

## DETEKSI SUARA CHORD PIANO MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK

Fajar Ferdiawan<sup>1</sup>, Budi Hartono<sup>2</sup>

<sup>12</sup>Program Studi Teknik Informatika, Universitas Stikubank Semarang

Jl. Tri Lomba Juang No 1 Semarang 50241

[1ferdiawanfajar06@gmail.com](mailto:ferdiawanfajar06@gmail.com), [2budihartono@edu.unisbank.ac.id](mailto:budihartono@edu.unisbank.ac.id)

### Abstract

Piano is the most popular musical instrument by the public besides the guitar, piano can be a good instrument to accompany singers even without the accompaniment of other musical instruments. Piano consists of notes that reach 6.5 octaves to more than 7 octaves, from the existing notes can be formed into chords according to the scale. There are several scales, namely pentatonic, chromatic, and diatonic, of the three scales the diatonic scale is the one that is often used. Diatonic scales also have 2 types, namely diatonic major and diatonic minor. Diatonic major scales are generally used by beginners to learn the piano. This research will classify piano major scale chords using the Convolutional Neural Network method. Convolutional Neural Network is used to detect and recognize objects in an image. This research also uses Keras library which is an artificial neural network that runs on TensorFlow to speed up the image processing process. The test results using 240 piano chord datasets produce the highest accuracy reaching 98%.

**Keywords :** Deep Learning, Piano, Audio Processing, Convolutional Neural Network, Python

### Abstrak

Piano merupakan alat musik yang paling digemari oleh masyarakat selain gitar, piano dapat menjadi instrument yang baik untuk mengiringi penyanyi walaupun tanpa iringan alat musik yang lain. Piano terdiri dari not yang mencapai 6.5 oktaf sampai lebih dari 7 oktaf, dari not yang ada dapat dibentuk menjadi *chord* yang sesuai dengan tangga nadanya. Ada beberapa tangga nada yaitu *pentatonic*, *chromatic*, serta *diatonic*, dari ketiga tangga nada tersebut tangga nada *diatonic*-lah yang sering dipakai. Tangga nada *diatonic* juga memiliki 2 jenis yaitu *diatonic major* dan *diatonic minor*. Tangga nada *diatonic major* umumnya digunakan pemula untuk belajar piano. Penelitian ini akan mengklasifikasikan *chord* piano *major scale* dengan menggunakan metode *Convolutional Neural Network*. *Convolutional Neural Network* digunakan untuk mendeteksi serta mengenali *object* pada sebuah gambar. Penelitian ini juga menggunakan *library Keras* yang merupakan jaringan syaraf tiruan yang berjalan diatas *TensorFlow* untuk mempercepat proses pengolahan citra. Hasil uji dengan menggunakan 240 *dataset chord* piano menghasilkan akurasi tertinggi mencapai 98%.

**Kata Kunci :** Deep Learning, Piano, Audio Processing, Convolutional Neural Network, Python

### 1. PENDAHULUAN

Nada yaitu bunyi yang beraturan serta memiliki frekuensi tertentu. Nada memiliki susunan yang dimulai dari nada dasar sampai nada oktaf dinamakan tangga nada. Selain itu tangga nada memiliki tiga jenis yaitu *Pentatonic*, *Chromatic*, dan *Diatonic*. Tangga nada *Diatonic* dibedakan menjadi dua jenis yaitu tangga nada mayor (*major scale*) dan tangga nada minor (*minor scale*). Skala mayor dimulai dari nada dasar Do dengan interval  $1-1-\frac{1}{2}-1-1-1-\frac{1}{2}$  sampai Do satu oktaf atau Do yang lebih tinggi dari nada

Do awal [1]. Sedangkan skala minor dimulai dari nada dasar La dengan interval  $1-\frac{1}{2}-1-1-1-\frac{1}{2}-1$  sampai La satu oktaf atau La yang lebih tinggi dari nada La awal.

