintah `sudo apt install openssh-server`. Fungsinya agar administrator dapat mengakses terminal Ubuntu Server dari perangkat lain tanpa harus berinteraksi langsung dengan mesin virtual target.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2026-05-08 10:52:05 WIB; 45min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 727 (sshd)
```

Gambar 4.6 Pengecekan Status OpenSSH

Setelah proses instalasi selesai, dilakukan pengecekan status layanan menggunakan perintah `systemctl status ssh`. Perintah tersebut digunakan untuk memastikan bahwa service SSH telah aktif dan berjalan dengan normal 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

sistem. Jika layanan berada pada status *active (running)*, maka server siap menerima koneksi remote dari host.

4.3.2 Instalasi dan Konfigurasi Web Server Apache2

Sebelum instalasi *web server* Apache2 dilakukan, dilakukan penyinkronan sistem dengan repositori server agar sistem operasi mengenali daftar, versi, dan dependensi paket perangkat lunak terbaru.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo apt update
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,484 kB]
Get:6 http://id.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [524 kB]
```

Gambar 4.7 Perintah Memperbarui Daftar Paket

Perintah `sudo` merupakan singkatan dari superuser `do` yang berfungsi memberikan hak akses administrator untuk keperluan proses instalasi paket, konfigurasi sistem, dan pengaturan firewall pada sistem operasi Linux. Perintah `apt` digunakan untuk mengelola perangkat lunak dari repositori resmi.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.20).
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
```

Gambar 4.8 Instalasi Web Server Apache2

Perintah `sudo apt install apache2` digunakan untuk memasang layanan *web server* Apache2 ke dalam sistem. Apache2 berfungsi sebagai *web server* utama yang menangani permintaan HTTP dari client dan menampilkan halaman web aplikasi.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo apt install php libapache2-mod-php php-mysql -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php8.1-mysql' instead of 'php-mysqli'
libapache2-mod-php is already the newest version (2:8.1+92ubuntu1).
php is already the newest version (2:8.1+92ubuntu1).
php8.1-mysql is already the newest version (8.1.2-1ubuntu2.23).
php8.1-mysql set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
```

Gambar 4.9 Instalasi PHP, libapache2, dan mysql

Selanjutnya, dilakukan instalasi PHP beserta library pendukung menggunakan perintah `sudo apt install php libapache2-mod-php php-mysql`. PHP digunakan



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

sebagai bahasa pemrograman sisi server untuk menjalankan aplikasi login dan sistem deteksi DDoS yang dikembangkan. Library libapache2-mod-php berfungsi menghubungkan Apache2 dengan interpreter PHP sehingga file berekstensi .php dapat diproses oleh *web server*. Sementara itu, ekstensi php-mysqli digunakan agar aplikasi PHP dapat terhubung dengan database MySQL menggunakan antarmuka MySQLi.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo ufw app list
[sudo] password for main:
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
```

Gambar 4.10 Konfigurasi Firewall UFW untuk Apache2

Pada tahap konfigurasi keamanan jaringan, digunakan perintah `sudo ufw app list` untuk menampilkan daftar profil aplikasi yang dapat diatur melalui firewall UFW. Setelah profil Apache2 tersedia, dilakukan konfigurasi izin akses menggunakan perintah `sudo ufw allow 'Apache'` atau `sudo ufw allow 'Apache Full'` agar layanan *web server* dapat menerima koneksi HTTP maupun HTTPS dari client pada jaringan internal.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install enable apache2
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo systemctl start apache2
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Fri 2026-05-08 11:23:32 WIB; 3min 34s ago
```

Gambar 4.11 Aktivasi Web Server Apache2

Perintah `sudo systemctl enable apache2` digunakan untuk mengaktifkan layanan Apache2. Selanjutnya, perintah `systemctl` untuk mengelola service atau layanan di sistem Linux berbasis systemd.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.3.3 Instalasi MySQL dan Konfigurasi Database

Sebelum melakukan instalasi database MySQL, dilakukan pembaruan repositori paket menggunakan perintah `sudo apt update`.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo apt install mysql-server -y
```

Gambar 4.12 Instalasi MySQL Server

Instalasi MySQL dilakukan dengan perintah `sudo apt install mysql-server-y` untuk memasang layanan MySQL Server pada Ubuntu Server. Opsi `-y` digunakan agar proses instalasi berjalan otomatis tanpa perlu konfirmasi manual pengguna.

```
main@ubuntu-main:~/ddos_skripsi/ddos_detection$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor prese
   Active: active (running) since Fri 2026-05-08 10:52:23 WIB; 50min ago
   Main PID: 757 (mysqld)
   Status: "Server is operational"
   Tasks: 37 (limit: 4554)
   Memory: 460.2M
   CPU: 1min 13.021s
   CGroup: /system.slice/mysql.service
           └─757 /usr/sbin/mysqld
```

Gambar 4.13 Pengecekan Status MySQL

Setelah instalasi selesai, dilakukan pengecekan status layanan menggunakan perintah `systemctl status mysql`. Informasi yang ditampilkan meliputi status layanan *active (running)*, Main PID sebagai identitas proses utama MySQL, jumlah task yang berjalan, penggunaan memori (memory), serta penggunaan CPU oleh service database.

```
main@ubuntu-main:~/ddos_skripsi$ sudo mysql -u root -p
[sudo] password for main:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.45-0ubuntu0.22.04.1 (Ubuntu)
```

Gambar 4.14 Pengaksesan MySQL dari Terminal

Untuk mengakses terminal administrasi database, digunakan perintah `mysql -u root -p`. Parameter `-u root` menunjukkan bahwa login dilakukan menggunakan akun administrator database (root), sedangkan opsi `-p` digunakan untuk meminta



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 Politeknik Negeri Jakarta

input password sebelum akses diberikan. Setelah autentikasi berhasil, pengguna akan masuk ke terminal interaktif MySQL yang digunakan untuk menjalankan perintah SQL dan melakukan konfigurasi database.

```
mysql> use ddos_detection;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_ddos_detection |
+-----+
| classification            |
| dynamic_k                 |
| entropy_log               |
| raw_traffic               |
| reevaluation_log          |
| suspicious_ip             |
| threshold                 |
+-----+
7 rows in set (0.18 sec)
```

Gambar 4.15 Daftar Tabel Database

Selanjutnya, perintah `CREATE DATABASE ddos_detection;` digunakan untuk membuat sebuah database baru dengan nama `ddos_detection`. Perintah `USE ddos_detection;` digunakan untuk memilih database tersebut yang akan digunakan dalam sesi MySQL saat ini. Perintah `SHOW TABLES;` untuk menampilkan seluruh tabel yang terdapat di dalamnya.

Secara keseluruhan, terdapat dua struktur tabel pada database `ddos_detection` yang dibedakan berdasarkan tujuan penyusunan datanya, yaitu tabel untuk lingkungan *web server* dan tabel untuk sistem deteksi DDoS. Pemisahan ini dilakukan agar pencatatan dan pengambilan data yang dilakukan *web server* maupun sistem deteksi DDoS dapat berjalan secara efisien.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Tabel Lingkungan Web Server

Tabel `raw_traffic` digunakan sebagai penyimpanan utama data seluruh trafik HTTP yang masuk ke *web server*.

```
mysql> desc raw_traffic;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
timestamp	datetime	NO	MUL	NULL	
ip_address	varchar(45)	NO	MUL	NULL	
endpoint	varchar(255)	YES		NULL	
method	varchar(10)	YES		NULL	
user_agent	text	YES		NULL	
payload_size	int	YES		0	

Gambar 4.16 Struktur Tabel `raw_traffic`

Data trafik HTTP yang disimpan meliputi waktu *request* masuk, alamat IP klien, endpoint yang diakses, metode HTTP, user agent, serta ukuran *payload request*. Data pada tabel ini menjadi sumber utama dalam proses analisis trafik menggunakan metode *entropy* berbasis *sliding window*.

2. Tabel Sistem Deteksi DDoS

Tabel `entropy_log`, `dynamic_k`, `threshold`, `suspicious_ip`, `reevaluation_log`, dan `classification` digunakan untuk penyimpanan dan pemanggilan data dari proses-proses yang ada di dalam skrip sistem deteksi DDoS.

```
mysql> desc entropy_log;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
unique_ip	int	YES		NULL	
entropy	double	NO		NULL	
total_request	int	YES		NULL	
timestamp	datetime	NO	MUL	NULL	
normalized_entropy	double	YES		NULL	

6 rows in set (0.04 sec)

Gambar 4.17 Struktur Tabel `entropy_log`

Tabel `entropy_log` digunakan untuk menyimpan hasil perhitungan *entropy* pada setiap *window* pengamatan. Informasi yang disimpan meliputi jumlah IP unik



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

(*unique_ip*), nilai *entropy*, total *request* dalam *window*, *timestamp*, serta nilai *normalized entropy*. Nilai *normalized entropy* digunakan sebagai indikator distribusi trafik untuk menentukan apakah kondisi trafik bersifat normal atau mencurigakan.

```
mysql> desc dynamic_k;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
timestamp	datetime	NO	MUL	NULL	
emax	double	YES		NULL	
ptotal	int	YES		NULL	
k_dynamic	double	YES		NULL	

5 rows in set (0.01 sec)

Gambar 4.18 Struktur Tabel *dynamic_k*

Tabel *dynamic_k* digunakan untuk menyimpan hasil perhitungan parameter *dynamic K* yang berfungsi sebagai faktor penyesuaian *threshold* secara adaptif. Tabel ini mencatat nilai *entropy* maksimum (*emax*), total *request* (*ptotal*), serta nilai *k_dynamic* yang dihitung berdasarkan kondisi trafik pada *window* tertentu.

```
mysql> desc threshold;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
timestamp	datetime	YES	MUL	NULL	
mean	double	YES		NULL	
stddev	double	YES		NULL	
k_dynamic	double	YES		NULL	
threshold	double	YES		NULL	

6 rows in set (0.00 sec)

Gambar 4.19 Struktur Tabel *threshold*

Selanjutnya, tabel *threshold* digunakan untuk menyimpan hasil perhitungan *threshold* deteksi. Data yang disimpan meliputi nilai rata-rata *normalized entropy* (*mean*), standar deviasi (*stddev*), nilai *dynamic K*, serta *threshold* akhir yang digunakan pada proses klasifikasi trafik. *Threshold* ini dihitung berdasarkan



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

histori beberapa *window* sebelumnya sehingga sistem mampu beradaptasi terhadap perubahan pola trafik jaringan.

```
mysql> desc suspicious_ip;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
timestamp	datetime	YES		NULL	
ip	varchar(45)	YES		NULL	
request_count	int	YES		NULL	
probability	double	YES		NULL	

5 rows in set (0.00 sec)

Gambar 4.20 Struktur Tabel *suspicious_ip*

Tabel *suspicious_ip* digunakan untuk menyimpan daftar alamat IP mencurigakan dari hasil perbandingan pertama yang hasilnya berstatus SUS. Tabel ini mencatat alamat IP, jumlah *request* yang dikirimkan, serta probabilitas distribusi *request* dari masing-masing IP. Data tersebut digunakan dalam proses perhitungan *entropy* khusus terhadap kelompok IP mencurigakan (ESIP).

```
mysql> desc reevaluation_log;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
window_id	int	NO		NULL	
timestamp	datetime	YES	MUL	NULL	
entropy_esip	double	YES		NULL	
mean_nesip	double	YES		NULL	
stddev_nesip	double	YES		NULL	
threshold_nesip	double	YES		NULL	
final_result	varchar(10)	YES		NULL	
normalized_esip	double	YES		NULL	

9 rows in set (0.00 sec)

Gambar 4.21 Struktur Tabel *reevaluation_log*

Tabel *reevaluation_log* digunakan untuk menyimpan hasil tahap revaluasi terhadap trafik yang sebelumnya diklasifikasikan sebagai *suspicious*. Tabel ini mencatat nilai *entropy* ESIP, rata-rata ESIP, standar deviasi ESIP, *threshold* ESIP, *normalized* ESIP, serta hasil akhir klasifikasi.



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 Politeknik Negeri Jakarta

```
mysql> desc classification;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
timestamp	datetime	YES	MUL	NULL	
normalized_entropy	double	YES		NULL	
threshold	double	YES		NULL	
result	varchar(10)	YES		NULL	
final_result	varchar(10)	YES		NULL	
delta	double	YES		NULL	

Gambar 4.22 Struktur Tabel classification

Tabel classification digunakan untuk menyimpan hasil klasifikasi tahap pertama dan tahap kedua berdasarkan perbandingan normalized entropy dan *threshold*. Tabel ini menyimpan nilai normalized entropy, *threshold*, selisih nilai (delta), hasil klasifikasi awal (result), serta hasil akhir deteksi (final_result). Hasil klasifikasi awal umumnya berupa status NORMAL atau SUS sebelum dilakukan proses revaluasi IP.

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$db = "ddos_detection";

$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Gambar 4.23 Koneksi PHP ke MySQL

Gambar 4.23 menunjukkan file config.php yang digunakan untuk mengatur koneksi antara aplikasi PHP dengan database MySQL pada sistem deteksi DDoS. Pada file ini didefinisikan parameter koneksi berupa host, username, password, dan nama database ddos_detection. Konfigurasi dibuat terpusat agar dapat digunakan kembali oleh seluruh modul sistem tanpa perlu menuliskan koneksi secara berulang.

Koneksi database dilakukan menggunakan library MySQLi melalui objek new mysqli(), kemudian disimpan pada variabel \$conn. Variabel tersebut digunakan



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

untuk menjalankan proses query seperti penyimpanan log trafik, perhitungan *entropy*, dan hasil klasifikasi. Selain itu, terdapat pengecekan `connect_error` untuk memastikan koneksi berhasil dilakukan. Jika koneksi gagal, sistem akan menampilkan pesan error dan menghentikan proses program.

4.3.4 Pembuatan Web Server

Pembuatan *web server* dimulai dengan pembuatan *codebase* halaman login utama (`login.php`) seperti pada gambar berikut:

```
<?php
require_once "/home/main/ddos_skripsi/ddos_detection/logger.php";

$username = $_POST['username'] ?? '';
$password = $_POST['password'] ?? '';

$valid_user = "admin";
$valid_pass = "12345";

if($username == $valid_user && $password == $valid_pass){
    header("Location: dashboard.php");
    exit;
}else{
    echo "
    <script>
        alert('Login gagal! Username atau password salah.');
```

Gambar 4.24 Penambahan Include `logger.php` di dalam `login.php`

Pada gambar 4.24, dilakukan penambahan `require_once "logger.php"` pada file `login.php`. Penambahan ini bertujuan agar setiap *request* yang masuk ke endpoint login secara otomatis dicatat oleh sistem sebelum proses autentikasi dijalankan. Dengan mekanisme tersebut, seluruh aktivitas akses terhadap halaman login, baik berhasil maupun gagal, dapat direkam sebagai data trafik untuk kebutuhan analisis deteksi DDoS.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
<?php
require_once "config.php";

// data yang dicatat: timestamp, ip_address, endpoint, method,
user_agent, payload_size
$timestamp = date("Y-m-d H:i:s");

if (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) {
    $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
} else {
    $ip = $_SERVER['REMOTE_ADDR'];
}

$endpoint = $_SERVER['REQUEST_URI'];
$method = $_SERVER['REQUEST_METHOD'];
$user_agent = $_SERVER['HTTP_USER_AGENT'] ?? 'unknown';
$payload_size = $_SERVER['CONTENT_LENGTH'] ?? 0;
```

Gambar 4.25 Tipe Data yang Dicatat dalam Sebuah Request

Gambar 4.25 adalah logger.php yang berfungsi sebagai modul pencatatan trafik yang mengambil beberapa informasi penting dari setiap *request* HTTP, meliputi waktu *request* (*timestamp*), alamat IP client (*ip_address*), endpoint yang diakses (*endpoint*), metode *request* (*method*), informasi perangkat pengguna (*user_agent*), serta ukuran *payload request* (*payload_size*). Informasi tersebut digunakan sebagai sumber data utama dalam proses analisis pola trafik jaringan berbasis *entropy*.

