BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi yang semakin cepat, kecerdasan yang awalnya hanya dimiliki oleh makhluk hidup, sekarang mulai dikembangkan dan diimplementasikan ke dalam komputer. Hal tersebut biasa disebut dengan Kecerdasan Buatan. Kecerdasan Buatan atau dalam bahasa Inggris disebut *Artificial Intelligence* (*AI*) didefinisikan sebagai salah satu cabang ilmu pengetahuan yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Teknologi kecerdasan buatan memiliki banyak kegunaan sehingga mampu membuat komputer dan perangkat selular untuk melakukan verifikasi wajah, pengenalan suara, pengenalan sidik jari, memberikan rekomendasi video, dan masih banyak lainnya. Selain diimplementasikan di ranah teknologi, kecerdasan buatan juga telah diimplementasikan di ranah lainnya, seperti kesehatan, otomotif, hingga permusikan.

Namun, meskipun implementasi kecerdasan buatan dalam dunia permusikan tergolong banyak, namun dalam hal akurasi masih kurang akurat seperti *Chord Recognition*. *Chord Recognition* adalah sebuah sistem atau aplikasi untuk mengetahui atau mengukur akor secara tepat. Untuk membangun sistem tersebut, dibutuhkan metode-metode agar akurasi yang dihasilkan tinggi. Beberapa metode yang sering digunakan adalah metode *Convolutional Neural Network (CNN)* dan *K-Nearest Neighbor (KNN)*.

Convolutional Neural Network (CNN) adalah salah satu metode *Machine Learning* dari pengembangan *Multi Layer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. *CNN* dikatakan pengembangan lebih lanjut dari *MLP* karena *CNN* menggunakan metode yang mirip dengan *MLP*, namun menggunakan dimensi yang lebih banyak. *CNN* juga sering digunakan sebagai ekstraktor fitur yang kuat dari data yang telah disatukan, seperti gambar. Metode ini dapat diperluas ke berbagai tugas klasifikasi sinyal audio dengan merepresentasikan sinyal *input* dalam domain frekuensi waktu.

Terdapat beberapa peneliti terdahulu yang telah melakukan penelitian serupa, seperti *Filip Korzeniowski* dan *Gerhard Widmer* telah melakukan penelitian tentang pengenalan

akor. Penelitian tersebut berjudul ***FEATURE LEARNING FOR CHORD RECOGNITION: THE DEEP CHROMA EXTRACTOR***. Pada penelitian tersebut, *Filip* dan *Gerhard* melakukan pelatihan data menggunakan *chromogram* terhadap *Short-Time Fourier Transform (STFT)* sebagai data latih. *STFT* merupakan sebuah metode yang mengubah data mentah audio ke bentuk satuan signal menjadi frekuensi dalam satuan waktu. Penelitian tersebut juga membandingkan hasil akurasi *chromogram* dengan *spectogram*. Tingkat akurasi yang didapat menggunakan *chromogram* yaitu sebesar 69.2%, sementara tingkat akurasi untuk model yang menggunakan *spectogram* adalah sebesar 78.8%.

Beberapa studi penelitian menggunakan metode *CNN* pada model arsitektur untuk melakukan klasifikasi terhadap data berupa suara. Penelitian tersebut menerapkan *CNN* asli yang diperluas secara fungsional untuk *input spectogram* suara dan menunjukkan bahwa arsitektur *CNN* mengungguli bentuk-bentuk dasar sebelumnya dari *Deep Learning Network* (*DNN*) yang terhubung penuh pada pengenalan telepon dan tugas-tugas besar pengenalan suara vokal. Adapun sebuah penelitian untuk melakukan klasifikasi kelaparan, kesakitan dan mengantuk pada suara bayi menangis menggunakan *CNN* terhadap *spectogram MFCC* dengan tingkat akurasi 78.5%.

Beberapa studi penelitian lainnya, seperti yang dilakukan oleh I Gede Harsemadi, Made Sudarman, dan Nyoman Pramaita yakni memanfaatkan algoritma *KNN* dalam mengelompokkan musik terhadap suasana hati. Sistem yang dibangun dalam penelitian tersebut akan menerima masukan data berupa file musik format mono *.wav, yang selanjutnya melakukan proses pengelompokan terhadap musik dengan menggunakan klasifikasi *KNN*. Sistem tersebut akan menghasilkan keluaran berupa label jenis mood yaitu, *contentment*/ kepuasan, *exuberance*/ gembira, *depression*/ depresi dan *anxious*/ cemas; kalut. Secara umum hasil akurasi sistem dengan menggunakan algoritma klasifikasi K-NN cukup baik yaitu 86,55% pada nilai k = 3, serta waktu pemrosesan klasifikasi rata-rata 0,01021 detik per-file musik.

1.2 Identifikasi Masalah

Berdasarkan latar belakang telah diuraikan, maka penelitian ini akan membandingkan metode mana yang lebih baik dan lebih optimal antara metode *CNN* dan metode *KNN* dalam melakukan pengenalan terhadap data lagu yang dimainkan dengan menggunakan alat musik

Piano. Data latih yang digunakan adalah suara instrumen *piano* dalam bentuk format *mp3* dan *wav*. Data latih akan direkam dengan bantuan aplikasi *Audacity* dan akan dipisah menjadi potongan-potongan lagu berdasarkan jenis akor dan not pada *piano*.

1.3 Batasan Masalah

Hasil penelitian ini memiliki implikasi baik untuk penelitian dan pengembangan lebih lanjut. Agar pengerjaan lebih terarah, dalam penelitian ini terdapat beberapa batasan masalah, diantaranya :

a. Jumlah instrumen yang digunakan adalah instrumen *piano*.
b. Format *file* instrumen yang dapat diterima hanya *file* dengan format *mp3* dan *wav.*
c. Data yang dianalisis terdiri dari 12 kelas Major dan 12 kelas Minor.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengukur tingkat akurasi dari metode *Convolutional Neural Network (CNN)* dalam melakukan pengenalan terhadap data lagu yang dimainkan dengan menggunakan alat musik Piano.

1.5 Kegunaan Penelitian

Penelitian ini memiliki beberapa kegunaan, diantaranya :

a. Kegunaan Teoritis pada penelitian ini dibagi menjadi dua bagian, yakni kegunaan penelitian untuk peneliti dan kegunaan penelitian untuk peneliti selanjutnya.

1. Peneliti

Penelitian ini dapat digunakan sebagai tambahan pengetahuan tentang metode *CNN* dan metode *KNN* dalam melakukan pengenalan otomatis. Selain itu, juga menambah pengetahuan peneliti tentang bahasa pemrograman *Python*.

2. Peneliti Selanjutnya

Penelitian ini dapat digunakan untuk sebagai referensi untuk penelitian selanjutnya bagi peneliti lain yang berminat untuk mempelajari Kecerdasan Buatan. Selain itu, penelitian ini juga dapat digunakan untuk mengimplementasikan metode *Machine Learning*.

b. Kegunaan praktis pada penelitian ini adalah untuk pemusik amatir.

Penelitian ini dapat digunakan sebagai bahan acuan untuk menentukan akor dari sebuah lagu atau nada. Selain itu, penelitian ini juga sangat membantu para pemusik amatir yang masih buta dengan not nada.

BAB II

TELAAH PUSTAKA

2.1 Tinjauan Pustaka Penelitian

Penelitian ini dilakukan dengan terlebih dahulu membahas sejumlah teori dan konsep yang berhubungan dengan kedua metode yang akan dibandingkan (Metode *KNN* dan Metode CNN). Landasan teori yang dibahas terdiri dari *audio*, *akor, bass, Beat Tracking*, *Chromagram*, Metode *KNN* dan Metode *CNN*.

2.1.1 Audio

Audio adalah sebuah deret waktu, dimana sumbu y adalah amplitudo arus yang sesuai dengan sebuah membran *loudspeaker* dan sumbu x adalah sumbu yang sesuai dengan satuan waktu data tersebut.[1] Telinga manusia hanya dapat mendengar bunyi dengan rentang frekuensi antara 20 Hz hingga 20 KHz (20.000Hz). Angka 20 Hz sebagai frekuensi suara terendah yang dapat didengar, sedangkan 20 KHz merupakan frekuensi tertinggi yang dapat didengar.[2] Gelombang suara mengandung sejumlah komponen penting, seperti amplitudo, panjang gelombang, dan frekuensi. Komponen-komponen tersebut mampu membuat suara yang satu berbeda dengan suara yang lain.

Amplitudo sendiri merupakan kekuatan atau daya gelombang sebuah sinyal. Nilai amplitudo diinterpretasikan sebagai volume. Semakin besar nilai dari sebuah amplitudo, maka semakin keras suara yang dihasilkan. Sebaliknya, jika nilai dari suatu amplitudo semakin kecil, maka suara yang dihasilkan semakin lemah. Frekuensi adalah jumlah dari siklus yang terjadi dalam satu detik dan memiliki satuan *Hertz* (Hz). Getaran gelombang suara yang cepat membuat frekuensi semakin tinggi. Misalnya, menyanyi nada tinggi membuat tali suara pada pita suara bergetar secara cepat.

---

[1] Ingo Miersawa, Katharina Morik. Automatic Feature Extraction for Classifying Audio Data,(Jerman: Artificial Intelligence Unit, University of Dortmund, 2005), h.1.
[2] Sri Waluyanti, Buku Direktorat PSMK Untuk Tehnik Audio Video, (Jakarta: Direktorat Pembinaan SMK, 2008), h.1.

2.1.2 Akor

Akor adalah sebuah kombinasi tiga nada atau lebih yang dibunyikan secara bersamaan. Akor tiga not yang berjarak tiga *scale* pada *root* sampai not ketiga disebut *triad*. Untuk mempermudah pemusik dalam menentukan apakah suatu kombinasi dari tiga nada tertentu merupakan sebuah *triad* atau bukan, maka pemusik dapat membentuk diagram *circle of thirds*. *Scale* atau skala nada merupakan jarak yang menandakan lokasi suatu nada dengan titik mulai tertentu. Terdapat empat jenis *triad* yang dibedakan berdasarkan skala nadanya, diantaranya :

A. Major Triad

Major triad merupakan tiga nada yang dibunyikan dengan skala nada 1-2-1,5. Akor *major* menunjukan identitas suatu bagian lagu lebih dekat dengan *root* karena ada satu buah akor *major* yang menempati *Scale I*.

B. Minor Triad

Minor triad merupakan tiga nada yang dibunyikan dengan skala 1-1,5-2.

C. Diminished Triad

Diminished triad merupakan tiga nada yang dibunyikan dengan skala 1-1,5-1,5.

D. Augmented Triad

Augmented triad merupakan tiga nada yang dibunyikan dengan skala 1-1,5-2,5.

Akor *seventh* merupakan akor empat not yang kelas pitchnya dapat diatur sebagai sepertiga. Skala nada akor *major seventh* adalah 1-2-1,5-2. Sedangkan skala nada untuk *minor seventh* adalah 1-1,5-2-2. Sama halnya dengan *triad*, kelas nada yang dimiliki oleh akor *seventh* menempati posisi yang berdekatan (rumpun empat kelas) pada lingkaran pertiga. Keempat anggota akor ketujuh adalah *root*, *third*, *fifth*, dan *seventh*. Terdapat suatu cara untuk melakukan teknik *blocking* pada akor *minor seventh*. Dengan memanfaatkan teknik *blocking*, maka *root* dapat berperan sebagai nada *bass*. Sementara *third, fifth,* dan *seventh* sebagai *major triad*.

2.1.3 Beat Tracking

Beat tracking digunakan untuk menentukan contoh waktu dalam rekaman *audio*, di mana pendengar manusia cenderung mengetuk kakinya ke musik.[3] *Beat tracking* pada rancangan menggunakan metode *dynamic programming*. *Dynamic programming* atau *dynamic optimization* bukan merupakan rumus matematika yang mampu memberikan jawaban dengan hanya sekedar memberikan input.[4] Sebaliknya, *dynamic programming* adalah kombinasi pemikiran terstruktur dan pola pikir analitis untuk menyelesaikan sebuah permasalahan.

Untuk melakukan *beat tracking*, hal yang harus dilakukan terlebih dahulu adalah mencari sinyal yang berupa *onset*. *Onset* merupakan lokasi dimana sinyal berada pada nilai yang tinggi secara tiba-tiba.[5] *Onset* yang ditemukan kemudian dilakukan penandaan. *Onset* yang ditemukan bisa saja merupakan *false positive.* Untuk mengurangi *false positive* tersebut, maka harus dilakukan perhitungan untuk menemukan urutan umum terpanjang dari *onset*.[6]

2.1.4 Chromagram

Chromagram adalah transformasi properti frekuensi-waktu sinyal menjadi prekursor pitch yang berubah-ubah untuk sementara waktu. Transformasi ini didasarkan pada pengamatan persepsi tentang sistem pendengaran dan telah terbukti memiliki beberapa sifat matematika yang menarik. *Chromagram* memperluas konsep chroma untuk memasukkan dimensi waktu. Seperti halnya kita menggunakan spektrogram untuk menyimpulkan properti tentang distribusi energi sinyal dari frekuensi dan waktu, chromagram dapat digunakan untuk menyimpulkan properti tentang distribusi energi sinyal terhadap kroma dan waktu.

