Rancang Bangun Aplikasi Deteksi *Pitch* Suara Berbasis Android Dengan Algoritma *Fast Fourier Transform Dan Convolutional Neural Network*

Daffa Ian Nabil, Mera Kartika Delimayanti
Teknik Informatika, Teknik Informatika dan Komputer, Politeknik Negeri Jakarta
Depok, Jawa Barat

daffa.ian.nabil.tik22@mhsw.pnj.ac.id, mera.kartika@tik.pnj.ac.id

Abstract - The music creation process often begins with humming as a form of initial idea expression. However, for both novice and professional composers, manually transcribing humming sounds into digital notation is a complex and time-consuming process. This study aims to design and build a mobile-based pitch detection application on the Android platform capable of converting human voice input into the Musical Instrument Digital Interface (MIDI) format. The developed system uses a hybrid approach combining digital signal processing and Deep Learning. The Fast Fourier Transform (FFT) or Short-Time Fourier Transform (STFT) algorithm is applied to extract monophonic voice signal features into spectrum images (spectrograms). Furthermore, a Convolutional Neural Network (CNN) is used to classify the spectrogram images to precisely recognize pitch levels. The model was trained using the MIR-QBSH dataset and implemented into a mobile device using the LiteRT framework for computational efficiency on the user side. The final result of this research is an application that can help musicians document melodic ideas automatically into MIDI files, which can then be further edited in audio production software.

Keywords: Android, Convolutional Neural Network (CNN), Fast Fourier Transform (FFT), MIDI, Pitch Detection

Abstrak*-- Proses penciptaan musik sering kali diawali dengan senandung (humming) sebagai bentuk ekspresi ide awal. Namun, bagi komposer pemula maupun profesional, melakukan transkripsi manual dari suara senandung menjadi notasi digital merupakan proses yang rumit dan memakan waktu. Penelitian ini bertujuan untuk merancang dan membangun aplikasi deteksi pitch suara berbasis mobile pada platform Android yang mampu mengonversi input suara manusia menjadi format Musical Instrument Digital Interface (MIDI). Sistem yang dibangun menggunakan pendekatan hibrida yang menggabungkan Digital Signal Processing dan Deep Learning. Algoritma Fast Fourier Transform (FFT) atau Short-Time Fourier Transform (STFT) diterapkan untuk mengekstraksi fitur sinyal suara monofonik menjadi citra spektrum (spektrogram). Selanjutnya, Convolutional Neural Network (CNN) digunakan untuk mengklasifikasikan citra spektrogram tersebut guna mengenali tinggi rendahnya pitch secara presisi. Model dilatih menggunakan dataset MIR-QBSH dan diimplementasikan ke dalam perangkat mobile menggunakan framework LiteRT untuk efisiensi komputasi di sisi pengguna. Hasil akhir dari penelitian ini adalah aplikasi yang dapat membantu musisi mendokumentasikan ide melodi secara otomatis ke dalam file MIDI, yang kemudian dapat disunting lebih lanjut pada perangkat lunak produksi audio.

Kata kunci: Android, Convolutional Neural Network (CNN), Fast Fourier Transform (FFT), MIDI, Pitch Detection

I. PENDAHULUAN

Musik merupakan cabang seni yang masyarakat gunakan sebagai media ekspresi yang mempunyai kapasitas untuk menyampaikan dan mengatur emosi [1]. Menurut laporan IFPI Global Music Report 2025 terdapat 752 miliar akun pengguna yang aktif melakukan *subscription* untuk mendengarkan musik [2]. Seorang yang menciptakan atau mengarang ide karya seni musik menjadi bentuk komposisi disebut komposer [3]. Untuk mendapatkan ide awal dalam menciptakan musik komposer dapat menggunakan teknik *humming* [4]. Namun, komposer pemula maupun professional, untuk melakukan transkripsi dari suara *humming* bukanlah hal yang mudah dan memakan cukup banyak waktu [5]. Masalah ini membutuhkan solusi teknologi yang dapat menjembatani *input* suara *humming* menjadi data digital yang akurat.

Untuk mendeteksi *pitch* dari data suara dapat dilakukan dengan algoritma *Fast Fourier Transform* (FFT) [6]. Algoritma *Fast Fourier Transform* (FFT) bekerja dengan mengkonversi sinyal suara dalam domain waktu menjadi sinyal suara dalam domain frekuensi [7]. Proses konversi tersebut memungkinkan komputer untuk memproses data audio mentah dengan kompleksitas komputasi yang efisien [8]. *Fast Fourier Transform* (FFT) sudah terbukti handal dan akurat untuk suara alami termasuk suara manusia [9]. *Fast Fourier Transform* (FFT) digunakan untuk ekstraksi fitur yang berupa citra spektrum berbentuk spektogram [10].