```
$stmt = $conn->prepare("
    INSERT INTO raw_traffic
    (timestamp, ip_address, endpoint, method, user_agent, payload_size)
    VALUES (?, ?, ?, ?, ?, ?)
");

$stmt->bind_param("sssssi",
    $timestamp,
    $ip,
    $endpoint,
    $method,
    $user_agent,
    $payload_size
);

?>
```

Gambar 4.26 Skrip Pencatatan Data ke dalam Tabel raw_traffic

Gambar 4.26 adalah skrip lanjutan logger.php untuk proses pencatatan yang dilakukan dengan mengambil data *request* melalui variabel global `$_SERVER`, kemudian menyimpannya ke dalam tabel `raw_traffic` menggunakan prepared



Hak Cipta :

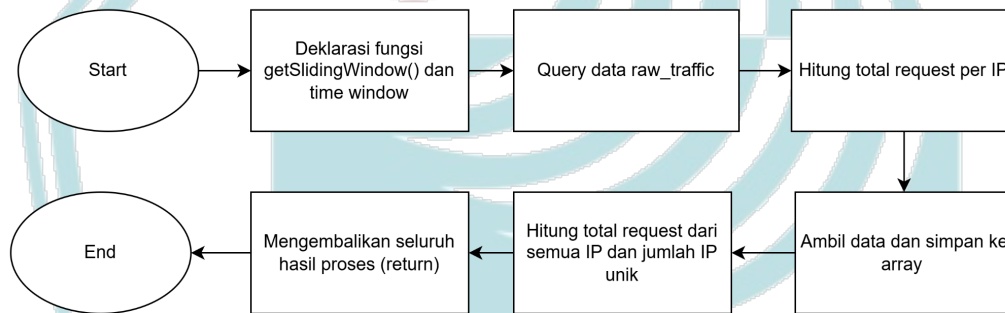
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

statement MySQLi. Penggunaan prepared statement bertujuan meningkatkan keamanan query database dan mengurangi risiko manipulasi input. Setiap *request* yang diterima server akan menghasilkan satu record baru pada tabel *raw_traffic*, sehingga sistem memiliki histori trafik yang dapat digunakan untuk proses monitoring maupun deteksi serangan.

4.3.5 Implementasi Perhitungan Entropy

1. Pengambilan Data Menggunakan Sliding Window

Proses pengambilan data dilakukan menggunakan mekanisme *sliding window* yang terdapat pada file *sliding-window.php*. Logika alur proses diilustrasikan dalam *flowchart* berikut:



Gambar 4.27 Flowchart Pengambilan Data dalam Window

Pada tahap awal, dilakukan deklarasi fungsi *getSlidingWindow()* dan penetapan *time window* yang digunakan. Sistem lalu melakukan query pengambilan data *raw_traffic* berdasarkan total *request* per IP dan disimpan ke dalam *array \$data*. Setelahnya, dilakukan perhitungan seluruh *request* dari semua IP dan penjumlahan alamat IP unik dalam *window* tersebut. Lalu data dikembalikan ke skrip utama sistem deteksi. Proses detail tentang pengambilan data dalam *window* terdapat pada potongan kode 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
<?php
function getSlidingWindow($conn, $window_size = 30)
{
    $query = "
    SELECT ip_address, COUNT(*) as request_count
    FROM raw_traffic
    WHERE timestamp >= NOW() - INTERVAL $window_size SECOND
    GROUP BY ip_address
    ";

    $result = $conn->query($query);

    $data = [];
    $total_requests = 0;

    while ($row = $result->fetch_assoc()) {
        $data[] = $row;
        $total_requests += $row['request_count'];
    }

    $unique_ip = count($data);

    return [
        'window_size' => $window_size,
        'data' => $data,
        'total_requests' => $total_requests,
        'unique_ip' => $unique_ip
    ];
}
?>
```

Gambar 4.28 Skrip Pengambilan dan Penyimpanan Data dalam Window

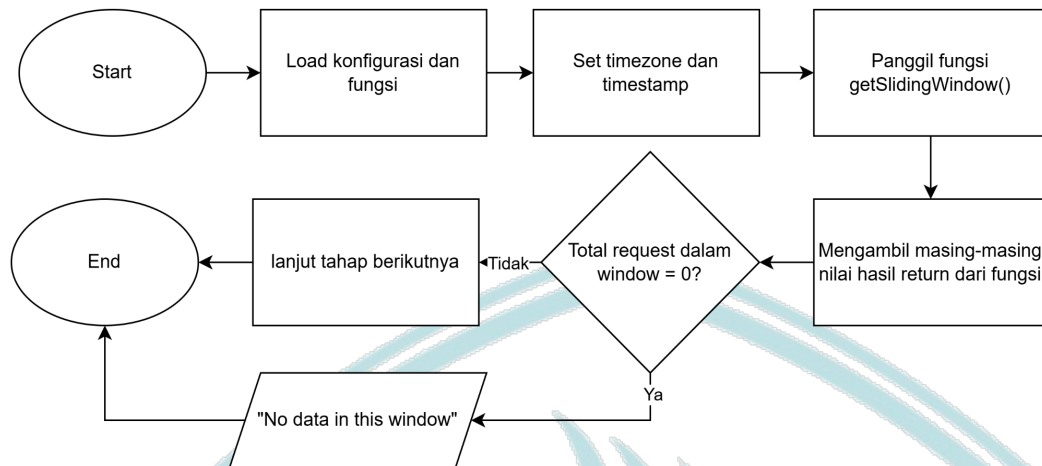
Fungsi `getSlidingWindow()` berada di dalam file `sliding-window.php`. Fungsi ini bertugas mengambil data trafik jaringan dari database berdasarkan interval waktu tertentu. Pada penelitian ini, ukuran *window* ditetapkan selama 30 detik, sehingga sistem hanya memproses *request* yang tercatat dalam rentang waktu 30 detik terakhir. Lalu fungsi menjalankan query SQL terhadap tabel `raw_traffic` untuk mengambil alamat IP beserta jumlah *request* yang dilakukan pada interval tersebut. Data kemudian dikelompokkan berdasarkan alamat IP menggunakan perintah `GROUP BY ip_address` sehingga diperoleh distribusi jumlah *request* dari masing-masing IP.

Selain mengambil data trafik, fungsi ini juga menghitung total keseluruhan *request* dalam satu *window* serta jumlah IP unik yang terdeteksi. Seluruh hasil tersebut kemudian dikembalikan ke file `main-function.php` dalam bentuk *array* asosiatif yang berisi ukuran *window*, data trafik, total *request*, dan jumlah IP unik.



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 Politeknik Negeri Jakarta



Gambar 4.29 Flowchart Pemanggilan Fungsi dan Validasi Window Kosong

Pada tahap awal dalam main-function.php, dilakukan *loading* config.php (untuk membuka koneksi database) dan sliding-window.php. Setelah pengaturan *timezone* dan *timestamp*, pemanggilan fungsi `getSlidingWindow()` dilakukan untuk mengambil data dalam *array return*. Terakhir, sistem melakukan pengecekan untuk memastikan *window* memiliki data trafik. Proses detail tentang pemanggilan fungsi terdapat pada potongan kode berikut:

```

<?php
$start_time = microtime(true);
require_once "../config.php";
require_once "sliding-window.php";
require_once "save-performance-log.php";
date_default_timezone_set('Asia/Jakarta');
$current_timestamp = date("Y-m-d H:i:s");

// 0. AMBIL DATA SLIDING WINDOW
$window = getSlidingWindow($conn);

$window_size = $window['window_size'];
$data = $window['data'];
$total_requests = $window['total_requests'];
$n = $window['unique_ip'];

if ($total_requests == 0) {
    die("No data in this window\n");
}
  
```

Gambar 4.30 Skrip Pemanggilan Fungsi `getSlidingWindow`

Sistem memanggil fungsi `getSlidingWindow()` untuk mengambil data trafik jaringan dalam rentang waktu tertentu menggunakan metode *sliding window*. Fungsi tersebut mengembalikan beberapa informasi berupa ukuran *window* (`window_size`), data trafik per alamat IP (`data`), total *request* dalam satu *window*



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 Politeknik Negeri Jakarta

(total_requests), serta jumlah IP unik (unique_ip) yang akan digunakan pada tahap selanjutnya, seperti perhitungan *entropy*. Sistem juga melakukan validasi terhadap total *request* pada *window*. Jika tidak terdapat trafik yang tercatat (total_requests = 0), maka proses deteksi dihentikan dan sistem menampilkan pesan “No data in this window”.

```
// 1. BUAT WINDOW RECORD

$stmt = $conn->prepare("
    INSERT INTO window_log (timestamp, window_size)
    VALUES (?, ?)
");

$stmt->bind_param("si", $current_timestamp, $window_size);
$stmt->execute();

$window_id = $conn->insert_id;
```

Gambar 4.31 Skrip Pemanggilan Fungsi getSlidingWindow

Selanjutnya, sistem memasuki melakukan pembuatan *window record*. Pada tahap ini dilakukan penyimpanan informasi *timestamp* dan ukuran *window* ke tabel *window_log* menggunakan prepared statement. Setelah proses insert berhasil dilakukan, sistem mengambil nilai *window_id* menggunakan *\$conn->insert_id*. Nilai tersebut digunakan sebagai relasi atau identitas unik *window* untuk proses pencatatan pada tabel-tabel berikutnya, seperti *entropy*, *threshold*, klasifikasi, dan re-evaluation.

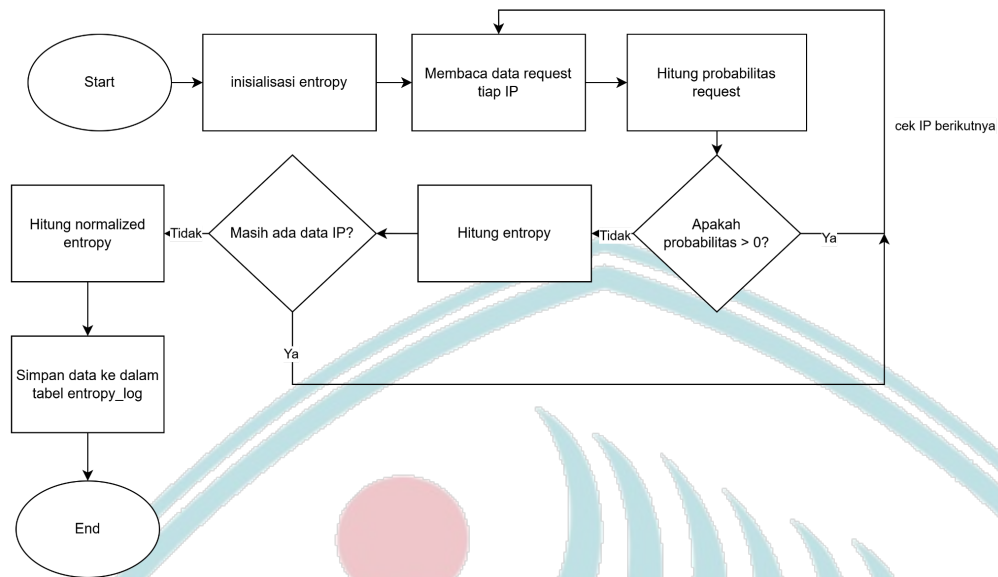
2. Perhitungan Entropy dan Normalized Entropy

Pada tahap ini, sistem melakukan perhitungan *entropy* berdasarkan distribusi jumlah *request* dari setiap alamat IP pada *window* yang sedang dianalisis. Setelah nilai *entropy* didapatkan, dilakukan tahap normalisasi agar nilai tersebut berada pada rentang 0-1.



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 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 Politeknik Negeri Jakarta



Gambar 4.32 Flowchart Perhitungan Entropy dan Normalized Entropy

Proses dimulai dengan inialisasi nilai *entropy*, kemudian sistem membaca data *request* tiap IP dan menghitung probabilitas kemunculan *request* terhadap total trafik. Jika nilai probabilitas lebih besar dari nol, maka sistem menghitung kontribusi *entropy* dari IP tersebut. Proses ini dilakukan berulang hingga seluruh data IP selesai diproses.

Setelah *entropy* diperoleh, sistem menghitung *normalized entropy* untuk menormalkan nilai *entropy* agar nilainya berada pada rentang 0-1. Selanjutnya, hasil perhitungan *entropy* dan *normalized entropy* disimpan ke tabel *entropy_log* sebagai data histori untuk proses deteksi berikutnya. Proses detail tentang perhitungan *entropy* dan *normalized entropy* terdapat pada potongan kode berikut:

```
// 2. HITUNG ENTROPY
$entropy = 0;
foreach ($data as $row) { //1
    $p = $row['request_count'] / $total_requests; //2
    if ($p > 0) { //3
        $entropy -= $p * log($p, 2); //4
    }
}
$normalized_entropy = ($n > 1) ? ($entropy / log($n, 2)) : 0;
```

Gambar 4.33 Skrip Perhitungan Entropy dan Normalized Entropy

Pada tahap ini, sistem menghitung distribusi probabilitas *request* dari setiap alamat IP terhadap total *request* dalam satu *window*. Nilai probabilitas tersebut



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

digunakan dalam rumus Shannon Entropy untuk mengukur tingkat keragaman distribusi trafik jaringan, yang kemudian dinormalisasi agar nilainya berada pada rentang 0 hingga 1.

Proses transformasi persamaan pada subbab 2.4.1 dan 2.4.2 ke dalam skrip PHP pada gambar 4.33 dilakukan sebagai berikut:

1. Inisialisasi Nilai Awal: Variabel \$entropy = 0; dideklarasikan di awal fungsi sebagai kontainer kosong untuk menampung hasil penjumlahan dari rumus sigma (Σ) pada Shannon Entropy.
2. Perhitungan Probabilitas $P(x_i)$: Dikonversi ke dalam baris kode \$p = \$row['request_count'] / \$total_requests. Kode ini membagi jumlah *request* dari satu IP spesifik dengan total seluruh *request* yang ada di dalam *window* aktif.
3. Pencegahan Nilai Tidak Terdefinisi ($p > 0$): Kondisi if ($p > 0$) ditambahkan untuk menghindari error komputasi akibat nilai $\log_s(0)$ yang tidak terdefinisi.
4. Konversi Rumus Shannon Entropy: Persamaan $-p(x_i)\log_2(p(x_i))$ diturunkan langsung ke dalam sintaks PHP menjadi \$entropy -= \$p * log(\$p, 2). Tanda minus (-) diperlukan karena logaritma dari nilai peluang (yang berkisar antara 0 dan 1) selalu menghasilkan bilangan negatif, sehingga tanda tersebut berfungsi membalikkan hasilnya agar nilai akhir *entropy* menjadi positif.
5. Konversi Rumus Normalized Entropy (H_{norm}): Persamaan untuk menormalisasi nilai *entropy* diwujudkan pada baris \$normalized_entropy = (\$n > 1) ? (\$entropy / log(\$n, 2)) : 0. Sintaks ini membagi total variabel \$entropy dengan nilai maksimum *entropy*-nya ($\log_2(n)$). Kondisi $n > 1$ ditambahkan sebagai validasi untuk mencegah pembagian dengan nol apabila hanya terdapat 1 IP unik ($\log_2(1) = 0$) di dalam *window* tersebut.



Hak Cipta :

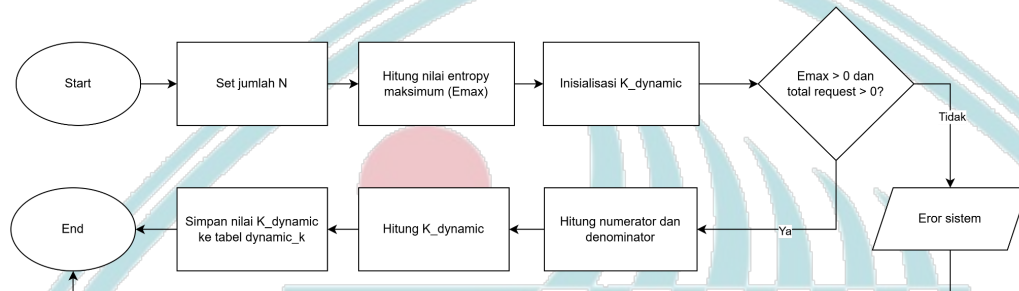
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

Hasil perhitungan berupa jumlah IP unik (\$n) , *entropy*, *normalized entropy*, *total request*, dan *timestamp* kemudian disimpan ke tabel *entropy_log*.

4.3.6 Implementasi Perhitungan K-Dinamis dan Threshold

1. Perhitungan K-dinamis

K-dinamis diperlukan sebagai parameter sensitivitas untuk nilai threshold yang akan dihitung di tahap selanjutnya.