Terdapat dua masalah yang muncul dalam mengembangkan konsep *chromagram* ini, yakni definisi dan bagaimana cara menghitung *chromagram*. Chroma adalah pemetaan banyak-ke-satu (*many-to-one*) frekuensi, sementara chromagram adalah ukuran kekuatan sinyal sebagai fungsi dari chroma dan waktu. Oleh karena itu, *chromagram* dapat didefinisikan sebagai pemetaan banyak-ke-satu (*many-to-one*) kekuatan sinyal pada frekuensi milik kelas chroma yang sama. Pembagian antara transformasi langsung dari sinyal asli dan

---

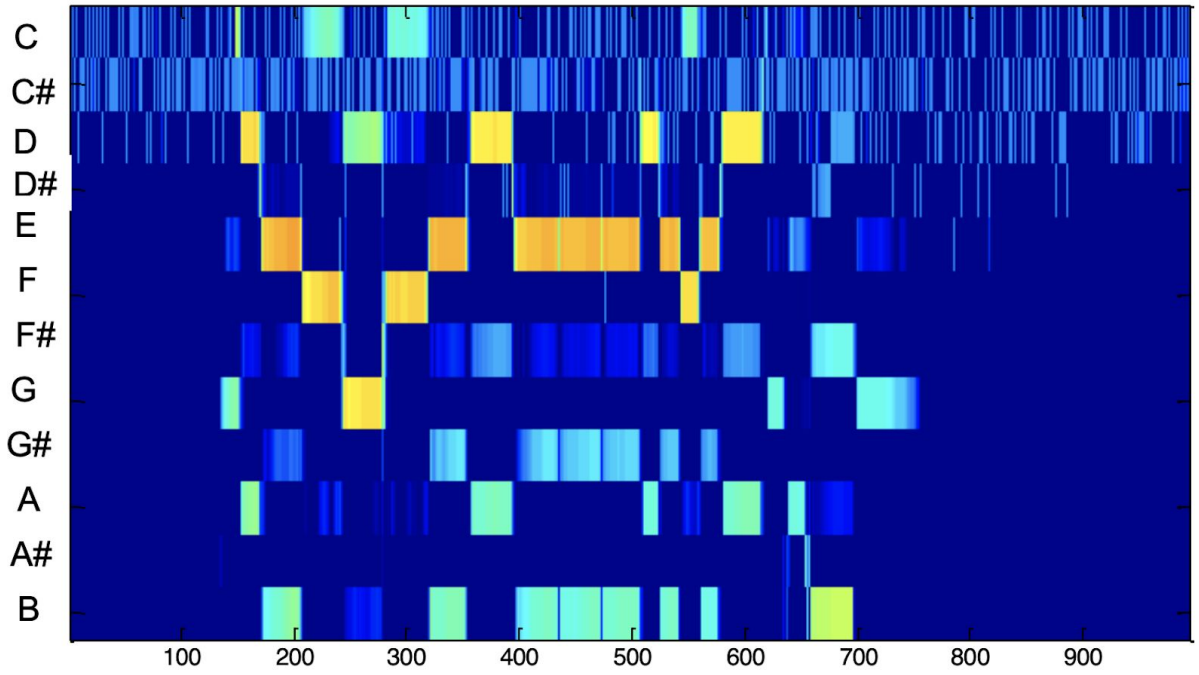
[3] Tavish Srivastava, Solve Interview Case Studies 10x Faster Using Dynamic Programming, Septhttps://www.analyticsvidhya.com/blog/2016/05/ase-studies-10x-faster-using-dynamic-programming/, 20 September 2019.

[4] ibid.

[5] Faizan Shaikh, Learn Audio Beat Tracking for Music Information Retrieval, https://www.analyticsvidhya.com/blog/2018/02/audio-beat-tracking-for-music-information-retrieval/, 20 September 2019.

[6] ibid.

transformasi dari gambar frekuensi-waktu dapat digunakan untuk menghitung chromagram. Contoh *chromagram* dapat dilihat dengan jelas pada Gambar 1.3.



Gambar 1.3 *Chromagram* dari sebuah alat musik Clarinet [https://users.cs.northwestern.edu/~pardo/courses/eecs352/lectures/MPM16-topic7-Spectrogram-Chroma-Cepstra.pdf]

2.1.5 Windowing

Windowing berfungsi untuk meminimalisir sinyal yang tak kontinu pada awal dan akhir masing-masing *frame*.[7] Nilai panjang filter didapatkan dengan menggunakan rumus di bawah. Nilai panjang tersebut akan dipakai untuk membuat respon *impulse* di simulasi dan implementasi. Nilai panjang filter juga akan berpengaruh kepada faktor *roll-off* atau faktor kelandaian dari suatu filter. Rumus untuk menghitung *window* dapat dilihat pada persamaan 1.

$$Y_t(n) = x_t(n)\ w(n), 0 \leq n \leq N-1 \qquad (1)$$

[7] Marwa A.Nasr, et.al., "Speaker identification based on normalized pitch frequency and Mel Frequency Cepstral Coefficient", International Journal of Speech Technology, Vol.21, Issue 4, (Desember, 2018), h.941-951.

Rumus untuk menghitung *Hamming Window* dapat dilihat pada persamaan 2.

$$w(n) = 0.54 - 0.46 \cos cos \left(\frac{2\pi n}{N-1}\right),\ 0 \leq n \leq N-1 \qquad (2)$$

Keterangan:

N = Panjang setiap *frame*

$x_t(n)$ = Sinyal masukan ke-n pada *frame* ke-t

$w(n)$ = Fungsi *Hamming Window*

$Y_t(n)$ = Nilai hasil *windowing* untuk sinyal masukan ke-n pada *frame* ke-t

Data mentah dari masukan yang pertama kali dibaca oleh bahasa pemrograman *Python* adalah sinyal dalam domain waktu. Data sinyal dalam domain waktu tersebut perlu diubah ke dalam satuan frekuensi. Proses untuk mengubah data mentah yang berbentuk sinyal tersebut diubah dengan menggunakan metode *Discrete Fourier Transform* (DFT). Pada tahap ini, setiap *frame* yang terdiri dari N sampel dikonversi dari domain waktu ke domain frekuensi.[8] Rumus dari DFT dapat dilihat di dalam persamaan 3.[9]

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\left(\frac{-j2\pi nk}{N}\right)},\ k = 0, 1, \ldots, N-1 \qquad (3)$$

Keterangan:

$X(k)$ = Nilai transformasi Fourier (FT) ke-k

$x(n)$ = *Input* signal ke-n

N = Panjang *input* signal

Hasil proses *windowing* pada persamaan 1 kemudian dimasukan sebagai input signal ke-n, sehingga menghasilkan persamaan 4.[10]

$$Z_t(k) = \sum_{n=0}^{N-1} Y_t(n)\, e^{\left(\frac{-j2\pi nk}{N}\right)},\ k = 0, 1, \ldots, N-1 \qquad (4)$$

$$e^{i\theta} = \cos cos\ \theta + \sin sin\ \theta \qquad (5)$$

---

[8] Agus Buono, Wisnu Jatmiko, dan Benyamin Kusumoputro, "Perluasan Metode MFCC 1D ke 2D sebagai Ekstraksi Ciri pada Sistem Identifikasi Pembicara Menggn Hidden Markov Model (HMM)", Makara, Sains, Vol. 13, Nomor 1, (April, 2009), h.87.

[9] Todor Dimitrov Ganchev, Speaker Recognition, Patras: Department of Electrical and Computer Engineering University of Patras (Unpublished Doctoral Dissertation), 2005, h.68.

[10] Ranny, Sistem Pengenalan Pembicara Menggunakan Mel Frequency Cepstrum Coefficients, Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara (Skripsi tidak dipublikasikan), 2005.

Berdasarkan rumus *Euler* pada persamaan 5,[11] dengan e adalah bilangan eksponen, i adalah bilangan imajiner, dan cos dengan sin adalah fungsi trigonometri. Jika persamaan 5 disubstitusikan ke persamaan 4 maka rumus *DFT* menjadi persamaan 6 di bawah ini.[12]

$$Z_t(k) = \sum_{n=0}^{N-1} Y_t(n) \left[\cos cos \left(\frac{2\pi nk}{N}\right) - j \sin sin \left(\frac{2\pi nk}{N}\right) \right] \qquad (6)$$

$k = 0, 1, \dots, N-1$

Keterangan:

$Z_t(k)$ = Nilai FT yang mengandung bilangan kompoleks

$Y_t(n)$ = Nilai *hamming window* ke-n pada frame ke-t

j = Bilangan imajiner

Hasil perhitungan pada persamaan 6 mengandung bilangan riil dan imajiner. Oleh karena itu, nilai Fourier spektrum didapat dari nilai *magnitude* yang dibentuk kedua bilangan riil dan imajiner. Besar nilai *magnitude* dari persamaan $z = x + y$ tersebut dihitung di dalam persamaan 7.[13]

$$r = M(n) = \sqrt{x^2 + y^2} \qquad (7)$$

Keterangan:

z = Nilai FT yang mengandung bilangan kompleks

$M(n)$ = *Magnitude*

x = koefisien bagian riil

y = Koefisien bagian imajiner

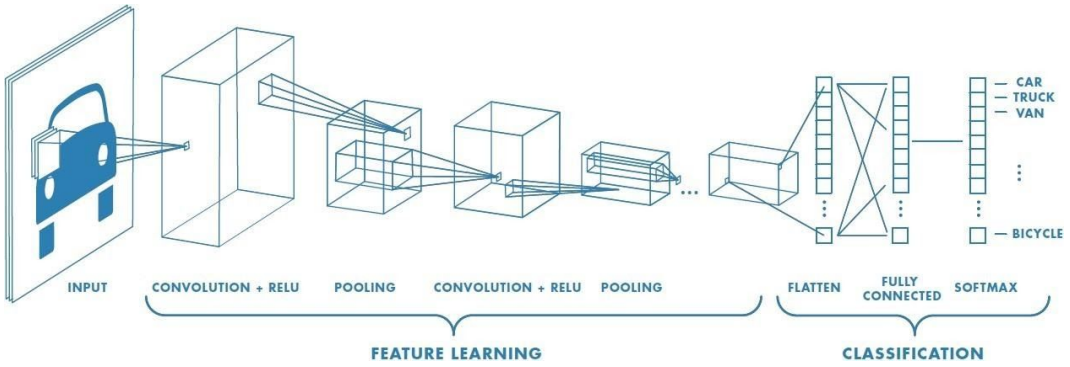
2.1.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (*CNN*) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*.[14] *CNN* bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah image. *CNN* terdiri dari neuron yang memiliki *weight*, bias, dan fungsi aktivasi. Cara kerja *CNN* memiliki kesamaan pada *MLP*, namun dalam *CNN* setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti *MLP* yang setiap neuron hanya

[11] Martin A Moskowitz, A Course in Complex Analysis in One Variable, (Singapore: World Scientific Publishing Co., 2002), h.41.
[12] Bella, op.cit., h. 28.
[13] Tom Apostol, Mathematical analysis, (Cambridge: Addison-Wesley, 1981), h.18.

[14] Samuel Sena, Pengenalan Deep Learning Part7: Convolutional Neural Network (CNN), https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94, 7 September 2019.

berukuran satu dimensi. Pada *CNN*, data yang dipropagasikan oleh jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada *CNN* berbeda. Pada *CNN* operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi. Karena sifat proses konvolusi, maka *CNN* hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara. Arsitektur dari *CNN* dibagi menjadi dua bagian besar, *Feature Extraction Layer* dan *Fully-Connected Layer* (MLP). Arsitektur *CNN* dapat dilihat pada Gambar 1.4.



Gambar 1.4 Arsitektur *Convolutional Neural Network*

[Medium, Pengenalan Deep Learning Part 7: Convolutional Neural Network (CNN), https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94]

I Wayan Suartika Eka Putra(2016) dalam penelitiannya yang berjudul Klasifikasi Citra Menggunakan *Convolutional Neural Network (CNN)* pada *Caltech 101* mengimplementasikan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi citra objek yaitu *CNN*. Pada penelitian yang dilakukan, peneliti mengimplementasikan metode *CNN* yang terdiri dari dua tahap. Tahap pertama adalah klasifikasi citra menggunakan *feedforward*. Tahap kedua merupakan tahap pembelajaran dengan metode *backpropagation*. Hasil uji coba dari klasifikasi citra objek dengan tingkat confusion yang berbeda pada basis data Caltech 101 menghasilkan rata-rata nilai akurasi tergolong tinggi, sehingga dapat disimpulkan bahwa metode CNN yang digunakan pada penelitian tersebut mampu melakukan klasifikasi dengan baik.