*Chord* merupakan kumpulan dari beberapa nada yang dibunyikan secara sendiri-sendiri atau bersama sehingga menciptakan suara yang harmonis. *Chord* memiliki beberapa jenis berdasarkan nada penyusunnya, contohnya *chord major*, *minor*, *augmented*, *diminished*, dan lain-lain. Biasanya *chord* direpresentasikan dengan huruf yaitu C, E, F, G, A, B dan juga C# atau Db, D# atau Eb, F# atau Gb, G# atau Ab, dan A#

atau Bb. Selain itu *chord* juga memiliki perpindahan atau progresi. Progresi *chord* berfungsi agar saat *chord* digunakan untuk membentuk sebuah lagu dinamika dari lagu tersebut tidak berantakan.

Piano merupakan salah satu instrument penghasil nada yang disukai oleh masyarakat selain gitar. Piano sendiri dapat dikatakan sebagai instrumen dasar dalam belajar bermusik. *Range* dari piano bisa mencapai 6.5 oktaf sampai lebih dari 7 oktaf, dengan *range* seluas ini piano dapat memainkan beragam unsur seperti bass, harmoni, melodi, dan lain-lain. Dari penjabaran tersebut banyak orang yang tertarik untuk belajar mengenai piano. Akan tetapi, kurangnya pengetahuan dasar mengenai teori musik menjadikan sebagian orang kesulitan untuk belajar piano. Terlebih lagi mahalnnya biaya untuk mengikuti les musik menjadi salah satu alasan yang menyertainya.

Pada era ini kemajuan teknologi semakin berkembang pesat terlebih lagi dibidang *machine learning*. *Machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu [2]. *Machine learning* sendiri telah mengalami perkembangan menjadi *deep learning*. *Deep learning* terinspirasi dari jaringan syaraf manusia yang bekerja melalui *hidden layer*. Perkembangan *deep learning* sendiri saat ini dipermudah dengan banyaknya *Application Program Interface* (API) serta *library*. Salah satu *library* yang sedang hangat dibicarakan adalah *Tensorflow*.

*Tensorflow* merupakan perpustakaan perangkat lunak yang dikembangkan untuk pembelajaran mesin dan penelitian jaringan syaraf tiruan [3]. *Tensorflow* adalah framework yang berkaitan erat dengan *deep learning*. *Tensorflow* dapat digunakan dalam mendeteksi gambar ataupun suara. Salah satu jenis metode yang dapat digunakan dengan *Tensorflow* yaitu *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) merupakan algoritma *deep learning* yang sangat populer karena dapat menghilangkan kebutuhan untuk ekstraksi fitur secara manual, dapat dilatih untuk tugas pengenalan data baru, dan juga memiliki beberapa model. Oleh karena itu penelitian ini akan berfokus pada penerapan *framework Tensorflow* dan metode *Convolutional Neural Network* (CNN) untuk mendeteksi *chord* piano serta nada pembentuk *chord* dari suara piano agar dapat mempermudah dalam mengenali *chord* dan juga nada pembentuk *chord*.

## 2. TINJAUAN PUSTAKA DAN TEORI

### 2.1 Tinjauan Pustaka

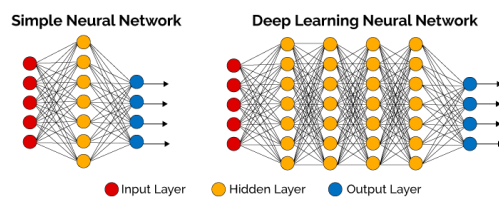
Penelitian mengenai “Pengenalan Suara Paru Dengan *Convolutional Neural Network* (CNN)” yang dilakukan oleh Muhammad Hasbi Ashshiddieqy (2020). Penelitian ini bertujuan untuk mendeteksi kelainan pada suara paru dengan mengubah data suara pernafasan menjadi data spektogram dengan menggunakan algoritma *Short-Time Fourier Transform* (STFT). Data suara yang telah dikonversi akan diklasifikasi dengan algoritma *Convolutional Neural Network* (CNN). Hasil pengujian arsitektur CNN dengan urutan *convolutional layer* kemudian *pooling layer* sejumlah empat lapisan dan *fully connected layer* sejumlah tiga lapis menghasilkan kinerja terbaik dengan *learning rate* bernilai 0.0005 menghasilkan akurasi sebesar 84.8% pada data *train*. Pada penelitian ini pembagian data menjadi salah satu faktor yang membuat kinerja model menjadi tidak maksimal karena data pada kelas normal sejumlah 52.18%, kelas *crackle* sejumlah 27.1%, kelas *wheeze* sejumlah 12.83%, dan kelas *crackle* dan *wheeze* sejumlah 7.26% [4].