Gambar 4.34 Flowchart Perhitungan K-Dinamis

Proses dimulai dengan menentukan jumlah histori *window* (*N_window*) dan menghitung nilai *entropy* maksimum (*Emax*) berdasarkan jumlah IP unik pada *window* saat ini. Jika nilai *Emax* dan *total request* lebih besar dari nol, maka sistem menghitung nilai *numerator* dan *denominator* untuk memperoleh nilai *K_dynamic*. Terakhir, nilai *K-dinamis* yang didapat disimpan ke dalam tabel *dynamic_k*. Proses detail tentang perhitungan *K-dinamis* terdapat pada potongan kode berikut:

```

// 3. HITUNG DYNAMIC K

$N_window = 20;
$emax = ($n > 0) ? log($n, 2) : 0; //1
$k_dynamic = 0;
if ($emax > 0 && $total_requests > 0) {
    $numerator = pow(log(1 + $emax), 2); //2
    $denominator = ($emax + $total_requests) * log(1 + $total_requests); //3
    $k_dynamic = ($numerator / $denominator); //4
}
  
```

Gambar 4.35 Skrip Perhitungan K-Dinamis

Proses transformasi persamaan *K-dinamis* pada subbab 2.5.2 ke dalam skrip PHP pada gambar 4.35 dilakukan sebagai berikut:



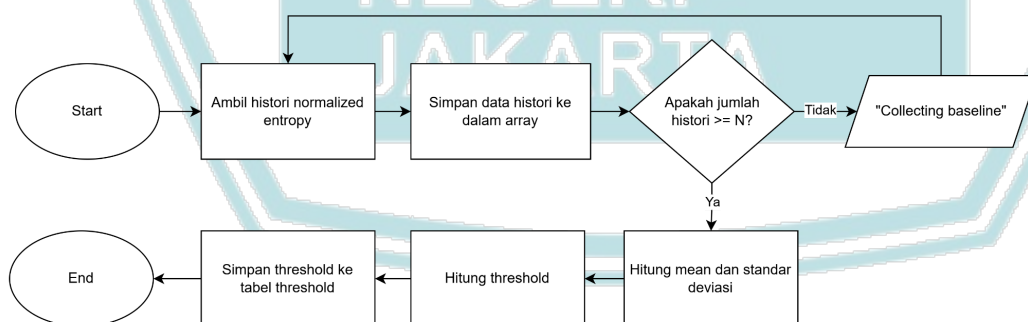
Hak Cipta :

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Perhitungan Nilai Entropy Maksimum (E_{max}): Persamaan untuk mencari batas atas *entropy* ($\log_2(n)$) dikonversi ke dalam baris $\$n = (\$n > 0) ? \log(\$n, 2) : 0$. Kondisi $\$n > 0$ digunakan agar fungsi hanya mengeksekusi jika jumlah IP unik lebih besar dari nol.
2. Konversi Rumus Pembilang (Numerator): Bagian pembilang dari persamaan K-dinamis, yaitu $(\ln(1 + E_{max}))^2$ dikonversi ke dalam baris $\$numerator = \text{pow}(\log(1 + \$n), 2)$. Fungsi $\log()$ di PHP secara bawaan mempresentasikan logaritma natural ($\ln$) dan fungsi $\text{pow}(..., 2)$ digunakan untuk melakukan operasi kuadrat.
3. Konversi Rumus Penyebut (Denominator): Bagian penyebut dari persamaan K-dinamis, yaitu $(E_{max} + P_{total}) \cdot \ln(1 + P_{total})$ dikonversi ke dalam baris $\$denominator = (\$n + \$total_requests) * \log(1 + \$total_requests)$. Variabel $\$total_requests$ digunakan sebagai representasi total seluruh trafik (P_{total}) dalam satu *window*.
4. Perhitungan K-dinamis: Untuk mendapatkan nilai K-dinamis, dilakukan pembagian pembilang dengan penyebutnya ($\$numerator / \$denominator$).

Setelah diperoleh, nilai $K_{dynamic}$ bersama informasi *timestamp*, E_{max} , dan total *request* disimpan ke tabel *dynamic_k* sebagai histori parameter deteksi.

2. Perhitungan Threshold



Gambar 4.36 Flowchart Perhitungan Threshold

Pada tahap perhitungan threshold, sistem mengambil histori nilai *normalized entropy* dari beberapa *window* sebelumnya sebanyak N . Nilai tersebut kemudian disimpan ke dalam *array* untuk proses perhitungan statistik. Sistem melakukan pengecekan apakah jumlah histori telah mencukupi. Jika histori belum memenuhi



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 Politeknik Negeri Jakarta

jumlah minimum *window*, maka sistem menampilkan pesan “Collecting baseline” dan proses dihentikan sementara sampai jumlah histori memenuhi.

Apabila jumlah histori telah mencukupi, sistem menghitung nilai rata-rata (mean) dan standar deviasi dari *normalized entropy*. Nilai *threshold* kemudian diperoleh menggunakan kombinasi mean, standar deviasi, dan nilai K-dinamis. Hasil *threshold* selanjutnya disimpan ke tabel *threshold* untuk digunakan pada tahap klasifikasi trafik. Proses detail tentang perhitungan *threshold* terdapat pada potongan kode berikut:

```
// 4. HITUNG THRESHOLD

$result_nev = $conn->query("SELECT normalized_entropy FROM entropy_log
ORDER BY timestamp DESC LIMIT $N_window");
$nev = [];
while ($row = $result_nev->fetch_assoc()) { $nev[] =
$row['normalized_entropy']; }

if (count($nev) >= $N_window) {
    $mean = array_sum($nev) / count($nev); //1
    $variance = 0;
    foreach ($nev as $val) { $variance += pow($val - $mean, 2); } //2
    $stddev = sqrt($variance / count($nev)); //2

    $threshold = $mean - ($k_dynamic * $stddev); //3
} else {
    echo "Collecting baseline... (" . count($nev) . "/" . $N_window) . "\n";
    savePerformanceLog(
        $conn,
        $window_id,
        $current_timestamp,
        $start_time
    );
    $conn->close();
    exit;
}
```

Gambar 4.37 Skrip Perhitungan Threshold

Tahap berikutnya adalah perhitungan *threshold* deteksi. Sebelum menghitung nilai statistik, sistem melakukan query untuk mengambil data *\$N window* terakhir (dalam skrip ini ditetapkan *\$N_window = 20*) dari tabel *entropy_log* secara berurutan (descending) berdasarkan *timestamp* terbaru untuk dimasukkan ke dalam *array* *\$nev*. Proses transformasi perhitungan *threshold* pada subbab 2.5.3 ke dalam skrip PHP pada gambar 4.37 dilakukan sebagai 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Perhitungan Nilai Mean ($M_{entropy}$): Persamaan untuk mencari nilai mean dari *array normalized entropy* (*nev*) dikonversi ke dalam baris $\$mean = array_sum(\$nev) / count(\$nev)$.
2. Perhitungan Standar Deviasi ($\sigma_{entropy}$): Perulangan *foreach* digunakan untuk mengambil setiap nilai pada *array* *\$nev* yang disimpan sementara ke dalam variabel *\$val*. Pada setiap iterasi, sistem menghitung selisih nilai *\$val* dengan *\$mean*. Selisih lalu dipangkatkan dua ($pow(..., 2)$) agar seluruh hasil bernilai positif. Hasil kuadrat selisih tersebut lalu dijumlahkan ke dalam variabel *\$variance*. Selanjutnya standar deviasi dihitung dalam persamaan $\$stddev = \sqrt{\$variance / count(\$nev)}$. Fungsi $\sqrt{}$ digunakan untuk menghitung akar kuadrat dari *varians*.
3. Perhitungan Threshold: Nilai *threshold* didapatkan dari baris $\$threshold = \$mean - (\$k_dynamic * \$stddev)$. Operasi pengurangan (-) digunakan karena penurunan nilai *normalized entropy* (menjauhi satu) mengindikasikan terjadinya pemusatan trafik.

Jika histori belum memenuhi jumlah minimum *\$N_window*, maka sistem menampilkan pesan “Collecting baseline” hingga data *window* mencukupi. Hasil *threshold* selanjutnya disimpan ke tabel *threshold* untuk digunakan pada proses klasifikasi trafik normal dan trafik mencurigakan pada tahap berikutnya.

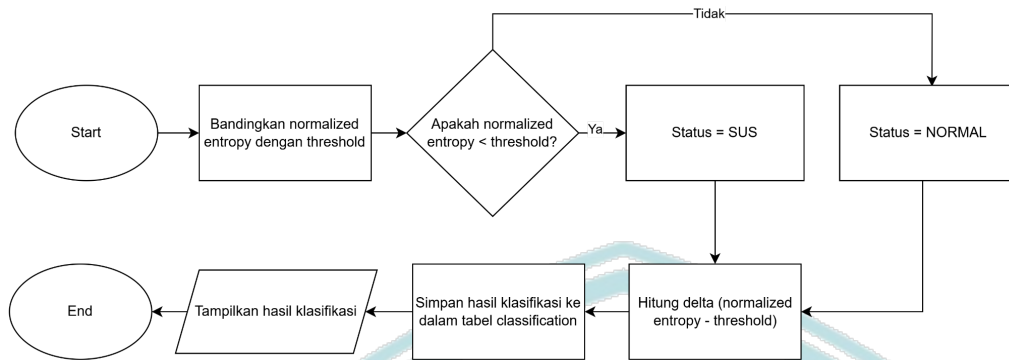
4.3.7 Klasifikasi Hasil Deteksi Tahap Pertama

Pada tahap klasifikasi pertama, sistem membandingkan nilai *normalized entropy* dengan *threshold* yang telah dihitung sebelumnya. Jika nilai *normalized entropy* lebih kecil dari *threshold*, maka trafik dikategorikan sebagai trafik mencurigakan. Sebaliknya, jika nilai *normalized entropy* berada di atas *threshold*, maka trafik dikategorikan sebagai trafik normal.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.38 Flowchart Klasifikasi Tahap Pertama

Selanjutnya, sistem menghitung nilai selisih (delta) antara *normalized entropy* dan *threshold* untuk mengetahui jarak nilai terhadap batas deteksi. Hasil klasifikasi awal kemudian disimpan ke tabel *classification* beserta informasi *normalized entropy*, *threshold*, *delta*, *timestamp*, dan hasil klasifikasi. Setelah data berhasil disimpan, sistem menampilkan hasil deteksi tahap pertama pada terminal. Proses detail tentang klasifikasi tahap pertama terdapat pada potongan kode berikut:

```
// 5. KLASIFIKASI

$result_status = ($normalized_entropy < $threshold) ? "SUS" : "NORMAL";
$delta = $normalized_entropy - $threshold;

$final_result = $result_status;

// Simpan Hasil Klasifikasi tahap 1
$stmt = $conn->prepare("
INSERT INTO classification
(window_id, timestamp, normalized_entropy, threshold, delta, result,
final_result)
VALUES (?, ?, ?, ?, ?, ?, ?)
");

$stmt->bind_param("isdddss",
    $window_id,
    $current_timestamp,
    $normalized_entropy,
    $threshold,
    $delta,
    $result_status,
    $final_result // default dulu
);
```

Gambar 4.39 Skrip Klasifikasi Tahap Pertama

Setelah *threshold* berhasil diperoleh, sistem memasuki tahap klasifikasi trafik tahap pertama. Pada tahap ini, sistem membandingkan nilai *normalized entropy* dengan *threshold* untuk menentukan kondisi trafik jaringan. Jika nilai *normalized*



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

entropy lebih kecil dari *threshold*, maka trafik dikategorikan sebagai trafik mencurigakan (SUS). Sebaliknya, jika nilai *normalized entropy* lebih besar atau sama dengan *threshold*, maka trafik dikategorikan sebagai trafik normal.

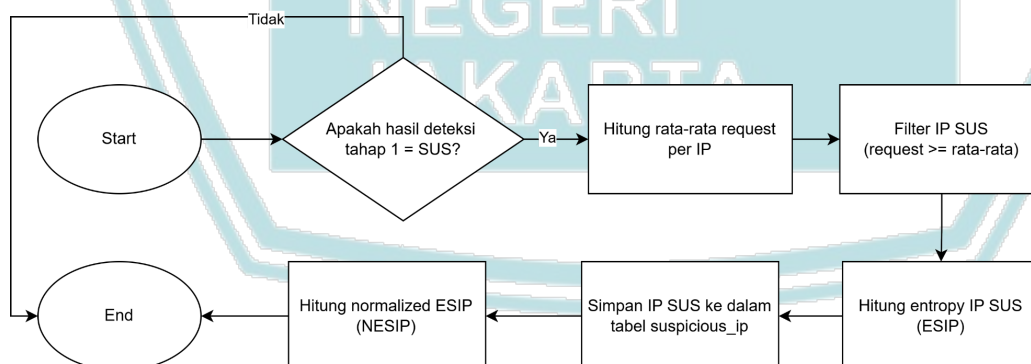
Selain menentukan status trafik, sistem juga menghitung nilai delta, yaitu selisih antara *normalized entropy* dan *threshold*. Nilai ini digunakan untuk menunjukkan seberapa jauh nilai *entropy* terhadap batas deteksi yang ditetapkan sistem.

Hasil klasifikasi awal kemudian disimpan ke tabel *classification* yang berisi informasi *window ID*, *timestamp*, *normalized entropy*, *threshold*, delta, hasil klasifikasi, dan hasil akhir sementara (*final_result*). Pada tahap ini, nilai *final_result* masih mengikuti hasil klasifikasi awal karena proses re-evaluation belum dilakukan. Setelah proses penyimpanan berhasil, sistem mengambil ID hasil klasifikasi (*classification_id*) yang akan digunakan sebagai relasi pada tahap re-evaluation berikutnya. Sistem juga menampilkan hasil deteksi tahap pertama pada terminal sebagai monitoring proses deteksi secara real-time.

4.3.8 Implementasi Tahap Revaluasi IP

1. Perhitungan Entropy dan Normalized Entropy IP Mencurigakan

Tahap re-evaluation dilakukan hanya ketika hasil klasifikasi tahap pertama menunjukkan status mencurigakan (SUS).



Gambar 4.40 Flowchart Perhitungan Entropy IP Mencurigakan

Pada tahap ini, sistem menghitung rata-rata *request* per IP dalam *window* saat ini, kemudian melakukan *filtering* untuk memilih IP yang memiliki jumlah *request* sama dengan atau lebih besar dari rata-rata sebagai IP mencurigakan. Selanjutnya,



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 Politeknik Negeri Jakarta

sistem menghitung *entropy* IP mencurigakan (ESIP) dan *normalized entropy*-nya berdasarkan distribusi *request* dari IP-IP hasil *filtering*. Data IP mencurigakan juga disimpan ke tabel *suspicious_ip* untuk keperluan pelacakan sumber trafik mencurigakan. Proses detail tentang perhitungan *entropy* IP mencurigakan terdapat pada potongan kode berikut:

```
if ($result_status == "SUS") {
    // 6. IDENTIFIKASI SUSIP & HITUNG ESIP

    $normalized_esip = 0;
    $entropy_esip = 0;
    $filtered_susip = [];

    $avg_req = $total_requests / $n; //1

    $filtered_susip = [];
    $total_req_filtered = 0;

    foreach ($data as $row) { //2
        // Hanya ambil IP yang requestnya sama atau di atas rata-rata
        if ($row['request_count'] >= $avg_req) {
            $filtered_susip[] = $row;
            $total_req_filtered += $row['request_count'];
        }
    }
    $n_susip = count($filtered_susip); //2

    if ($n_susip > 0) {
        foreach ($filtered_susip as $s) {
            $p_prime = $s['request_count'] / $total_req_filtered; //3
            if ($p_prime > 0) {
                $entropy_esip -= $p_prime * log($p_prime, 2); //4
            }

            $stmt = $conn->prepare("INSERT INTO suspicious_ip
(window_id, timestamp, ip, request_count, probability) VALUES (?, ?, ?,
?, ?)");
            $stmt->bind_param("issdd", $window_id, $current_timestamp,
            $s['ip_address'], $s['request_count'], $p_prime);
            $stmt->execute() or die($stmt->error);
        }
        $normalized_esip = ($n_susip > 1) ? ($entropy_esip /
log($n_susip, 2)) : 0; //4
    }
}
```

Gambar 4.41 Skrip Perhitungan Entropy IP Mencurigakan

Setelah trafik diklasifikasikan sebagai mencurigakan (SUS) pada tahap pertama, sistem melanjutkan ke tahap identifikasi IP mencurigakan (SUSIP) dan perhitungan *entropy* IP mencurigakan (ESIP).

1. Perhitungan Rata-rata Request per IP: Sebelum menyaring data, sistem menghitung rata-rata *request* per IP (\$avg_req) di dalam *window* tersebut dengan membagi total paket trafik terhadap jumlah IP unik keseluruhan.



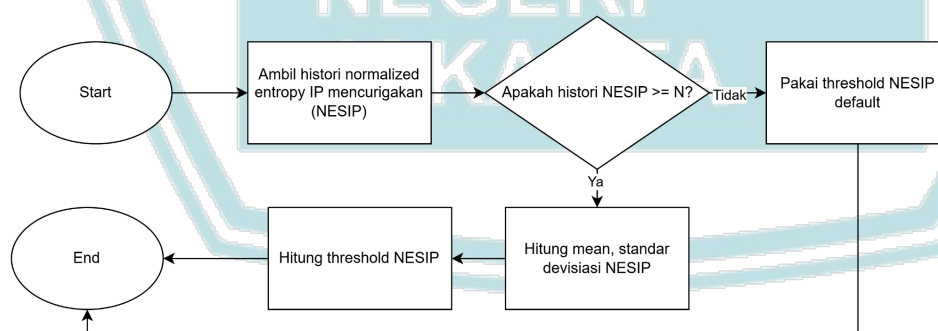
Hak Cipta :

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 kritis 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 Politeknik Negeri Jakarta

Operasi matematika ini diwujudkan melalui baris kode $\$avg_req = \$total_requests / \$n;$

2. Penyaringan IP Mencurigakan: Perulangan foreach digunakan untuk memilih IP yang memiliki jumlah *request* sama/lebih besar dari rata-rata *request* dalam *window* tersebut (if ($\$row['request_count'] \geq \$avg_req$)). IP yang memenuhi kondisi disimpan ke $\$filtered_susip[]$, total *request* dari IP mencurigakan tersebut dijumlahkan ke $\$total_req_filtered$, dan total jumlah IP-nya dijumlahkan ke $\$n_susip$.
3. Perhitungan Probabilitas IP Mencurigakan: Karena lingkungan data berubah menjadi hanya kelompok IP mencurigakan, probabilitas baru dihitung ulang. Jika terdapat minimal satu IP yang lolos penyaringan sebelumnya (if ($\$n_susip > 0$)), dilakukan pembagian jumlah *request* IP ke-i dengan total *request* seluruh IP mencurigakan ($\$p_prime = \$s['request_count'] / \$total_req_filtered$).
4. Perhitungan Entropy dan Normalized Entropy IP Mencurigakan: Perhitungan ini dikonversi ke dalam baris kode $\$entropy_esip = \$p_prime * \log(\$p_prime, 2);$. Selanjutnya, nilai entropy tersebut dinormalisasi pada baris kode $\$normalized_esip = (\$n_susip > 1) ? (\$entropy_esip / \log(\$n_susip, 2)) : 0;$.