Berbeda dengan Dzikry Maulana Hakim dan Ednawati Rainarli(2019) dalam penelitiannya yang berjudul *Convolutional Neural Network* untuk Pengenalan Citra Notasi Musik. Peneliti melakukan pengujian untuk melakukan pengenalan citra notasi musik dengan dua cara. Pertama, peneliti menguji performa CNN menggunakan data notasi musik yang telah dipotong dan yang kedua peneliti melakukan pengujian menggunakan sebaris notasi musik. Nilai akurasi yang didapatkan untuk pengenalan sebaris notasi musik tidak terlalu besar, yaitu 26,19%. Walaupun untuk proses segmentasi masih belum maksimal dalam memotong setiap notasi, namun metode CNN bekerja sangat baik untuk mengenali setiap notasi musik yang telah dipotong dengan benar. Hal ini ditunjukkan dari nilai akurasi yang mencapai 95,56%.

2.1.7 Music Information Retrieval

Music Information Retrieval (*MIR*) terutama berkaitan dengan ekstraksi dan inferensi fitur yang berarti dari musik seperti sinyal audio, representasi simbolik atau sumber eksternal seperti halaman web. Berikut merupakan contoh tugas *MIR*, antara lain *fingerprinting*, *cover song detection*, *genre recognition*, *key detection*, *beat tracking*, *symbolic melodic similarity*, *source separation*, *pitch tracking*, *tempo estimation*, dan masih banyak lagi. *MIR* digunakan pada sistem untuk menyiapkan data mentah menjadi bentuk data yang dapat dilakukan observasi oleh model. Berikut merupakan beberapa fitur pada *MIR* yang digunakan oleh sistem :

A. Beat Tracking

Sistem memanfaatkan fitur beat tracking yang disediakan oleh pustaka librosa yang menghasilkan keluaran dalam bentuk list. Nilai yang dihasilkan dalam satuan detik.

B. Harmonic Separation

Fitur harmonic separation berfungsi untuk memisahkan data masukan berupa harmonic dan percussive. Contoh dari harmonic adalah nada yang dibunyikan oleh piano dan suara vokal, sedangkan percussive merupakan kumpulan data suara yang dimainkan dengan cara dipukul, digesek, maupun dikocok.

2.2 Penelitian Terdahulu

I Wayan Suartika Eka Putra(2016) dalam penelitiannya yang berjudul Klasifikasi Citra Menggunakan *Convolutional Neural Network (CNN)* pada *Caltech 101* mengimplementasikan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi citra objek yaitu *CNN*. Pada penelitian yang dilakukan, peneliti mengimplementasikan metode *CNN* yang terdiri dari dua tahap. Tahap pertama adalah klasifikasi citra menggunakan *feedforward*. Tahap kedua merupakan tahap pembelajaran dengan metode *backpropagation*. Sebelum melakukan klasifikasi, terlebih dahulu dilakukan praproses dengan metode *wrapping* dan *cropping* untuk memfokuskan objek yang akan diklasifikasi. Selanjutnya dilakukan *training* menggunakan metode *feedforward* dan *backpropagation*. Terakhir adalah tahap klasifikasi menggunakan metode *feedforward* dengan bobot dan bias yang diperbarui. Hasil uji coba dari klasifikasi citra objek dengan tingkat confusion yang berbeda pada basis data Caltech 101 menghasilkan rata-rata nilai akurasi tergolong tinggi, sehingga dapat disimpulkan bahwa metode CNN yang digunakan pada penelitian tersebut mampu melakukan klasifikasi dengan baik.

Danny Lionel, Rudy Adipranata dan Endang Setyati (2016) dalam penelitian yang berjudul Klasifikasi Genre Musik Menggunakan Metode Deep Learning Convolutional Neural Network dan MelSpektrogram mengimplementasikan metode Mel-spectrogram. Dimana Mel spektrogram merupakan hasil pemetaan fitur yang telah diambil oleh metode MFCC, yang akan diklasifikasikan dan dimasukkan ke dalan *Convolutional Neural Network*. Yang akan dibedakan activation function nya yaitu *ReLU* dan *ELU*. Penelitian ini menunjukkan bahwa pengambilan fitur dari audio dengan menggunakan *MFCC* merupakan metode yang benar dan dalam hasil pengujian, banyaknya dataset, iterasi training, dan spesifikasi komputer sangat mempengaruhi tingkat akurasi dan lama pembuatan neural network model yang optimal. Dalam hasil penelitian ini telah diuji beberapa kali didapatkan hasil akurasi yang paling optimal yaitu 99%.

Dzikry Maulana Hakim dan Ednawati Rainarli (2019) pada penelitiannya yang berjudul *Convolutional Neural Network* untuk Pengenalan Citra Notasi Musik melakukan penelitian untuk mengenali citra notasi musik dengan metode *CNN*. Pada penelitian ini, untuk pengenalan notasi musik digunakan *Convolutional Neural Network (CNN)*. Arsitektur CNN yang dipakai adalah kernel 3x3, jumlah layer pada feature learning sebanyak 3 *convolutional*

layer, 3 *pooling layer*, filter pada convolutional layer 64,128, 256, dan jumlah neuron pada hidden layer sebanyak 7168. Pengujian dilakukan dengan dua cara, yang pertama menguji performa *CNN* menggunakan data notasi musik yang telah dipotong dan yang kedua adalah melakukan pengujian menggunakan sebaris notasi musik. Nilai akurasi yang didapatkan untuk pengenalan sebaris notasi musik tidak terlalu besar, yaitu 26,19%. Walaupun untuk proses segmentasi masih belum maksimal dalam memotong setiap notasi, namun metode *CNN* bekerja sangat baik untuk mengenali setiap notasi musik yang telah dipotong dengan benar. Hal ini ditunjukkan dari nilai akurasi yang mencapai 95,56%.

Nick Collins melakukan penelitian dengan membuat *library SuperCollider Music Information Retrieval* (*SCMIR*). *SCMIR* adalah set ekstensi untuk bahasa pemrograman audio yang berfungsi untuk melakukan analisis otomatis sebuah *file* audio. *Library* ini mendukung teknologi pencarian informasi musik umum, termasuk untuk pemrosesan *batch* di seluruh file suara. *Library* memiliki kelebihan yang diambil dari mode *Non-Real-time scsynth* dan *plug-in* untuk ekstraksi fitur cepat, serta fitur tambahan untuk perhitungan intensif. Hasil dari penelitian ini adalah Nick Collins dapat meningkatkan hubungan antara sintesis pencocokan fitur, mendengarkan mesin dalam sistem interaktif, dan tugas MIR, melintasi *real time task* dan *non-real time* dengan memprioritaskan adopsi teknologi MIR dalam sistem interaktif. Selain itu, *library SCMIR* juga membawa teknologi pencarian informasi musik ke bahasa pemrograman komputer musik yang sudah terkenal.

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Judul Jurnal	Tujuan Penelitian	Hasil
1	I Wayan Suartika Eka Putra	Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101	Mengimplementasikan salah satu metode machine learning yang dapat digunakan untuk klasifikasi citra objek yaitu CNN	Hasil uji coba dari klasifikasi citra objek dengan tingkat confusion yang berbeda pada basis data Caltech 101 menghasilkan rata-rata nilai akurasi tergolong tinggi, sehingga dapat disimpulkan bahwa

				metode CNN yang digunakan pada penelitian tersebut mampu melakukan klasifikasi dengan baik
2	Danny Lionel, Rudy Adipranata dan Endang Setyati	Klasifikasi Genre Musik Menggunakan Metode Deep Learning Convolutional Neural Network dan MelSpektrogram	Mengimplementasikan metode agar suatu audio file dapat dikenali secara otomatis dari fitur-fitur yang telah diekstrak sebelumnya dengan bantuan MFCC (Mel Frequency Cepstral Coefficients) dan ANN (Artificial Neural Network).	Penelitian ini menunjukkan bahwa pengambilan fitur dari audio dengan menggunakan *MFCC* merupakan metode yang benar dan dalam hasil pengujian, banyaknya dataset, iterasi training, dan spesifikasi komputer sangat mempengaruhi tingkat akurasi dan lama pembuatan neural network model yang optimal. Dalam hasil penelitian ini telah diuji beberapa kali didapatkan hasil akurasi yang paling optimal yaitu 99%.
3	Dzikry Maulana Hakim dan Ednawati Rainarli	Convolutional Neural Network untuk Pengenalan Citra Notasi Musik	Mengetahui cara agar sistem dapat mendeteksi sebuah notasi musik dan kemudian mengenali notasi musik tersebut	Pengujian dilakukan dengan dua cara, yang pertama menguji performa *CNN* menggunakan data notasi musik yang telah dipotong

				dan yang kedua adalah melakukan pengujian menggunakan sebaris notasi musik. Nilai akurasi yang didapatkan untuk pengenalan sebaris notasi musik tidak terlalu besar, yaitu 26,19%. Walaupun untuk proses segmentasi masih belum maksimal dalam memotong setiap notasi, namun metode *CNN* bekerja sangat baik untuk mengenali setiap notasi musik yang telah dipotong dengan benar. Hal ini ditunjukkan dari nilai akurasi yang mencapai 95,56%.
4	Nick Collins	SCMIR: A SUPERCOLLIDER MUSIC INFORMATION RETRIEVAL LIBRARY	Membuat sebuah *library extension set* untuk menganalisis *file* audio secara otomatis	Peneliti berhasil mengembangkan *library* SCMIR sehingga dapat meningkatkan hubungan antara sintesis pencocokan fitur, mendengarkan mesin dalam sistem interaktif, dan mampu melintasi pekerjaan *realtime* dan *non-realtime* dengan memprioritaskan adopsi teknologi MIR dalam sistem interaktif.
5	Florian Krebs,	RHYTHMIC PATTERN	Menyelidiki penggunaan	Penelitian yang dilakukan menunjukkan bahwa

	Sebastian Bo ¨ck, and Gerhard Widmer	MODELING FOR BEAT AND DOWNBEAT TRACKING IN MUSICAL AUDIO	pemodelan pola ritmis untuk menyimpulkan struktur metrik dalam rekaman audio musikal	menghitung fitur onset setidaknya untuk dua band frekuensi yang berbeda meningkatkan kinerja pelacakan suram secara signifikan dibandingkan dengan fitur tunggal yang mencakup seluruh rentang frekuensi. Dalam perbandingan dengan enam sistem referensi, pemodelan gaya tarian secara eksplisit sebagai pola ritmik terbukti mengurangi kesalahan oktaf (mendeteksi setengah atau tempo ganda) dalam pelacakan ketukan. Selain itu, pelacakan suram ditingkatkan secara substansial dibandingkan dengan varian yang hanya memodelkan meter dan dua sistem referensi.
6	Albert Parlys, Ajub Ajulian Zahra, dan Achmad Hidayatno	Penggolongan Lagu Berdasarkan Spektogram dengan *Convolutional Neural Network*	Penelitian ini akan membahas perancangan sebuah sistem untuk menggolongan lagu berdasarkan spektogram. Masukan sistem berupa lagu dengan format audio MP3 yang diubah ke dalam bentuk spektogram kemudian dilatih menggunakan Convolutional Neural Network. Ciri lagu akan diperoleh kemudian diklasifikan ke dalam lima genre berbeda yaitu pop, rock, classic, dubstep, dan reggae	Berdasarkan hasil pelatihan dan pengujian dengan filter 3x3 didapat nilai akurasi penggolongan lagu sebesar 100% pada 750 data latih dan 98% pada 50 lagu data uji. Algoritme pembelajaran terbaik pada pelatihan dengan filter yang sama adalah algoritme Adam yang lebih cepat dibandingkan dengan Adadelta, Adagrad, dan SGD

BAB III
METODE PENELITIAN

3.1 Metode Penelitian

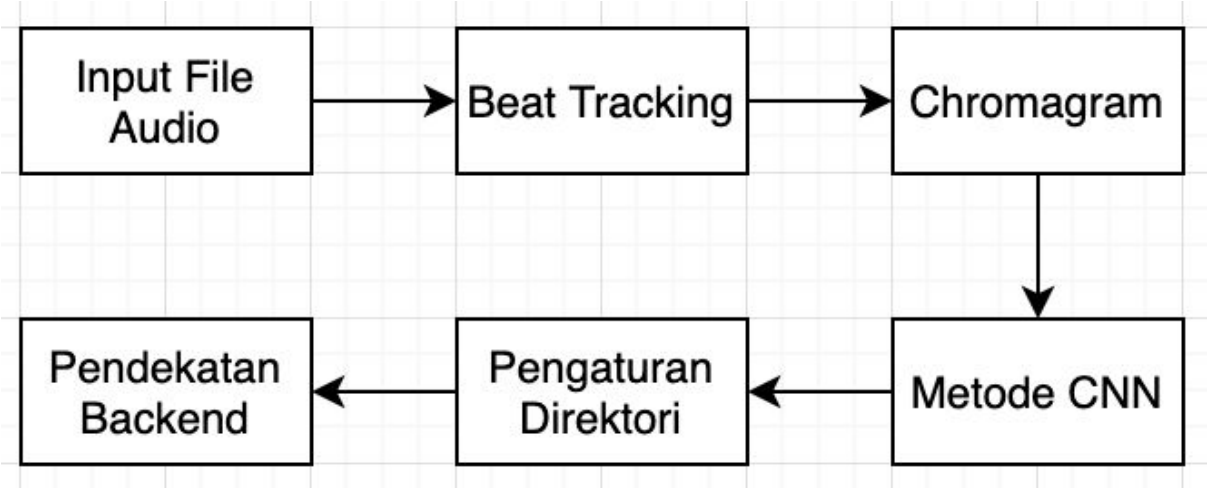
Dalam melakukan penelitian ini, tahapan pertama yang dilakukan adalah pengumpulan data. Tahap ini bertujuan untuk memberikan gambaran tentang tahapan-tahapan apa saja yang akan dilakukan pada proses penelitian. Teknik yang digunakan pada tahapan pengumpulan data adalah studi literatur atau biasa disebut dengan studi pustaka. Studi literatur atau studi pustaka merupakan sebuah proses pengumpulan data dan informasi berupa teori-teori yang berhubungan dengan masalah yang diteliti. Studi literatur dilakukan dengan mencari jurnal ilmiah, paper ilmiah, dan beberapa tesis yang tersebar di internet. Adapun data-data yang dibutuhkan untuk dalam penelitian ini, diantaranya teori tentang *CNN* dan *KNN*, teori dan proses tentang *Beat Tracking*, teori tentang *Chromagram* dan cara mengubah audio ke bentuk *chromagram*, teori tentang bahasa pemrograman *Python* versi 3, dan beberapa implementasi tentang *backend* pada sebuah aplikasi.