Penelitian mengenai “Deteksi Penyakit Kulit Wajah Menggunakan *Tensorflow* Dengan Metode *Convolutional Neural Network*” yang dilakukan oleh Indah Widhi Prastika (2021). Penelitian ini bertujuan untuk mendeteksi penyakit pada kulit wajah secara *realtime* berbasis Android dengan menggunakan data *training* sebanyak 700 gambar dari kulit wajah manusia. Data gambar kemudian diklasifikasikan menggunakan *Convolutional Neural Network* (CNN) dengan *epoch* sebesar 50. Setelah data diklasifikasikan kemudian data tersebut diolah dengan *Tensorflow lite* agar dapat dijadikan aplikasi Android. Dari sistem yang telah dibuat diperoleh ketepatan tertinggi yaitu 99,91% dengan persentase rata-rata ketepatan sebesar 80% [5].

Penelitian mengenai “*Deep Learning* Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma *Convolutional Neural Network* (CNN) Dengan *Tensorflow*” yang dilakukan oleh Wulan Anggraini (2020). Penelitian ini bertujuan untuk mendeteksi wajah yang berhijab dengan *dataset* berjumlah 300 gambar wajah dengan dua ekspresi yaitu datar dan tersenyum. Data yang telah dipersiapkan diuji coba dengan nilai *epoch* dan *learning rate* yang berbeda-beda. Dari hasil penelitian dihasilkan nilai *accuracy* untuk data *training* sebesar 92% sedangkan untuk data *testing* sebesar 87% [6].

## 2.2 Deep Learning

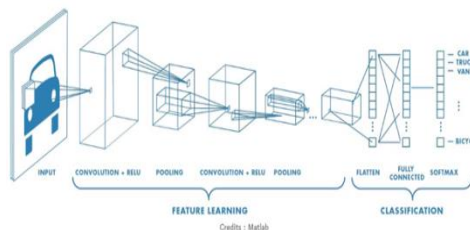
*Deep learning* merupakan bagian dari *machine learning* serta menggunakan *artificial neural network* yang berlapis-lapis. *Deep learning* sendiri adalah metode dari pembelajaran mesin yang meniru cara kerja dasar dari otak manusia dengan jumlah jaringan syaraf dari algoritmanya yang sangat banyak. *Deep learning* sering digunakan dalam pengenalan citra, klasifikasi teks, pengenalan suara, dan lain-lain. Tidak seperti jaringan syaraf tiruan *deep learning* memiliki dua sampai ratusan *hidden layer* sehingga *deep learning* dapat mempelajari data secara mandiri dan optimal [7].



Gambar 1. Perbedaan Layer Pada Jaringan Syaraf Tiruan Dan Deep Learning.

## 2.3 Convolutional Neural Network

*Convolutional Neural Network* merupakan metode *machine learning* yang dikembangkan dari *Multi Layer Perceptron* yang digunakan untuk mengolah data *image*. CNN terdiri dari beberapa lapisan yaitu *Convolution*, *Pooling*, dan *Fully Connected* [8]. CNN memiliki *Application Program Interface* yang dapat dilihat dari gambar dibawah ini.



Gambar 2. Lapisan Pada CNN.

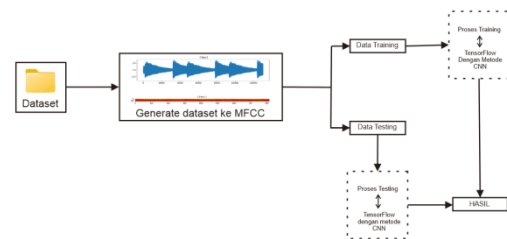
## 2.4 Python

*Python* merupakan bahasa pemrograman yang dikembangkan khusus agar sebuah kode dapat mudah dibaca. *Python* juga memiliki *library* yang lengkap sehingga dapat memudahkan *programer* dalam membuat sebuah aplikasi dengan kode yang sederhana.