2. Perhitungan Threshold revaluasi IP



Gambar 4.42 Flowchart Perhitungan Threshold IP Mencurigakan

Sistem selanjutnya melakukan perhitungan threshold khusus tahap revaluasi. Setelah mengambil histori nilai *normalized entropy* IP mencurigakan sebelumnya, dilakukan pengecekan apakah histori lebih dari atau sama dengan N. Jika histori



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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

belum mencukupi, sistem menggunakan nilai *threshold default*. Jika histori mencukupi, maka sistem menghitung mean, standar deviasi, dan *threshold* NESIP (*threshold* khusus revaluasi IP). Proses detail tentang perhitungan *threshold* IP revaluasi terdapat pada potongan kode berikut:

```
// 7. HITUNG THRESHOLD ESIP (dari histori NESIP sebelumnya)
$nesip_values = [];
$result_nesip = $conn->query("SELECT normalized_esip FROM
reevaluation_log ORDER BY timestamp DESC LIMIT $N_window"); //1

while ($row = $result_nesip->fetch_assoc()) {
    $nesip_values[] = $row['normalized_esip'];
}

$mean_nesip = 0;
$stddev_nesip = 0;
$threshold_nesip = 0;

if (count($nesip_values) >= $N_window) { //2
    $mean_nesip = array_sum($nesip_values) / count($nesip_values);
    $sum_sq = 0;
    foreach ($nesip_values as $val) { $sum_sq += pow($val -
$mean_nesip, 2); }
    $stddev_nesip = sqrt($sum_sq / count($nesip_values));

    $threshold_nesip = $mean_nesip-($k_dynamic*$stddev_nesip); //3
} else {
    $threshold_nesip = 0.962;
}
```

Gambar 4.43 Skrip Perhitungan Threshold IP Mencurigakan

Tahap berikutnya adalah perhitungan *threshold* revaluasi IP berdasarkan histori nilai *normalized entropy* IP mencurigakan dari proses sebelumnya. Proses pembentukan skrip PHP pada gambar 4.43 dijabarkan sebagai berikut:

1. Ekstraksi Data Normalized Entropy IP Mencurigakan (NESIP): Sistem mengambil sejumlah histori nilai *normalized_esip* dari tabel *reevaluation_log* sebanyak jumlah *window baseline* (*N_window*) yang telah ditentukan.
2. Perhitungan Rata-rata dan Standar Deviasi NESIP: Jika data *normalized_esip* memenuhi batas minimum N (*if* (*count*(\$*esip_values*) >= \$*N_window*)), dilakukan perhitungan rata-rata (*\$mean_nesip*) dan standar deviasi (*\$stddev_nesip*) yang akan digunakan pada proses perhitungan *threshold*. Jika batas minimum N tidak terpenuhi, maka digunakan



Hak Cipta :

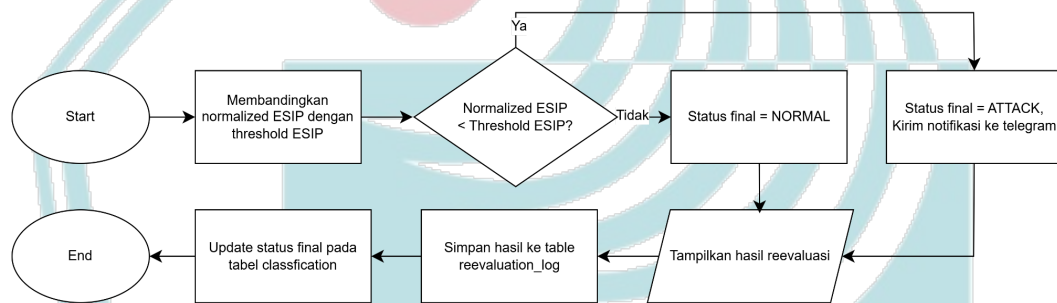
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

threshold default sebesar 0.962 yang didapat dari rata-rata NESIP pada keadaan normal.

3. Perhitungan Threshold revaluasi IP: Threshold dihitung sesuai dengan persamaan pada subbab 2.5.3. Konversinya ke dalam skrip PHP terdapat pada baris kode `$threshold_esip = $mean_esip - ($k_dynamic * $stddev_esip);`.

3. Hasil Klasifikasi Tahap Kedua

Tahap berikutnya adalah klasifikasi tahap kedua dengan membandingkan nilai *normalized ESIP* terhadap *threshold ESIP*.



Gambar 4.44 Flowchart Klasifikasi Tahap Kedua

Jika *normalized ESIP* berada di bawah *threshold ESIP*, maka trafik dikategorikan sebagai serangan dan sistem mengirimkan notifikasi ke Telegram. Seluruh hasil re-evaluation kemudian disimpan ke tabel *reevaluation_log* dan hasil akhir pada tabel *classification* diperbarui.

```
// 8. FINAL DECISION

if ($normalized_esip < $threshold_nesip) {
    $final_result = "ATTACK";
    include "send-telegram.php";
} else {
    $final_result = "NORMAL";
}

echo "TAHAP 2: NESIP: $normalized_esip | ThresNESIP: $threshold_nesip | FINAL: $final_result\n";
```

Gambar 4.45 Skrip Klasifikasi Tahap Kedua

Pada tahap klasifikasi kedua, sistem membandingkan nilai *normalized ESIP* dengan *threshold ESIP* untuk menentukan hasil akhir deteksi. Jika nilai



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 Politeknik Negeri Jakarta

normalized ESIP lebih kecil dari *threshold* ESIP, maka trafik dikategorikan sebagai serangan (ATTACK). Dalam kondisi tersebut, sistem juga menjalankan file `send-telegram.php` untuk mengirimkan notifikasi serangan ke Telegram.

Sebaliknya, jika normalized ESIP berada di atas *threshold* ESIP, maka trafik dikategorikan sebagai normal. Hasil deteksi kedua kemudian ditampilkan pada terminal untuk memudahkan observasi proses deteksi secara *near real-time*.

```
// 9. SIMPAN REEVALUATION

if ($n_susip > 1) {
    $stmt = $conn->prepare("
        INSERT INTO reevaluation_log
        (window_id, timestamp, entropy_esip, normalized_esip,
        mean_nesip, stddev_nesip, threshold_nesip, final_result)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)
    ");
    $stmt->bind_param("isddddds", $window_id, $current_timestamp,
    $entropy_esip, $normalized_esip, $mean_nesip, $stddev_nesip,
    $threshold_nesip, $final_result);
    $stmt->execute() or die($stmt->error);
}

// update hasil
$stmt = $conn->prepare("
    UPDATE classification
    SET final_result = ?
    WHERE id = ?
");
$stmt->bind_param("si", $final_result, $inserted_id);
$stmt->execute() or die($stmt->error);
```

Gambar 4.46 Skrip Penyimpanan Hasil revaluasi IP

Selanjutnya, seluruh hasil keseluruhan revaluasi IP seperti *entropy* ESIP, normalized ESIP, mean ESIP, standar deviasi ESIP, *threshold* ESIP, dan hasil akhir deteksi disimpan ke tabel `reevaluation_log`. Sistem juga melakukan pembaruan nilai `final_result` pada tabel `classification` agar hasil klasifikasi akhir sesuai dengan hasil re-evaluation tahap kedua.

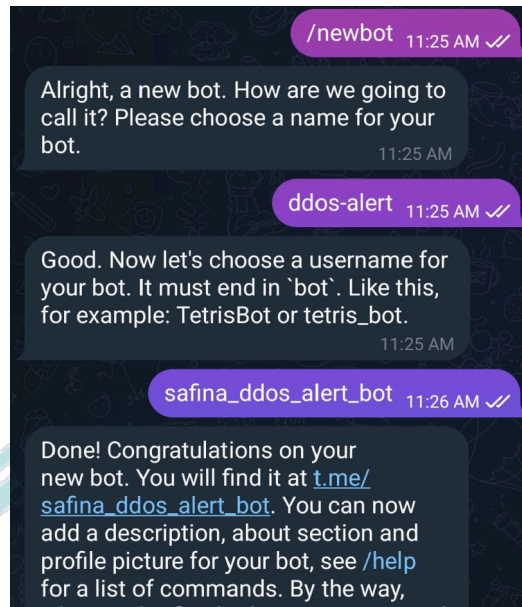
4.3.9 Implementasi Pengiriman Alert ke Telegram

Sebelum sistem dapat mengirimkan notifikasi otomatis ke Telegram, diperlukan proses konfigurasi bot Telegram terlebih dahulu. Bot dibuat melalui akun resmi Telegram bernama BotFather dengan menggunakan perintah `/newbot`.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.47 Pembuatan Bot Telegram

Setelah bot berhasil dibuat, Telegram akan memberikan bot *token* yang berfungsi sebagai identitas sekaligus autentikasi bot ketika mengakses Telegram Bot API. Selain itu, sistem juga memerlukan *chat ID* sebagai tujuan pengiriman pesan notifikasi. *Chat ID* dapat diperoleh dengan mengirim pesan ke bot terlebih dahulu, kemudian mengakses *endpoint* Telegram API `getUpdates` untuk melihat identitas *chat* yang aktif. Nilai bot *token* dan *chat ID* tersebut kemudian disimpan pada *file* `send-telegram.php` agar sistem dapat mengirimkan *alert* secara otomatis ketika serangan DDoS terdeteksi.

```
<?php
$botToken = "xxx";
$chatId = "xxx";
$susip_text = "";

foreach ($filtered_susip as $s) {
    $susip_text .= "- {$s['ip_address']} ({$s['request_count']}
req)\n";
}

$message = "
🚨 *DDoS ATTACK DETECTED* 🚨

*Time:* $current_timestamp
*Total Request:* $total_requests
*NE:* $normalized_entropy
*Threshold:* $threshold
*NESIP:* $normalized_esip
*Threshold NESIP:* $threshold_nesip
```



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 Politeknik Negeri Jakarta

```
*Suspicious IPs:*
$susip_text
";

$url = "https://api.telegram.org/bot$botToken/sendMessage";

$dataTelegram = [
    'chat_id' => $chatId,
    'text' => $message,
    'parse_mode' => 'Markdown'
];

$options = [
    'http' => [
        'header' => "Content-type:
application/x-www-form-urlencoded\r\n",
        'method' => 'POST',
        'content' => http_build_query($dataTelegram),
    ],
];

$context = stream_context_create($options);
$response = file_get_contents($url, false, $context);

if ($response === FALSE) {
    echo "Failed to send Telegram alert!\n";
} else {
    echo "Telegram alert sent!\n";
}
?>
```

Gambar 4.48 Skrip Pengiriman *Alert* ke Telegram

File `send-telegram.php` pada gambar 4.48 berfungsi untuk mengirimkan notifikasi otomatis ke Telegram ketika sistem mendeteksi serangan DDoS dengan status akhir “ATTACK”. Modul ini memanfaatkan Telegram Bot API untuk mengirim pesan berisi informasi hasil deteksi, seperti waktu deteksi, jumlah *request* dalam *window*, nilai *Normalized Entropy* (NE), *Threshold*, nilai *Normalized Entropy Suspicious IP* (NESIP), serta daftar IP mencurigakan yang teridentifikasi pada proses re-evaluasi. Pengiriman notifikasi dilakukan secara otomatis setelah proses klasifikasi tahap kedua menghasilkan keputusan serangan (ATTACK).

4.3.10 Instalasi Alat Pengujian

1. Instalasi Locust

Sebelum melakukan instalasi Locust, dilakukan pengecekan jika Python dan pip sudah terinstal menggunakan perintah “`Python3 --version`” dan “`pip --version`”.



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 Politeknik Negeri Jakarta

```
server2@server2:~/ddos_detection$ sudo apt update
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
74 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Gambar 4.49 Penginstalan Python3 dan Modul Pip

Perintah `sudo apt update` dilakukan untuk memastikan seluruh *package* sistem berada pada versi terbaru sehingga dapat mengurangi potensi konflik dependensi saat instalasi Locust dilakukan.

```
server2@server2:~/ddos_detection$ sudo apt install python3 python3-pip -y
[sudo] password for server2:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.7).
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
```

Gambar 4.50 Penginstalan Python3 dan Modul Pip

Perintah `sudo apt install python3 python3-pip` digunakan untuk menginstal bahasa pemrograman Python serta modul pip sebagai *package manager* Python. Jika proses berhasil, terminal akan menampilkan nomor versi Python dan pip tanpa pesan *error*.

```
server2@server2:~/ddos_detection$ python3 -m venv venv
```

Gambar 4.51 Pembuatan *Virtual Environment*

Setelah sistem diperbarui, dilakukan pembuatan *virtual environment* menggunakan modul `venv`. *Virtual environment* digunakan untuk memisahkan dependensi proyek dengan dependensi sistem utama sehingga instalasi *package* hanya berlaku pada lingkungan proyek tertentu. Pada gambar 4.51, *virtual environment* bernama “venv” dibuat menggunakan *command* `python3 -m venv venv`.



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 Politeknik Negeri Jakarta

```
server2@server2:~/ddos_detection$ source venv/bin/activate
(venv) server2@server2:~/ddos_detection$
```

Gambar 4.52 Pengaktifan Virtual Environment

Jika pengaktifan *virtual environment* berhasil, terminal akan menampilkan nama *environment* pada awal baris terminal. Selanjutnya perintah pip install locust dilakukan didalam *virtual environment*.

```
(venv) server2@server2:~/ddos_detection$ locust -V
locust 1.4.3
```

Gambar 4.53 Pengecekan Versi Locust

Jika Locust berhasil terinstal, perintah locust -V dapat dilakukan untuk mengecek versi Locust.

2. Instalasi GoldenEye

Berikut adalah tahapan untuk instalasi GoldenEye:

```
server2@server2:~/ddos_detection$ source venv/bin/activate
(venv) server2@server2:~/ddos_detection$ git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye'...
```

Gambar 4.54 Cloning Repositori GoldenEye dari Github

Pada gambar 4.54, pengunduhan GoldenEye dilakukan di dalam *virtual environment*. Setelah *cloning* selesai, perintah cd GoldenEye/ dilakukan untuk berpindah ke dalam direktori tersebut untuk mengeksekusi *tool*-nya.

4.4 Pengujian

4.4.1. Deskripsi Pengujian

Tahap pengujian dilakukan untuk mengevaluasi dan membuktikan efektivitas sistem deteksi serangan DDoS berbasis *entropy* yang memanfaatkan parameter K-dinamis dan mekanisme revaluasi IP pada web *server* Ubuntu. Pengujian bertujuan untuk membuktikan bahwa sistem mampu mendeteksi anomali trafik, membedakan sumber trafik legal dan serangan DDoS, serta mengetahui dampak implementasi sistem terhadap penggunaan sumber daya *server*.

Lingkungan pengujian dibangun menggunakan *server* Ubuntu dalam jaringan lokal yang terdiri atas satu *virtual machine* (VM) sebagai web *server* target dan



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 Politeknik Negeri Jakarta

empat VM sebagai penghasil trafik. *Server* target menjalankan layanan Apache2 serta sistem deteksi yang dikembangkan pada penelitian ini. Sementara itu, VM pembuat trafik digunakan untuk menghasilkan trafik legal menggunakan Locust dan trafik serangan HTTP *Flood* menggunakan GoldenEye. Untuk mensimulasikan variasi sebaran paket yang dinamis, skrip Locust dikonfigurasi agar memanipulasi *header* HTTP secara acak dari kumpulan alamat IP yang ditentukan.

Secara umum, pengujian pada penelitian ini dibagi menjadi tiga bagian. Pengujian pertama dilakukan untuk memverifikasi fungsionalitas sistem dalam mendeteksi kondisi trafik normal, *flash crowd*, dan serangan DDoS berdasarkan nilai *entropy*, *threshold*, K-dinamis, serta mekanisme revaluasi IP. Pengujian kedua dilakukan untuk mengevaluasi performa sistem deteksi menggunakan *confusion matrix* yang terdiri atas nilai *accuracy*, *precision*, *recall*, dan F1-score. Pengujian ketiga dilakukan untuk menganalisis dampak penerapan sistem deteksi terhadap performa *server* berdasarkan penggunaan CPU dan RAM selama sistem beroperasi.

Hasil dari seluruh pengujian tersebut digunakan untuk menjawab rumusan masalah penelitian mengenai implementasi, performa, dan dampak sistem deteksi DDoS berbasis *entropy* dengan K-dinamis dan revaluasi IP pada web *server* Ubuntu.

4.4.2. Prosedur Pengujian

4.4.2.1 Pengujian Fungsionalitas Sistem Deteksi Serangan DDoS Berbasis *Entropy* Menggunakan K-dinamis dan revaluasi IP

Pengujian ini bertujuan untuk memverifikasi fungsionalitas sistem deteksi dalam memproses trafik HTTP yang masuk ke web *server*. Pengujian dilakukan untuk memastikan sistem mampu menghitung nilai *entropy*, membentuk *threshold* menggunakan metode K-dinamis, mengidentifikasi *suspicious* IP, melakukan revaluasi IP, serta menghasilkan keputusan akhir sesuai dengan karakteristik trafik yang diterima. Proses pengujian terbagi ke dalam 3 skenario kondisi trafik sebagai 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

A. Pengujian Kondisi Trafik Normal

Skenario trafik normal disimulasikan dengan cara seluruh permintaan berasal dari pengguna legal. Trafik dihasilkan menggunakan *tool* Locust yang dijalankan pada empat VM dengan detail sebagai berikut:

Tabel 4.4 Jumlah VM Pada Kondisi Trafik Normal

Nama <i>Hostname</i>	Role
server1 (192.168.100.11)	Locust (10 <i>users</i> , 5 <i>spawn-rate</i>)
server2 (192.168.100.12)	Locust (10 <i>users</i> , 5 <i>spawn-rate</i>)
server3 (192.168.100.13)	Locust (10 <i>users</i> , 5 <i>spawn-rate</i>)
server4 (192.168.100.14)	Locust (10 <i>users</i> , 5 <i>spawn-rate</i>)

Masing-masing VM dikonfigurasi menggunakan 10 *user* virtual dengan spawn rate 5 user per detik. Tiap konfigurasi Locust di dalam VM Simulasi dijalankan selama 5 menit dengan target akses menuju halaman login.php pada web *server*. Hasil pengujian digunakan sebagai acuan perilaku sistem pada kondisi operasional normal.

B. Pengujian Kondisi Trafik *Flash Crowd*

Skenario dilakukan untuk menguji efektivitas sistem deteksi ketika terjadi lonjakan trafik normal yang masif, tetapi masih berasal dari *user* legal. Pada kondisi ini, permintaan dari *user* legal berasal dari *tool* Locust yang dijalankan dari empat VM (server1 hingga server4) dengan detail berikut:

Tabel 4.5 Jumlah VM Pada Kondisi Trafik *Flash Crowd*

Nama <i>Hostname</i>	Role
server1 (192.168.100.11)	Locust (50 <i>users</i> , 10 <i>spawn-rate</i>)
server2 (192.168.100.12)	Locust (50 <i>users</i> , 10 <i>spawn-rate</i>)
server3 (192.168.100.13)	Locust (50 <i>users</i> , 10 <i>spawn-rate</i>)
server4 (192.168.100.14)	Locust (50 <i>users</i> , 10 <i>spawn-rate</i>)



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Masing-masing VM menggunakan Locust untuk mengirim simulasi permintaan ke web *server* target dengan konfigurasi 50 user dan spawn rate 10 user per detik. Simulasi dijalankan selama 5 menit.

C. Pengujian Kondisi Trafik Serangan

Skenario serangan digunakan untuk mengevaluasi kemampuan sistem dalam mendeteksi serangan DDoS. Pada skenario ini, *server1* berperan sebagai pengguna legal yang menghasilkan trafik menggunakan Locust. Sementara itu, *server2*, *server3*, dan *server4* berperan sebagai penyerang yang menjalankan GoldenEye.

Tabel 4.6 Jumlah VM Pada Kondisi Serangan

Nama <i>Hostname</i>	Role
<i>server1</i> (192.168.100.11)	Locust (50 <i>users</i> , 10 <i>spawn-rate</i>)
<i>server2</i> (192.168.100.12)	GoldenEye (5 <i>workers</i> , 100 <i>socket</i>)
<i>server3</i> (192.168.100.13)	GoldenEye (3 <i>workers</i> , 150 <i>socket</i>)
<i>server4</i> (192.168.100.14)	GoldenEye (2 <i>workers</i> , 150 <i>socket</i>)

Simulasi dijalankan selama 5 menit dengan target halaman login.php pada web *server*. Detail simulasi sebagai berikut:

Tabel 4.7 Detail Simulasi Kondisi Serangan

Waktu (Menit)	Trafik Aktif
0 sampai 1	<i>Flash crowd</i>
1 sampai 3	<i>Flash crowd</i> dan serangan
3 sampai 5	<i>Flash crowd</i>

Skenario dibuat sedemikian rupa sehingga simulasi merepresentasikan terjadinya suatu lonjakan trafik akibat serangan. Hasil pengujian digunakan untuk mengevaluasi kemampuan sistem dalam mengidentifikasi trafik serangan dan membedakannya dari trafik legal yang berjalan secara bersamaan. Parameter 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

diamati dalam pengujian adalah nilai *entropy*, *normalized entropy*, K-dinamis, *threshold* adaptif, hasil deteksi, serta hasil perhitungan pada revaluasi IP.

4.4.2.2 Pengujian Performa Sistem Deteksi dengan *Confusion Matrix*

Tujuan dari pengujian ini untuk mengevaluasi sistem dalam mengklasifikasikan alamat IP asal dari permintaan HTTP yang masuk, yakni membedakan antara pengguna legal dan sumber serangan. *Window* waktu yang berada pada periode aktifnya GoldenEye dikategorikan sebagai kondisi serangan (positif), sedangkan *window* waktu yang hanya berisi trafik dari Locust dikategorikan sebagai kondisi normal (negatif). Hasil klasifikasi yang dihasilkan sistem kemudian dibandingkan dengan kondisi sebenarnya untuk memperoleh nilai TP, FP, TN, dan FN:

- *True Positive* (TP): Sistem menghasilkan hasil akhir “ATTACK” pada *window* waktu ketika GoldenEye sedang aktif mengirim serangan.
- *False Positive* (FP): Sistem menghasilkan hasil akhir “ATTACK” pada *window* waktu ketika trafik yang masuk hanya berasal dari Locust.
- *True Negative* (TN): Sistem menghasilkan hasil akhir “NORMAL” pada *window* waktu ketika trafik yang masuk hanya berasal dari Locust.
- *False Negative* (FN): Sistem menghasilkan hasil akhir “NORMAL” pada *window* waktu ketika GoldenEye sedang aktif mengirim serangan. Hal ini menandakan serangan lolos dari deteksi.

Selanjutnya, komponen tersebut digunakan sebagai dasar perhitungan metrik akurasi, *precision*, *recall*, dan *F1-score* sistem deteksi sesuai persamaan pada subbab 2.8.

4.4.2.3 Pengujian Dampak Sistem Deteksi Terhadap Sumber Daya Server

Tujuan dari pengujian ini adalah untuk mengetahui bagaimana dampak sistem deteksi terhadap performa CPU dan RAM *virtual machine*. Terdapat tiga mode uji yang digunakan, yaitu:

Tabel 4.8 Mode Uji Sumber Daya *Server*

Mode	Deskripsi
Idle	Apache2 aktif, tanpa trafik
Normal	Trafik Locust
Serangan	Trafik Locust + GoldenEye

Besar penggunaan CPU dan RAM diperoleh menggunakan skrip python yang dijalankan selama proses pengujian berlangsung. *Library* psutil digunakan untuk merekam data penggunaan sumber daya setiap 1 detik, yang kemudian dikonversikan ke dalam satuan persentase dan *megabyte* (MB). Pada akhir sesi pengujian, akumulasi data yang tersimpan di dalam memori akan disusun menjadi struktur tabel dan disimpan ke dalam *file* csv.

4.4.3 Data Hasil Pengujian

4.4.3.1 Data Hasil Pengujian Sistem Deteksi Serangan DDoS Berbasis Entropy Menggunakan K-dinamis dan revaluasi IP

A. Hasil Pengujian Kondisi Trafik Normal

Pengujian pada kondisi normal untuk mengamati kemampuan sistem menghadapi beban trafik rendah hingga sedang. Berdasarkan simulasi yang dilakukan, sistem mencatat total *request* yang masuk sebanyak 21.825 *request*.

Tabel 4.9 Hasil Perhitungan Parameter Deteksi Pada Kondisi Normal

Fase	<i>Entropy</i>	<i>Normalized Entropy</i>	K-dinamis	<i>Threshold</i> Tahap 1	Hasil Tahap 1
Awal	6,4756	0,96494	0,00829	0,9498	NORMAL
Tengah	7,1213	0,95541	0,00211	0,95458	NORMAL
Akhir	7,12298	0,96373	0,0025	0,95748	NORMAL

Berdasarkan tabel 4.9, pada fase awal diperoleh nilai *entropy* sebesar 6,4756 dengan *normalized entropy* sebesar 0,96494, nilai K-dinamis 0,00829, dan *threshold* tahap 1 sebesar 0,9498. Pada fase tengah, nilai *entropy* meningkat



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 Politeknik Negeri Jakarta

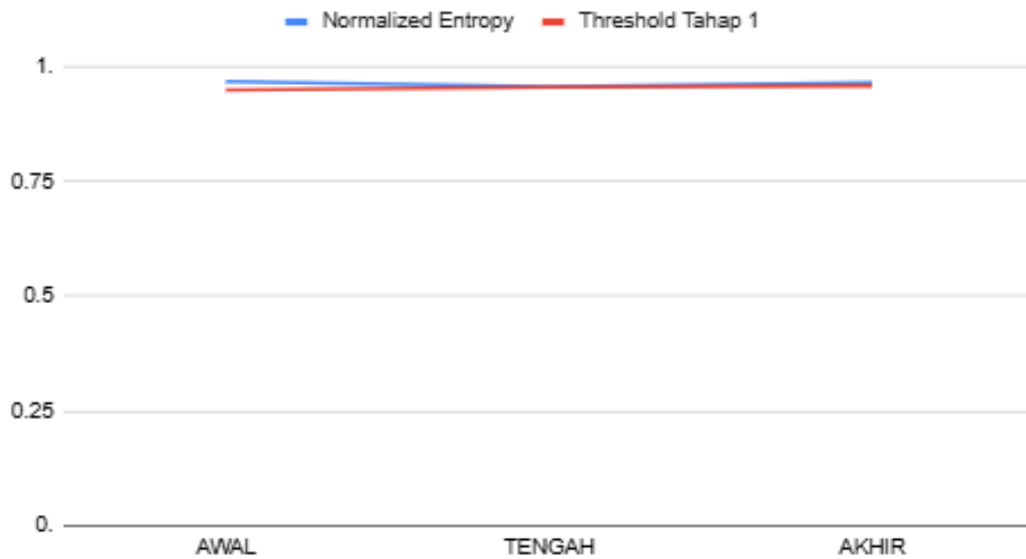


Hak Cipta :

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 Politeknik Negeri Jakarta

menjadi 7,1213 dengan *normalized entropy* 0,95541, nilai K-dinamis 0,00211, dan *threshold* 0,95458. Selanjutnya pada fase akhir, nilai *entropy* tercatat sebesar 7,12298 dengan *normalized entropy* 0,96373, nilai K-dinamis 0,0025, dan *threshold* 0,95748. Seluruh fase pada kondisi normal menghasilkan klasifikasi NORMAL pada tahap pertama deteksi.

Normalized Entropy and Threshold Tahap 1



Gambar 4.55 Grafik Hasil Perhitungan Parameter Deteksi Kondisi Normal

Grafik pada gambar 4.55 memperlihatkan perbandingan antara nilai *normalized entropy* dan *threshold* tahap 1 pada ketiga fase. Garis *normalized entropy* tampak berada secara konsisten di atas garis *threshold* sepanjang pengujian. Pada fase awal, jarak antara kedua garis sedikit lebar karena sistem masih dalam tahap penyesuaian sensitivitas *threshold*. Memasuki fase tengah, jarak antara *normalized entropy* dan *threshold* menjadi sangat dekat karena *threshold* menyesuaikan pola trafik yang stabil. Pada fase akhir, *normalized entropy* kembali berada sedikit lebih tinggi dari *threshold* sehingga kedua garis tidak bersilangan.

B. Hasil Pengujian Kondisi Trafik *Flash Crowd*

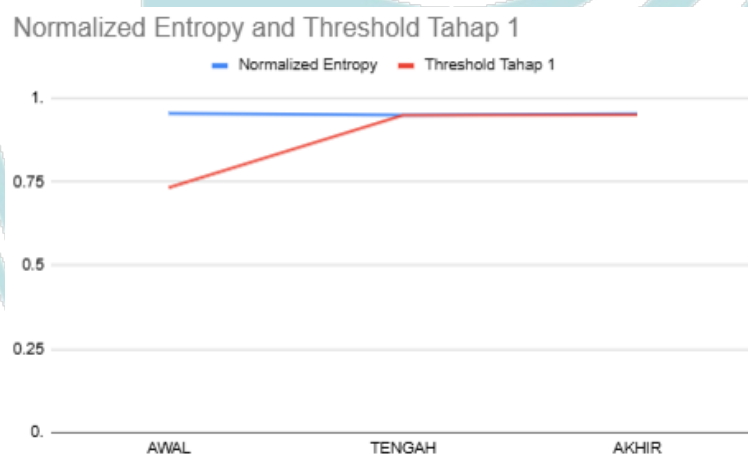
Pengujian pada kondisi *flash crowd* untuk mengamati kemampuan sistem menghadapi lonjakan beban trafik normal yang tinggi. Berdasarkan simulasi yang dilakukan, sistem mencatat total *request* yang masuk sebanyak 65.678.



Tabel 4.10 Hasil Perhitungan Parameter Deteksi Pada Kondisi *Flash Crowd*

Fase	<i>Entropy</i>	<i>Normalized Entropy</i>	K-dinamis	<i>Threshold Tahap 1</i>	Hasil Tahap 1
Awal	7,02227	0,95374	0,0055	0,73212	NORMAL
Tengah	7,56357	0,94906	0,00046	0,94793	NORMAL
Akhir	7,24385	0,95276	0,00287	0,94941	NORMAL

Berdasarkan tabel 4.10, pada fase awal diperoleh nilai *entropy* sebesar 7,02227 dengan *normalized entropy* sebesar 0,95374, nilai K-dinamis 0,0055, dan *threshold* tahap 1 sebesar 0,73212. Pada fase tengah, nilai *entropy* meningkat menjadi 7,56357 dengan *normalized entropy* 0,94906, nilai K-dinamis 0,00046, dan *threshold* 0,94793. Selanjutnya pada fase akhir, nilai *entropy* tercatat sebesar 7,24385 dengan *normalized entropy* 0,95276, nilai K-dinamis 0,00287, dan *threshold* 0,94941. Seluruh fase pada kondisi *flash crowd* menghasilkan klasifikasi NORMAL pada tahap pertama deteksi.



Gambar 4.56 Grafik Hasil Perhitungan Parameter Deteksi Kondisi *Flash Crowd*

Pada gambar 4.56, garis *threshold* fase awal terlihat sangat rendah sementara garis *normalized entropy* berada pada posisi tinggi, sehingga jarak keduanya sangat lebar. Memasuki fase tengah, kedua garis saling mendekat karena *threshold* menyesuaikan diri terhadap lonjakan trafik. Pada fase akhir, kedua garis tetap berada pada jarak yang relatif kecil, tetapi garis *normalized entropy* masih di atas *threshold* sehingga garis tidak saling bersilangan.



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 Politeknik Negeri Jakarta

C. Hasil Pengujian Kondisi Trafik Serangan

Pengujian pada kondisi serangan bertujuan untuk mengevaluasi kemampuan sistem dalam mendeteksi serangan DDoS yang berlangsung bersamaan dengan trafik legal. Berdasarkan simulasi yang dilakukan, sistem mencatat total *request* yang masuk sebanyak 51.797.

Tabel 4.11 Hasil Perhitungan Parameter Deteksi Pada Kondisi Serangan

Fase	<i>Entropy</i>	<i>Normalized Entropy</i>	K-dinamis	<i>Threshold</i> Tahap 1	Hasil Tahap 1
Awal	6,25453	0,93788	0,01056	0,94111	SUS
Tengah	5,15614	0,67414	0,0008	0,78025	SUS
Akhir	6,24269	0,84311	0,00245	0,78995	NORMAL

Berdasarkan Tabel 4.11, pada fase awal diperoleh nilai *entropy* sebesar 6,25453 dengan *normalized entropy* sebesar 0,93788, nilai K-dinamis 0,01056, dan *threshold* tahap 1 sebesar 0,94111 sehingga menghasilkan klasifikasi SUS. Pada fase tengah, nilai *entropy* tercatat sebesar 5,15614 dengan *normalized entropy* 0,67414, nilai K-dinamis 0,0008, dan *threshold* 0,78025 sehingga menghasilkan klasifikasi SUS. Selanjutnya pada fase akhir, nilai *normalized entropy* sebesar 0,84311 dan *threshold* 0,78995 sehingga sistem menghasilkan klasifikasi NORMAL. Fase awal dan tengah memasuki tahap revaluasi IP.

Tabel 4.12 Hasil Perhitungan Tahap Revaluasi IP Pada Trafik Mencurigakan

Fase	<i>Entropy</i> Tahap 2	<i>Normalized Entropy</i> Tahap 2	<i>Threshold</i> Tahap 2	Hasil Tahap 2
Awal	4,99373	0,89806	0,81156	NORMAL
Tengah	2,08151	0,58691	0,73246	ATTACK

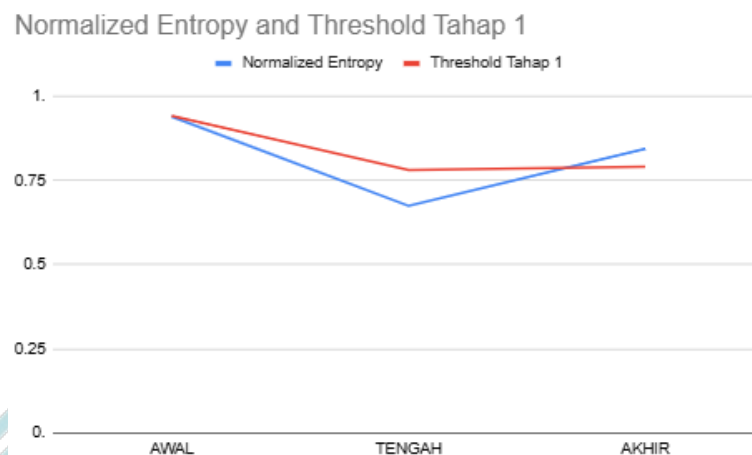
Tabel 4.12 menunjukkan bahwa pada fase awal, nilai *entropy* tahap 2 sebesar 4,99373 dengan *normalized entropy* tahap 2 (NESIP) sebesar 0,89806 dan *threshold* tahap 2 sebesar 0,81156, sehingga menghasilkan klasifikasi NORMAL. Sedangkan pada fase tengah, didapat nilai *entropy* tahap 2 sebesar 2,08151 dengan



Hak Cipta :

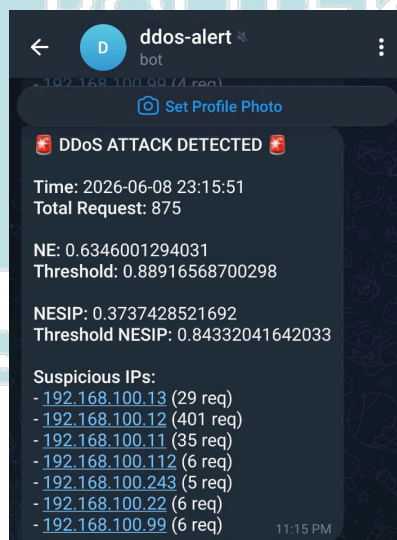
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

normalized entropy tahap 2 (NESIP) sebesar 0,58691 dan *threshold* tahap 2 sebesar 0,73246, sehingga menghasilkan klasifikasi ATTACK.



Gambar 4.57 Grafik Hasil Perhitungan Parameter Deteksi Kondisi Serangan

Pada grafik 4.57, garis *normalized entropy* fase awal berada sedikit dibawah garis *threshold*. Namun, pada fase tengah, garis *normalized entropy* mengalami penurunan yang tajam hingga berada jauh di bawah garis *threshold*. Pada fase akhir, garis *normalized entropy* kembali berada di atas *threshold* setelah periode serangan aktif berakhir.



Gambar 4.58 Hasil Pengiriman Notifikasi ke Telegram

Pada gambar 4.58, pengiriman notifikasi ke bot telegram dilakukan ketika hasil tahap revaluasi IP menyatakan ATTACK beserta informasi 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.4.3.2 Data Hasil Pengujian Performa Sistem Deteksi dengan Confusion Matrix

Perhitungan *confusion matrix* dilakukan dengan mengamati ketepatan hasil klasifikasi tiap *window* sesuai dengan kondisi sebenarnya pada saat pengujian berlangsung.

A. Hasil Pengujian *Confusion Matrix* Kondisi Normal

Tabel 4.13 Hasil *Confusion Matrix* Kondisi Normal

Klasifikasi	Jumlah <i>Window</i>
<i>True Positive</i> (TP)	0
<i>False Positive</i> (FP)	1
<i>True Negative</i> (TN)	60
<i>False Negative</i> (FN)	0