Setelah mendapatkan citra spektrum melalui algoritma *Fast Fourier Transform* (FFT), *Convolutional Neural Network* (CNN) diterapkan untuk meningkatkan akurasi klasifikasi *pitch*. CNN memiliki kemampuan unggul dalam mengekstraksi fitur pola dari representasi citra spektrum (*spectrogram*) yang dihasilkan oleh FFT [10]. Dalam skema ini, FFT bertugas mengubah sinyal suara menjadi citra spektrm, yang kemudian dipelajari oleh CNN untuk mengenali tinggi rendahnya *pitch* secara presisi [11]. Integrasi antara ekstraksi fitur FFT dan klasifikasi CNN terbukti memberikan hasil pengenalan *pitch* yang lebih akurat dibandingkan metode konvensional [10].

Penerapan teknologi ini pada platform berbasis *mobile* memberikan keuntungan fleksibilitas yang tinggi karena pengguna dapat merekam ide melodi kapan saja dan di mana saja melalui perangkat telepon pintar [12]. Hasil deteksi *pitch* dari algoritma FFT selanjutnya dapat dikonversi menjadi format *Musical Instrument Digital Interface* (MIDI) untuk kebutuhan dokumentasi musik digital [16]. Format MIDI ini memungkinkan data notasi suara yang telah diproses untuk disunting kembali atau digunakan sebagai *controller virtual* pada perangkat lunak produksi audio [13]. Oleh karena itu, integrasi antara pemrosesan sinyal audio dan aplikasi *mobile* menjadi solusi efisien untuk membantu komposer mendokumentasikan ide.

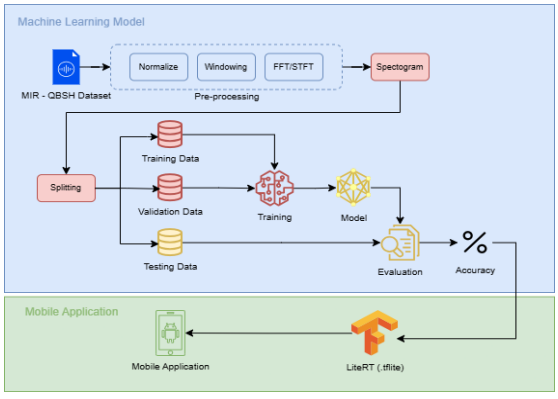
Penelitian ini merancang bangun aplikasi deteksi *pitch* suara berbasis aplikasi yang dapat dijalankan di perangkat android dengan menggunakan algoritma *Fast Fourier Transform* (FFT) dan *Convolutional Neural Network* (CNN). Penulis membangun sebuah sistem yang menerima *input* suara pengguna melalui *file* audio ke aplikasi. Aplikasi ini kemudian memproses sinyal suara tersebut untuk mendeteksi *pitch* secara akurat. Hasil akhir penelitian ini merancang dan membangun aplikasi berbasis *mobile* yang dapat digunakan untuk mendeteksi *pitch* suara dan melakukan konversi data suara tersebut ke *file Musical Instrument Digital Interface* (MIDI).

II. METODOLOGI PENELITIAN

Penelitian ini menerapkan pendekatan kuantitatif yang dikombinasikan dengan metode eksperimental. Pendekatan kuantitatif dipilih karena penelitian ini berfokus pada pengukuran data secara objektif, yaitu mengukur tingkat akurasi, presisi, dan error dari sistem yang dibangun. Metode eksperimental diterapkan untuk melakukan serangkaian uji coba terhadap kinerja model dalam mengenali pitch suara manusia.

A. Perancangan Model

Model *Convolutional Neural Network* (CNN) merupakan model yang menerima citra sebagai *input* dan teks sebagai *output*. Model deteksi *pitch* dalam penelitian ini merupakan model yang menerima sinyal suara sebagai *input* serta menghasilkan notasi musik digital sebagai *output*. Model jenis ini menggabungkan *Digital Signal Processing* dan *Deep Learning*. Pada penelitian ini, algoritma *Fast Fourier Transform* (FFT) digunakan untuk mengubah suara menjadi citra spektrogram, yang kemudian akan diproses oleh *Convolutional Neural Network* (CNN) untuk diklasifikasikan menjadi kelas nada.