## 3. METODE PENELITIAN

Penelitian ini sepenuhnya menggunakan menggunakan *platform kaggle.com* untuk

membuat pemodelan dari metode *Convolutional Neural Network* untuk mengklasifikasikan *chord* piano. *Kaggle* merupakan situs yang digunakan untuk membuat model guna menganalisa serta memprediksi suatu *dataset*. *Kaggle* sendiri mempunyai beberapa bahasa pemrograman yang dapat digunakan yaitu *Python*, *R*, dan lain-lain. Program pada *Kaggle* ditulis dengan bahasa pemrograman *Python*. *Python* merupakan bahasa pemrograman interpretative yang berfokus pada tingkat keterbacaan kode, *Python* juga adalah salah satu bahasa pemrograman yang paling populer untuk *Data Science*, *Machine Learning*, dan *IoT*.



Gambar 3. Tahapan Penelitian.

Pada gambar 3 dapat dilihat alur penelitian dari pengumpulan *dataset* hingga hasil yang menyatakan nilai akurasi dari proses klasifikasi yang telah dilakukan.

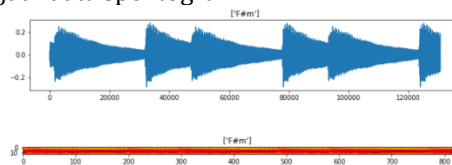
## 3.1 Kebutuhan Data

Data berupa *file* suara dari tiap-tiap *chord* piano yang berjumlah 24 *chord*. Data *chord* piano dibuat dengan aplikasi *FL Studio 20* dengan total sebanyak 480 data dengan sampel dari setiap *chord* sebanyak 20 sampel dengan durasi 15-20 detik. *File* sampel berekstensi *WAV* dengan *sample rate* sebesar 44Khz dan hanya memiliki satu sinyal *audio (mono)*. Tiap sampel yang ada dimainkan dengan panjang pendek penekanan nada dari penyusun *chord* dengan berbeda-beda. Data akan dibagi menjadi data *training* dan data *testing* dengan persentase 80% untuk data *training* dan 20% untuk data *testing*.

## 4. HASIL DAN PEMBAHASAN

Klasifikasi dilakukan dengan menggunakan metode *Convolutional Neural Network* dengan mengubah sinyal *audio* menjadi data spektrogram dengan *Mel-Frequency Cepstral Coefficient* (MFCC). *Mel-frequency cepstral coefficients* (MFCC) adalah representasi transformasi linear kosinus dari spektrum daya log waktu singkat dari sinyal ucapan pada skala frekuensi *Mel non-linear* [9]. Data spektrogram merupakan data yang akan digunakan pada proses klasifikasi.

Berikut contoh data *audio* yang telah diubah menjadi data spektrogram :



Gambar 4. Data Spektrogram.

Sebelum citra data spektrogram akan diproses dengan pemodelan *Convolutional Neural Network* ada beberapa *library* yang akan digunakan antara lain :

1. *NumPy* (*Numerical Python*) merupakan *library* yang digunakan untuk *scientific computing* serta memiliki fungsi untuk membentuk objek *N-dimensional array*.
2. *Pandas* (*Python for Data Analysis*) merupakan *library* yang digunakan untuk persiapan data, manipulasi data, serta pembersihan data. *Pandas* sendiri mempunyai dua objek yaitu *DataFrame* dan *Series*.
3. *Python\_speech\_features* adalah *library* yang digunakan untuk ekstraksi fitur dan pelatihan jaringan syaraf tiruan. *Library* ini juga digunakan untuk mengkonversi data kedalam bentuk *Mel-frequency cepstral coefficients* (MFCC).
4. *Librosa* adalah *library* yang digunakan untuk menganalisis sinyal *audio*. *Librosa* sendiri dapat berfungsi juga untuk fitur ekstraksi *Mel-frequency cepstral coefficients* (MFCC).
5. *Matplotlib* adalah *library* yang digunakan untuk visualisasi data contohnya *plot* grafik.
6. *Seaborn* merupakan *library* yang digunakan untuk visualisasi data yang dibangun diatas *Matplotlib*.
7. *Keras* merupakan *library* yang digunakan untuk menyederhanakan model dari kerangka *deep learning*. *Keras* dijalankan diatas *framework TensorFlow*.