Berdasarkan hasil *confusion matrix* pada tabel 4.13, dapat dihitung nilai performa sistem sebagai berikut:

- *Accuracy*: Seberapa sering sistem mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{0 + 60}{0 + 60 + 1 + 0}$$

$$Accuracy = 98,36\%$$

Nilai *Precision*, *Recall*, dan *F1-Score* tidak dapat dihitung (N/A) karena uji serangan menggunakan GoldenEye tidak dilakukan.

B. Hasil Pengujian *Confusion Matrix* Kondisi *Flash Crowd*

Tabel 4.14 Hasil *Confusion Matrix* Kondisi *Flash Crowd*

Klasifikasi	Jumlah <i>Window</i>
<i>True Positive</i> (TP)	0
<i>False Positive</i> (FP)	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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Klasifikasi	Jumlah <i>Window</i>
<i>True Negative</i> (TN)	57
<i>False Negative</i> (FN)	0

Berdasarkan hasil *confusion matrix* pada tabel 4.14, dapat dihitung nilai performa sistem sebagai berikut:

- *Accuracy*: Seberapa sering sistem mengklasifikasikan dengan benar

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{0 + 57}{0 + 57 + 2 + 0}$$

$$Accuracy = 96,61\%$$

Nilai *Precision*, *Recall*, dan *F1-Score* tidak dapat dihitung (N/A) karena uji serangan menggunakan GoldenEye tidak dilakukan.

C. Hasil Pengujian Confusion Matrix Kondisi Serangan

Tabel 4.15 Hasil Confusion Matrix Kondisi Serangan

Klasifikasi	Jumlah <i>Window</i>
<i>True Positive</i> (TP)	16
<i>False Positive</i> (FP)	0
<i>True Negative</i> (TN)	59
<i>False Negative</i> (FN)	8

Berdasarkan hasil *confusion matrix* pada tabel 4.15, dapat dihitung nilai performa sistem sebagai berikut:

- *Accuracy*: Persentase seluruh *window* yang diklasifikasi dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{16 + 59}{16 + 59 + 0 + 8} = 90,36\%$$



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- *Precision*: Persentase ketepatan klasifikasi serangan yang benar merupakan serangan.

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{16}{16 + 0} = 100\%$$

- *Recall*: Persentase kemampuan sistem menemukan seluruh serangan yang benar-benar terjadi.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{16}{16 + 8} = 66,67\%$$

- *F1-Score*: Persentase keseimbangan antara *precision* dan *recall*.

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1\ Score = 2 \cdot \frac{100 \cdot 66,67}{100 + 66,67} = 80\%$$

4.4.3.3 Data Hasil Pengujian Dampak Sistem Deteksi Terhadap Sumber Daya Server

Berikut merupakan data penggunaan sumber daya server ketika web server dan sistem deteksi berjalan bersamaan dalam tiga kondisi pengujian:

A. Hasil Pengujian Dampak Sumber Daya Server Kondisi Normal

Tabel 4.16 Hasil Pemantauan Sumber Daya Kondisi Normal

	CPU (%)	RAM (%)	RAM (MB)
Rata-rata	48,60	47,34	1,851.91
Maksimum	85,90	47,70	1,864.16
Minimum	0,00	45,80	1,796.43

Pada tabel 4.16, selama pengujian kondisi trafik normal sistem menggunakan sumber daya CPU dengan rata-rata sebesar 48,60% dan mencapai penggunaan maksimum sebesar 85,90%. Selain itu, penggunaan RAM relatif stabil dengan rata-rata 47,34% atau setara dengan 1,851.91 MB, serta penggunaan maksimum



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 Politeknik Negeri Jakarta

sebesar 47,70% atau 1,864.16 MB. Kecilnya selisih antara nilai rata-rata dan maksimum pada penggunaan RAM menunjukkan bahwa kebutuhan memori sistem cenderung konstan selama pengujian berlangsung.

Tabel 4.17 Hasil Pemantauan Sumber Daya Kondisi Flash Crowd

	CPU (%)	RAM (%)	RAM (MB)
Rata-rata	73,56	48,88	1,912.46
Maksimum	98,90	49,30	1,930.11
Minimum	0,00	45,00	1,758.87

Berdasarkan tabel 4.17, selama pengujian kondisi trafik *flash crowd* sistem menggunakan sumber daya CPU dengan rata-rata sebesar 73,56% dan mencapai penggunaan maksimum sebesar 98,90%. Nilai tersebut lebih tinggi dibandingkan kondisi normal karena *server* menerima lonjakan trafik legal dalam jumlah besar secara bersamaan dari beberapa klien. Dari sisi memori, penggunaan RAM relatif stabil dengan rata-rata sebesar 48,88% atau setara dengan 1,912.46 MB dan penggunaan maksimum sebesar 49,30% atau 1,930.11 MB. Hasil ini menunjukkan bahwa sistem deteksi mampu beroperasi pada kondisi *flash crowd* dengan konsumsi sumber daya yang masih terkendali, meskipun membutuhkan pemanfaatan CPU yang lebih tinggi untuk memproses jumlah *request* yang lebih besar dibandingkan kondisi normal.

Tabel 4.18 Hasil Pemantauan Sumber Daya Kondisi Serangan

	CPU (%)	RAM (%)	RAM (MB)
Rata-rata	53,90	48,38	1,892.90
Maksimum	100,00	49,70	1,944.09
Minimum	0,00	44,90	1,756.66

Berdasarkan tabel 4.18, selama pengujian kondisi serangan DDoS sistem menggunakan sumber daya CPU dengan rata-rata sebesar 53,90% dan mencapai penggunaan maksimum sebesar 100,00%. Penggunaan CPU maksimum tersebut menunjukkan bahwa pada beberapa periode tertentu sistem mengalami beban pemrosesan yang sangat tinggi akibat masuknya trafik serangan secara intensif. Namun demikian, rata-rata penggunaan CPU masih berada pada tingkat 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

lebih rendah dibandingkan kondisi *flash crowd*. Dari sisi memori, penggunaan RAM berada pada rata-rata 48,38% atau setara dengan 1,892.90 MB dan mencapai maksimum sebesar 49,70% atau 1,944.09 MB. Hasil ini menunjukkan bahwa mekanisme deteksi berbasis *entropy*, K-dinamis, dan revaluasi IP mampu menangani trafik serangan tanpa menyebabkan peningkatan konsumsi memori yang signifikan, meskipun pada beberapa waktu tertentu memerlukan kapasitas CPU yang tinggi untuk melakukan proses analisis dan klasifikasi trafik.

4.4.4 Analisa Data Pengujian Fungsionalitas Sistem Deteksi Serangan DDoS

Berdasarkan hasil pengujian yang disajikan pada subbab 4.4.3.1, sistem deteksi menunjukkan kemampuan yang sesuai dengan rancangan metode *entropy*, K-dinamis, dan revaluasi IP dalam membedakan karakteristik trafik normal, *flash crowd*, dan serangan DDoS. Analisis dilakukan berdasarkan perubahan nilai *entropy*, *normalized entropy*, K-dinamis, dan *threshold* adaptif, serta respons mekanisme revaluasi IP terhadap pola trafik yang diamati.

Pada kondisi trafik normal, nilai *entropy* mengalami peningkatan dari fase awal menuju fase tengah dan kemudian relatif stabil pada fase akhir. Nilai *normalized entropy* pada seluruh fase juga berada pada tingkat yang tinggi (0,95—0,97) dan selalu berada di atas *threshold* tahap pertama (0,95—0,96). Hal ini menunjukkan bahwa distribusi *request* pada kondisi normal masih tersebar dan tidak menunjukkan pola pemusatan trafik. Nilai K-dinamis mengalami penurunan pada fase tengah dan tetap berada pada nilai yang kecil pada fase akhir (0,0025), sehingga *threshold* dapat menyesuaikan karakteristik trafik normal. Dengan kondisi tersebut, sistem menghasilkan klasifikasi NORMAL pada seluruh fase pengujian.

Pada kondisi *flash crowd*, volume *request* yang lebih besar menyebabkan nilai *entropy* berada pada tingkat yang tinggi. Meskipun terjadi lonjakan trafik legal, nilai *normalized entropy* pada fase awal, tengah, dan akhir ($\approx 0,95$) tetap berada di atas *threshold*. Pada fase awal, *threshold* masih berada pada nilai yang lebih rendah, kemudian meningkat pada fase tengah dan akhir mengikuti pola trafik



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

yang terbentuk. Hal ini menunjukkan bahwa mekanisme *threshold* dinamis mampu beradaptasi terhadap lonjakan trafik legal tanpa langsung mengklasifikasikannya sebagai serangan. Seluruh fase pada kondisi *flash crowd* menghasilkan klasifikasi NORMAL pada tahap pertama deteksi.

Pada kondisi serangan DDoS, pola trafik menunjukkan perubahan yang lebih jelas dibandingkan kondisi normal dan *flash crowd*. Pada fase awal, nilai *normalized entropy* berada sedikit di bawah *threshold* dengan selisih sekitar 0,001, sehingga sistem mengklasifikasikan trafik sebagai SUS pada tahap pertama. Namun, setelah dilakukan revaluasi IP, nilai *normalized entropy* tahap kedua masih berada di atas *threshold* tahap kedua, sehingga hasil akhir pada fase awal diklasifikasikan sebagai NORMAL. Hal ini menunjukkan bahwa mekanisme revaluasi IP berperan dalam memverifikasi ulang trafik yang terdeteksi mencurigakan agar tidak langsung dikategorikan sebagai serangan. Memasuki fase tengah, nilai *normalized entropy* mengalami penurunan yang lebih signifikan (0,67) dan berada jauh di bawah *threshold* tahap pertama (0,78). Kondisi tersebut menunjukkan adanya pemusatan distribusi *request* pada sejumlah alamat IP tertentu. Pada tahap revaluasi IP, nilai *normalized entropy* tahap kedua (0,59) juga berada di bawah *threshold* tahap kedua (0,73), sehingga sistem mengonfirmasi kondisi tersebut sebagai ATTACK. Sementara itu, pada fase akhir, nilai *normalized entropy* kembali berada di atas *threshold* tahap pertama dan sistem menghasilkan klasifikasi NORMAL. Hal ini menunjukkan bahwa setelah periode serangan aktif berakhir, pola distribusi trafik kembali mendekati karakteristik trafik legal.

4.4.5 Analisa Data Pengujian Performa Sistem Deteksi dengan *Confusion Matrix*

Berdasarkan hasil pengujian pada subbab 4.4.3.2, performa sistem dievaluasi menggunakan *confusion matrix* untuk mengukur kemampuan sistem dalam membedakan trafik normal dan trafik serangan. Karena tidak terdapat kondisi serangan aktif pada skenario normal dan *flash crowd*, metrik *precision*, *recall*, dan *F1-Score* hanya dapat dihitung pada skenario serangan, sedangkan evaluasi pada dua skenario lainnya difokuskan pada nilai akurasi dan jumlah *false positive*.



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 Politeknik Negeri Jakarta

Pada tabel 4.13 kondisi normal, dari 61 *window* yang dianalisis,, diperoleh nilai *true negative* (TN) = 60 dan *false positive* (FP) = 1. Nilai *accuracy* sebesar 98,36% menunjukkan bahwa sistem mampu mengklasifikasikan hampir seluruh trafik sah secara tepat. Satu *false positive* yang terjadi kemungkinan disebabkan oleh fluktuasi distribusi *request* sesaat yang menyebabkan nilai *normalized entropy* melewati *threshold* secara sementara. Meskipun tahap revaluasi IP telah berjalan pada *window* tersebut, hasil akhirnya tetap tercatat sebagai ATTACK dalam evaluasi sistem.

Pada tabel 4.14 kondisi *flash crowd*, diperoleh TN = 57 dan FP = 2. Nilai *accuracy* sebesar 96,61% menunjukkan bahwa sistem tetap mampu membedakan lonjakan trafik sah dari kondisi serangan meskipun volume *request* meningkat secara cepat. Meskipun terdapat dua *false positive*, nilai akurasi yang masih tinggi menunjukkan bahwa mekanisme K-dinamis dan revaluasi IP mampu mengurangi dampak lonjakan trafik legal terhadap proses deteksi. Hasil ini menjadi indikator bahwa sistem dapat menghadapi fenomena *flash crowd*.

Pada tabel 4.15 kondisi serangan DDoS, sistem memproses total 83 *window* yang mencakup periode serangan aktif maupun periode trafik normal yang berlangsung bersamaan. Dari jumlah tersebut, terdapat 24 *window* aktual serangan (*True Positive* + *False Negative*) dan 59 *window* trafik normal (*True Negative* + *False Positive*). Sistem menghasilkan 16 TP, 59 TN, 0 FP, dan 8 FN, dengan nilai akurasi sebesar 90,36%. Nilai *Precision* sebesar 100% berarti setiap keputusan ATTACK yang dikeluarkan sistem selalu benar dan tidak ada trafik sah yang salah diidentifikasi sebagai serangan. Nilai *Recall* sebesar 66,67% menandakan bahwa sistem berhasil mendeteksi 16 dari 24 *window* serangan aktif. Delapan *window* serangan yang tidak terdeteksi sebagai serangan (*False Negative*) dapat terjadi karena trafik serangan berjalan bersamaan dengan trafik legal, sehingga proses deteksi yang dilakukan menggunakan *sliding window* dapat membuat komposisi trafik pada setiap *window* berbeda.

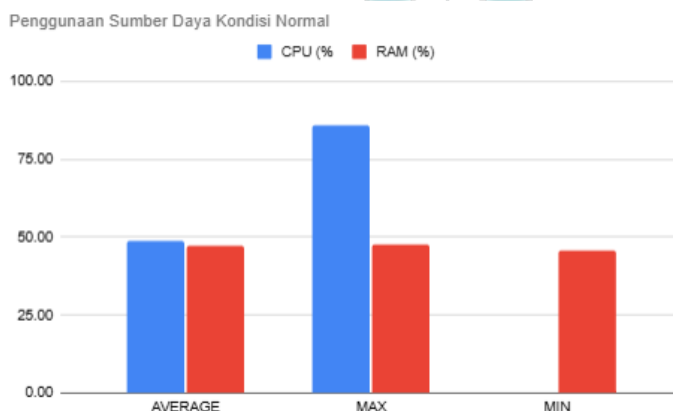


Hak Cipta :

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

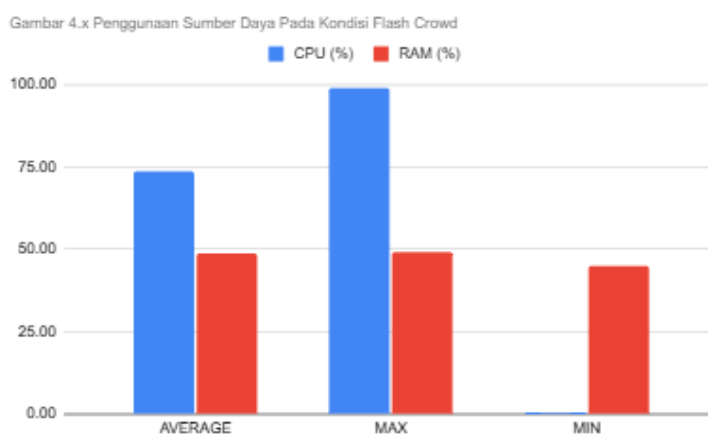
4.4.6 Analisa Data Pengujian Dampak Sistem Deteksi Terhadap Sumber Daya Server

Analisis dampak sistem deteksi terhadap sumber daya *server* dilakukan dengan membandingkan penggunaan CPU dan RAM pada tiga kondisi pengujian sebagai berikut:



Gambar 4.59 Penggunaan Sumber Daya Pada Kondisi Normal

Pada kondisi normal, penggunaan CPU rata-rata sebesar 48,60%, sedangkan penggunaan RAM rata-rata berada pada 47,34%. Nilai tersebut menunjukkan bahwa sistem deteksi dapat berjalan tanpa menyebabkan penggunaan sumber daya yang berlebihan saat kondisi trafik legal berlangsung.



Gambar 4.60 Penggunaan Sumber Daya Pada Kondisi Flash Crowd

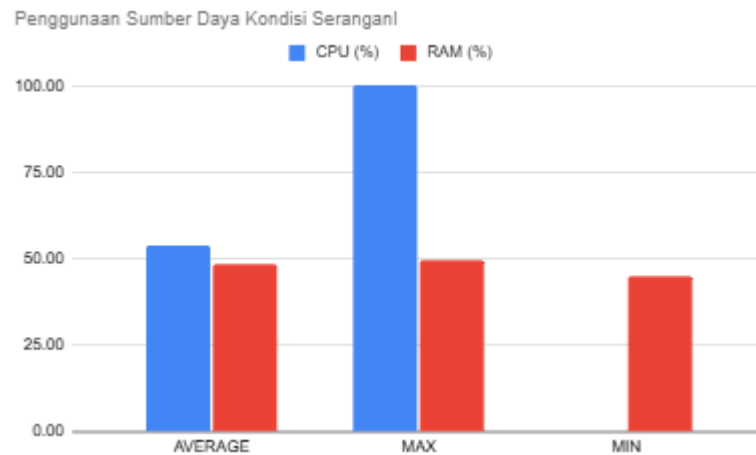
Pada kondisi *flash crowd*, penggunaan CPU meningkat dengan rata-rata 73,56% dan penggunaan maksimum mencapai 98,90%. Peningkatan ini terjadi karena *server* harus memproses jumlah *request* legal yang jauh lebih besar dibandingkan



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 Politeknik Negeri Jakarta

kondisi normal. Meskipun demikian, penggunaan RAM relatif stabil dengan rata-rata 48,88% atau sekitar 1.912,46 MB. Hasil ini menunjukkan bahwa beban utama yang ditimbulkan oleh lonjakan trafik lebih banyak mempengaruhi proses komputasi CPU dibandingkan konsumsi memori.



Gambar 4.61 Penggunaan Sumber Daya Pada Kondisi Serangan

Pada kondisi serangan DDoS, penggunaan CPU rata-rata tercatat sebesar 53,90% dengan puncak penggunaan mencapai 100%, sedangkan penggunaan RAM rata-rata sebesar 48,38% atau sekitar 1.892,90 MB dan maksimum 1.944,09 MB. Berdasarkan data di atas, dampak sistem deteksi terhadap sumber daya server lebih terlihat pada penggunaan CPU dibandingkan RAM. Kondisi *flash crowd* menghasilkan rata-rata CPU tertinggi sebesar 73,56%, sedangkan kondisi serangan menghasilkan rata-rata CPU sebesar 53,90% dengan nilai maksimum mencapai 100%. Hal ini menunjukkan bahwa pada beberapa waktu tertentu, *server* tetap dapat mengalami lonjakan beban komputasi ketika memproses trafik berintensitas tinggi. Sementara itu, penggunaan RAM pada seluruh kondisi relatif stabil, yaitu berada pada rentang rata-rata 47,34% hingga 48,88% dan tidak melebihi 49,70% pada nilai maksimum. Dengan demikian, sistem deteksi tidak menyebabkan peningkatan konsumsi memori yang signifikan, tetapi tetap memerlukan perhatian pada penggunaan CPU saat beban trafik meningkat.



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 Politeknik Negeri Jakarta

BAB V

PENUTUP

5.1 Simpulan

Sistem deteksi serangan DDoS berbasis *entropy* menggunakan parameter K-dinamis dan mekanisme revaluasi IP telah berhasil dibangun dan dijalankan pada *web server* Ubuntu. Pengujian fungsionalitas menunjukkan bahwa sistem mampu memproses data trafik secara *near real-time* melalui pendekatan *sliding window*, menghitung nilai *entropy* dan *normalized entropy*, membentuk *threshold* dinamis, melakukan klasifikasi awal, serta menjalankan revaluasi IP pada trafik yang terdeteksi mencurigakan. Pada kondisi normal dan *flash crowd*, sistem secara umum menghasilkan klasifikasi NORMAL, sedangkan pada kondisi serangan sistem mampu mengidentifikasi fase serangan aktif melalui penurunan nilai *normalized entropy* dan hasil revaluasi IP.

Berdasarkan pengujian performa menggunakan *confusion matrix* pada kondisi serangan, sistem menghasilkan tingkat *accuracy* sebesar 90,36%, *precision* 100%, *recall* 66,67%, dan F1-score 80%. Nilai *recall* sebesar 66,67% menunjukkan bahwa masih terdapat beberapa *window* pada periode aktif serangan yang belum terdeteksi sebagai ATTACK. Hal ini menunjukkan bahwa sistem masih memiliki keterbatasan dalam mendeteksi seluruh *window* serangan, khususnya ketika komposisi trafik pada suatu *window* belum cukup menekan nilai *normalized entropy* hingga melewati *threshold* deteksi.

Analisis penggunaan sumber daya server menunjukkan bahwa penggunaan RAM relatif stabil pada seluruh kondisi pengujian dan berada di bawah 50%. Pada kondisi serangan, penggunaan RAM rata-rata tercatat sebesar 48,38%, sedangkan penggunaan CPU rata-rata sebesar 53,90% dengan nilai maksimum mencapai 100%.



5.2. Saran

Berdasarkan keterbatasan dan hasil yang diperoleh selama proses pengujian sistem deteksi serangan DDoS ini, terdapat beberapa poin penting yang dapat disarankan sebagai acuan untuk pengembangan penelitian selanjutnya guna meningkatkan performa dan fungsionalitas sistem:

1. Pengembangan Skenario dan Jenis Serangan Lain: Penelitian selanjutnya diharapkan dapat menguji efektivitas algoritma menggunakan variasi serangan DDoS lain selain HTTP *Flood*.
2. Mekanisme Mitigasi Otomatis: Sistem deteksi ini dapat diintegrasikan dengan modul mitigasi atau pemblokiran otomatis (*automated firewall blocking*), seperti otomatisasi iptables atau Fail2ban, untuk langsung memblokir alamat IP yang telah dinyatakan positif sebagai penyerang setelah tahap revaluasi.

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Baldini, G. dan Amerini, I. 2022. Online Distributed Denial of Service (DDoS) Intrusion Detection Based on Adaptive Sliding Window and Morphological Fractal Dimension. *Computer Networks*, 210, p.108923. doi:<https://doi.org/10.1016/j.comnet.2022.108923>.
- Catillo, M., Pecchia, A. dan Villano, U. 2022. No more DoS? An empirical study on defense techniques for web server Denial of Service mitigation. *Journal of Network and Computer Applications*, 202, p.103363. doi:<https://doi.org/10.1016/j.jnca.2022.103363>.
- Chandra, A.Y. 2019. Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. *Jurnal Sistem dan Informatika (JSI)*, 14(1), pp.48–56. doi:<https://doi.org/10.30864/jsi.v14i1.248>.
- Cheema, A., Tariq, M., Hafiz, A., Khan, M.M., Ahmad, F. dan Anwar, M. 2022. Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review. *Security and Communication Networks*, 2022(1), pp.1–15. doi:<https://doi.org/10.1155/2022/8379532>.
- Deressu, T.F., Beyene, A.M. dan Gemechu, A.Y. 2026. A Survey of Application Layer Distributed Denial of Service Detection: Approaches, Models, and Datasets. *Artificial Intelligence Review*, 59. doi:<https://doi.org/10.1007/s10462-026-11528-3>.
- Ding, D., Savi, M. dan Siracusa, D. 2021. Tracking Normalized Network Traffic Entropy to Detect DDoS Attacks in P4. *IEEE Transactions on Dependable and Secure Computing*, 19(6), pp.1–1. doi:<https://doi.org/10.1109/tdsc.2021.3116345>.
- Duan, X., Fu, Y. dan Wang, K. 2022. Network traffic anomaly detection method based on multi-scale residual classifier. *Computer Communications*, 198. doi:<https://doi.org/10.1016/j.comcom.2022.10.024>.
- Erbani, J., Portier, P.-É., Egyed-Zsigmond, E. dan Nurbakova, D. 2024. Confusion Matrices: A Unified Theory. *IEEE Access*, [online] 12, pp.181372–181419. doi:<https://doi.org/10.1109/access.2024.3507199>.
- Fan, C., Kaliyamurthy, N.M., Chen, S., Jiang, H., Zhou, Y. dan Campbell, C. 2021. Detection of DDoS Attacks in Software Defined Networking Using



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 Politeknik Negeri Jakarta

Entropy. Applied Sciences, 12(1), p.370.
doi:https://doi.org/10.3390/app12010370.

Goel, M., Singh, S., Garg, A. dan Roy, N.R. 2023. Comparative Study of DDoS Attacks & Tools and Their Analysis. International Conference on IoT, Communication and Automation Technology (ICICAT), [online] 11, pp.1–8. doi:https://doi.org/10.1109/icicat57735.2023.10263744.

Hemmati, Z., Mirjalily, G. dan Mohtajollah, Z. 2021. Entropy-based DDoS Attack Detection in SDN using Dynamic Threshold. 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS), pp.1–8. doi:https://doi.org/10.1109/icspis54653.2021.9729355.

Irawan, B., Rahaman, M., Nur Sheha, K., Erzed, N. dan Herwanto, A. 2025. Evaluating the Effectiveness of Center of Internet Security Benchmark for Hardening Linux Servers Against Cyber Attacks. Journal of Social Research, [online] 4(6), pp.1172–1183. doi:https://doi.org/10.55324/josr.v4i6.2544.

Januhandini, R. dan Oktiawati, U.Y. 2025. Analisis Kinerja Web Server Apache, Nginx, dan Caddy dengan Metode Stress Testing Menggunakan Autocannon. Journal of Internet and Software Engineering, 6(2), pp.96–100. doi:https://doi.org/10.22146/jise.v6i2.15011.

Khumaidi, A. 2021. Simulasi Entropi Shannon, Entropi Renyi, dan informasi pada kasus Spin Wheel. Aksioma: Jurnal Matematika dan Pendidikan Matematika, 12(1), pp.120–128. doi:https://doi.org/10.26877/aks.v12i1.6893.

Kurniawan, A. dan Matin, I.M.M. 2025. Analisis Keamanan Website Kota Depok Menggunakan Metode Vulnerability Assessment. MULTINETICS, 11(1), pp.62–74. doi:https://doi.org/10.32722/multinetics.v11i1.7226.

Nasrulah, D.O., Hertaliando, A.W., Tullah, R. dan Ferawati. 2025. Evaluasi Perbandingan Sistem Operasi Linux untuk Pemanfaatan Client-Server: Studi Kasus Ubuntu dan Debian. MALCOM: Indonesian Journal of Machine Learning and Computer Science, [online] 5(1), pp.242–248. doi:https://doi.org/10.57152/malcom.v5i1.1702.

Park, S., Kim, Y., Choi, H., Kyung, Y. dan Park, J. 2021. HTTP DDoS Flooding Attack Mitigation in Software-Defined Networking. IEICE Transactions on Information and Systems, E104.D(9), pp.1496–1499. doi:https://doi.org/10.1587/transinf.2021edl8022.



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 Politeknik Negeri Jakarta

Pebrianto, J. dan Suryani, V. 2025. Adaptive DDoS Attack Detection: Entropy-Based Model with Dynamic Threshold and Suspicious IP Reevaluation. IEEE Access, 13, pp.55858–55876. doi:<https://doi.org/10.1109/access.2025.3553144>.

Shrivastava, S. dan SB, P. 2020. Comprehensive Review of Load Testing Tools. International Research Journal of Engineering and Technology (IRJET), 07(05).

Solichin, A. dan Nugroho, L. 2024. Deteksi Dini Gangguan Jaringan Distributed Denial Of Service (DDoS) Menggunakan Metode Shannon Entropy Pada Software Defined Network (SDN). Jurnal Teknologi Informasi dan Ilmu Komputer, 11(3), pp.461–474. doi:<https://doi.org/10.25126/jtiik.938188>.

Swaminathan, S. dan Tantri, B.R. 2024. Confusion Matrix-Based Performance Evaluation Metrics. African Journal of Biomedical Research, 27(4s), pp.4023–4031. doi:<https://doi.org/10.53555/ajbr.v27i4s.4345>.

Vishwakarma, R. and Jain, A.K. 2019. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication systems*, [online] 73. doi:10.1007/s11235-019-00599-z.

Yenugula, M., Kodam, R. dan He, D. 2019. Performance and load testing: Tools and challenges. International Journal of Engineering in Computer Science, 1(1), pp.57–62. doi:<https://doi.org/10.33545/26633582.2019.v1.i1a.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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP PENULIS



Safina Ayu Rasya, lahir di Jakarta pada tanggal 16 Agustus 2004. Lulusan dari MAN 13 Jakarta. Sekarang sedang melakukan pendidikan di Politeknik Negeri Jakarta, Jurusan Teknik Informatika dan Komputer, Program Studi Teknik Multimedia dan Jaringan.





- 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 Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1: Hasil Pengujian Fungsionalitas

A. Hasil Pengujian Kondisi Normal

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.99112	0.92594	NORMAL	NULL	NULL	NORMAL
0.98859	0.92983	NORMAL	NULL	NULL	NORMAL
0.97911	0.93287	NORMAL	NULL	NULL	NORMAL
0.97631	0.93558	NORMAL	NULL	NULL	NORMAL
0.97385	0.9382	NORMAL	NULL	NULL	NORMAL
0.97054	0.94056	NORMAL	NULL	NULL	NORMAL
0.96823	0.94282	NORMAL	NULL	NULL	NORMAL
0.96547	0.94531	NORMAL	NULL	NULL	NORMAL
0.95826	0.94744	NORMAL	NULL	NULL	NORMAL
0.95764	0.94987	NORMAL	NULL	NULL	NORMAL
0.95615	0.9525	NORMAL	NULL	NULL	NORMAL
0.95581	0.95532	NORMAL	NULL	NULL	NORMAL
0.95576	0.95818	SUS	6.34466	0.80266	NORMAL
0.95959	0.96137	SUS	5.3375	0.81907	NORMAL
0.95777	0.96366	SUS	5.29829	0.8369	NORMAL
0.95625	0.96552	SUS	5.0904	0.85344	NORMAL
0.95221	0.96747	SUS	6.32464	0.87086	NORMAL
0.95497	0.96711	SUS	5.50161	0.8862	NORMAL
0.95622	0.96667	SUS	5.50154	0.89299	NORMAL
0.95332	0.96434	SUS	5.44645	0.89976	NORMAL
0.95549	0.96256	SUS	5.52668	0.90648	NORMAL
0.9584	0.96105	SUS	5.59578	0.91257	NORMAL
0.96657	0.9551	NORMAL	NULL	NULL	NORMAL
0.96635	0.95627	NORMAL	NULL	NULL	NORMAL
0.96428	0.95722	NORMAL	NULL	NULL	NORMAL
0.96207	0.95828	NORMAL	NULL	NULL	NORMAL



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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.94223	0.95547	SUS	5.20263	0.93989	NORMAL
0.94263	0.95432	SUS	5.03577	0.94353	NORMAL
0.96535	0.9539	NORMAL	NULL	NULL	NORMAL
0.96657	0.9551	NORMAL	NULL	NULL	NORMAL
0.94091	0.95218	SUS	5.29475	0.96132	ATTACK
0.95007	0.95189	SUS	5.56312	0.96429	NORMAL
0.9551	0.95186	NORMAL	NULL	NULL	NORMAL
0.94991	0.95137	SUS	5.54169	0.96852	NORMAL
0.95183	0.95108	NORMAL	NULL	NULL	NORMAL
0.95255	0.95089	NORMAL	NULL	NULL	NORMAL
0.95155	0.95086	NORMAL	NULL	NULL	NORMAL
0.95006	0.95061	SUS	5.54866	0.96873	NORMAL
0.95891	0.95075	NORMAL	NULL	NULL	NORMAL
0.95936	0.95105	NORMAL	NULL	NULL	NORMAL
0.95667	0.95111	NORMAL	NULL	NULL	NORMAL
0.95568	0.95097	NORMAL	NULL	NULL	NORMAL
0.95846	0.95107	NORMAL	NULL	NULL	NORMAL
0.96035	0.95138	NORMAL	NULL	NULL	NORMAL
0.96415	0.95193	NORMAL	NULL	NULL	NORMAL
0.96663	0.95275	NORMAL	NULL	NULL	NORMAL
0.96535	0.9539	NORMAL	NULL	NULL	NORMAL
0.96657	0.9551	NORMAL	NULL	NULL	NORMAL
0.96635	0.95627	NORMAL	NULL	NULL	NORMAL
0.96428	0.95722	NORMAL	NULL	NULL	NORMAL
0.96207	0.95828	NORMAL	NULL	NULL	NORMAL
0.96327	0.95894	NORMAL	NULL	NULL	NORMAL
0.96462	0.95942	NORMAL	NULL	NULL	NORMAL
0.9657	0.96021	NORMAL	NULL	NULL	NORMAL
0.96342	0.96079	NORMAL	NULL	NULL	NORMAL



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 Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.96598	0.96146	NORMAL	NULL	NULL	NORMAL
0.96494	0.96213	NORMAL	NULL	NULL	NORMAL
0.96739	0.963	NORMAL	NULL	NULL	NORMAL
0.96842	0.96347	NORMAL	NULL	NULL	NORMAL
0.96397	0.9637	NORMAL	NULL	NULL	NORMAL
0.96415	0.96408	NORMAL	NULL	NULL	NORMAL

B. Hasil Pengujian Kondisi Flash Crowd

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.98485	0.49802	NORMAL	NULL	NULL	NORMAL
0.9822	0.53023	NORMAL	NULL	NULL	NORMAL
0.96464	0.55892	NORMAL	NULL	NULL	NORMAL
0.95202	0.58664	NORMAL	NULL	NULL	NORMAL
0.95174	0.61427	NORMAL	NULL	NULL	NORMAL
0.94494	0.64135	NORMAL	NULL	NULL	NORMAL
0.94725	0.6712	NORMAL	NULL	NULL	NORMAL
0.94669	0.70101	NORMAL	NULL	NULL	NORMAL
0.94755	0.73058	NORMAL	NULL	NULL	NORMAL
0.94667	0.75965	NORMAL	NULL	NULL	NORMAL
0.94667	0.78864	NORMAL	NULL	NULL	NORMAL
0.94678	0.8172	NORMAL	NULL	NULL	NORMAL
0.94618	0.84472	NORMAL	NULL	NULL	NORMAL
0.94828	0.87144	NORMAL	NULL	NULL	NORMAL
0.94558	0.89601	NORMAL	NULL	NULL	NORMAL
0.94241	0.90532	NORMAL	NULL	NULL	NORMAL
0.94228	0.90693	NORMAL	NULL	NULL	NORMAL
0.94153	0.90631	NORMAL	NULL	NULL	NORMAL
0.94654	0.95373	SUS	0.94585	0.89599	NORMAL
0.94407	0.95094	SUS	0.94845	0.89689	NORMAL



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 Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.95093	0.94924	NORMAL	NULL	NULL	NORMAL
0.94685	0.94748	SUS	0.95296	0.89965	NORMAL
0.95152	0.94682	NORMAL	NULL	NULL	NORMAL
0.95488	0.94697	NORMAL	NULL	NULL	NORMAL
0.95288	0.94702	NORMAL	NULL	NULL	NORMAL
0.94849	0.9472	NORMAL	NULL	NULL	NORMAL
0.94811	0.94724	NORMAL	NULL	NULL	NORMAL
0.94731	0.94728	NORMAL	NULL	NULL	NORMAL
0.94759	0.94728	NORMAL	NULL	NULL	NORMAL
0.94824	0.94736	NORMAL	NULL	NULL	NORMAL
0.9473	0.94739	SUS	0.94922	0.91052	NORMAL
0.94946	0.94752	NORMAL	NULL	NULL	NORMAL
0.95061	0.94774	NORMAL	NULL	NULL	NORMAL
0.95249	0.94795	NORMAL	NULL	NULL	NORMAL
0.94903	0.94813	NORMAL	NULL	NULL	NORMAL
0.94691	0.94835	SUS	0.95273	0.91107	NORMAL
0.94818	0.94865	SUS	0.95049	0.92404	NORMAL
0.9472	0.94893	SUS	0.94816	0.93229	NORMAL
0.94919	0.94906	NORMAL	NULL	NULL	NORMAL
0.94697	0.94921	SUS	0.95618	0.93106	NORMAL
0.94886	0.9491	SUS	0.95068	0.9308	NORMAL
0.94763	0.94914	SUS	0.95419	0.93137	NORMAL
0.94275	0.9487	SUS	0.95264	0.94033	NORMAL
0.94587	0.94825	SUS	0.9467	0.94982	ATTACK
0.9474	0.94798	SUS	0.9497	0.95724	ATTACK
0.95336	0.94822	NORMAL	NULL	NULL	NORMAL
0.95309	0.94847	NORMAL	NULL	NULL	NORMAL
0.95303	0.94876	NORMAL	NULL	NULL	NORMAL
0.95039	0.9489	NORMAL	NULL	NULL	NORMAL



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 Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.95138	0.94905	NORMAL	NULL	NULL	NORMAL
0.95017	0.9492	NORMAL	NULL	NULL	NORMAL
0.9515	0.9493	NORMAL	NULL	NULL	NORMAL
0.95331	0.94943	NORMAL	NULL	NULL	NORMAL
0.95213	0.94941	NORMAL	NULL	NULL	NORMAL
0.95527	0.94972	NORMAL	NULL	NULL	NORMAL
0.95589	0.95017	NORMAL	NULL	NULL	NORMAL
0.95667	0.95059	NORMAL	NULL	NULL	NORMAL
0.97906	0.95214	NORMAL	NULL	NULL	NORMAL
0.96042	0.95252	NORMAL	NULL	NULL	NORMAL

C. Hasil Pengujian Kondisi Serangan

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.9954	0.94079	NORMAL	NULL	NULL	NORMAL
0.97527	0.94317	NORMAL	NULL	NULL	NORMAL
0.94529	0.94365	NORMAL	NULL	NULL	NORMAL
0.93845	0.94376	SUS	0.91243	0.78335	NORMAL
0.92319	0.94318	SUS	0.90328	0.78917	NORMAL
0.91914	0.94257	SUS	0.908	0.7943	NORMAL
0.91841	0.942	SUS	0.87083	0.80088	NORMAL
0.91914	0.94148	SUS	0.88375	0.80486	NORMAL
0.91107	0.94025	SUS	0.88377	0.80927	NORMAL
0.91728	0.93946	SUS	0.89863	0.81492	NORMAL
0.92538	0.93939	SUS	0.91533	0.82052	NORMAL
0.92419	0.93955	SUS	0.90825	0.83998	NORMAL
0.9181	0.93956	SUS	0.89633	0.85835	NORMAL
0.88597	0.93797	SUS	0.86168	0.87562	ATTACK
0.76335	0.93023	SUS	0.56454	0.88932	ATTACK
0.70897	0.91945	SUS	0.51921	0.88779	ATTACK



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 Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.66174	0.90539	SUS	0.44851	0.86712	ATTACK
0.6346	0.88917	SUS	0.37374	0.84332	ATTACK
0.63074	0.8707	SUS	0.41137	0.8162	ATTACK
0.63588	0.85249	SUS	0.46037	0.79066	ATTACK
0.6438	0.83491	SUS	0.52255	0.76825	ATTACK
0.6411	0.8182	SUS	0.54946	0.74827	ATTACK
0.65542	0.8037	SUS	0.61496	0.73064	ATTACK
0.65343	0.78945	SUS	0.71615	0.7152	NORMAL
0.6507	0.77583	SUS	0.60753	0.70539	ATTACK
0.64471	0.76211	SUS	0.64397	0.69061	ATTACK
0.65455	0.74891	SUS	0.59171	0.67741	ATTACK
0.67908	0.73691	SUS	0.64137	0.66345	ATTACK
0.69025	0.72588	SUS	0.64437	0.65134	ATTACK
0.68742	0.71439	SUS	0.6361	0.63938	ATTACK
0.68453	0.70236	SUS	0.64021	0.62625	NORMAL
0.65796	0.68906	SUS	0.6155	0.61251	NORMAL
0.66968	0.67665	SUS	0.63983	0.59788	NORMAL
0.66582	0.66566	NORMAL	NULL	NULL	NORMAL
0.6642	0.66071	NORMAL	NULL	NULL	NORMAL
0.65441	0.65799	SUS	0.62208	0.58508	NORMAL
0.66099	0.65795	NORMAL	NULL	NULL	NORMAL
0.58823	0.65563	SUS	0.65886	0.57312	NORMAL
0.50506	0.64934	SUS	0.83899	0.57785	NORMAL
0.51735	0.64341	SUS	0.8392	0.59383	NORMAL
0.52911	0.63767	SUS	0.84093	0.61336	NORMAL
0.53952	0.63259	SUS	0.85234	0.63672	NORMAL
0.54806	0.62722	SUS	0.85438	0.65877	NORMAL
0.57916	0.6235	SUS	0.77159	0.67847	NORMAL
0.61183	0.62156	SUS	0.90037	0.69092	NORMAL



© Hak Cipta milik 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 Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.68219	0.62342	NORMAL	NULL	NULL	NORMAL
0.74057	0.62771	NORMAL	NULL	NULL	NORMAL
0.87597	0.6375	NORMAL	NULL	NULL	NORMAL
0.91624	0.64876	NORMAL	NULL	NULL	NORMAL
0.90872	0.6598	NORMAL	NULL	NULL	NORMAL
0.91077	0.6711	NORMAL	NULL	NULL	NORMAL
0.90669	0.68352	NORMAL	NULL	NULL	NORMAL
0.90607	0.69533	NORMAL	NULL	NULL	NORMAL
0.90976	0.70751	NORMAL	NULL	NULL	NORMAL
0.9076	0.71967	NORMAL	NULL	NULL	NORMAL
0.90664	0.73227	NORMAL	NULL	NULL	NORMAL
0.91214	0.74483	NORMAL	NULL	NULL	NORMAL
0.91342	0.76109	NORMAL	NULL	NULL	NORMAL
0.9148	0.78158	NORMAL	NULL	NULL	NORMAL
0.91448	0.80146	NORMAL	NULL	NULL	NORMAL
0.91681	0.82086	NORMAL	NULL	NULL	NORMAL
0.91628	0.83972	NORMAL	NULL	NULL	NORMAL
0.91532	0.85812	NORMAL	NULL	NULL	NORMAL
0.91771	0.87507	NORMAL	NULL	NULL	NORMAL
0.91798	0.89042	NORMAL	NULL	NULL	NORMAL
0.91995	0.90234	NORMAL	NULL	NULL	NORMAL
0.92288	0.9115	NORMAL	NULL	NULL	NORMAL
0.92581	0.914	NORMAL	NULL	NULL	NORMAL
0.92478	0.91442	NORMAL	NULL	NULL	NORMAL
0.92471	0.91522	NORMAL	NULL	NULL	NORMAL
0.92415	0.91589	NORMAL	NULL	NULL	NORMAL
0.9164	0.91638	NORMAL	NULL	NULL	NORMAL
0.91631	0.91689	SUS	0.89553	0.70836	NORMAL
0.90938	0.91687	SUS	0.8878	0.72239	NORMAL



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

normalized_entropy	threshold_stage1	stage1_result	normalized_esip	threshold_nesip	final_result
0.90393	0.91669	SUS	0.88607	0.73096	NORMAL
0.89963	0.91633	SUS	0.85702	0.74484	NORMAL
0.89877	0.91566	SUS	0.8412	0.75545	NORMAL
0.89597	0.91478	SUS	0.82917	0.76791	NORMAL
0.91213	0.91464	SUS	0.89939	0.77721	NORMAL
0.91907	0.91485	NORMAL	NULL	NULL	NORMAL
0.91343	0.91466	SUS	0.88694	0.78955	NORMAL
0.96218	0.91684	NORMAL	NULL	NULL	NORMAL
0.96507	0.91903	NORMAL	NULL	NULL	NORMAL

Lampiran 6: Dokumentasi Eksekusi Pengujian

- Terminal Server Target