Gambar 1. Alur Rancangan Penelitiian

B. Pengumpulan dan Pre-processing Dataset

Dataset yang digunakan dalam penelitian adalah MIR-QBSH yang dikembangkan oleh Prof. Jyh-Shing Roger Jang beserta tim peneliti dari *Music Information Retrieval* (MIR) Lab, National Tsing Hua University (NTHU), Taiwan [14]. Setiap data audio di dalam *dataset* ini memiliki pasangan label

pitch dalam format .pv. *Dataset* dieksplorasi dan dipilah agar hanya suara laki-laki saja yang diambil dengan total 3279 *file* audio serta 3279 *file* anotasi. Data suara dari *dataset* harus melalui tahap pra-pemrosesan (*pre-processing*) agar dapat dikenali oleh model. Tujuan utama tahap ini adalah mengubah sinyal audio satu dimensi menjadi representasi visual (*spectrogram*).

Gambar 2. Data *Pre-Processing*

Data audio dibaca menggunakan *library soundfile* dan librosa, kemudian *sample rate* diatur sesuai dengan *dataset* yaitu sebesar 8000 Hz. Selanjutnya, dilakukan proses ekstraksi fitur terhadap sinyal audio tersebut menggunakan metode *Short-Time Fourier Transform* (STFT). Parameter pengaturan yang digunakan untuk proses STFT ini adalah *n_fft* = 4096 dan *hop_length* = 256. Hasil ekstraksi STFT tersebut kemudian dikonversi ke dalam skala desibel menggunakan fungsi librosa.amplitude_to_db dengan parameter *ref=np.max* untuk normalisasi. Selanjutnya, hanya bagian rentang *pitch* yang sesuai dengan batasan saja yaitu C3 sampai B5 (37 kelas) yang digunakan sebagai *input*.

C. Arsitektur Convolutional Neural Network (CNN)

Model *deep learning* yang digunakan pada penelitian ini adalah model *Convolutional Neural Network* (CNN). CNN digunakan karena mampu mempelajari pola lokal pada fitur spektrogram audio. Arsitektur model terdiri dari beberapa *convolution layer*, *batch normalization*, *max pooling*, *flatten layer*, *dense layer*, *dropout*, dan *output layer*. *Output layer* menggunakan aktivasi *softmax* dengan jumlah kelas 37. Kelas 0 digunakan untuk *silent*, sedangkan kelas 1 sampai 36 digunakan untuk merepresentasikan *pitch* pada rentang C3 sampai B5. Model dikompilasi menggunakan *optimizer* Adam, *loss function* *sparse_categorical_crossentropy*, dan metrik *accuracy*.

D. Pembagian *Dataset* dan Evaluasi

Dataset yang telah diproses dibagi menggunakan metode *5-Fold Cross Validation* dengan proporsi pembagian data latih sebesar 72%, data validasi 18%, dan data *testing* 10%. Metode ini digunakan agar evaluasi model lebih stabil dan tidak bergantung pada satu pembagian data saja. Evaluasi kinerja model *deep learning* dilakukan menggunakan metrik standar klasifikasi *machine learning*, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Ketepatan prediksi model dalam mengenali setiap kelas nada juga dianalisis lebih lanjut melalui pemetaan *Confusion Matrix* untuk memvisualisasikan distribusi kesalahan klasifikasi..

E. Integrasi dan Pengembangan Aplikasi Mobile

Model *Convolutional Neural Network* (CNN) yang telah dilatih dikonversi ke format .tflite dan diintegrasikan ke dalam aplikasi untuk melakukan inferensi (klasifikasi *pitch*) secara *offline* di perangkat pengguna. *Output* dari proses inferensi ini kemudian diproses untuk digabungkan menjadi sebuah *file* MIDI berdasarkan *hop_length*, *sampling rate*, dan tempo yang diberikan pengguna.