Citra data spektrogram akan diproses oleh pemodelan *Convolutional Neural Network*. Model dari metode *Convolutional Neural Network* juga mempengaruhi hasil dari akurasi data. Ada beberapa tahapan dari pemodelan metode *Convolutional Neural Network* yaitu sebagai berikut :

1. Tahapan *convolution* pertama menggunakan *kernel 3x3* dengan 16 *filter*. Tahapan ini akan menghasilkan kombinasi dari dua matiks yang berbeda sehingga menghasilkan nilai satu matriks yang baru.

Kemudian ditambahkan aktivasi *Retrified Linear Unit* (RELU) untuk mengubah nilai negatif menjadi nol. Kemudian tambahkan *stride* untuk mengatur jumlah *pixel* yang bergeser diatas matriks. Berikan *padding* agar hasil ukuran dari tahap konvolusi pertama tetap sama nilainya. Langkah terakhir masukkan nilai *input\_shape*.

2. Tahapan *convolution* kedua merupakan lanjutan dari tahapan *convolution* pertama. Tahapan ini sedikit berbeda dengan tahapan pertama karena menggunakan 32 *filter*. Tahapan *convolutional* kedua juga menggunakan aktivasi *Retrified Linear Unit* (RELU) serta *stride* tapi tidak menambahkan *input\_shape*.
3. Tahapan *pooling* digunakan untuk mengurangi ukuran matriks. Dalam penelitian ini aktivasi *pooling* yang digunakan adalah *maxpooling*. *Maxpooling* mengambil nilai maksimum dari pergesaran *kernel*. *Kernel maxpooling* yang digunakan pada penelitian ini adalah 2x2.
4. Tahapan *dropout* digunakan untuk mencegah *overfitting* serta mempercepat proses *learning*. *Overfitting* adalah kondisi dimana data dalam proses pelatihan menghasilkan nilai akurasi terbaik, sedangkan apabila dilakukan tes dengan data yang lain hasil akurasi menjadi kacau. *Dropout* bekerja secara acak untuk mengatur unit tersembunyi pada setiap pembaruan fase pelatihan. Dalam penelitian kali ini *dropout* digunakan beberapa kali agar menghasilkan akurasi terbaik dengan nilai *dropout* sebesar 0.1 atau 10%.
5. Tahapan *flatten* atau *fully connected* digunakan untuk mengubah hasil dari *maxpooling* menjadi *vector*. *Vector* tersebut akan digunakan sebagai *input fully connection layer* dengan satu *hidden layer*. Kemudian tambahkan fungsi *dense* untuk menambahkan *layer* yang akan digunakan sebagai *layer classification*.
6. Tahapan *softmax* digunakan untuk mendeteksi label pada data.
7. Tahapan *compile model* yang bersi *categorical crossentropy* yang digunakan untuk klasifikasi multikelas dan juga pengoptimalan menggunakan *Adam*.

Pada tahapan *convolutional layer* pertama filter yang digunakan adalah 16 dengan kernel 3x3 untuk mendapatkan parameter 160 dilakukan perhitungan  $(3 \times 3 \times 1 + 1) 16 = 160$ . Hasil dari *convolutional layer* pertama akan diteruskan ke *convolutional layer* kedua dengan filter sebesar 32 dengan kernel 3x3 dengan



perhitungan nilai parameter  $((3 \times 3 \times 16) + 1) 32 = 4640$ . Selanjutnya hasil dari *convolutional layer* kedua akan diteruskan ke *layer maxpooling* untuk mengurangi ukuran matriks. Hasil dari *maxpooling* akan dilanjutkan ke *layer dropout* agar proses klasifikasi tidak mengalami *overfitting*. *Overfitting* adalah keadaan dimana data untuk pelatihan merupakan data terbaik sehingga saat dilakukan *test* dengan data yang berbeda dapat mengurangi akurasi. Selanjutnya parameter akan diubah bentuk dari matriks menjadi *vector* sehingga menghasilkan nilai 78144. Parameter dari proses sebelumnya akan diteruskan ke *layer dense* dengan 128 untuk mendapatkan parameter sebesar 10002560 dengan perhitungan  $(78144 \times 128) + 128 = 10002560$ . Selanjutnya akan diteruskan ke *layer dropout* kedua agar proses klasifikasi tidak mengalami *overfitting*. Dari *layer dropout* akan diteruskan ke *layer* terakhir yaitu *dense* dengan filter sebesar 24 sesuai dengan jumlah *chord* dari data untuk menghasilkan parameter 1560 dengan perhitungan  $(64 \times 24) + 24 = 1560$ . Berikut merupakan tampilan pemodelan *Convolutional Neural Network* :