3.1.1 Teori Penelitian

Menurut Kamus Besar Bahasa Indonesia (KBBI), kecerdasan adalah kesempurnaan perkembangan akal budi (seperti kepandaian atau ketajaman pemikiran). Helfi Nasution berpendapat bahwa Kecerdasan Buatan atau dalam bahasa Inggris disebut *Artificial Intelligence* (*AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan (Helfi Nasution, 2012), sedangkan menurut Dedi Nugraha dan Sri Winati (2014), kecerdasan buatan adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi.

3.2 Kerangka Pemikiran

Kerangka pemikiran dalam penelitian ini memiliki hasil akhir yakni mengetahui akurasi dari metode *CNN* dalam mengenali sebuah akor dengan memanfaatkan *Chromagram*. Kerangka pemikiran yang akan dilakukan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Kerangka Pemikiran Penelitian

Langkah awal yang akan dilakukan adalah menerima *input* file audio yang memiliki format .wav. Selanjutnya akan dilakukan *beat tracking*. Pada proses *beat tracking*, file audio yang telah diunggah akan dipotong tiap detik dan akan dianalisa bentuk gelombangnya sehingga terbentuklah sebuah *onset*. Setelah mendapatkan *onset* dari proses *beat tracking*, selanjutnya *onset* tersebut akan diubah ke bentuk *chromagram*. Pada proses *chromagram*, *onset* yang telah didapatkan akan dibentuk ke bentuk gambar. Gambar tersebut nantinya akan dilatih dengan menggunakan metode *CNN* untuk menemukan akor yang sesuai. Setelah menemukan akor yang sesuai, selanjutnya melakukan proses pengaturan direktori. Proses ini dibutuhkan agar penempatan *parenthesis* dan *child package* bagian *backend* dan *frontend* dapat berkomunikasi satu sama lain. Dan terakhir adalah ada proses pendekatan *backend*. Pendekatan *backend* dibutuhkan agar memudahkan dalam mengakses fungsi yang ada pada pada *script Python*.

3.2.1 Audio

Audio adalah sebuah deret waktu, dimana sumbu y adalah amplitudo arus yang sesuai dengan sebuah membran *loudspeaker* dan sumbu x adalah sumbu yang sesuai dengan satuan waktu data tersebut.[15] Telinga manusia hanya dapat mendengar bunyi dengan rentang frekuensi antara 20 Hz hingga 20 KHz (20.000Hz). Angka 20 Hz sebagai frekuensi suara terendah yang dapat didengar, sedangkan 20 KHz merupakan frekuensi tertinggi yang dapat

[15] Ingo Miersawa, Katharina Morik. Automatic Feature Extraction for Classifying Audio Data,(Jerman: Artificial Intelligence Unit, University of Dortmund, 2005), h.1.

didengar.[16] Gelombang suara mengandung sejumlah komponen penting, seperti amplitudo, panjang gelombang, dan frekuensi. Komponen-komponen tersebut mampu membuat suara yang satu berbeda dengan suara yang lain.

Audio yang digunakan dalam penelitian ini adalah audio yang berisi suara piano dan bass serta audio yang berbentuk format *wav. Audio tersebut nantinya akan diolah melalui beberapa tahapan, seperti *Preprocessing* dan *Modeling*. Proses *preprocessing* terbagi atas 2 tahap, yakni *Beat Tracking* dan membentuk gambar *chroma*. Sedangkan untuk proses *modeling* akan langsung dengan menggunakan metode *CNN*.

3.2.2 Beat Tracking

Beat tracking digunakan untuk menentukan contoh waktu dalam rekaman *audio*, di mana pendengar manusia cenderung mengetuk kakinya ke musik.[17] *Beat tracking* pada rancangan menggunakan metode *dynamic programming*. *Dynamic programming* adalah kombinasi pemikiran terstruktur dan pola pikir analitis untuk menyelesaikan sebuah permasalahan. Proses *beat tracking* ini dibutuhkan untuk membentuk ketukan dalam setiap detik dari file audio yang diunggah.

3.2.3 Chromagram

Chromagram adalah transformasi properti frekuensi-waktu sinyal menjadi prekursor pitch yang berubah-ubah untuk sementara waktu. Transformasi ini didasarkan pada pengamatan persepsi tentang sistem pendengaran dan telah terbukti memiliki beberapa sifat matematika yang menarik. *Chromagram* memperluas konsep chroma untuk memasukkan dimensi waktu. Seperti halnya kita menggunakan spektrogram untuk menyimpulkan properti tentang distribusi energi sinyal dari frekuensi dan waktu, chromagram dapat digunakan untuk menyimpulkan properti tentang distribusi energi sinyal terhadap kroma dan waktu.

---

[16] Sri Waluyanti, Buku Direktorat PSMK Untuk Tehnik Audio Video, (Jakarta: Direktorat Pembinaan SMK, 2008), h.1.
[17] Tavish Srivastava, Solve Interview Case Studies 10x Faster Using Dynamic Programming, Septhttps://www.analyticsvidhya.com/blog/2016/05/ase-studies-10x-faster-using-dynamic-programming/, 20 September 2019.

3.2.4 Convolutional Neural Network (CNN)

Convolutional Neural Network (*CNN*) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*.[18] *CNN* bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah image. Metode *CNN* ini digunakan untuk melatih sebuah sistem dalam mengenali akor pada setiap bar.

3.2.5 Pengaturan Direktori

Tujuan dari pengaturan direktori adalah agar penempatan *parenthesis* dan *child package* bagian *backend* dan *frontend* dapat berkomunikasi satu sama lain. Semua direktori *package* dibuat di dalam direktori berjenis Git. Direktori aplikasi ini dapat diunduh melalui *link* yang terdapat halaman *website* Git.

Direktori tersebut akan memiliki tiga buah direktori *child* (sub direktori) dengan nama "*app*", "*app_settings*", dan "*backend*". Selain ketiga sub direktori tersebut, terdapat enam buah *file* yaitu, "*.env*", "*.flaskenv*", "*.gitignore*", "*README.md*", "*requirements.txt*", dan "*run.py*".

Direktori "*app*" merupakan bagian dari pendekatan *frontend* yang bertujuan untuk menyediakan tampilan kepada pengguna dengan basis web. Sementara itu, direktori "*backend*" merupakan pendekatan *backend* yang bertujuan untuk menyediakan model dan *preprocessing* data masukan. Direktori "*app_settings*" bertujuan untuk melakukan inisialisasi pengaturan *Redis Server* dan *session id* dengan tujuan agar kedua pendekatan tersebut dapat berkomunikasi satu sama lain.

File "*README.md*" dibuat dengan tujuan memberikan penjelasan aplikasi secara singkat pada situs Github. File "*requirements.txt*" berfungsi untuk menuliskan pustaka *Python* yang dibutuhkan aplikasi ini. File "*requirements.txt*" dapat digunakan pengguna untuk melakukan pemasangan pustaka yang dibutuhkan aplikasi dengan memasukan perintah "*pip install -r requirements.txt*" setelah mengunduh direktori aplikasi dari Github. File terakhir yang ada pada direktori "*web*" adalah "*run.py*". File ini berfungsi untuk menjalankan keseluruhan aplikasi melalui perintah "*flask run*".

---

[18] Samuel Sena, Pengenalan Deep Learning Part7: Convolutional Neural Network (CNN), https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94, 7 September 2019.

3.2.6 Pendekatan *Backend*

Pendekatan *backend* pada aplikasi ini ditempatkan pada direktori "*backend*" dan dilengkapi dengan file "*__init__.py*" di dalamnya. Hal ini dilakukan agar direktori "*app*" milik pendekatan *frontend* dapat mengakses fungsi yang ada pada pendekatan *backend* dengan melakukan perintah import backend pada *script Python*. Model CNN pada aplikasi menggunakan 1655 citra *chromagram* sebagai data latih, dan 347 citra *chromagram* sebagai data validasi.

Pembuatan model menggunakan pustaka *Jupyter*, *Librosa*, dan *Keras*. Data latih berbentuk *wav* dan *mp3* yang dipisah dan dikelompokan setiap dua ketukan menggunakan pustaka *Librosa*. Setiap ketukan yang terbentuk berupa detik dengan tipe data *float*. Setiap detik ketukan kemudian disimpan ke dalam file berbentuk ".txt". Setelah membentuk file "*.txt*", dilakukan *labelling* akor pada masing-masing pasangan ketukan yang dibantu oleh Anastashia Aurelia, pelajar piano dari Associated Board of the Royal School of Music (ABRSM) yang sudah berada pada *Grade* 7.

Akor pada aplikasi berjumlah 24 yang terdiri dari 12 akor *major* dan 12 akor *minor*. Setelah *labelling* dilakukan, dilakukan visualisasi citra *spectogram MFCC* berdasarkan detik ketukan, dan akor yang ada pada file "*.txt*". Visualisasi tersebut kemudian diletakkan di dalam direktori masing-masing kelas untuk dijadikan data latih. Sebanyak 20 persen dari jumlah masing-masing kelas dijadikan data validasi. Data latih dimasukan ke dalam direktori "*data_train*" dan data validasi ke dalam direktori "*data_test*". Banyak citra data latih dan data validasi masing-masing kelas dapat dilihat pada Tabel 3.1.

Tabel 3.1 Citra Data Latih Dan Data Validasi

Kelas	Data Latih	Data Validasi
A	48	12
A#	50	9
A#m	248	15
Am	84	27
B	61	14
Bm	53	13

(Sambungan)

C	76	16
C#	44	11
C#m	77	16
Cm	85	22
D	79	25
D#	57	13
D#m	24	8
Dm	48	12
Em	56	13
F	56	13
F#	67	13
F#m	54	13
Fm	74	17
G	131	17
G#	54	19
G#m	54	13
Gm	75	16
Jumlah	1655	347

Setelah data latih dan data validasi ditempatkan pada direktori dan kelasnya masing-masing, tahap selanjutnya membuat model *CNN* menggunakan pustaka *Keras*. Proses pelatihan menggunakan *Jupyter Notebook*. Model dilakukan proses kompilasi menggunakan *optimizer RMSprop* yang ada pada pustaka *Keras* dengan nilai *learning rate* sebesar 0.00001. Tahap selanjutnya dalam pembuatan model adalah dengan memasukan *path* direktori data latih dan data validasi ke dalam *Keras*. Dimensi citra yang digunakan sebagai data masukan ke dalam model memiliki dimensi 200x200 dan berbentuk tiga dimensi (*RGB*).

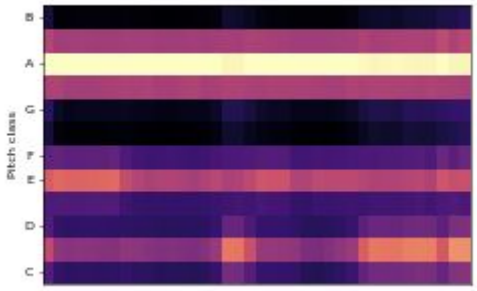
Ukuran *batch* data latih dan data validasi masing-masing adalah sebesar 32 data dan 257 data dengan mode *shuffle* bernilai *true*, sehingga data latih diproses sebanyak 32 data

yang dipilih secara acak oleh *Keras* dengan jumlah pengulangan sebanyak banyak data latih dibagi besar *batch* yang disebut sebagai *steps per epoch*. Sementara itu, *steps per epoch* yang dipilih untuk data validasi adalah sebanyak dua kali dengan mode *shuffle* bernilai true. Langkah selanjutnya dalam proses pembuatan model adalah dengan melakukan proses pelatihan. Model yang dilatih disimpan di dalam file berbentuk ".h5" agar dapat digunakan kembali. Proses pelatihan berhasil mendapatkan nilai *accuracy* 87 persen, *loss accuracy* 31 persen, *validation loss* 1.7 persen, dan *validation accuracy* sebanyak 96 persen.

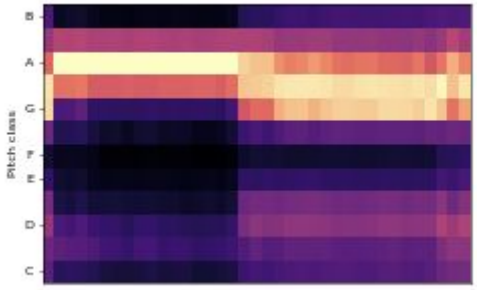
Selanjutnya pengaturan *session id*. *Session id* dibentuk pada saat pengguna pertama kali mengunjungi halaman web aplikasi dan juga setelah pengguna menggunakan fitur pengenalan akor. *Session id* yang dibentuk berguna pada penamaan data masukan pengguna. Penamaan data masukan yang unik berdasarkan *session id* diperlukan agar data pada saat aplikasi diakses oleh pengguna lebih dari satu tidak saling tumpang tindih.

Redis Server diinisialisasi oleh *backend* terlebih dahulu pada saat aplikasi dijalankan pertama kali. *Redis Server* pada aplikasi berguna untuk menyimpan *session id* yang sedang aktif agar dapat diakses baik dari segi *backend* maupun *frontend.* Selain untuk menyimpan *session id*, *Redis Server* juga berfungsi sebagai media komunikasi antara *frontend* dengan *backend* pada saat pengguna memilih fitur. Fitur-fitur yang dapat dipilih oleh pengguna adalah memilih data masukan dari direktori pengguna atau melakukan rekaman secara langsung. Perbedaan fitur ini diperlukan suatu variabel pembeda agar fungsi yang dijalankan oleh *backend* tepat sesuai pilihan pengguna.

Model yang sudah dilatih dapat diakses melalui file ".h5" yang sudah disimpan terlebih dahulu pada saat proses pelatihan model, sehingga pada saat pengguna melakukan fitur pengenalan dalam aplikasi, model dapat digunakan kembali. Data masukan pengguna pada saat menggunakan fitur pengenalan pertama-tama dibentuk ".txt" berdasarkan pasangan detik setiap ketukan pada data. Setiap pasangan detik tersebut dibentuk citra *spectogram*. Masing-masing dari citra yang dibentuk kemudian dilakukan proses pengenalan oleh model. Contoh data latih dan data validasi dapat dilihat pada Gambar 3.2 dan Gambar 3.3. Untuk mengetahui akor pada gambar *chromagram*, yang perlu dilihat adalah baris yang paling terang (berwarna putih kekuning-kuningan). Pada data latih, dapat dilihat bahwa akor A yang paling terang dan diikuti oleh akor E dan akor Dm. Setelah dilakukan validasi, ternyata benar bahwa akor yang dilatih adalah akor A. Hal tersebut dapat diketahui dari baris yang paling terang pada data validasi yakni akor A.



Gambar 3.1 Contoh Data Latih Akor A



Gambar 3.2 Contoh Data Validasi Akor A

BAB IV
HASIL DAN PEMBAHASAN

Hasil dari kerangka penelitian adalah mengukur akurasi data akor yang dihasilkan pada *Chord Recognition* dengan menggunakan metode *CNN*, kemudian menarik kesimpulan dari hasil yang telah didapatkan. Pada bab ini disajikan beberapa hal yang berkaitan dengan proses, hasil, dan pembahasan hasil penelitian, diantaranya, hasil pengujian prediksi model menggunakan *Confusion Matrix*, dan hasil pengujian akurasi akor beserta pembahasannya.

4.1 Pengujian Prediksi Model Menggunakan *Confusion Matrix*

Confusion matrix berbentuk tabel yang sering digunakan untuk memberikan keterangan performa dari model klasifikasi terhadap sekumpulan data tes.[19] Pada aplikasi sistem terdapat 1192 data latih dan 247 data validasi dengan kelas sebanyak 24. Terdapat tiga variabel untuk melakukan perhitungan tingkat akurasi terhadap klasifikasi yang dilakukan oleh model. Ketiga variabel itu adalah *precision*, *recall*, dan *F1-score*. Untuk mendapatkan nilai dari ketiga variabel tersebut dibutuhkan *confusion matrix* yang memiliki nilai-nilai sebagai berikut:[20]

1. *True Negative* (TN)

   TN merupakan jumlah prediksi benar untuk data salah.
2. *True Positive* (TP)

   TP merupakan jumlah prediksi benar untuk data benar.
3. *False Negative* (FN)

   FN merupakan jumlah prediksi salah untuk data salah.
4. *False Positive* (FP)

   FP merupakan jumlah prediksi salah untuk data benar.

---

[19] Data School, Simple Guide to Confusion Matrix, https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/, 14 Desember 2019.

[20] Muthu, Understanding the Classification Report through Sklearn, https://muthu.co/understanding-the-classification-report-in-sklearn/, 15 Desember 2019.

4.2 Hasil dan Pembahasan

4.2.1 Hasil Pengujian Model

Hasil *confusion matrix* pada pengujian dapat dilihat pada Tabel 4.1. Berdasarkan nilai dari *confusion matrix* tersebut, maka nilai *precision*, *recall*, dan *F1-score* dapat dilihat pada Tabel 4.2. Hasil dari pengujian, nilai akurasi yang didapatkan sebesar 89% terhadap 72 data yang diuji. Pada Tabel 4.2 dapat dilihat bahwa prediksi perbedaan akor dapat mudah dilakukan, sementara itu prediksi jenis akor lebih sukar dibedakan. Sebagai contoh untuk kelas akor A#m terdapat kesalahan prediksi menjadi kelas akor A#. Terdapat juga kesalahan prediksi pada kelas akor G#. Model memprediksi kelas akor G# menjadi kelas akor G dan kelas akor G#m. Hal ini dikarenakan akor *major* dan *minor* hanya memiliki perbedaan nada kedua sebesar setengah langkah. Akor A# (A# *major*) terdiri dari nada A#, D, dan F, sedangkan akor A#m (A# *minor*) terdiri dari nada A#, C#, dan F. Perbedaan nada yang tidak terlalu signifikan mengakibatkan sistem sukar membedakan akor *minor* dan *major*.

Tabel 4.1 Hasil *Confusion Matrix*

	A	A#	A#m	Am	B	Bm	C	C#	C#m	Cm	D	D#m	Dm	E	Em	F	F#	F#m	Fm	G	G#	G#m	Gm
A	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A#	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A#m	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Am	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bm	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	3	0	0	1	0	0	0	1	0	0	0	0	0	0	2	0	0
C#	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C#m	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cm	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
D#m	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Dm	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
Em	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
F	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0	0
F#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
F#m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
Fm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
G#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0
G#m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0
Gm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Tabel 4.2 Perhitungan *Precision*, *Recall*, *F1-score*, dan *Support*

	precision	recall	f1-score	support
A	1.00	1.00	1.00	3
A#	0.67	1.00	0.80	2
A#m	1.00	0.75	0.86	4
Am	0.67	1.00	0.80	2
B	1.00	1.00	1.00	6
Bm	1.00	1.00	1.00	3
C	1.00	0.43	0.60	7
C#	1.00	1.00	1.00	3
C#m	1.00	1.00	1.00	3
Cm	0.67	1.00	0.80	2
D	1.00	1.00	1.00	3
D#m	1.00	1.00	1.00	3
Dm	1.00	1.00	1.00	3
E	0.67	1.00	0.80	2
Em	1.00	1.00	1.00	3
F	1.00	0.60	0.75	5
F#	1.00	1.00	1.00	3
F#m	1.00	1.00	1.00	3
Fm	0.67	1.00	0.80	2
G	1.00	1.00	1.00	3
G#	0.00	0.00	0.00	0
G#m	1.00	0.75	0.86	4
Gm	1.00	1.00	1.00	3
accuracy			0.89	72

4.2.2 Hasil Pengujian Akurasi Akor C Major dan C Minor

Terdapat 30 visualisasi yang didapat dari data lagu masukan akor C dengan nilai *F1* yang didapat sebesar 95 persen. Terdapat beberapa kelas akor lain yang diprediksi oleh model yaitu, kelas akor C *minor* dan kelas akor F *major*. Kesalahan ini dikarenakan akor C *major* dan akor C *minor* hanya memiliki satu buah nada yang berbeda. Akor C *major* terdiri dari nada C, E, dan G. Sementara akor C *minor* terdiri dari nada C, D#, dan G. Frekuensi nada E adalah 329.63 Hz sedangkan frekuensi nada D# adalah 311.13 Hz. Kedua frekuensi tersebut saling berdekatan sehingga model salah mengenali akor C *major* sebagai akor C *minor* sebanyak dua kali. Detil hasil pengujian pada kelas akor C *major* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor C *Major*

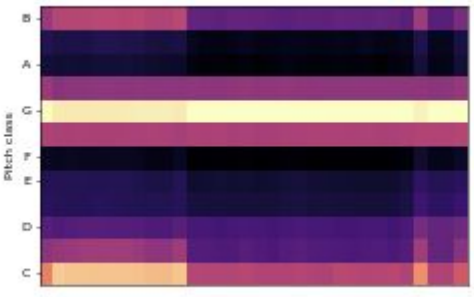
	Precision	Recall	F1-Score	Support
C	0,90	1,00	0,95	27
Cm	0,00	0,00	0,00	2
F	0,00	0,00	0,00	1
Macro Avg	0,30	0,33	0,32	-

Weighted Avg	0,82	0,90	0,85	-

Pengujian akor C *minor* mendapatkan nilai *F1* sebesar 77 persen dari total 45 data masukan. Model salah memprediksi akor C *minor* sebanyak 12 prediksi untuk akor C *major*. Hal ini dikarenakan kemiripan antara komposisi nada yang membentuk akor C *minor* dan C *major* saling berdekatan. Selain kesalahan prediksi C *major* dari model, terdapat juga kesalahan prediksi pada kelas akor D# *major*. Komposisi nada penyusun akor C *minor* adalah C, D#, dan G. Komposisi nada penyusun akor D# *major* adalah D#, G, dan A#. Kesamaan komposisi nada penyusun akor C *minor* dan D# *major* adalah nada D# dan G. Perhitungan *precision*, *recall*, dan *F1-score* akor C *minor* dapat dilihat pada Tabel 4.4. Untuk data latih akor C *major* dapat dilihat pada Gambar 4.1 dan untuk data validasi C *major* dapat dilihat pada Gambar 4.2.

Tabel 4.4 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor C *Minor*

	Precision	**Recall**	**F1-Score**	**Support**
Cm	0,62	1,00	0,77	28
C	0,00	0,00	0,00	28
D#	0,00	0,00	0,00	5
Macro Avg	0,21	0,33	0,26	-
Weighted Avg	0,39	0,62	0,48	-

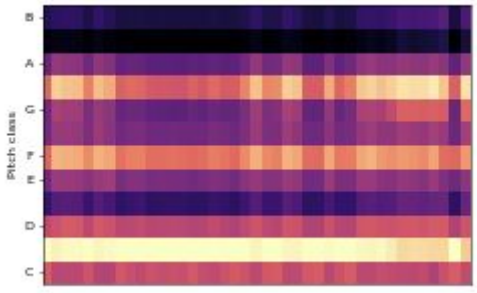


Gambar 4.1 Data Latih Akor C *Major*

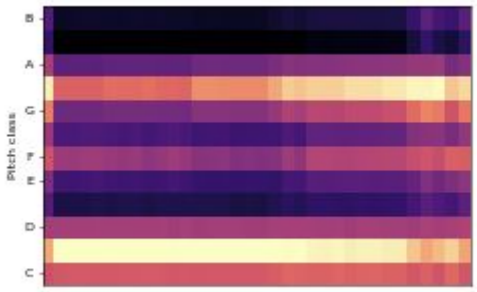
Gambar 4.2 Data Validasi Akor C *Major*

4.2.3 Hasil Pengujian Akurasi Akor C# Major dan C# Minor

Nilai F1 yang diperoleh dari pengujian akor C# *major* adalah sebesar 91 persen dari total 49 data masukan. Model salah memprediksi akor C# *major* sebagai akor C# *minor* dan akor F *minor*. Komposisi nada penyusun akor C# *major* adalah C#, F, dan G#. Komposisi nada penyusun akor C# *minor* adalah C#, E, dan G#. Komposisi nada penyusun akor F *minor* adalah F, G#, dan C. Kesalahan terbanyak dari prediksi model adalah pada akor F *minor* dimana terdapat kemiripan komposisi nada penyusun antara kedua akor tersebut. Besar frekuensi nada C adalah 261.63 Hz, sedangkan besar frekuensi nada C# adalah 277.18 Hz. Kedekatan antara kedua nada tersebut memuat model salah memprediksi. Rincian hasil pengujian terhadap akor C# *major* dapat dilihat pada Tabel 4.5. Untuk data latih akor C# *major* dapat dilihat pada Gambar 4.3 dan untuk data validasi C# *major* dapat dilihat pada Gambar 4.4.



Gambar 4.3 Data Latih C# *Major*



Gambar 4.4 Data Validasi C# *Major*

Tabel 4.5 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor C# *Major*

	Precision	Recall	F1-Score	Support
C#	0,84	1,00	0,91	41
C#m	0,00	0,00	0,00	3
Fm	0,00	0,00	0,00	5
Macro Avg	0,28	0,33	0,30	-
Weighted Avg	0,70	0,84	0,76	-

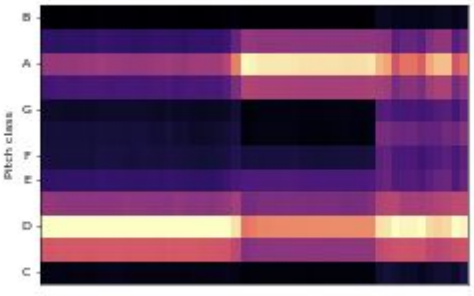
Baris yang terang pada data latih C# *major* adalah baris akor C# *major*, akor F minor, dan akor C#m. Setelah dilakukan validasi, hasil baris yang terang adalah baris akor C# *major*, A *minor*, dan F. Nilai F1 yang diperoleh dari pengujian kelas akor C# *minor* adalah sebesar 80 persen. Model salah memprediksi akor C# *minor* sebagai akor C# *major* dan E *major*. Kesalahan terbanyak prediksi adalah akor E *major* yaitu sebesar 12 kesalahan prediksi. Komposisi nada penyusun akor C# *minor* adalah C#, E, dan G#. Komposisi nada penyusun akor E *major* adalah E, G#, dan B. Terdapat kesamaan sebanyak dua nada yang ada pada akor C# *minor* dan E *major*, yaitu nada E dan G#. Kesalahan pengenalan akor C# *minor* lainnya adalah kesalahan prediksi kelas akor C# *major* Kesamaan penyusun nada tersebut membuat model salah memprediksi. Rincian hasil pengujian terhadap akor C# *minor* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor C# *Minor*

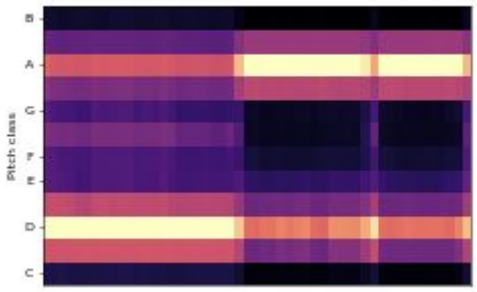
	Precision	Recall	F1-Score	Support
C#m	0,67	1,00	0,80	41
C#	0,00	0,00	0,00	8
Fm	0,00	0,00	0,00	12
Macro Avg	0,22	0,33	0,27	-
Weighted Avg	0,45	0,67	0,54	-

4.2.4 Hasil Pengujian Akurasi Akor D Major dan D Minor

Nilai F1 yang diperoleh dari pengujian kelas akor D *major* adalah sebesar 83 persen dari total 37 data. Pengujian pada akor D *major* berhasil memperoleh nilai *F1-score* sebesar 83 persen. Model salah memprediksi akor D *major* sebagai akor D *minor* dan akoe B *major*. Kesalahan pengenalan akor *major* dan *minor* terjadi kembali pada pengujian ini, sedangkan kesalahan mengenali akor D *major* sebagai B *major* dikarenakan adanya kedekatan antara kedua komposisi nada penyusun akor-akor tersebut. Komposisi nada penyusun akor D *major* adalah D, F#, dan A. Komposisi nada penyusun akor B *major* adalah B, D#, dan F#. Frekuensi nada D dan D# berturut-turut adalah 293.66 Hz dan 311.13 Hz. Rincian hasil pengujian terhadap akor D *major* dapat dilihat pada Tabel 4.7. Untuk data latih akor D *major* dapat dilihat pada Gambar 4.5 dan untuk data validasi D *major* dapat dilihat pada Gambar 4.6.



Gambar 4.5 Data Latih D *Major*



Gambar 4.6 Data Validasi D *Major*

Tabel 4.7 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor D *Major*

	Precision	Recall	F1-Score	Support
D	0,70	1,00	0,83	26
B	0,00	0,00	0,00	3
Dm	0,00	0,00	0,00	8
Macro Avg	0,23	0,33	0,28	-
Weighted Avg	0,49	0,70	0,58	-

Nilai F1 yang diperoleh dari hasil pengujian pengenalan kelas akor D *minor* yaitu sebesar 75 persen dari total 48 data. Model salah memprediksi akor D *minor* sebagai kelas akor A# *major*, D *major*, dan F *major*. Komposisi nada penyusun akor D *minor* adalah D, F, dan A. Komposisi nada penyusun A# *major* adalah A#, D, dan F. Komposisi nada penyusun F *major* adalah F, A, dan C. Kesalahan pengenalan untuk kelas-kelas tersebut dikarenakan komposisi nada penyusun antara masing-masing akor sama dan berdekatan. Rincian hasil pengujian terhadap akor D *minor* dapat dilihat pada Tabel 4.8

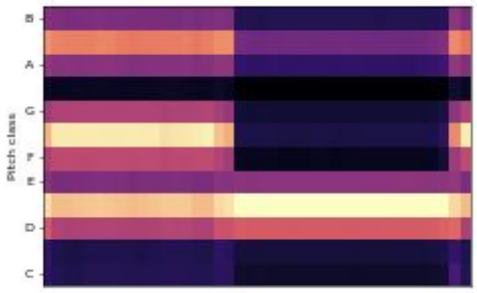
Tabel 4.8 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor D *Minor*

	Precision	Recall	F1-Score	Support
D	0,70	1,00	0,83	26
B	0,00	0,00	0,00	3
Dm	0,00	0,00	0,00	8
Macro Avg	0,23	0,33	0,28	-

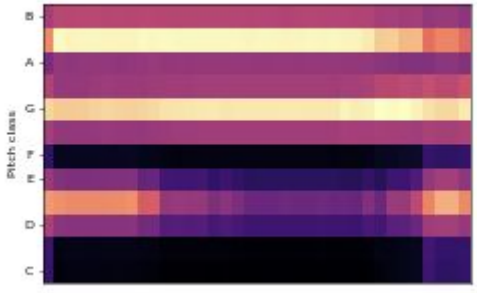
Weighted Avg	0,49	0,70	0,58	-

4.2.5 Hasil Pengujian Akurasi Akor D# Major dan D# Minor

Nilai F1 yang diperoleh dari hasil pengujian kelas akor D# *major* yaitu sebesar 81 persen dari total 22 data observasi akor. Model salah mengenali kelas akor D# *major* sebagai kelas akor B *major*, D# *minor*, dan G *minor*. Kesalahan pengenalan paling banyak adalah mengenali kelas akor G *minor* sebagai akor D# *major*. Komposisi nada penyusun akor D# *major* adalah D#, G, dan A#. Komposisi nada penyusun akor G *minor* adalah G, A#, dan D. Kemiripan komposisi kedua akor tersebut membuat model salah memprediksi akor D# *major* sebagai akor G *minor* sebanyak empat pengenalan. Rincian hasil pengujian terhadap akor D# *major* dapat dilihat pada Tabel 4.9. Untuk data latih akor D# *major* dapat dilihat pada Gambar 4.7 dan untuk data validasi D# *major* dapat dilihat pada Gambar 4.8.



Gambar 4.7 Data Latih D# *Major*



Gambar 4.8 Data Latih D# *Major*

Tabel 4.9 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor D# *Major*

	Precision	Recall	F1-Score	Support
D#	0,68	1,00	0,81	15
B	0,00	0,00	0,00	2
D#m	0,00	0,00	0,00	1
Gm	0,00	0,00	0,00	4
Macro Avg	0,17	0,25	0,20	-
Weighted Avg	0,46	0,68	0,55	-

Nilai F1 yang diperoleh dari hasil pengujian kelas akor D# *minor* adalah sebesar 82 persen dari total 62 data observasi. Model salah memprediksi akor D# *minor* sebagai akor F# *major* dan G# *major*. Kesalahan terbanyak adalah mengenali akor F# *major*. Pada pengujian akor ini, tidak terdapat kesalahan pengenalan akor *minor* sebagai akor *major* dalam *root* yang sama. Komposisi nada penyusun akor D# *minor* adalah D#, F#, dan A#. Komposisi nada penyusun akor F# *major* adalah F#, A#, dan C#. Komposisi nada penyusun akor G# *major* adalah G#, C, dan D#. Ketiga kelas akor tersebut memiliki nada penyusun akor masing-masing yang mirip sehingga terdapat kesalahan pengenalan dari model pada sistem aplikasi. Rincian hasil pengujian terhadap akor D# *minor* dapat dilihat pada Tabel 4.10

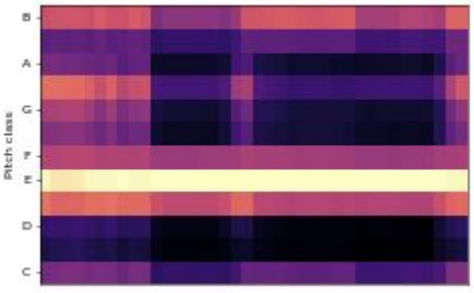
Tabel 4.10 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor D# *Minor*

	Precision	Recall	F1-Score	Support
D#m	0,69	1,00	0,82	43
F#	0,00	0,00	0,00	12
G#	0,00	0,00	0,00	7
Macro Avg	0,23	0,33	0,27	-
Weighted Avg	0,48	0,69	0,57	-

4.2.6 Hasil Pengujian Akurasi Akor E Major dan E Minor

Nilai F1 yang diperoleh dari hasil pengujian kelas akor E *major* adalah sebesar 89 persen dari total 21 data observasi akor. Kesalahan pada pengujian ini dikarenakan kemiripan antara akor *major* dan akor *minor* pada *root* yang sama. Komposisi nada penyusun akor E *major* adalah E, G#, dan G. Sedangkan, komposisi nada penyusun akor E *minor* adalah E, G, dan B. Komposisi nada penyusun yang sama antara akor E *major* dan E *minor* adalah pada naga E dan B. Selain memiliki kesamaan pada nada E dan B, nada G# dan G memiliki frekuensi yang dekat satu sama lain. Rincian hasil pengujian terhadap akor E *major* dapat dilihat pada Tabel 4.11. Untuk data latih akor E *major* dapat dilihat pada Gambar 4.9 dan untuk data validasi E *major* dapat dilihat pada Gambar 4.10.

Gambar 4.9 Data Latih E *Major*



Gambar 4.10 Data Validasi E *Major*

Tabel 4.11 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor E *Major*

	Precision	Recall	F1-Score	Support
E	0,81	1,00	0,89	17
Em	0,00	0,00	0,00	4

Macro Avg	0,40	0,50	0,45	-
Weighted Avg	0,66	0,81	0,72	-

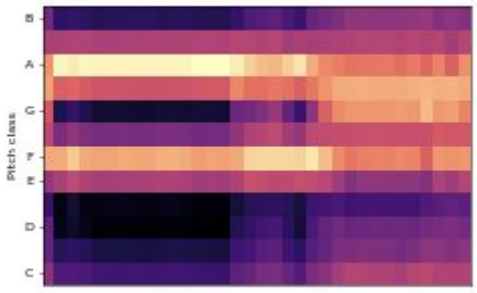
Nilai F1 yang diperoleh dari hasil pengujian kelas akor E *minor* adalah sebesar 84 persen dari total 18 data masukan observasi. Kesalahan yang didapat dari pengenalan oleh model sistem aplikasi adalah dikarenakan kemiripan antara akor *major* dan akor *minor*. Ciri kesalahan yang ada pada pengujian pengenalan kelas akor E *minor* sama dengan kesalahan pada kelas akor E *major* dimana ciri komposisi penyusun kedua akor tersebut memiliki keserupaan nada yang tidak jauh. Rincian hasil pengujian terhadap akor E *minor* dapat dilihat pada Tabel 4.12

Tabel 4.12 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor E *Minor*

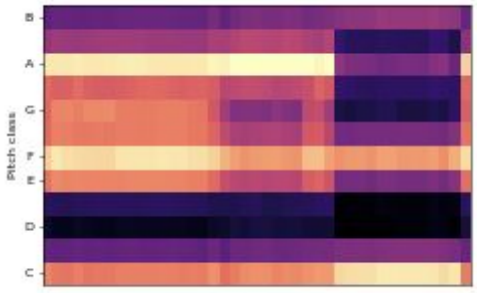
	Precision	**Recall**	**F1-Score**	**Support**
Em	0,72	1,00	0,84	13
E	0,00	0,00	0,00	5
Macro Avg	0,36	0,50	0,42	-
Weighted Avg	0,52	0,72	0,61	-

4.2.7 Hasil Pengujian Akurasi Akor F Major dan F Minor

Nilai F1 pada kelas ini sebesar 99 persen dari total 41 data. Model salah mengenali akor F *major* sebagai akor A# *minor*. Pada pengujian ini, model memiliki kesalahan pengenalan yang tidak biasa yaitu A# *minor* dimana korelasi susunan nada pembentuk akor F *major* dan A# *minor* tidak terlalu dekat. Komposisi akor F *major* terdiri dari nada F, A, dan C. Sedangkan, komposisi akor A# *minor* terdiri dari nada A#, D, dan F. Kesamaan komposisi nada yang ada pada kedua akor tersebut hanya pada nada F. Selain itu, terdapat juga komposisi kedekatan antara komposisi nada penyusun akor F *major* dan A# *minor* yaitu pada nada A dan A#. Kedua nada tersebut berdekatan dengan frekuensi nada A sebesar 440.00 Hz dan frekuensi nada A# sebesar 466.16 Hz. Rincian hasil pengujian terhadap akor F *major* dapat dilihat pada Tabel 4.13. Untuk data latih akor F *major* dapat dilihat pada Gambar 4.11 dan untuk data validasi F *major* dapat dilihat pada Gambar 4.12.



Gambar 4.11 Data Latih F *Major*



Gambar 4.12 Data Validasi F *Major*

Tabel 4.13 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor F *Major*

	Precision	**Recall**	**F1-Score**	**Support**
F	0,98	1,00	0,99	40
A#m	0,00	0,00	0,00	1
Macro Avg	0,49	0,50	0,49	-
Weighted Avg	0,95	0,98	0,96	-

Nilai F1 yang diperoleh dari hasil pengujian pengenalan kelas akor F *minor* adalah sebesar 85 persen dari total 35 data masukan observasi. Kesalahan yang didapatkan dari pengujian ini adalah salah mengenali kelas akor F *minor* sebagai akor F *major*. Kesalahan ini diakibatkan oleh kesamaan komposisi susunan akor antara akor *major* dan akor *minor*. Model

mengenali akor F *minor* sebagai akor F *major* yaitu sebanyak 9 data observasi. Rincian hasil pengujian terhadap akor F *minor* dapat dilihat pada Tabel 4.14.

Tabel 4.14 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor F *Minor*

	Precision	Recall	F1-Score	Support
Fm	0,74	1,00	0,85	26
F	0,00	0,00	0,00	9
Macro Avg	0,37	0,50	0,43	-
Weighted Avg	0,55	0,74	0,63	-

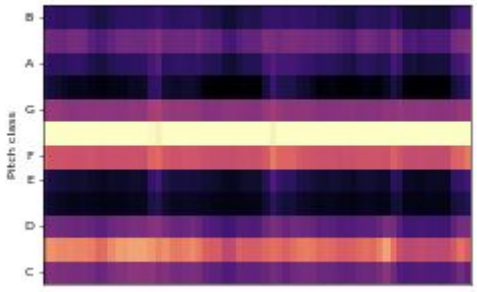
4.2.8 Hasil Pengujian Akurasi Akor F# Major dan F# Minor

Nilai F1 pada pengenalan kelas akor F# *major* sebesar 84 persen dari total 38 data. Kesalahan pengenalan pada pengujian ini adalah A# *major* dan B *major* dengan jumlah masing-masing sebesar 6 dan 9 kesalahan. Komposisi nada akor F# *major* adalah F#, A#, dan C#. Komposisi nada akor A# *major* adalah A#, D, dan F. Komposisi nada akor B *major* adalah B, D#, dan F#. Akor F# *major* dan akor A# *major* memiliki kemiripan komposisi pada nada A# *major* dan F dengan F#. Besar frekuensi nada F adalah 249.23 Hz, sedangkan besar frekuensi nada F# adalah 369.99 Hz. Kesalahan pengenalan pada pengujian ini cukup besar pada kelas B *major*. Kemiripan komposisi nada akor antara akor F# *major* dengan B *major* adalah pada nada F# dan A# dengan B. Besar frekuensi A# adalah 466.16 Hz, sedangkan besar frekuensi B adalah 493.88 Hz. Rincian hasil pengujian terhadap akor F# *major* dapat dilihat pada Tabel 4.15. Untuk data latih akor F# *major* dapat dilihat pada Gambar 4.13 dan untuk data validasi F# *major* dapat dilihat pada Gambar 4.14.

Tabel 4.15 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor F# *Major*

	Precision	Recall	F1-Score	Support
F#	0,72	1,00	0,84	38
A#	0,00	0,00	0,00	6
B	0,00	0,00	0,00	9
Macro Avg	0,23	0,33	0,28	-
Weighted Avg	0,51	0,72	0,60	-

Gambar 4.13 Data Latih F# *Major*



Gambar 4.14 Data Validasi F# *Major*

Nilai F1 pada kelas akor ini mendapatkan akurasi sebesar 100 persen dari total 21 data masukan observasi. Model berhasil melakukan pengenalan akor F# *minor* tanpa terdapat kesalahan seperti yang ada pada kelas lainnya. Rincian hasil pengujian terhadap akor F# *minor* dapat dilihat pada Tabel 4.16.

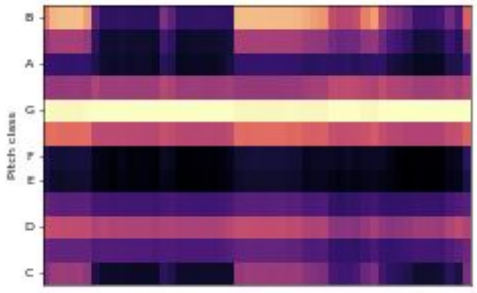
Tabel 4.16 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor F# *Minor*

	Precision	**Recall**	**F1-Score**	**Support**
F#m	1,00	1,00	1,00	21
Macro Avg	1,00	1,00	1,00	-
Weighted Avg	1,00	1,00	1,00	-

4.2.9 Hasil Pengujian Akurasi Akor G Major dan G Minor

Nilai F1 yang diperoleh dari pengujian terhadap kelas akor G *major* adalah sebesar 83 persen dari total 17 data masukan observasi. Model salah mengenali akor G *major* sebagai akor E *minor* dan akor G *minor*. Kemiripan yang ada antara akor *major* dan akor *minor* di dalam *root* yang sama terjadi pada pengujiian akor G *major*. Rincian hasil pengujian terhadap akor G *major* dapat dilihat pada Tabel 4.17. Untuk data latih akor G *major* dapat dilihat pada Gambar 4.15 dan untuk data validasi G *major* dapat dilihat pada Gambar 4.16.

Gambar 4.15 Data Latih G *Major*



Gambar 4.16 Data Validasi G *Major*

Tabel 4.17 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor G *Major*

	Precision	**Recall**	**F1-Score**	**Support**
G	0,71	1,00	0,83	12
Em	0,00	0,00	0,00	1
Gm	0,00	0,00	0,00	4
Macro Avg	0,24	0,33	0,28	-

Weighted Avg	0,50	0,71	0,58	-

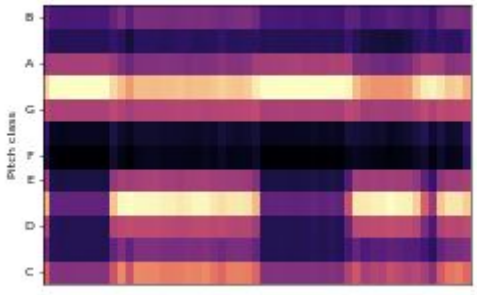
Pada pengujian akor G *minor*, Model berhasil memberikan pengenalan dengan nilai F1 sebesar 98 persen dari total 42 data. Pada pengujian ini, model memiliki kesalahan pengenalan yaitu akor C *major*. Akor G *minor* terdiri dari nada G, A#, dan D. Sedangkan, akor C *major* terdiri dari nada C, E, dan G. Kesamaan antara akor G *minor* dan akor C *major* hanya satu nada yaitu nada G. Kesalahan yang ada pada pengujian ini sama dengan kesalahan pada pengujian F *major*, dimana kesalahan pengenalan memiliki korelasi antara komposisi kedua akor yang tidak begitu jauh antara nilai akor sesungguhnya dan nilai akor pada pengenalan yang salah. Rincian hasil pengujian terhadap akor G *minor* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor G *Minor*

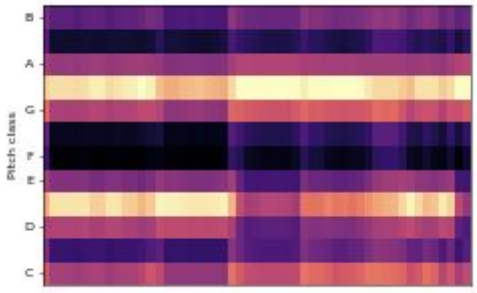
	Precision	**Recall**	**F1-Score**	**Support**
Gm	0,95	1,00	0,98	40
C	0,00	0,00	0,00	2
Macro Avg	0,48	0,50	0,49	-
Weighted Avg	0,91	0,95	0,93	-

4.2.10 Hasil Pengujian Akurasi Akor G# Major dan G# Minor

Pengujian ini berhasil memperoleh nilai F1 sebesar 92 persen dari total 35 data. Kesalahan pengenalan pada pengujian akor G# *major* adalah C *major* dan F *minor*. Komposisi nada pada akor G# *major* adalah G#, C, dan D#. Komposisi nada akor C *major* adalah C, E, dan G. Sedangkan, komposisi nada akor F *minor* adalah F, G#, dan C. Terdapat kemiripan komposisi akor G# *major* dengan F *minor* yaitu pada nada G# dan C. Sedangkan, kemiripan komposisi antara akor G# *major* dan C *major* terletak pada nada C dan G dengan G#. G memiliki besar frekuensi 392.00 Hz dan G# memiliki besar frekuensi 415.30 Hz. Kedua nada tersebut sangat dekat sehingga model CNN pada sistem aplikasi salah mengenali G# *major* dengan C *major*. Rincian hasil pengujian terhadap akor G# *major* dapat dilihat pada Tabel 4.19. Untuk data latih akor G# *major* dapat dilihat pada Gambar 4.17 dan untuk data validasi G# *major* dapat dilihat pada Gambar 4.18.



Gambar 4.17 Data Latih G# *Major*



Gambar 4.18 Data Validasi G# *Major*

Tabel 4.19 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor G# *Major*

	Precision	**Recall**	**F1-Score**	**Support**
Gm	0,86	1,00	0,92	30
Fm	0,00	0,00	0,00	4
F	0,00	0,00	0,00	1
Macro Avg	0,29	0,33	0,31	-
Weighted Avg	0,73	0,86	0,79	-

Nilai F1 yang diperoleh dari pengujian kelas akor G# *minor* adalah sebesar 86 persen dari total 25 data masukan observasi. Model salah memprediksi akor G# *minor* sebagai akor

G# *major* dikarenakan kemiripan antara akor *major* dan *minor* dalam *root* yang sama. Kesalahan pengenalan lainnya adalah mengenali kelas akor G# *minor* sebagai kelas akor B *major*. Komposisi nada penyusun akor G# *minor* adalah G#, B, dan D#. Komposisi nada penyusun akor B *major* adalah B, D#, dan F#. Terdapat dua buah nada yang sama antara kedua penyusun akor tersebut. Rincian hasil pengujian terhadap akor G# *minor* dapat dilihat pada Tabel 4.20.

Tabel 4.20 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor G# *Minor*

	Precision	Recall	F1-Score	Support
G#m	0,76	1,00	0,86	19
B	0,00	0,00	0,00	4
C#m	0,00	0,00	0,00	2
Macro Avg	0,25	0,33	0,29	-
Weighted Avg	0,58	0,76	0,66	-

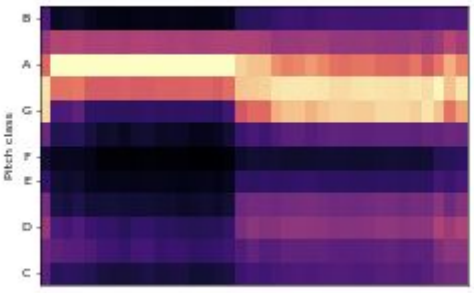
4.2.11 Hasil Pengujian Akurasi Akor A Major dan A Minor

Nilai F1 yang didapat dari pengujian pengenalan akor A *major* adalah sebesar 98 persen dari total 32 data masukan observasi. Pada pengeujian kelas akor ini, kesalahan pengenalan yang didapat sebanyak satu buah data observasi. Model salah mengenali akor A *major* sebagi akor A *minor* dikarenakan kemiripan antara nada penyusun akor *major* dan akor *minor* dalam *root* yang sama. Komposisi nada penyusun akor A *major* adalah A, C#, dan E. Sedangkan, komposisi nada penyusun akor A *minor* adalah A, C, dan E. Frekuensi nada C# adalah 277.18 Hz dan frekuensi nada C adalah 261.63 Hz. Kedekatan kedua frekuensi tersebut membuat perbedaan kedua kelas akor yang dekat antara satu sama lain dan menyebabkan model salah memprediksi. Rincian hasil pengujian terhadap akor A *major* dapat dilihat pada Tabel 4.21. Untuk data latih akor A *major* dapat dilihat pada Gambar 4.19 dan untuk data validasi A *major* dapat dilihat pada Gambar 4.20.