III. HASIL DAN PEMBAHASAN

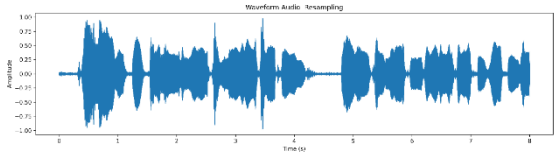
A. Ekstraksi Fitur dan Pemrosesan Data

Tahap *pre-processing* dilakukan untuk mengolah *dataset* sebelum dilatih menggunakan arsitektur *deep learning Convolutional Neural Network* (CNN). Tahap awal yang dilakukan adalah proses pelabelan (*labelling*) yang bertujuan untuk membaca anotasi MIDI dari data audio dan label *pitch*, serta membatasi rentang nada yang digunakan sesuai target penelitian, yaitu C3 sampai B5. *Pitch* yang berada di luar rentang nada tersebut dipetakan menjadi kelas *silent* (kelas 0). Untuk menjaga keseimbangan distribusi data saat pelatihan model, kelas *silent* dibatasi maksimal 30.000 data, sedangkan kelas *pitch* C3 (MIDI 48 / kelas 1) dibatasi maksimal 45.000 data. *Dataset* yang digunakan terdiri dari *file* audio berformat .wav dengan durasi 8 detik dan *sample rate* 8000 Hz, serta *file* anotasi label *pitch* berformat .pv yang memiliki panjang 250 data *pitch*. Setiap *file* audio dicari pasangan labelnya berdasarkan kesamaan nama *file*. Apabila *file* label tidak ditemukan, maka data tersebut dilewati. Untuk menyelaraskan dimensi antara sinyal audio dan label *pitch*, dilakukan perhitungan *window hop length* menggunakan rumus persamaan berikut:

$$H = \frac{f_s \times T}{N}$$

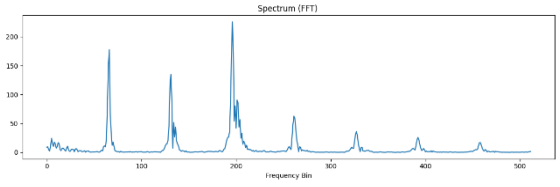
di mana H merepresentasikan *hop length*, f_s adalah *sampling rate* (8000 Hz), T adalah durasi audio (8 detik), dan N merupakan jumlah anotasi *pitch* (250).

Proses *labelling* dilanjutkan dengan membaca *file* .pv menggunakan fungsi *np.loadtxt* dan membulatkan nilai *pitch* menjadi bilangan bulat agar sesuai dengan format klasifikasi kelas nada. Label *pitch* kemudian diselaraskan dengan panjang fitur audio agar jumlah *frame* fitur dan label sejajar. Setelah pemetaan kelas *silent* dan pembatasan jumlah sampel pada kelas dominan selesai dilakukan, tahap ini menghasilkan sebuah *array* label akhir bernama *final_pv* yang berisi pengkategorian kelas *silent* dan kelas *pitch* sebagai target pelatihan model. Selanjutnya, dilakukan ekstraksi fitur pada setiap audio yang memiliki pasangan label dengan membaca *file* menggunakan *library soundfile*. Apabila audio memiliki format stereo (lebih dari satu *channel*), sinyal dikonversi menjadi monofonik dengan mengambil nilai rata-rata antar *channel*, kemudian disamakan frekuensi samplingnya menjadi 8000 Hz menggunakan fungsi *librosa.resample*.



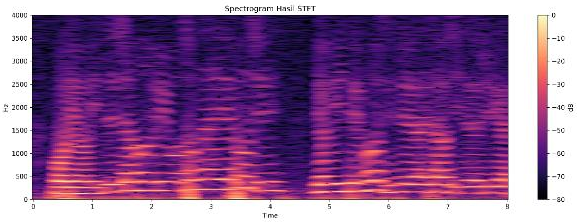
Gambar 3. *Waveform Audio Dataset*

Setelah format dan spesifikasi audio diseragamkan, ekstraksi fitur dilakukan menggunakan metode *Short-Time Fourier Transform* (STFT) untuk mengamati perubahan frekuensi terhadap waktu. Parameter STFT yang diterapkan adalah *N_FFT* = 4096 dan *hop length* = 256. Output awal dari proses STFT ini berupa matriks FFT berdimensi 2049 × T, di mana 2049 merupakan jumlah komponen frekuensi (*frequency bin*) dan T adalah jumlah *frame* waktu. Dari matriks tersebut dapat diekstrak sebuah vektor FFT yang menunjukkan distribusi energi pada domain frekuensi untuk satu *frame* waktu tertentu.

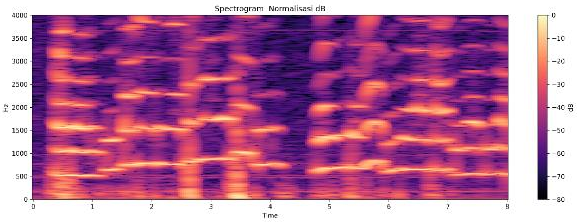


Gambar 4. FFT Vector

Nilai *magnitude* yang dihasilkan dari proses STFT kemudian dikonversi ke dalam skala desibel (dB) menggunakan fungsi *librosa.amplitude_to_db* dengan parameter *ref=np.max*. Langkah normalisasi amplitudo ini bertujuan untuk mengubah rentang energi menjadi relatif terhadap amplitudo maksimum, sehingga rentang nilai fitur menjadi jauh lebih stabil dan memudahkan model CNN dalam mempelajari pola frekuensi. Hasil normalisasi ini menghasilkan representasi waktu-frekuensi visual yang lebih jelas dalam bentuk spektrogram.