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)              (None, 815, 13, 16)      168
conv2d_1 (Conv2D)            (None, 815, 13, 32)      4648
max_pooling2d (MaxPooling2D) (None, 487, 6, 32)       0
dropout (Dropout)            (None, 487, 6, 32)       0
flatten (Flatten)            (None, 78144)            0
dense (Dense)                (None, 128)              10002560
dropout_1 (Dropout)          (None, 128)              0
dense_1 (Dense)              (None, 64)               8256
dropout_2 (Dropout)          (None, 64)               0
dense_2 (Dense)              (None, 24)               1560
-----
Total params: 10,017,176
Trainable params: 10,017,176
Non-trainable params: 0

```

Gambar 5. Tampilan Pemodelan CNN.

Hasil dari pelatihan setiap *layer* dapat dilihat pada gambar 5 jumlah dari total parameter yang dilatih mencapai 10017176. Untuk memperoleh nilai akurasi yang tinggi tidak lepas dari pengaruh parameter-parameter yang ada. Parameter tersebut akan diuji guna membandingkan pemodelan terbaik dari *Convolutional Neural Network* dengan memperhatikan nilai dari parameter tersebut.

### 1. Pengaruh jumlah data

Guna menguji pengaruh dari jumlah data, maka akan diuji 3 bagian yaitu 120, 240, dan 480 data. Masing masing data akan dibagi menjadi 80% untuk data *training* dan 20% untuk data *testing*. Berikut hasil pengujian dari pengaruh jumlah data :

TABEL 1. PENGARUH JUMLAH DATA

Data	Accur acy	Loss	Average time(s)/ste p
120	96%	0.0671	47
240	98%	0.0332	57
480	93%	0.1834	47

### 2. Pengaruh jumlah random sampling

*Random sampling* bekerja dengan mengambil *n* sampel dimana *n* merupakan nilai yang bebas. *n* sendiri akan mengambil posisi dimanapun dalam *file audio* secara acak dalam setiap iterasi. Guna menguji pengaruh dari random sampling, maka akan diuji 3 bagian yaitu 1000, 3000, dan 6000. Berikut hasil pengujian dari pengaruh random sampling :

TABEL 2. PENGARUH JUMLAH  
RANDOM SAMPLING

Rand om samp ling	Accur acy	Loss	Average time(s)/s tep
1000	93%	0.1834	47
3000	92%	0.0054	140
6000	98%	0.0332	57

### 3. Pengaruh epoch

*Epoch* adalah proses dimana dataset akan dilatih dengan *neural network*. Guna menguji pengaruh dari *epoch*, maka akan diuji 3 nilai *epoch* yaitu 30, 50, dan 100. Berikut hasil pengujian dari pengaruh nilai *epoch* :

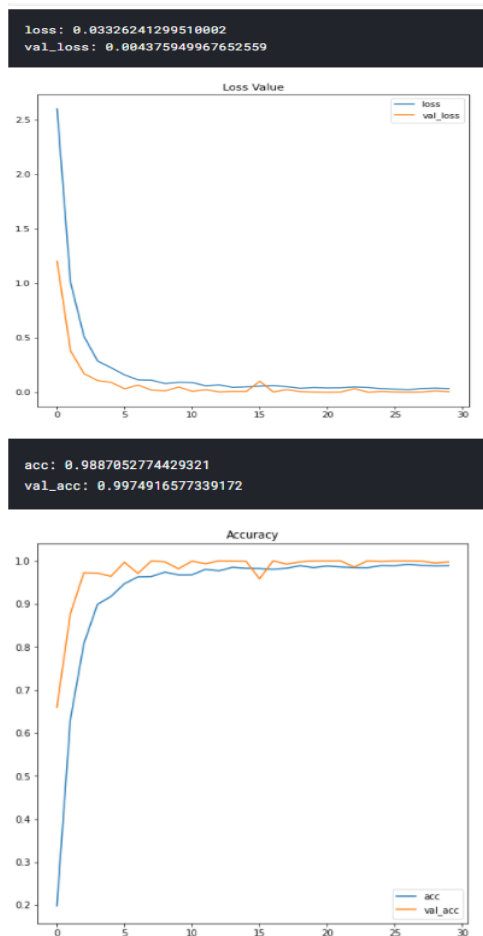
TABEL 3. PENGARUH NILAI EPOCH

Epoch	Accur acy	Loss	Average time(s)/s tep
30	98%	0.0332	57

50	96%	0.0671	47
100	92%	0.0054	140

Berdasarkan pengujian nilai parameter-parameter yang telah dilakukan diperoleh nilai akurasi tertinggi yaitu 98%. Nilai akurasi tersebut dihasilkan dari 240 jumlah data, 6000 *random sampling*, dan 30 *epoch*. Masing-masing parameter mempengaruhi satu sama lain, dengan jumlah data dan nilai *random sampling* yang semakin banyak maka akan semakin akurat tetapi akan memakan waktu yang lebih lama serta akan memakan daya CPU yang besar pula. Hal tersebut juga dipengaruhi oleh nilai *epoch*, semakin tinggi nilai *epoch* maka nilai akurasi akan semakin tinggi tetapi hal tersebut harus diimbangi dengan jumlah data yang digunakan untuk menghindari *overfitting*. Jumlah data sebanyak 240 sebanding dengan nilai *epoch* sehingga *overfitting* berhasil dihindari dan menghasilkan nilai akurasi terbaik.

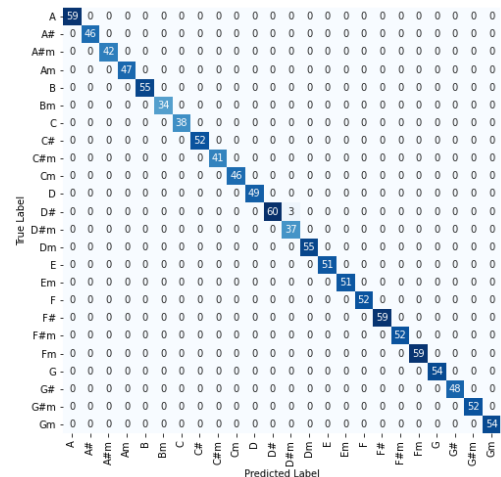
Berikut merupakan grafik dari nilai *loss* dan *accuracy* yang dihasilkan dari pengujian terbaik :



Gambar 6. Grafik Loss Dan Accuracy.

Dari grafik *loss* dan *accuracy* diatas dapat dilihat bahwa dengan *Dataset* sebanyak 240 data, nilai *random sampling* sebesar 6000, serta jumlah *epoch* sebesar 30 menghasilkan nilai *loss* sebesar 0.033 atau 3% dan nilai *val\_loss* sebesar 0.004 atau 0.4%. sementara itu nilai dari *acc* sebesar 0.98 atau 98% dan *val\_acc* sebesar 0.99 atau 99%.

Evaluasi dari kinerja pemodelan *Convolutional Neural Network* dapat dilihat melalui gambar *confusion matrix* dibawah ini :



Gambar 7. Confusion Matrix.

*Confusion matrix* diatas menunjukkan bahwa semua *chord* piano terklasifikasi dengan baik serta memiliki tingkat ketepatan yang tinggi. Penelitian sebelumnya yang dilakukan oleh Ismail Hasanain W (2021) yaitu tentang "Klasifikasi Suara Paru-Paru Menggunakan *Convolutional Neural Network* (CNN)" menghasilkan nilai akurasi sebesar 74% dari keseluruhan penelitian [10]. Sedangkan pada penelitian ini berhasil menghasilkan nilai akurasi sebesar 98%.

## 5. Kesimpulan Dan Saran

Berdasarkan penelitian yang telah dilakukan yaitu "Deteksi Suara *Chord* Piano Menggunakan Metode *Convolutional Neural Network*" dimana penelitian dengan pemodelan CNN dengan masukan *audio file* dengan durasi 15-20 detik dari tiap sampel dengan jumlah data sebanyak 240 data *audio chord* piano. Nilai *random sampling* yang digunakan adalah 6000 serta *epoch* yang digunakan sebesar 30 menghasilkan nilai *accuracy* tertinggi dari pemodelan CNN yang telah dibuat yaitu 98%. Nilai *accuracy* yang tinggi diimbangi dengan nilai *random sampling* serta nilai *epoch* yang paling sesuai yaitu 6000 untuk nilai *random sampling* serta 30 untuk nilai

*epoch*. Hal ini dikarenakan semakin tinggi nilai *random sampling* maka akan semakin banyak potongan *audio* yang akan diuji. Akan tetapi nilai *random sampling* yang semakin tinggi membutuhkan kecepatan CPU yang tinggi. Sementara *epoch* dengan nilai 30 digunakan agar tidak memakan waktu yang lama ketika proses *training*. Apabila nilai *epoch* semakin tinggi maka nilai akurasi yang dihasilkan akan menurun karena nilai *epoch* tidak sebanding dengan jumlah *Dataset*.

Saran pada penelitian selanjutnya dapat dihasilkan sistem yang mampu mendeteksi suara *chord* piano secara *realtime* berbasis android ataupun web serta menambahkan *dataset* dari tangga nada lain seperti tangga nada *Pentatonic* dan *Chromatic* serta menambahkan *chord* lain seperti *augmented* dan *diminished*.

## 6. UCAPAN TERIMA KASIH

Terimakasih penulis ucapkan kepada :

1. Allah SWT atas anugerah dan karunia yang telah diberikan kepada saya
2. Kedua orang tua serta kakak saya yang selalu memberi dukungan moral dan materil.
3. Bapak Budi Hartono, S.Kom., M.Kom. selaku pembimbing dan penulis kedua dalam penelitian ini.
4. Teman – teman mahasiswa FTI Unisbank yang saya cintai.

## Daftar Pustaka:

- [1] T. Safaat, "Implementasi *Fast Fourier Transform* Pada Pengenalan Nada Piano Berbasis Android," Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2016.
- [2] P. R. Aningtiyas, "Pembuatan Aplikasi Deteksi Objek Menggunakan Tensorflow Object Detection API Dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra-Terlatih," *J. Ilm. KOMPUTASI*, vol. 19, no. September, pp. 421–430, 2020.
- [3] S. R. Dewi, "Deep Learning Object Detection Pada Video," Universitas Islam Indonesia Yogyakarta, 2018.
- [4] M. H. Ashshiddieqy, "Klasifikasi Suara Paru Dengan Convolutional Neural Network (CNN)," *eProceedings Eng.*, vol. 07, no. 02, pp. 8506–8512, 2020, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12750/12473>.
- [5] I. W. Prastika, "Deteksi penyakit kulit wajah menggunakan tensorflow dengan metode convolutional neural network," *MISI (Jurnal Manaj. Inform. Sist. Informasi)*, vol. 4, no. 2, pp. 84–91, 2021.
- [6] W. Anggraini, "Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow," Universitas Islam Negeri Ar-Raniry Darussalam Banda Aceh, 2020.
- [7] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/20000000039.
- [8] N. Giarsyani, A. F. Hidayatullah, and R. Rahmadi, "KOMPARASI ALGORITMA MACHINE LEARNING DAN DEEP LEARNING UNTUK NAMED ENTITY RECOGNITION: STUDI KASUS DATA KEBENCANAAN," *Jire (Jurnal Inform. Rekayasa Elektronik)*, vol. 3, no. 1, pp. 48–57, 2020.
- [9] D. Lionel, R. Adipranata, and E. Setyati, "Klasifikasi Genre Musik Menggunakan Metode Deep Learning Convolutional Neural Network dan Mel-Spektrogram," *J. Infra Petra*, vol. 7, no. 1, pp. 51–55, 2019.
- [10] I. W. Hasanain and A. Rizal, "Klasifikasi Suara Paru-Paru Menggunakan Convolutional Neural Network (CNN)," *e-Proceeding Eng.*, vol. 8, no. 2, pp. 3218–3223, 2021.