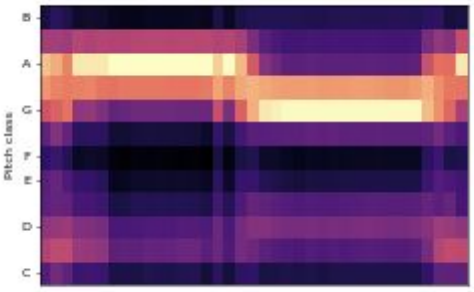
Tabel 4.21 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor A *Major*

	Precision	Recall	F1-Score	Support
A	0,97	1,00	0,98	31

Am	0,00	0,00	0,00	1
Macro Avg	0,48	0,50	0,49	-
Weighted Avg	0,94	0,97	0,95	-



Gambar 4.19 Data Latih A *Major*



Gambar 4.20 Data Validasi A *Major*

Nilai F1 yang didapat dari pengujian pengenalan kelas akor A *minor* adalah sebesar 80 persen dari total 24 data masukan observasi. Model salah mengenali akor A *minor* sebagai akor A *major* sebanyak 6 data observasi. Selain kesalahan tersebut, model juga salah mengenali akor A *minor* sebagai akor C *major* sebanyak dua data observasi. Kesalahan pengenalan akor A *minor* pada pengujian ini dikarenakan kemiripan komposisi nada susunan akor-akor tersebut. Rincian hasil pengujian terhadap akor A *minor* dapat dilihat pada Tabel 4.22.

Tabel 4.22 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor A *Minor*

	Precision	**Recall**	**F1-Score**	**Support**
Am	0,67	1,00	0,80	16
A	0,00	0,00	0,00	6

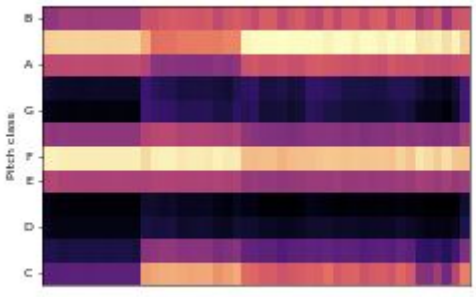
C	0,00	0,00	0,00	2
Macro Avg	0,22	0,33	0,27	-
Weighted Avg	0,44	0,67	0,55	-

4.2.12 Hasil Pengujian Akurasi Akor A# Major dan A# Minor

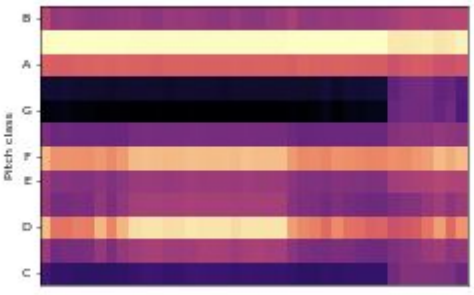
Nilai F1 yang diperoleh dari pengujian pengenalan kelas akor A# *major* adalah sebesar 82 persen dari total 43 data masukan observasi. Model salah mengenali akor A# *major* sebagai akor A# *minor* dan akor D *minor*. Komposisi nada penyusun akor A# *major* adalah A#, D, dan F. Komposisi nada penyusun akor A# *minor* adalah A#, C#, dan F. Komposisi nada penyusun akor D *minor* adalah D, F, dan A. Kesamaan komposisi nada akor-skor tersebut membuat perbedaan ciri antar akor tersebut tidak jauh dan menyebabkan model salah mengenali akor A# *major* sebanyak 30 persen pada pengujian ini. Rincian hasil pengujian terhadap akor A# *major* dapat dilihat pada Tabel 4.23. Untuk data latih akor A# *major* dapat dilihat pada Gambar 4.21 dan untuk data validasi A# *major* dapat dilihat pada Gambar 4.22.

Tabel 4.23 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor A# *Major*

	Precision	**Recall**	**F1-Score**	**Support**
A#	0,70	1,00	0,82	30
A#m	0,00	0,00	0,00	8
Dm	0,00	0,00	0,00	5
Macro Avg	0,23	0,33	0,27	-
Weighted Avg	0,49	0,70	0,57	-



Gambar 4.21 Data Latih A# *Major*



Gambar 4.22 Data Validasi A# *Major*

Nilai F1 yang didapat dari pengujian pengenalan kelas akor A# *minor* adalah sebesar 100 persen dari total 41 data masukan observasi. Dengan ini, model dengan baik memberikan prediksinya terhadap kelas akor A# *minor* tanpa mengalami kesalahan pengenalan yang paling banyak dialami oleh kelas-kelas lainnya yaitu kesalahan mengenali akor *major* sebagai akor *minor* ataupun sebaliknya. Rincian hasil pengujian terhadap akor A# *minor* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor A# *Minor*

	Precision	**Recall**	**F1-Score**	**Support**
A#m	1,00	1,00	1,00	41
Macro Avg	1,00	1,00	1,00	-
Weighted Avg	1,00	1,00	1,00	-

4.2.13 Hasil Pengujian Akurasi Akor B Major dan B Minor

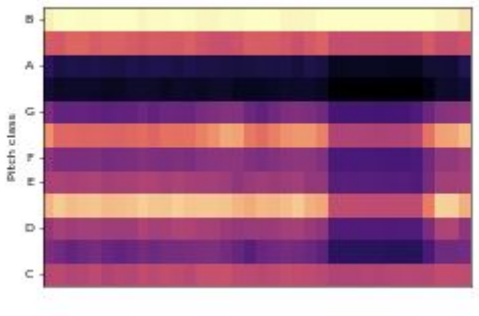
Nilai F1 yang didapat dari hasil pengujian pengenalan kelas akor B *major* adalah sebesar 86 persen dari total 54 data masukan observasi. Model salah memprediksi kelas akor B *major* sebagai akor D# *minor* dan G *minor*. Komposisi nada penyusun akor B *major* adalah B, D#, dan F#. Komposisi nada penyusun akor D# *minor* adalah D#, F#, dan B#. Komposisi nada penyusun akor G *minor* adalah G, A#, dan D. Kesamaan komposisi nada penyusun

antara akor B *major* dan akor D# *minor* adalah pada nada D# dan F#. Sementara itu, tidak ada kesamaan pada komposisi penyusun akor B *major* dan akor G *minor*. Namun, kedua akor tersebut mempunyai komposisi nada penyusun akor yang berdekatan. Nada-nada tersebut adalah A#, D, dan G. Ketiga nada yang ada didalam akor G *minor* semuanya berdekatan dengan ketiga nada yang ada pada akor B *major*. Kedekatan ini yang membuat model salah melakukan prediksi sebanyak 6 buah data observasi. Rincian hasil pengujian terhadap akor B *major* dapat dilihat pada Tabel 4.25. Untuk data latih akor B *major* dapat dilihat pada Gambar 4.23 dan untuk data validasi B *major* dapat dilihat pada Gambar 4.24.

Tabel 4.25 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor B *Major*

	Precision	**Recall**	**F1-Score**	**Support**
B	0,76	1,00	0,86	41
D#m	0,00	0,00	0,00	7
Gm	0,00	0,00	0,00	6
Macro Avg	0,25	0,33	0,29	-
Weighted Avg	0,58	0,76	0,66	-

Gambar 4.23 Data Latih B *Major*



Gambar 4.24 Data Validasi B *Major*

Nilai F1 yang didapat dari hasil pengujian kelas akor B *minor* adalah sebesar 86 persen dari 45 data observasi pengenalan. Terdapat kesalahan pengenalan akor B *minor* pada model yaitu, mengenali akor B *minor* sebagai akor D *major* dan akor B *major*. Kesalahan mengenali akor B *minor* menjadi akor B *major* dikarenakan kemiripan ciri akor *major* dan *minor* pada *root* yang sama. Komposisi nada penyusun akor B *minor* adalah B, D, dan F#. Komposisi nada penyusun akor D *major* adalah D, F#, dan A. Perbedaan komposisi nada penyusun akor B *minor* dan akor D *major* hanya pada nada B dan nada A. Kemiripan ciri komposisi inilah yang membuat model salah mengenali akor B *minor* sebagai akor D *major*. Rincian hasil pengujian terhadap akor B *minor* dapat dilihat pada Tabel 4.26.

Tabel 4.26 Perhitungan *Precision*, *Recall*, dan *F1-Score* Akor B *Minor*

	Precision	**Recall**	**F1-Score**	**Support**
Bm	0,76	1,00	0,86	34
B	0,00	0,00	0,00	7
D	0,00	0,00	0,00	4
Macro Avg	0,25	0,33	0,29	-
Weighted Avg	0,57	0,76	0,65	-

4.2.14 Hasil Pengujian Pengenalan Akor Data Lagu

Pengujian selanjutnya adalah menggunakan lagu “Fly Me To The Moon”. Akor-akor yang digunakan pada lagu “Fly Me To The Moon” adalah A *minor*, C *major*, D *minor*, E *minor*, F *major*, dan G *major*. Lagu ini dimainkan dengan nada dasar pada C *major* dengan

ketukan 4/4. Hasil rata-rata nilai F1 yang didapat dari pengujian pengenalan akor lagu "Fly Me To The Moon" adalah sebesar 59,33 persen dari total 64 data observasi pengenalan.

Nilai F1 pengenalan untuk kelas akor A *minor*, C *major*, D *minor*, E *minor*, F *major*, dan G *major* berturut-turut adalah 67 persen, 42 persen, 77 persen, 55 persen, 48 persen, dan 67 persen. Rincian hasil akurasi pengujian pengenalan akor lagu "Fly Me To The Moon" dapat dilihat pada Tabel 4.27 dan jumlah kesalahan prediksi pengujian pengenalan akor lagu "Fly Me To The Moon" dapat dilihat pada Tabel 4.28.

Tabel 4.27 Perhitungan *Precision*, *Recall*, dan *F1-Score* Pengujian Lagu "Fly Me To The Moon"

Y True	Precision	Recall	F1-Score	Total
Am	1,00	0,44	0,61	16
C	0,71	0,50	0,59	10
Dm	1,00	0,50	0,67	4
Em	1,00	0,60	0,75	5
F	0,55	0,35	0,43	17
G	1,00	0,62	0,77	29
Macro Avg	0,38	0,22	0,27	-
Weighted Avg	0,87	0,51	0,64	-

Tabel 4.28 Jumlah Kesalahan Prediksi Pengujian Pengenalan Akor Lagu "Fly Me To The Moon"

Y True	Y False	Jumlah Kesalahan
Am	A	7
	C	2
C	Cm	3
	F	2
Dm	D	1
	F	1
Em	E	2
F	Fm	7
	C	4
G	Gm	18

4.3.15 Hasil Rinci Pengujian

Dari pengujian akurasi yang telah dilakukan, dapat disimpulkan bahwa pengujian akurasi pada masing-masing kelas memberikan nilai F1 yang cukup baik. Besar nilai F1 pada masing-masing kelas dapat dilihat pada Tabel 4.29.

Tabel 4.29 Rincian Nilai F1 Keseluruhan Pengujian Pengenalan Akor Berdasarkan *Root*

Nomor	Kelas Akor	Nilai F1
1	C	0,95
2	Cm	0,77
3	C#	0,91
4	C#m	0,80
5	D	0,83
6	Dm	0,75
7	D#	0,81
8	D#m	0,82
9	E	0,89
10	Em	0,84
11	F	0,99
12	Fm	0,85
13	F#	0,84
14	F#m	1,00
15	G	0,83
16	Gm	0,98
17	G#	0,92
18	G#m	0,86
19	A	0,98
20	Am	0,80
21	A#	0,82
22	A#m	1,00
23	B	0,86

24	Bm	0,86
Rata-rata		0,87

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pengujian akurasi yang telah dilakukan, dapat disimpulkan bahwa pengujian akurasi pada masing-masing kelas memberikan nilai F1 yang cukup baik. Nilai F1 yang paling besar diperoleh pada pengujian akor F# *minor* dan A# *minor* yaitu sebesar 100 persen. Nilai F1 yang paling kecil diperoleh pada pengujian akor D *minor* yaitu sebesar 75 persen. Rata-rata nilai F1 dari keseluruhan pengujian pengenalan akor berdasarkan nada *root* adalah sebesar 87 persen.

5.2 Saran

Saran yang dapat diberikan untuk memperluas penelitian ini adalah sebagai berikut:

1. Meneliti pengenalan akor dari suara alat musik selain *piano* dan *bass*.
2. Melakukan penelitian menggunakan metode *KNN* dan membandingkan hasilnya dengan metode *CNN*.