Gambar 5. *Spectrogram* Hasil STFT



Gambar 6. *Spectrogram* Setelah Normalisasi dB

Setelah proses normalisasi selesai, dilakukan reduksi dimensi dengan hanya mengambil 512 komponen frekuensi pertama sebagai fitur masukan model CNN, sehingga ukuran akhir matriks fitur menjadi 512 × T. Pemangkasan dimensi ini dipilih karena rentang 512 komponen frekuensi pertama telah sepenuhnya mampu merepresentasikan frekuensi fundamental suara manusia yang dibutuhkan, sekaligus menekan kompleksitas komputasi agar ringan saat dijalankan di perangkat *mobile*. Tahap akhir dari pemrosesan ini adalah menyimpan pasangan fitur dan label ke dalam format *array biner* NumPy (.npy), dengan akhiran _X.npy untuk data fitur dan _y.npy untuk data label. Penyimpanan dalam format ini dilakukan agar proses pembacaan data dan pembagian *dataset* saat pelatihan model *deep learning* dapat berjalan secara cepat dan efisien.

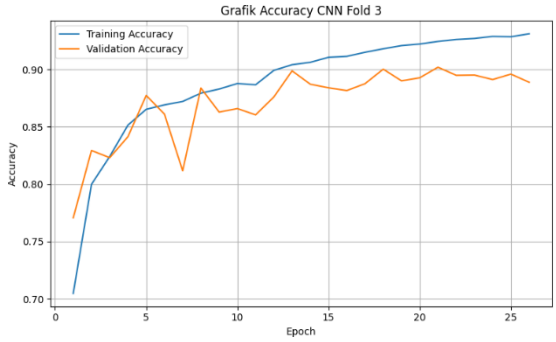
B. Hasil Pelatihan Model Deep Learning

Model dilatih dengan menggunakan metode *5-Fold Cross Validation*. Pada setiap *fold*, model dievaluasi untuk melihat tingkat akurasi generalisasinya terhadap data validasi. *Callback Early Stopping* diaktifkan untuk mencegah *overfitting*, sementara *ReduceLROnPlateau* digunakan untuk menyesuaikan *learning rate* ketika *loss* tidak mengalami perbaikan.

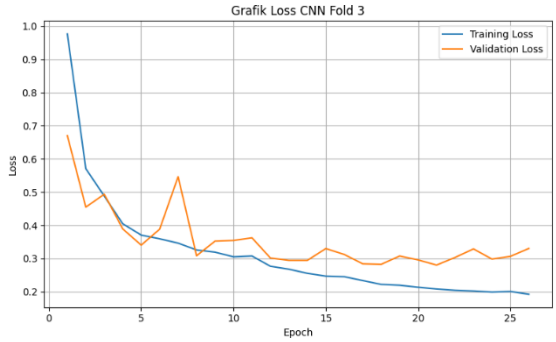
Tabel 1. Hasil 5-Fold Cross validation

Fold	***Validation Accuracy***
1	0.8813
2	0.8378
3	0.9018
4	0.8119
5	0.8553
Rata - Rata	0.8576
Standar Deviasi	0.0316

Dari hasil yang terlihat pada tabel 1, *Validation Accuracy* tertinggi diperoleh pada Fold 3 dengan nilai sebesar 90,18%. Rata-rata *Validation Accuracy* dari seluruh *Fold* adalah 85,76%, dengan standar deviasi sebesar 0,0316. Nilai standar deviasi yang relatif kecil menunjukkan bahwa performa model pada setiap *fold* cukup stabil.



Gambar 7. Grafik Accuracy Model CNN



Gambar 8. Grafik Loss Model CNN

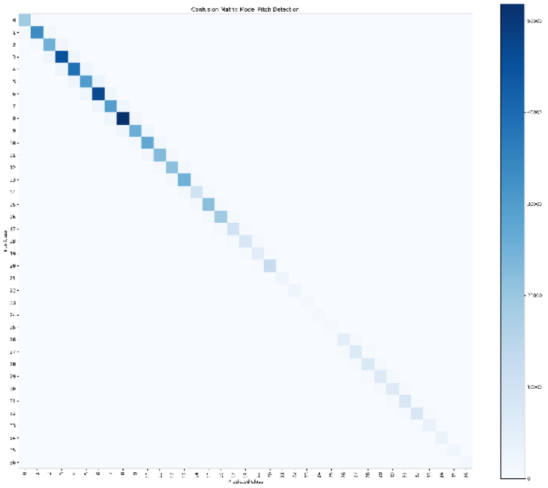
Dari pemantauan grafik *accuracy*, nilai *training accuracy* dan *validation accuracy* pada gambar 7 dan 8 mengalami peningkatan dan berada pada jarak yang relatif dekat, menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan. Fluktuasi pada *validation loss* menunjukkan sedikit ketidakstabilan pada data validasi, namun model dapat dikatakan telah mengalami proses pembelajaran yang baik.

C. Hasil Classification Report dan Confusion Matrix

Evaluasi kinerja model *deep learning* yang telah dilatih dilakukan secara menyeluruh dengan menganalisis *Classification Report* dan *Confusion Matrix*. *Classification Report* digunakan untuk menampilkan rincian metrik evaluasi standar, seperti nilai *precision*, *recall*, dan *F1-score* dari model terbaik pada setiap kelas nada. Sementara itu, *Confusion Matrix* dimanfaatkan untuk memvisualisasikan distribusi prediksi benar maupun salah, sehingga pola kesalahan klasifikasi antar kelas *pitch* dapat diidentifikasi secara lebih jelas.

Tabel 2. Ringkasan Classification Report Model

Metrik Evaluasi	Nilai
Accuracy	0,8817
Precision	0,8632
Recall	0,8545
F1-score	0,8582



Gambar 9. *Confusion Matrix*

Tabel 2 menunjukkan bahwa model mampu melakukan klasifikasi *pitch* dengan tingkat ketepatan (*accuracy*) sebesar 88,17%. Nilai *F1-score* sebesar 0,8582 menunjukkan keseimbangan yang baik antara *Precision* dan *Recall*. Berdasarkan *Confusion Matrix*, sebagian besar hasil prediksi klasifikasi *pitch* menempati garis diagonal utama,

membuktikan kemampuan model mengenali banyak kelas *pitch* dengan akurat. Beberapa kesalahan klasifikasi (*false positive* / *false negative*) yang muncul mayoritas terjadi pada nada-nada tinggi atau kelas *pitch* yang memiliki kedekatan karakteristik frekuensi.

D. Hasil Konversi ke Format MIDI

Setelah model menghasilkan keluaran berupa urutan klasifikasi *pitch* pada setiap *frame*, hasil tersebut diproses lebih lanjut untuk digabungkan (*merging*) menjadi representasi not MIDI berdasarkan durasi ketukan tempo. Proses konversi ini bertujuan untuk menerjemahkan data prediksi *frame* per detik menjadi sebuah *file* musik digital yang terstruktur dan dapat disunting. Untuk mengukur tingkat keberhasilan sistem dalam menerjemahkan audio menjadi notasi digital, pengujian konversi MIDI dilakukan pada 10 *file testing* guna membandingkan MIDI hasil inferensi aplikasi dengan MIDI *ground truth* dari *dataset*.

Tabel 3. Hasil Pengujian Konversi Midi

File	GT	TP	FP	FN	TN	*Overall Accuracy*
00015.wav	29	18	1	11	0	75%
00016.wav	36	15	18	21	0	43.48%
00017.wav	34	18	1	16	0	67.92%
00030.wav	25	21	4	4	0	84%
00033.wav	35	21	4	14	0	70%
00036.wav	40	26	8	14	0	70.27%
00037.wav	32	20	5	12	0	70.18%
00038.wav	31	22	2	9	0	80%
00041.wav	38	27	1	11	0	81.82%
00048.wav	22	16	0	6	0	84.21%
					Rata - Rata	72.69%

Sistem yang dibangun berhasil menjalankan proses konversi *file* MIDI dari sinyal audio dengan tingkat akurasi rata-rata sebesar 72,69% pada data pengujian. Kualitas keluaran dari hasil konversi MIDI ini terpantau sangat bergantung dan berbanding lurus dengan tingkat kejernihan sinyal audio *input* pengguna. Suara *humming* yang bersih (*low noise*) dan ditahan dengan durasi yang stabil memproduksi terjemahan notasi MIDI yang sangat rapi, sedangkan fluktuasi *pitch* yang terlalu cepat dapat memicu terbentuknya potongan notasi yang kurang akurat.

E. Perbandingan Model

Dalam penelitian ini, dilakukan eksperimen perbandingan skenario arsitektur model untuk mendapatkan konfigurasi yang paling optimal dalam pengenalan dan klasifikasi *pitch*. Skenario model yang diuji membandingkan dua pendekatan, yaitu FFT-CNN (model yang menerima masukan citra spektrogram hasil ekstraksi fitur *Fast Fourier Transform*) dan RAW-CNN (model yang memproses sinyal audio mentah pada domain waktu tanpa ekstraksi fitur FFT). Tujuan dari perbandingan ini adalah untuk membuktikan efektivitas dan pengaruh penggunaan teknik ekstraksi fitur FFT terhadap peningkatan akurasi serta ketepatan klasifikasi nada pada model *Convolutional Neural Network*.

Tabel 4. Perbandingan Hasil Evaluasi Model

Model	Accuracy	Precision	Recall	F1-Score
FFT-CNN	0.8816	0.8632	0.8544	0.8581
RAW-CNN	0.8713	0.8066	0.796	0.8

Hasil dari tabel 4, terlihat secara jelas bahwa penerapan ekstraksi fitur menggunakan *Fast Fourier Transform* (FFT) memberikan peningkatan performa yang signifikan terhadap kemampuan model. Model FFT-CNN memimpin kinerja keseluruhan dengan mencatatkan *Accuracy* sebesar 0,8816 dan *F1-Score* sebesar 0,8581, mengungguli model RAW-CNN yang hanya mencapai *Accuracy* 0,8713 dan *F1-Score* 0,8000. Peningkatan lompatan angka yang sangat tajam terlihat pada metrik *Precision* (dari 0,8066 menjadi 0,8632) dan *Recall* (dari 0,7960 menjadi 0,8544). Hal ini membuktikan bahwa konversi sinyal suara dari domain waktu ke domain frekuensi melalui FFT mampu memperjelas representasi fitur frekuensi fundamental suara manusia, sehingga membantu jaringan CNN dalam mempelajari pola nada dengan jauh lebih presisi dan mengurangi persentase kesalahan klasifikasi. Atas dasar keunggulan performa model ini, arsitektur FFT-CNN ditetapkan sebagai model final untuk dikonversi ke format LiteRT (.tflite) dan ditanamkan ke dalam aplikasi Android.

F. Hasil Pengujian Aplikasi Android

Tahap akhir dari pengembangan sistem adalah melakukan pengujian perangkat lunak pada

aplikasi Android yang telah diintegrasikan dengan model *deep learning*. Aplikasi Android yang dikembangkan kemudian diuji secara menyeluruh menggunakan teknik *Black Box Testing* guna memverifikasi 28 kasus uji fungsional yang terbagi atas berbagai *Activity* utama. Pengujian ini berfokus pada evaluasi kesesuaian spesifikasi fungsionalitas aplikasi tanpa melihat struktur kode internal, guna memastikan bahwa seluruh fitur dapat dioperasikan oleh pengguna tanpa kendala.

Tabel 5. Ringkasan Hasil Black Box Testing

Halaman/Fitur	Jumlah Kasus Uji	Sesuai	Tidak Sesuai	Persentase Keberhasilan
Halaman Utama	3	3	0	100%
Halaman Rekaman	10	10	0	100%
Halaman Riwayat	12	12	0	100%
Integrasi Model dan MIDI	3	3	0	100%
Total	28	28	0	100%

Hasil yang ditampilkan Tabel 5 secara jelas mengonfirmasi bahwa seluruh alur fungsionalitas aplikasi mampu bekerja secara sempurna dengan rasio keberhasilan mencapai 100%. Fitur-fitur esensial aplikasi, yang mencakup perekaman audio melalui *AudioRecord*, *input* tempo BPM, fungsi navigasi, serta pengelolaan *database* lokal (Room Database), telah berjalan sesuai dengan desain awal sistem. Di samping itu, modul pendukung seperti *import*/*export* MIDI serta proses inferensi *pitch* dari *file* TensorFlow Lite berhasil dieksekusi secara mulus tanpa memicu *crash* ataupun malfungsi pada perangkat pengguna.

IV. KESIMPULAN DAN SARAN

Aplikasi deteksi pitch suara berbasis Android dengan algoritma *Fast Fourier Transform* dan *Convolutional Neural Network* telah berhasil dirancang dan dibangun. Aplikasi ini dapat menerima input suara, melakukan proses deteksi *pitch*, serta mengonversi hasil klasifikasi menjadi *file Musical Instrument Digital Interface* (MIDI). Model terbaik yang digunakan dalam penelitian ini adalah FFT-CNN dengan *Accuracy* sebesar 0,8816 dan *F1-Score* sebesar 0,8581, mengungguli model RAW-CNN. Model memiliki kemampuan yang baik dalam melakukan klasifikasi *pitch* sesuai nada asli. Hasil pengujian fungsional aplikasi Android dengan *Black Box Testing* menunjukkan keberhasilan 100%, membuktikan kemampuan aplikasi mengelola perekaman suara hingga melakukan ekspor ke *file* berekstensi MIDI secara mandiri. Aplikasi ini dapat digunakan sebagai alat bantu otomatisasi dokumentasi ide melodi bagi komposer maupun pengguna lainnya.

Saran untuk pengembangan selanjutnya adalah menambahkan variasi *dataset* suara perempuan, menerapkan *preprocessing* audio lanjutan untuk ketahanan terhadap *noise*, dan menambahkan fitur visualisasi penyuntingan hasil MIDI secara langsung di dalam aplikasi.

V. REFERENSI

[1] Yuniar P, et al. Sejarah Musik sebagai Dasar Pengetahuan dalam Pembelajaran Teori Musik. Clef : Jurnal Musik dan Pendidikan Musik. 2022; 3(2): 141-150.

[2] IFPI. IFPI GLOBAL MUSIC REPORT 2025. International Federation of the Phonographic Industry. Report number: 1. 2025.

[3] Ongko ES, Handayaningrum W, Rahayu EW. Proses Kreatif Komponis Kontemporer Slamet Abdul Sjukur dalam Berkarya Seni. Jurnal Kajian Seni. 2022; 8(2): 132.

[4] Qiu Y, et al. Humming2Music: Being A Composer As Long As You Can Humming. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. California. 2023: 7163-7166.

[5] [Liu S, et al. Humtrans: A Novel Open-Source Dataset for Humming Melody Transcription and Beyond. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Seoul. 2024: 7915-7919.

[6] Rumpa LD, et al. Frequencies Analysis on Suling Te'dek using Fast Fourier Transform: Ethnic Musical Analysis. Applied Technology and Computing Science Journal. 2023; 6(1): 71-77.

[7] Septiawan RD, Rayes PR, Robbaniyyah NA. Simulasi Penghilangan Noise pada Sinyal Suara menggunakan Metode Fast Fourier Transfrom. Semeton Mathematics Journal. 2024; 1(1): 1-7.

[8] Coccoluto D, Cesarini V, Costantini G. OneBitPitch (OBP): Ultra-High-Speed Pitch Detection Algorithm Based on One-Bit Quantization and Modified Autocorrelation. Applied Sciences. 2023; 13(14): 8191.

[9] Brata IPBW, Darmawan IDMBA. Comparative study of pitch detection algorithm to detect traditional Balinese music tones with various raw materials. Journal of Physics: Conference Series. 2021; 1722(1): 012071.

[10] Rosmala D, Fadhilah MN. Audio Conversion for Music Genre Classification Using Short-Time Fourier Transform and Inception V3. ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika. 2025; 13(1): 84.

[11] Lin C, et al. Vowel classification with combining pitch detection and one-dimensional convolutional neural network based classifier for gender identification. IET Signal Processing. 2023; 17(5).

[12] Leo A, Kuswanto V, Damayanti L. Mobile Commerce untuk Sales Order dan Tracking Order berbasis MVC. ALGOR. 2022; 4(1): 118-126.

[13] Samosir Y, Suhartana IKG, Putra IGNAC. Konversi Suara ke MIDI Menggunakan Short Time Fourier Transform Sebagai Virtual Midi Controller Pada Digital Audio Workstation. JELIKU (Jurnal Elektronik Ilmu Komputer Udayana). 2023; 12(2): 451.

[14] J.-S. R. Jang, "MIR-QBSH Corpus," MIR Lab, Dept. of Computer Science, National Tsing Hua University (NTHU), Taiwan. [Online]. Available: http://mirlab.org/dataset/public/MIR-QBSH-corpus.rar [1, 2, 3, 4].

[15] [15] Raffel C, et al. mir_eval: A Transparent Implementation of Common MIR Metrics. Proceedings of the 15th International Society for Music Information Retrieval Conference. Taipei. 2014: 367-372.

[16] [16] Shao K, et al. Towards Improving Harmonic Sensitivity and Prediction Stability for Singing Melody Extraction. arXiv preprint. 2023.