```

server1@ubuntu-server1: ~/ddos_detection
(venv) server1@ubuntu-server1:~/ddos_detection$

server2@server2: ~
(venv) server2@server2:~$

main@ubuntu-main: ~/ddos_skripsi/ddos_detection/core
(classification_id, entropy_esip, normalized_esip, mean_nesip, stddev_nesip,
threshold_nesip, final_result)
VALUES (?, ?, ?, ?, ?, ?, ?)
);
$stmt->bind_param("idddddd", $inserted_id, $entropy_esip, $normalized_esip, $mean_nesip, $stddev_nesip, $threshold_nesip, $final_result);
$stmt->execute() or die($stmt->error);

// update hasil klasifikasi final
$stmt = $conn->prepare("
UPDATE classification
SET final_result = ?
WHERE id = ?
");
$stmt->bind_param("si", $final_result, $inserted_id);
$stmt->execute() or die($stmt->error);

// performance log (lama proses 1x iterasi main-function dalam microsecond)
savePerformanceLog(
    $conn,
    $window_id,
    $current_timestamp,
    $start_time
);
echo "-----\n";
$conn->close();
?>main@ubuntu-main:~/ddos_skripsi/ddos_detection/core$
main@ubuntu-main:~/ddos_skripsi/ddos_detection/core$

$
ilocust_fc.sh runilocust_normal.sh venv
  
```



- Terminal Server Pengirim Trafik

```

server1@ubuntu-server1: ~/ddos_detection
POST /login.php/login.php 111 52 373 93 | 4.70 0.00 69 0(0.00%)
Aggregated 111 52 373 93 | 4.70 0.00 69 0(0.00%)
Name Avg Min Max Median req/s failures/s # reqs # fails
Type ures/s
POST /login.php/login.php 110 52 373 96 | 4.80 0.00 79 0(0.00%)
Aggregated 110 52 373 96 | 4.80 0.00 79 0(0.00%)

server2@server2: ~/ddos_detection
POST /login.php/login.php 20 0(0.00%) | 170 75 443 120
Aggregated 20 0(0.00%) | 170 75 443 120 | 2.00
Type Name # reqs # fails | Avg Min Max Med | req/s fail
ures/s
POST /login.php/login.php 32 0(0.00%) | 136 51 443 99
Aggregated 32 0(0.00%) | 136 51 443 99 | 2.83

server3@server3: ~/ddos_detection
POST /login.php/login.php 33 0(0.00%) | 154 59 381 150
Aggregated 33 0(0.00%) | 154 59 381 150 | 3.67
Type Name # reqs # fails | Avg Min Max Med | req/s fail
ures/s
POST /login.php/login.php 42 0(0.00%) | 143 59 381 130
Aggregated 42 0(0.00%) | 143 59 381 130 | 4.00

server4@server4: ~/ddos_detection
Aggregated 4 0(0.00%) | 204 85 303 170 | 0.00
Type Name # reqs # fails | Avg Min Max Med | req/s fail
ures/s
POST /login.php/login.php 12 0(0.00%) | 171 71 303 150
Aggregated 12 0(0.00%) | 171 71 303 150 | 2.00
[2026-06-07 02:34:40,540] server4/INFO/locust.runners: All users spawned: {"Websit
eUser": 10} (10 total users)

```

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA