

PENERAPAN METODE *CONVOLUTIONAL NEURAL NETWORK* PADA PENDETEKSI GAMBAR NOTASI BALOK

SKRIPSI



Oleh :

Putri Navia Rena

11150910000033

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH

JAKARTA

2019 M / 1441 H

PENERAPAN METODE *CONVOLUTIONAL NEURAL NETWORK* PADA PENDETEKSI GAMBAR NOTASI BALOK

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar

Sarjana Komputer (S.Kom)



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH

JAKARTA

2019 M / 1441 H

LEMBAR PERSETUJUAN

**PENERAPAN METODE CONVOLUTIONAL NEURAL
NETWORK PADA SISTEM PENDETEKSI NOTASI BALOK**

Skripsi


**Sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh:

Putri Navia Rena
11150910000033

Menyetujui,

Pembimbing I



Fenty Eka Muzayyana Agustin, M.Kom.

NIP. 197608052009122003

Pembimbing II

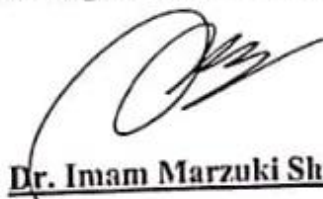


Nashrul Hakiem, S.Si., M.T., Ph.D.

NIP. 197106082005011005

Mengetahui,

Ketua Program Studi Teknik Informatika




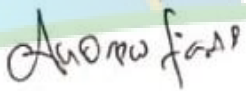
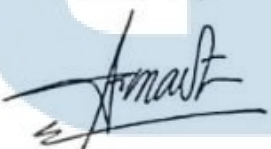
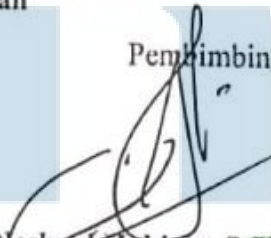
Dr. Imam Marzuki Shofi, MT.

NIP. 19720205 200801 1 010

PENGESAHAN UJIAN

Skripsi berjudul “Penerapan Metode *Convolutional Neural Network* pada Sistem Pendeteksi Notasi Balok” yang ditulis oleh Putri Navia Rena, NIM 11150910000033 telah diuji dan dinyatakan lulus dalam sidang munaqosyah Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta pada hari Senin, 21 Oktober 2019. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika.

Jakarta, Oktober 2019


Tim Penguji	
Penguji I	Penguji II
	
<u>Nenny Anggraini, MT</u>	<u>Andrew Fiade, M.Kom.</u>
NIDN. 0310097604	NIP, 198208112009121004
Tim Pembimbingan	
Pembimbing I	Pembimbing II
	
<u>Fenty Eka Muzayyana Agustin, M.Kom.</u>	<u>Nashrud Haklem, S.Si., M.T., Ph.D.</u>
NIP. 197608052009122003	NIP. 197106082005011005

Mengetahui

Dekan Fakultas Sains dan Teknologi

Ketua Program Studi Teknik Informatika


Prof. Dr. Lilv Surraya Eka Putri, M.Env.Stud
NIP. 196904042005012005

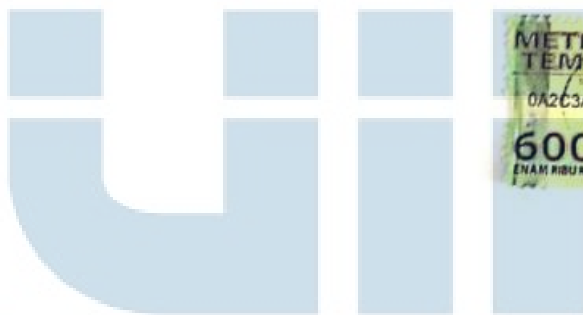

Dr. Imam Marzuki Shofi, MT
NIP. 197202052008011010

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa :

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila di kemudian hari terbukti karya ini bukan karya asli saya atau merupakan hasil jiplakan karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, Oktober 2019




Putri Navia Rena

11150910000033

PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI

Sebagai civitas akademik UIN Syarif Hidayatullah Jakarta, saya yang bertanda tangan dibawah ini:

Nama : Putri Navia Rena
NIM : 11150910000033
Program Studi : Teknik Informatika
Fakultas : Sains Dan Teknologi
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Islam Negeri Syarif Hidayatullah Jakarta Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right) atas karya ilmiah saya yang berjudul:

PENERAPAN METODE CONVOLUTIONAL NEURAL NETWORK PADA PENDETEKSI CITRA NOTASI BALOK

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Islam Negeri Syarif Hidayatullah Jakarta berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada tanggal : Oktober 2019

Yang menyatakan

(Putri Navia Rena)

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Puji syukur senantiasa dipanjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta nikmat-Nya sehingga penyusunan skripsi ini dapat diselesaikan. Sholawat serta salam senantiasa dihaturkan kepada junjungan kita baginda Nabi Muhammad SAW beserta keluarganya, para sahabatnya serta umatnya hingga akhir zaman. Penulisan skripsi ini mengambil tema dengan judul:

IMPLEMENTASI METODE *CONVOLUTIONAL NEURAL NETWORK* PADA PENDETEKSI NOTASI BALOK

Penyusunan skripsi ini adalah salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada program studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta. Adapun bahan penulisan skripsi ini adalah berdasarkan hasil penelitian dan beberapa sumber literatur.

Dalam penyusunan skripsi ini, telah banyak bimbingan dan bantuan yang didapatkan dari berbagai pihak sehingga skripsi ini dapat berjalan dengan lancar. Oleh karena itu, penulis ingin mengucapkan banyak terima kasih kepada :

1. Prof. Dr, Lily Suraya Eka Putri, M.Env.Stud selaku dekan Fakultas Sains dan Teknologi.
2. Dr. Imam Marzuki Shofi, MT selaku Ketua Program Studi Teknik Informatika.
3. Ibu Fenty Eka Muzayyana Agustin, M.Kom. dan Bapak Nashrul Hakiem, S.Si., M.T., Ph.D. selaku Dosen Pembimbing I dan II yang senantiasa meluangkan waktu dan memberikan bimbingan, bantuan, semangat dan motivasi dalam menyelesaikan skripsi ini.

4. Orang Tua tercinta, Bapak John Syafrudin dan Ibu Elvi Dayanti yang senantiasa memberikan dukungan moril dan materil. Tiada tutur kata selain terima kasih kepada kalian. Terima kasih, Alhamdulillah.
5. Seluruh dosen dan staff UIN Jakarta, khususnya Fakultas Sains dan Teknologi yang telah memberikan ilmu dan pengalaman yang berharga.
6. Para Staff Bisa AI terutama Bapak Octav, Bagas, Alfi dan Kak Rusnandi yang telah membantu memberikan ilmu terkait dan pembangunan sistem dipenelitian ini.
7. Orang-orang terdekat khususnya Yoan memberikan dukungan dan semangat yang tiada henti serta sepupu dan sahabat saya yang telah membantu dalam pengerjaan skripsi ini khususnya Ira, Mutia, Dewi, Rista, Wida, Riri, Qori, Intan.
8. Teman-teman Teknik Informatika angkatan 2015, teman-teman seperjuangan di Himpunan Mahasiswa Teknik Informatika (HIMTI), dan semua anak kelas TI B, teman-teman dekat DEMA U, serta teman-teman KKN yang tidak bisa disebutkan satu persatu.
9. Seluruh pihak yang tidak dapat disebutkan satu persatu yang secara langsung maupun tidak langsung telah membantu dalam menyelesaikan skripsi ini.

Penulisan skripsi ini masih jauh dari kata sempurna. Untuk itu, sangat diperlukan kritik dan saran yang membangun bagi penulis. Akhir kata, semoga laporan skripsi ini dapat bermanfaat bagi penulis dan orang lain.

Wassalamualaikum, Wr. Wb.

Jakarta, Oktober 2019

Penulis

Putri Navia Rena

11150910000033

Penulis : Putri Navia Rena (1115091000033)

Program Studi : Teknik Informatika

Judul : **PENERAPAN METODE *CONVOLUTIONAL NEURAL NETWORK* PADA PENDETEKSI NOTASI BALOK**

ABSTRAK

Notasi balok sering ditemukan pada partitur lagu baik dan dipakai sebagai standar notasi musik secara internasional. Untuk dapat membaca dan memahami notasi balok seseorang harus mempelajarinya dalam jangka waktu tertentu, sedangkan notasi huruf lebih mudah dan cepat dipahami dan dimengerti pembacaannya. Oleh karena itu dibuatlah sebuah sistem yang dapat membantu orang awam yang ingin mempelajari pembacaan musik dalam bentuk notasi balok.

Pembuatan sistem ini dibangun dengan menggunakan model *Convolutional Neural Network* dengan pembelajaran berbasis *Deep Learning* dimana sistem akan melalui pelatihan dan kemudian dapat mengenali gambar not balok sesuai dengan batasan yang ada. Hasil akurasi menjadi tolak ukur seberapa akurat pembelajaran yang diterima untuk mengenali not input pada sistem sebanyak 66,28% dan 69% setelah diuji coba pada data baru.

Kata Kunci : Not Balok, CNN, *Deep Learning*, *Image Classification*.

Daftar Pustaka : 21 Jurnal, 6 Buku, 3 Skripsi, 1 Website

Jumlah Halaman : VI BAB 63 halaman + xiv halaman + 32 gambar + 5 tabel

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
PENGESAHAN UJIAN	iii
PERNYATAAN ORISINALITAS	iv
PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI	v
KATA PENGANTAR	vi
ABSTRAK.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiv
BAB I.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Metode Penelitian.....	5
1.7 Sistematika Penulisan.....	6
BAB II.....	8
LANDASAN TEORI.....	8
2.1. <i>Digital Image</i> atau Citra Digital.....	11
2.2. Klasifikasi Gambar.....	14
2.3. Jaringan Syaraf Tiruan / <i>Artificial Neural Network (ANN)</i>	14
2.4. CNN (<i>Convolutional Neural Network</i>)	15
2.4.1. <i>Convolution Layer</i>	16
2.4.2. Operasi <i>Pooling</i>	21
2.4.3. <i>Fully-Connected Layer</i>	23
2.4.4. <i>Dropout Regularization</i>	24
2.4.5. <i>Softmax Layer</i>	26
2.5. <i>Artificial Intelligence (AI)</i>	26

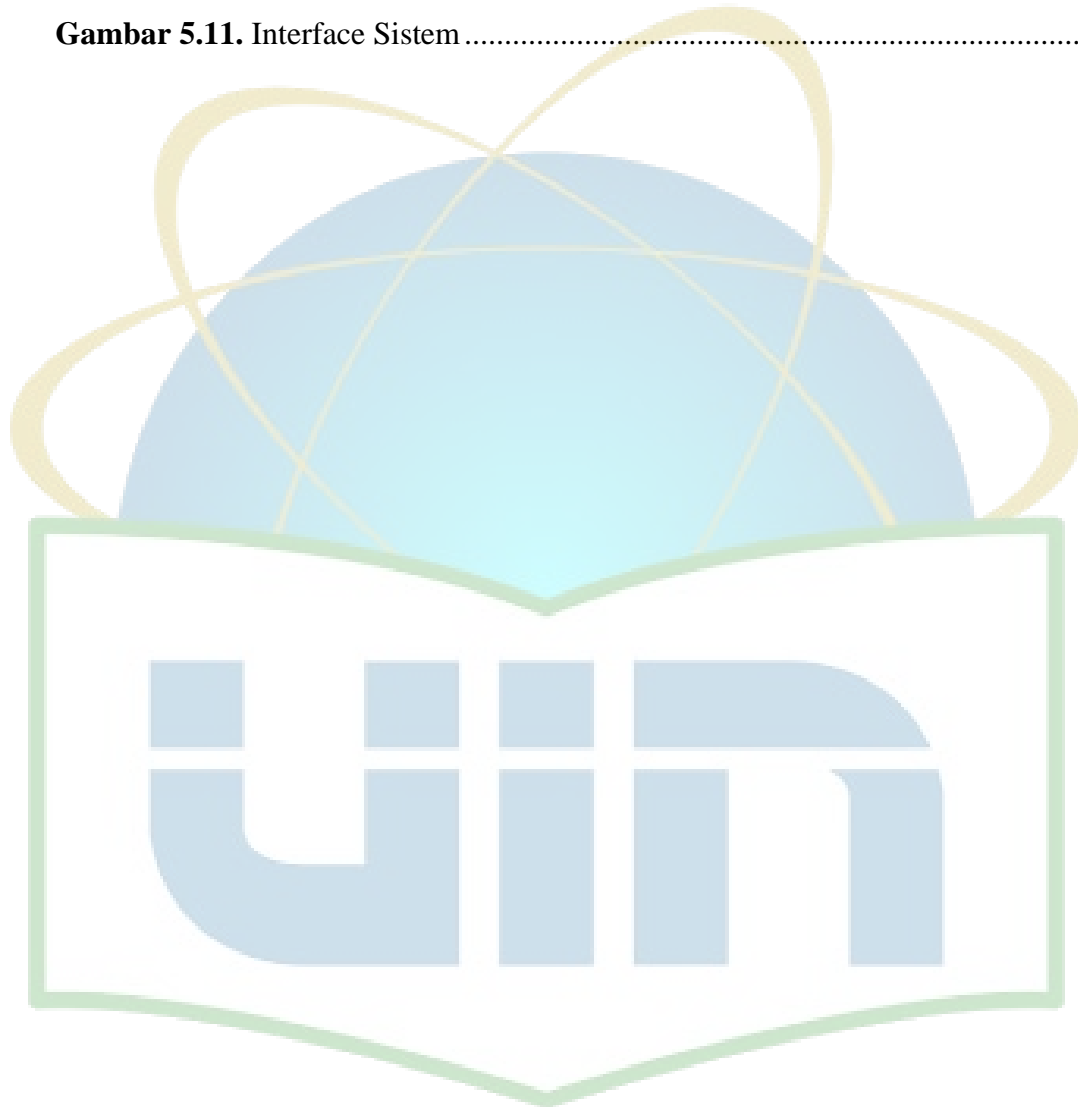
2.6.	<i>Machine Learning</i>	28
2.7.	DEEP LEARNING	29
2.8.	Precision dan Recall	31
2.9.	<i>Epoch</i>	31
2.10.	Python.....	31
2.11.	Keras	32
2.12.	Tensorflow	32
2.13.	Open CV	33
2.14.	<i>Google Colaboratory</i>	33
2.15.	Flask.....	34
2.16.	Partitur	34
2.17.	Notasi Balok	34
2.18.	Flask.....	39
BAB III	42
METODE PENELITIAN	42
3.1.	Metode Pengumpulan Data	42
3.1.1.	Studi Pustaka.....	42
3.1.2.	Observasi.....	42
3.2.	Metode Simulasi.....	43
3.2.1.	Formulasi Masalah (<i>Problem Formulation</i>)	43
3.2.2.	Model Pengkonsepan (<i>Conceptual Model</i>).....	43
3.2.3.	Pengumpulan Data Masukan/Keluaran (<i>Collection Input/Output Data</i>)	43
3.2.4.	<i>Preprocessing</i>	44
3.2.5.	Pemodelan (<i>Modelling</i>).....	44
3.2.6.	Pelatihan dan Pengujian	44
3.2.7.	Menghubungkan Model ke API.....	45
3.2.8.	Eksperimentasi (<i>Eksperimentation</i>)	45
3.2.9.	Analisis Output (<i>Output Analysis</i>).....	45
3.3.	Kerangka Berfikir Penelitian.....	46
BAB IV	47
IMPLEMENTASI	47

4.1.	Pengumpulan Data Notasi Balok	47
4.1.1.	Data dari Sebuah Lagu Berbentuk Notasi Balok	47
4.1.2.	Modifikasi data yang telah dikumpulkan.....	48
4.2.	Pembuatan Model menggunakan <i>Google Colab</i>	50
4.3.	<i>Preprocessing</i> dan <i>Labeling</i>	51
4.4.	Pemodelan	52
4.5.	Pelatihan dan Pengujian	53
4.6.	Membuat <i>base model</i> untuk dihubungkan ke <i>Interface</i>	54
1.1.1.	Menghubungkan Model ke API.....	54
1.1.2.	Eksperimentasi (<i>Eksperimentation</i>).....	54
BAB V	55
HASIL DAN PEMBAHASAN	55
5.1.	Pelabelan	55
5.2.	Membagi dataset untuk dilatih dan diuji dari dataset.....	56
5.3.	Arsitektur Jaringan	57
5.3.1.	Proses <i>Convolution Layer</i>	60
5.3.2.	Proses <i>Pooling</i>	61
5.3.3.	Proses <i>Fully Connected</i>	62
5.4.	Akurasi	62
5.5.	Precision Recall.....	65
5.6.	Pengujian.....	65
BAB VI	68
KESIMPULAN DAN SARAN	68
6.1.	Kesimpulan.....	68
6.2.	Saran.....	68
DAFTAR PUSTAKA	70

DAFTAR GAMBAR

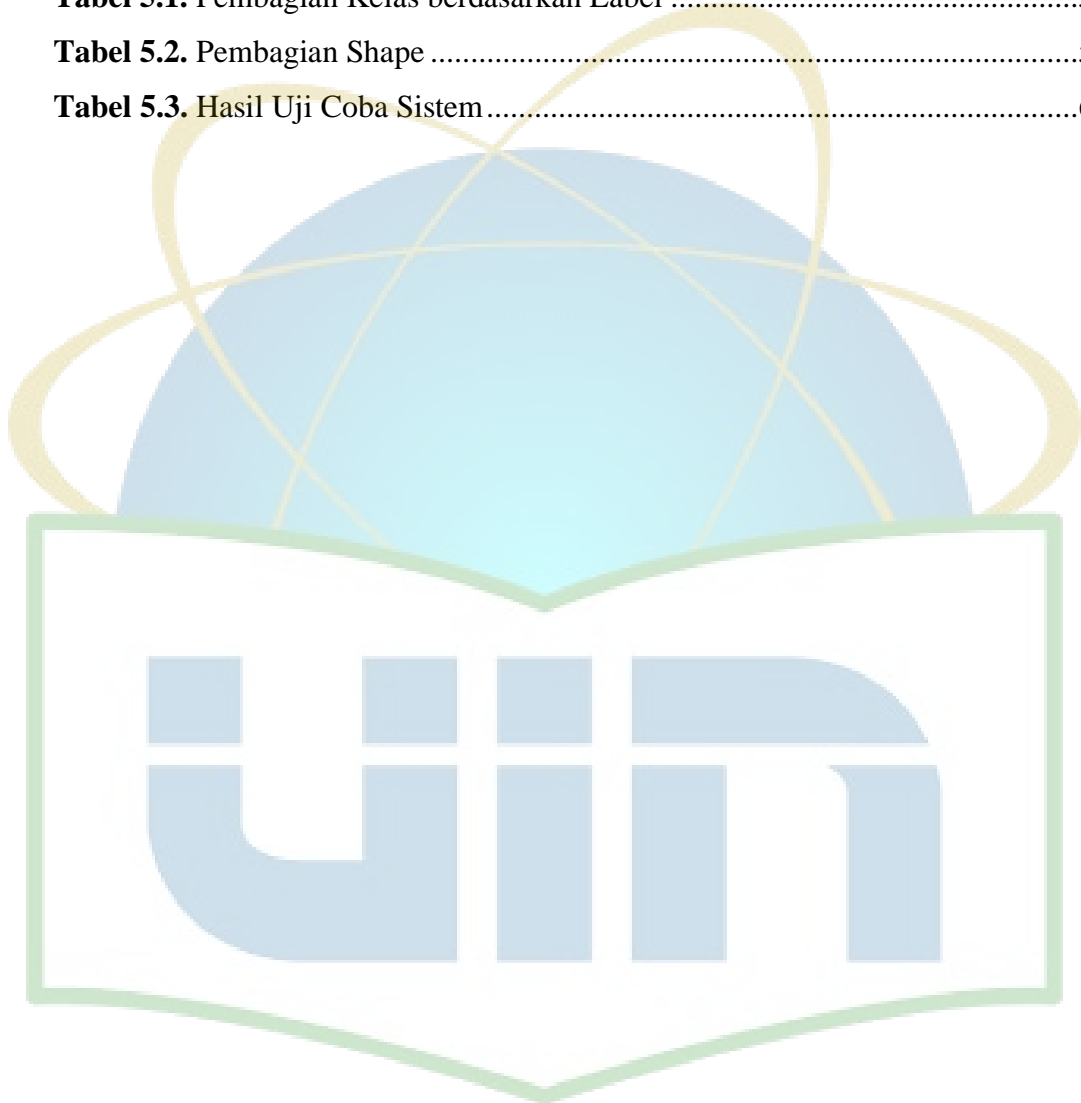
Gambar 2.1. Representasi Citra Digital dalam 2 Dimensi	13
Gambar 2.2. Arsitektur <i>Convolutional Neural Network</i>	16
Gambar 2.3. Proses pada convolutional layer	17
Gambar 2.4. Contoh conv layer sebelum bergeser	18
Gambar 2.5. Contoh conv layer setelah bergeser	19
Gambar 2.6. Hasil dari Convolutional Layer	20
Gambar 2.7. Operasi Max Pooling	22
Gambar 2.8. Dropout	25
Gambar 2.9. Diagram Kecerdasan Buatan	27
Gambar 2.10. Paranada (Musical Staff)	35
Gambar 2.11. Garis Bar/Bar Lines (kiri), garis bar ganda/Double Bar Line (kanan)	36
Gambar 2.12. Tanda kunci G/Treble (kiri), tanda kunci F/Bass(kanan)	37
Gambar 2.13. Tanda Birama (Time Signature)	37
Gambar 2.14. Posisi Not dalam Tanda Kunci Treble dan Bass	38
Gambar 2.15. Jenis-jenis Nilai Not	38
Gambar 2.16. Tanda Kres/Sharp (kiri), Tanda Mol/ Flat (tengah), Tanda Pugar/Natural(kanan)	39
Gambar 2.17. Tanda Tie	39
Gambar 3.1. Kerangka Berfikir Penelitian	46
Gambar 4.1. Contoh data gambar yang telah dicrop	48
Gambar 4.2. Contoh Gambar yang belum di modifikasi	48
Gambar 4.3. Contoh Gambar Setelah di Modifikasi	48
Gambar 5.1. Arsitektur Jaringan	57
Gambar 5.2. Model CNN	60
Gambar 5.3. Proses Konvolusi	60
Gambar 5.4. Fully Connected Layer	62
Gambar 5.5. Fully Connected Layer akhir	62

Gambar 5.6. Uji Akurasi	63
Gambar 5.7. Hasil Akurasi	63
Gambar 5.8. Grafik Model Accuracy	64
Gambar 5.9. Grafik Model Loss.....	64
Gambar 5.10. Precision Recall	65
Gambar 5.11. Interface Sistem	66



DAFTAR TABEL

Tabel 2.1. Perbandingan Penelitian Sejenis	10
Tabel 4.1. Daftar Gambar Berdasarkan Kelas.....	49
Tabel 5.1. Pembagian Kelas berdasarkan Label	56
Tabel 5.2. Pembagian Shape	57
Tabel 5.3. Hasil Uji Coba Sistem.....	66



BAB I

PENDAHULUAN

1.1 Latar Belakang

Musik sebagai bahasa yang universal merupakan media pengekspresian yang terdalam dan tidak terbatas dari setiap aksi sosial individu dalam lingkungannya dan lingkungan berkeaktivitas yang menyenangkan dan memuaskan (*self satisfaction/self fulfillment*), baik secara fisik, emosi, intelektual, moral, mental dan spiritual; yang dituangkan melalui bunyi, gerak dan struktur. Musik adalah bentuk ekspresi perasaan atau pikiran yang dikeluarkan secara teratur dalam bentuk bunyi. Bisa dikatakan, bunyi (suara) adalah elemen musik paling dasar. Suara musik yang baik adalah hasil interaksi dari tiga elemen, yaitu: irama, melodi, dan harmoni. Musik juga dikatakan sekumpulan suara yang digabungkan sehingga menjadi suatu irama dengan nada-nada (Hakim, 2017).

Ada beberapa cara untuk mengenal dan mengerti sebuah musik ataupun lagu. Satu diantaranya dengan cara membaca. Yang dibaca dapat berupa huruf, angka, ataupun gambar (Tombeng, Maramis, Alfred, & Puasa, 2018). Penulisan notasi musik menghasilkan tulisan musik atau disebut partitur. Partitur merupakan istilah umum yang medeskripsikan tentang kertas musik atau notasi musik. Dalam notasi musik, nada dilabangkan dengan not atau notasi (Hildayanti, Darmanto, & Informatika, 2019).

Satuan-satuan notasi disebut not. Jelasnya, notasi adalah sistem penulisan lagu, sedangkan not adalah satuan dari sistem penulisan tersebut. Yang umum di ketahui oleh masyarakat misalnya notasi angka, yang satuannya berupa angka. Di samping itu ada notasi balok, yang satuannya berupa gambar. Dengan demikian notasi merupakan perwujudan dari sebuah lagu, sedangkan not merupakan perwujudan dari nada. Jika nada dapat didengar, maka not akan terlihat sebagai lambang bagi nada dengan bebagai sifatnya (Tombeng et al., 2018).

Notasi balok adalah notasi yang satuannya berupa lambang gambar yang ditulis di atas paranada (Soeharto, 1992). Membaca notasi balok berarti membaca tulisan musik kemudian dimainkan melalui instrumen musik atau dinyanyikan ke dalam vokal. Kemampuan membaca notasi balok tidak mutlak harus dimiliki oleh seseorang yang sedang belajar musik karena sebuah studi menunjukkan bahwa kemampuan membaca notasi balok tidak berkembang paralel dengan *performance abilities*, yaitu “kemampuan bermain musik” (Syamsina, 2016). Itu sebabnya banyak orang yang tidak mengerti cara membaca notasi musik berbentuk balok. Maka dibuatlah sistem pendeteksi kunci nada pada sebuah citra partitur berdasarkan klasifikasi data.

Untuk merealisasikan sistem tersebut, perancangan deteksi citra dilakukan berdasarkan klasifikasi data notasi musik dengan batasan yang telah ditentukan. Teknologi informasi dan komunikasi saat ini berkembang sangat pesat dan telah menjadi bagian penting kehidupan manusia. Hal ini menyebabkan berbagai macam informasi digital terutama dalam bentuk gambar setiap hari terus mengalami pertumbuhan dan jumlahnya semakin besar. Mengelola data gambar dengan volume yang sangat besar tidak mudah dilakukan karena membutuhkan waktu yang sangat lama, sehingga dibutuhkan suatu sistem yang dapat mengelola dan mengelompokkan data gambar tersebut berdasarkan kelasnya.

Banyak macam metode yang dapat digunakan untuk klasifikasi seperti menggunakan metode algoritma *K-Nearest Neighbor* (K-NN) pada analisa rekomendasi penerima beasiswa (Noviana, Susanti, & Susanto, 2019), metode *data mining*, *Naive Bayes* dan K-NN pada penelitian Sistem Prediksi Produktifitas Pertanian Padi (Maesaroh, 2017), metode *Smooth Support Vector Machine* (SSVM) pada penelitian Klasifikasi Pasien *Diabetes Mellitus* (Nugroho, Tarno, & Prahutama, 2017). Selain metode tersebut baru-baru ini sedang marak penelitian mengenai klasifikasi menggunakan metode *Convolutional Neural Network* (CNN) salah satu

contoh penelitiannya adalah Klasifikasi Citra Wayang Punakawan (Salsabila, 2018). Pada penelitian tersebut membuktikan bahwa metode CNN adalah metode yang paling efektif dibanding dengan metode lainnya seperti metode KNN.

Berdasarkan penelitian yang telah ada mengenai klasifikasi menggunakan metode CNN (Arrofiqoh & Harintaka, 2018), CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis (Hu et al., 2015).

Sistem pendeteksi notasi balok ini dibuat berdasarkan dengan teknik pembelajaran *deep learning*. Berdasarkan (Zhang et al., 2018) pada penelitian (Arrofiqoh & Harintaka, 2018) metode CNN merupakan salah satu metode *deep learning* yang mampu melakukan proses pembelajaran mandiri untuk pengenalan objek, ekstraksi objek dan klasifikasi serta dapat diterapkan pada citra resolusi tinggi yang memiliki model distribusi nonparametrik. Teknik pembelajaran *deep learning* baru-baru ini menjadi domain yang berkembang pesat dan sekarang digunakan secara rutin untuk tugas klasifikasi dan prediksi, seperti pengenalan gambar dan suara, serta terjemahan. *Deep learning* adalah repertoire teknik pembelajaran mesin (ML), berdasarkan pada jaringan saraf tiruan. Aspek kuncinya dan dasar umumnya adalah istilah yang mendalam, yang berarti bahwa ada beberapa lapisan yang memproses banyak tingkat abstraksi, sebagai cara untuk mengekspresikan representasi yang kompleks dalam hal yang lebih sederhana representasi (Briot, n.d.).

Berdasarkan buku berjudul *Deep Learning* (Goodfellow, Bengio, & Courville, 2016) beberapa tahun belakangan ini teknik pembelajaran *deep learning* sering digunakan diberbagai pembuatan model penelitian, dikarenakan terbukti menghasilkan model yang menakjubkan. Hal tersebut dikarenakan Teknik ini memiliki proses komputasi yang kuat, serta proses pelatihan model yang lebih mendalam.

1.2 Rumusan Masalah

Berdasarkan uraian permasalahan pada latar belakang di atas, maka rumusan masalah pada makalah ini adalah bagaimana mendeteksi gambar notasi balok menggunakan metode *Convolutional Neural Network* sehingga menghasilkan kunci nada.

1.3 Batasan Masalah

Dalam penelitian ini, peneliti melakukan pembatasan masalah terhadap masalah penelitian yang akan dilakukan, yakni:

1. Metode pengumpulan data yang digunakan dalam penelitian ini adalah studi pustaka dan observasi.
2. Data yang dipakai untuk perancangan sistem pendeteksi notasi balok berdasarkan hasil dari deteksi objek dengan data set gambar notasi balok dari not c hingga c' pada kunci G.
3. Hasil dari sistem ini hanyalah mendeteksi dari dokumen lagu berformat notasi balok menjadi kunci nada yang memudahkan pembaca, belum sampai mengeluarkan suara musik yang dideteksi.

1.4 Tujuan

Tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah untuk mempermudah orang awam yang ingin mempelajari partitur untuk membaca notasi balok.

1.5 Manfaat

2.1.1. Bagi Mahasiswa

1. Dapat mempelajari lebih dalam mengenai perancangan dan pembuatan sistem pendeteksi notasi balok khususnya menggunakan metode *Deep Learning*
2. Dapat menerapkan ilmu-ilmu yang sudah didapat selama perkuliahan
3. Membandingkan teori yang telah didapat saat kuliah dengan masalah yang sebenarnya.

4. Mengamati teknik-teknik yang diterapkan di lapangan dalam bidang Teknik Informatika
5. Dapat mengembangkan penelitian yang sebelumnya telah dibuat.

2.1.2. Bagi Universitas

1. Memberikan gambaran tentang penguasaan materi pelajaran yang diperoleh selama masa perkuliahan.
2. Menambah referensi literatur ke pustakaan Universitas Islam Negeri Syarif Hidayatullah Jakarta.
3. Bahan referensi bagi mahasiswa Universitas Islam Negeri Syarif Hidayatullah Jakarta dalam pengembangan penelitian lain terkait penulisan ini.

2.1.3. Bagi Pembaca

1. Memberikan pengetahuan terkait perancangan dan pembuatan *Music generator* pada dokumen lagu berbentuk balok.
2. Memberikan kemudahan dalam pembacaan dan memahami dokumen lagu atau music dalam bentuk notasi balok menjadi notasi angka

1.6 Metode Penelitian

Dalam penyusunan tugas akhir berjudul penulis menggunakan metode sebagai berikut :

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data yang penulis lakukan adalah studi literatur dan pustaka. Penulis mengumpulkan data dari buku atau literatur sejenis yang berkaitan dengan penelitian sebagai referensi penulis dalam menyelesaikan skripsi ini. Selain data penelitian berupa literatur, penelitian ini juga membutuhkan data gambar partitur yang penulis kumpulkan dari media internet.

1.6.2 Metode Implementasi

Penulis melakukan implementasi pada pembuatan sistem pendeteksi notasi balok ini dalam 4 tahap, Analisis Kebutuhan sistem, Perancangan Sistem pendeteksi notasi balok dengan metode *Convolutional Neural Network*, Implementasi metode CNN untuk pendeteksi notasi balok, Pengujian sistem.

1.7 Sistematika Penulisan

Untuk memudahkan dalam penulisan skripsi, penulis menyusunnya ke dalam enam bab. Setiap bab-nya terdiri dari beberapa sub bab tersendiri. Bab tersebut secara keseluruhan saling berkaitan satu sama lain, dimana diawali dengan bab pendahuluan dan diakhiri bab penutup yang berisi kesimpulan dan saran. Secara singkat akan diuraikan sebagai berikut:

BAB I

PENDAHULUAN

Dalam bab ini akan dibahas mengenai latar belakang, batasan masalah, tujuan, dan manfaat serta sistematika penulisan.

BAB II

LANDASAN TEORI

Dalam bab ini akan dibahas mengenai berbagai teori yang mendasari analisis permasalahan dan berhubungan dengan topik yang dibahas.

BAB III

METODE PENELITIAN

Bab ini membahas mengenai metode penelitian yang akan digunakan dalam merancang dan membangun Sistem Pendeteksi Notasi Balok.

BAB IV

IMPLEMENTASI SISTEM PENDETEKSI NOTASI BALOK

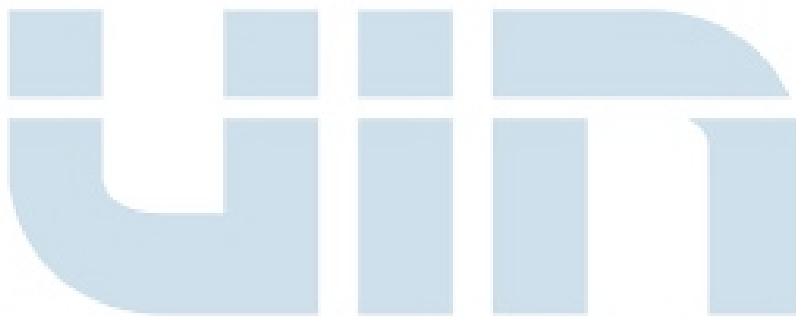
Pada bab ini membahas mengenai hasil dari analisis, perancangan, implementasi dan pengujian sesuai dengan metode yang dilakukan pada sistem yang dibuat serta hasil dari pengujian.

BAB V HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil dan pembahasan rancangan pembuatan sistem pendeteksi notasi balok menggunakan metode *Convolutional Neural Network*.

BAB VI PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diusulkan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih baik.



BAB II

LANDASAN TEORI

Bab ini akan membahas materi tentang teori yang terkait dan mendukung penelitian yang didapat dengan melakukan studi pustaka sebagai landasan teori dalam melakukan penelitian. Sebelum membahas materi teori yang terkait, pada bab ini akan dibahas terlebih dahulu penelitian terkait yang berhubungan dengan pembuatan sistem ini dan yang menjadi acuan penelitian ini.

Pada penelitian pertama berupa thesis milik (Salsabila, 2018) yang berjudul “Penerapan Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Wayang Punakawan”. Pada thesis ini, peneliti membuktikan penerapan metode CNN pada klasifikasi citra wayang Punakawan yang dimana menggunakan 4 kelas atau variabel, yaitu Wayang Gareng, Wayang Semar, Wayang Petruk, dan Wayang Bagong. Jumlah sampel yang dikumpulkan sebanyak 1200 citra Wayang Punakawan dengan 300 citra per variabelnya. Pada penelitian ini dihasilkan nilai akurasi yang tinggi, yaitu sebesar 91,6% yang dipengaruhi banyaknya data train yang digunakan.

Selanjutnya penelitian sejenis kedua yang merupakan jurnal dari (Rokhana, Priambodo, Karlita, Sunarya, & Yuniarno, 2019) yang berjudul “Convolutional Neural Network untuk Pendeteksian Patah Tulang Femur pada Citra Ultrasonik B-Mode”. Jurnal ini meneliti tentang bagaimana metode Convolutional Neural Network mendeteksi patah tulang femur pada citra Ultrasonik B-Mode. Penelitian ini menunjukkan bahwa CNN memiliki kinerja hasil yang lebih unggul disbanding dengan metode Support Vector Machine (SVM) dan k-Neural Network (k-NN). Penelitian ini diuji dan dilate menggunakan 4800 dataset, dengan 2 kelas dimana 3200 data untuk kelas tulang patah dan 1600 untuk kelas tidak patah. Selain itu penelitian ini diuji coba menggunakan 5 macam model CNN yang berbeda dan menghasilkan akurasi yang paling besar 95,3%.

Penelitian sejenis selanjutnya adalah sebuah thesis yang berjudul “Implementasi Deep Learning Berbasis Tensorflow untuk Pengenalan Sidik Jari” yang disusun oleh Royani Darma Nurfita (Nurfita & Ariyanto, 2018). Pada thesis ini, penulis meneliti bagaimana system yang dibuat dapat mengenali sidik jari. System ini dibuat menggunakan metode Convolutional Neural Network. Pada thesis ini digunakan dataset yang diambil dari website yang menggunakan sensor optical dan terdapat 80 gambar sebagai datasetnya dengan ukuran 24 x 24 px. Dengan tiga kali percobaan menggunakan learning-rate yang berbeda, hasil dari penelitian ini menghasilkan akurasi yang sempurna yaitu 100%.

Yang keempat adalah sebuah thesis berjudul “Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) pada Citra Wayang Golek” oleh Triano Nurhikmat (Nurhikmat, 2018). Pada thesis ini juga meneliti dan membuat system berdasarkan metode CNN untuk pengenalan wayang Golek dan citra digital sebagai inputannya. Data set yang digunakan berupa gambar yang dikumpulkan melalui metode Crawling dari search engine google. Jumlah sampel yang digunakan adalah 240 data dan jumlah untuk validasi sebanyak 60 data gambar. Data tersebut dibagi menjadi 3 label, yaitu Cepot, Gatot Kaca, dan Semar. Nilai akurasi yang didapatkan sebesar 95% pada training dan 90% pada testing.

Penelitian sejenis lain juga ada pada jurnal berjudul “Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi” oleh Erlyna Nour Arrofiqoh dan Harintaka (Arrofiqoh & Harintaka, 2018). Pada jurnal ini digunakan citra dengan resolusi yang tinggi dari teknologi UAV (Unmanned Aerial Vehicle). Digunakan metode deeplearning dengan CNN untuk mengimplementasi pendekatan klasifikasi yang dapat membedakan jenis tanaman sebanyak 5 macam jenis atau kelas, yaitu kelas tanaman padi, bawang merah, kelapa, pisang, dan cabai. Penelitian ini menggunakan dataset sebanyak 500 gambar berukuran 227 x 227 px dimana tiap kelasnya berjumlah 100 data. Hasil akurasi yang didapatkan pada penelitian ini adalah 100% pada proses learningnya, sedangkan pada data validasi menghasilkan 93% dan 82% pada data testnya.

Selain itu penelitian sejenis terdapat pada jurnal yang berjudul “Penggunaan Metode Backpropagation Artificial Neural Network dalam Sistem Pengenalan Notasi Balok menjadi MIDI” oleh Dewi Kusumaningsih dan Stefanus Adhie Putra Pramudita (Kusumaningsih, Adhie, & Pramudita, 2016). Pada penelitian ini merupakan sebuah *Machine Learning* untuk pengenalan pola dari sebuah citra menggunakan Jaringan Syaraf Tiruan (JST). JST yang digunakan adalah JST Backpropagation. Kasus yang dipilih pada penelitian ini adalah pengenalan notasi balok berbasis citra digital berformat .jpg. pada penelitian ini dibuat aplikasi yang akan mengenali pola notasi balok untuk diproses dan diubah menjadi notasi angka yang selanjutnya akan dapat dibuatkan sebuah file music dengan format MIDI.

Terakhir, penelitian terkait ada pada jurnal milik John Pierre Haumahu (Haumahu, 2019) yang berjudul “Implementasi Jaringan Syaraf Tiruan Untuk Pengenalan Pola Notasi Balok Menggunakan Metode *Backpropagation*”. Penelitian pada jurnal ini membahas bagaimana hasil yang ada saat metode *backpropagation* digunakan untuk pengenalan pola notasi balok. Pada penelitian ini, terdapat 7 gambar sebagai dataset dimana 6 gambar sebagai data latih dan 1 gambar tidak terpakai. Setelah melalui proses pengujian, dihasilkan presentase sebesar 91,2% untuk pengenalan gambar input dan sasarannya.

Setelah pembahasan mengenai penelitian yang sejenis, selanjutnya akan dibahas landasan teori yang digunakan pada pembuatan system dan penelitian ini. Berikut adalah tabel perbandingan penelitin yang sejenis.

Tabel 2.1. Perbandingan Penelitian Sejenis

Peneliti	Metode	Kasus	Jumlah DataSet	Kelas	Akurasi
(Salsabila, 2018)	CNN	Klasifikasi Citra Wayang Punakawan	1200	4	91,6%
(Rokhana,dkk)	CNN	Pendeteksian Patah Tulang Femur	4800	2	95,3%.
(Nurfita,dkk)	CNN	Pengenalan Sidik Jari	80	-	100%

Triano Nurhikmat	CNN	Image Classification pada Citra Wayang Golek	300	3	95%
Arrofiqoh & Harintaka	CNN	Klasifikasi Tanaman pada Citra Resolusi Tinggi	500	5	93%
Kusumaningsih	<i>Backpropagation</i>	Analisa Sistem Pengenalan Notasi Balok menjadi MIDI	-	-	-
John Pierre Haumahu	<i>Backpropagation</i>	Pengenalan Pola Notasi Balok	7	-	91,2%

2.1. *Digital Image* atau Citra Digital

Berdasarkan penelitian (Putri, 2018) *Digital Image* atau Citra digital adalah citra elektronik yang diambil dari beberapa jenis dokumen, yaitu berupa foto, buku, ataupun video dan suara. Proses yang digunakan untuk merubah citra analog menjadi suatu citra digital disebut sebagai proses digitasi. Digitasi adalah proses dimana mengubah suatu gambar, teks, atau suara yang berasal dari benda dapat dilihat ke dalam data elektronik dan dapat disimpan serta diproses untuk keperluan yang lainnya.

Citra digital merupakan sebuah representasi numerik (mayoritas biner) dari gambar 2 dimensi. Sebuah gambar dapat didefinisikan sebagai fungsi 2 dimensi $f(x,y)$ di mana x dan y merupakan titik koordinat bidang datar, dan harga dari fungsi f dari setiap pasangan titik koordinat (x,y) yang disebut dengan intensitas atau level keabuan (grey level) dari suatu gambar. Ketika nilai titik x,y dan nilai intensitas f terbatas dengan nilai diskrit, maka gambar tersebut akan dapat dikatakan sebagai sebuah citra digital (Gonzales, Rafael, & Woods, 2002).

Menurut Sutoyo, dalam konteks yang lebih luas, pengolahan citra digital lebih mengacu pada pemrosesan setiap dua data dimensi. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang bagaimana teknik

pengolahan sebuah citra. Citra yang dimaksud disini adalah sebuah gambar diam (foto) maupun gambar bergerak (Video). Sedangkan digital disini mempunyai maksud penting bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Dalam citra digital terdapat sebuah larik (*array*) yang berisi nilainilai real maupun kompleks yang di representasikan dengan derean bit tertentu (Nurhikmat, 2018).

Dalam sebuah komputer, citra digital dipetakan menjadi bentuk grid dan elemen piksel berbentuk matriks 2 dimensi. Setiap piksel-piksel tersebut memiliki angka yang mempresentasikan channel warna. Angka pada setiap piksel disimpan secara berurutan oleh sebuah komputer dan sering dikurangi untuk keperluan kompresi maupun pengolahan tertentu. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel. (*pixel* = picture element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh karena itu, citra dapat dituliskan kedalam sebuah matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Berdasarkan rumus diatas, suatu citra $f(x,y)$ dapat dituliskan kedalam fungsi matematis seperti berikut ini :

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x,y) \leq G-1$$

Dimana :

M = jumlah piksel baris pada array citra

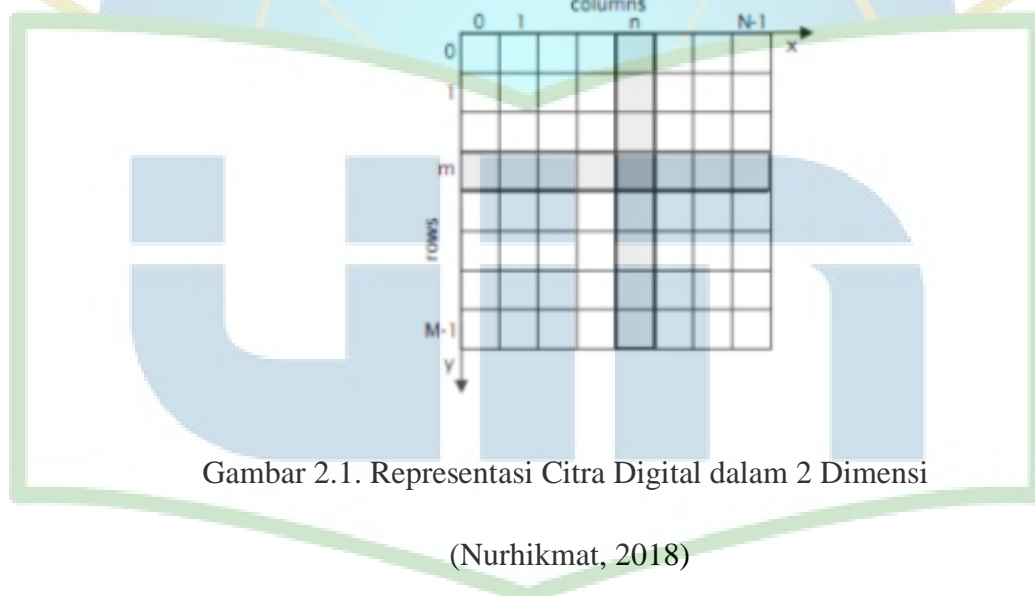
N = jumlah piksel kolom pada array citra

G = nilai skala keabuan (grayscale)

Besarnya nilai M , N , dan G biasanya merupakan perpangkatan dari dua seperti yang terlihat pada persamaan berikut :

$$M = 2^m ; N = 2^n ; G = 2^k$$

Dimana nilai m , n , dan k merupakan bilangan positif. Interval $(0, G)$ disebut dengan (*grayscale*). Besarnya nilai G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan) (Nurhikmat, 2018).



Pengolahan citra adalah suatu proses pengolahan citra dengan menggunakan komputer menjadi sebuah citra yang memiliki kualitas yang lebih baik. Tujuan dari pengolahan citra ini adalah memperbaiki kualitas suatu citra sehingga dapat diinterpretasi dengan mudah oleh manusia atau atau sebuah mesin (komputer) (Nurhikmat, 2018).

2.2. Klasifikasi Gambar

Klasifikasi gambar atau citra adalah sebuah pekerjaan untuk memasukkan citra dan menempatkan ke dalam suatu kategori. Ini merupakan salah satu dari permasalahan yang ada pada *Computer Vision* yang dapat disederhanakan dan memiliki berbagai macam aplikasinya. Salah satu aplikasi dalam klasifikasi citra adalah pengklasifikasian nama tempat pada suatu citra.

Setiap citra yang di input pada *training set data* diberikan label atau penamaan. Saat klasifikasi, label atau penamaan tersebut akan menjadi perbandingan dengan hasil hipotesis yang diberikan oleh model pembelajaran dan akan menghasilkan nilai error. Klasifikasi yang terawasi ini bisa sangat efektif dan akurat dalam mengklasifikasikan citra tempat maupun objek lainnya. Banyak metode dan algoritma yang dapat mendukung proses klasifikasi yang terawasi terutama dengan teknik *Deep Learning* (Putri, 2018).

2.3. Jaringan Syaraf Tiruan / Artificial Neural Network (ANN)

Jaringan Syaraf Tiruan / *Artificial Neural Network (ANN)* merupakan metode yang biasanya digunakan dalam peramalan maupun pengenalan pola. Pada peramalan jaringan syaraf tiruan biasa digunakan sebagai peramalan nilai tukar mata uang asing, peramalan harga saham, peramalan cuaca dan lain sebagainya, sedangkan untuk pengenalan pola biasanya jaringan syaraf tiruan digunakan untuk pengenalan pola huruf, pola tanda tangan hingga pola suara serta wajah (Kartika, Irawan, & Triyanto, 2016).

Berdasarkan (Nurhikmat, 2018), ANN merupakan sistem adatif yang dapat mengubah strukturnya untuk memecahkan suatu masalah berdasarkan informasi internal maupun eksternal. Menurut Pham dalam jurnal Hermantoro (Pham, 1994) mengatakan bahwa ANN bersifat fleksibel terhadap inputan data dan menghasilkan output respon konsisten. ANN telah banyak digunakan dalam area yang luas. Menurut Kumar & Haynes (Kumar, 2003) dalam jurnal Ulil Hamida (Hamida, 2014) menjelaskan, penerapan ANN dapat mengidentifikasi beberapa aplikasi yaitu:

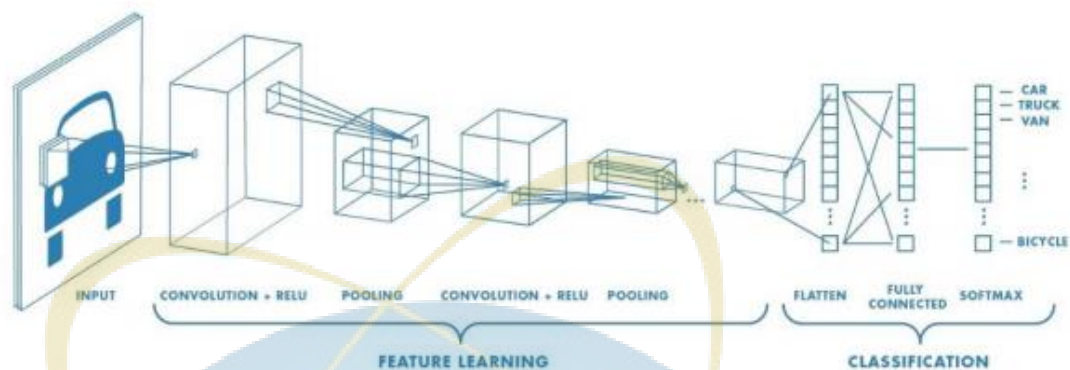
- Estimasi/prediksi (aproksimasi fungsi, peramalah)
- Pengenalan Pola (klasifikasi, diagnosis, dan analisis diskriminan)
- Klustering (pengelompokan tanpa adanya pengetahuan sebelumnya).

2.4. CNN (*Convolutional Neural Network*)

Convolution Neural Network bermula dari Yann LeCun dan teman-temannya berhasil melakukan klasifikasi citra kode zip menggunakan kasus khusus dari Feed Forward Neural Network Pada tahun 1989. Mulai pada tahun 2012, dimana perangkat keras *Graphical Processing Unit* (GPU) berkembang, DNN juga ikut berkembang dan bahkan sebuah CNN dapat melakukan pengenalan citra dengan akurasi yang sangat baik dan dapat menyaingi manusia pada dataset tertentu (Sugiarto, Kristian, & Setyaningsih, 2018).

Convolutional Neural Network (CNN) merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar (Ilmiah et al., 2018). *Convolutional Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. Pada CNN, setiap neuron direpresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. CNN termasuk dalam *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Putra, Wijaya, & Soelaiman, 2016). CNN hampir sama dengan neural network pada umumnya yang memiliki neuron yang memiliki bobot dan bias. CNN memiliki 1 tahap *training* (*Supervised Backpropagation*).

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa array yang biasa disebut feature map. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*. Berikut adalah jaringan arsitektur *Convolutional Neural Network* berdasarkan penelitian (Nurhikmat, 2018):



Gambar 2.2. Arsitektur *Convolutional Neural Network*

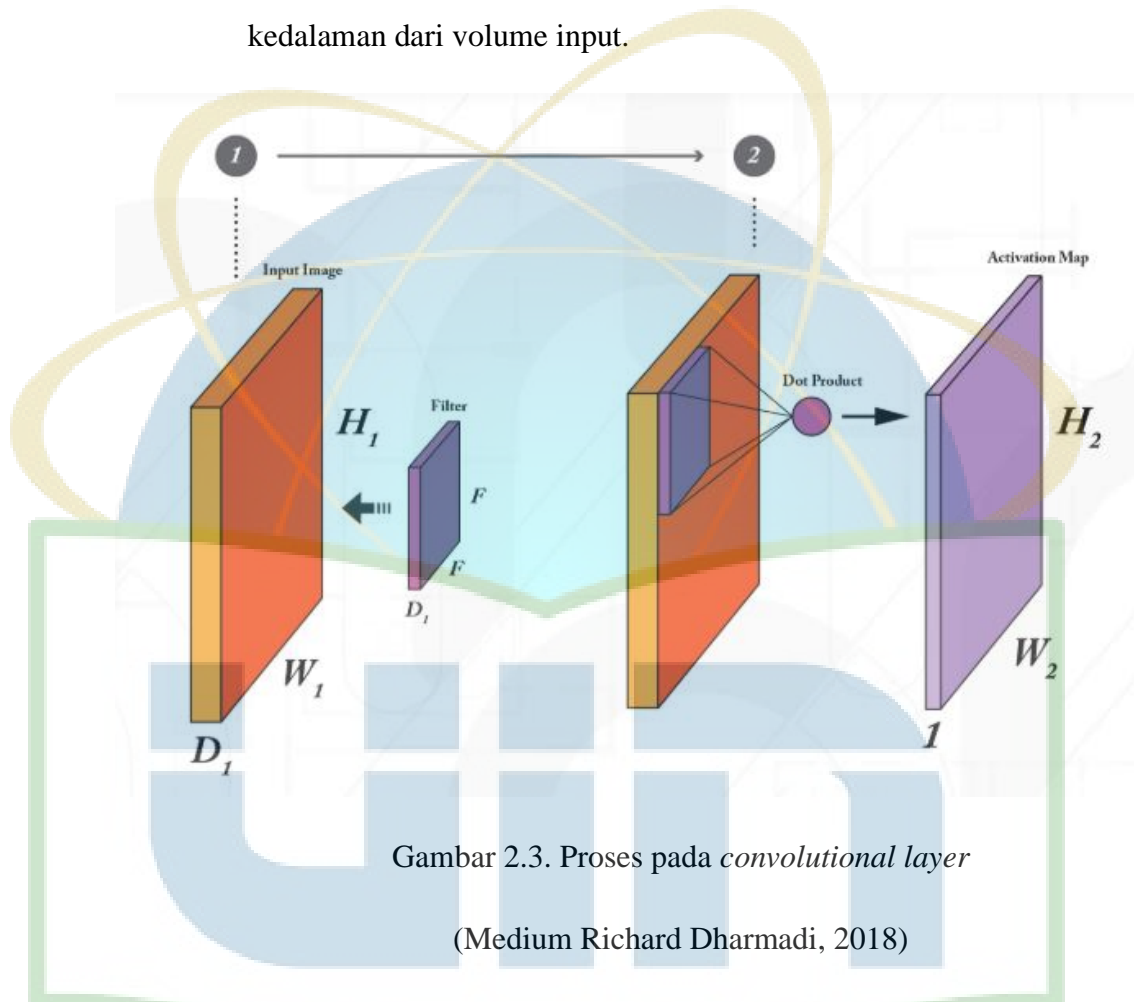
(Nurhikmat, 2018)

Berdasarkan Gambar 2.2., Tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

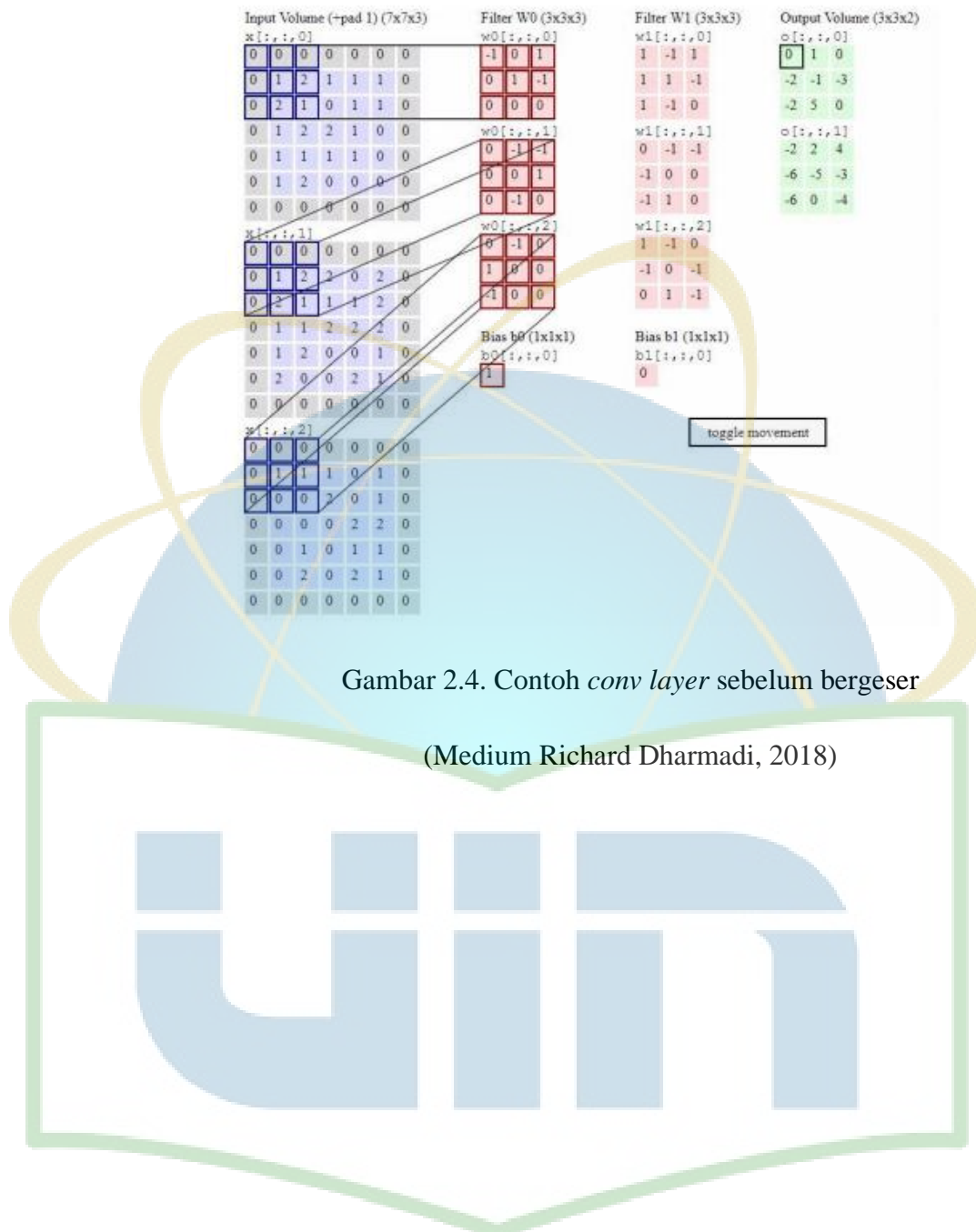
2.4.1. Convolution Layer

Pada proses konvolusi, filter menjadi peran utama untuk proses pelatihan mesin. Seperti layaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal tertentu. Filter ini diinisialisasi dengan nilai tertentu (random atau menggunakan teknik tertentu seperti Glorot), dan nilai dari filter inilah yang menjadi parameter yang akan di-*update* dalam proses *learning*.

Gambar input CNN selalu berbentuk kotak. Proses untuk gambar *non-rectangular* masih belum diketahui. Filter pun mengikuti karakteristik kotak tersebut. Parameter lapisan konvolusi terdiri dari satu set filter yang dapat dipelajari. Setiap filter kecil secara spasial (Panjang x lebar x tinggi), tetapi meluas melalui kedalaman dari volume input.

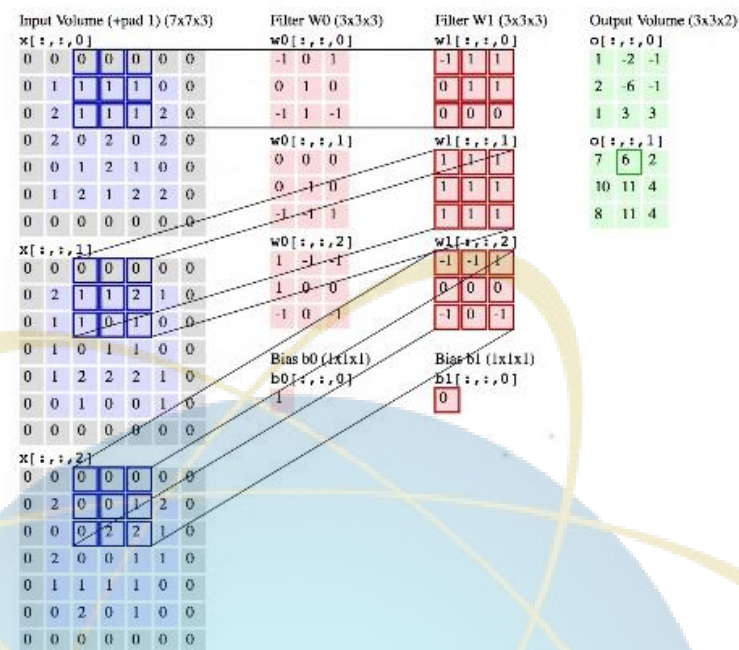


Pada satu gambar dihasilkan sebuah angka berdasarkan warna gambar yang nantinya menjadi identitas gambar tersebut. Dengan menggeser (*convolve*) filter di setiap kemungkinan posisi filter pada gambar, dihasilkan sebuah *activation map*.



Gambar 2.4. Contoh *conv layer* sebelum bergeser

(Medium Richard Dharmadi, 2018)

Gambar 2.5. Contoh *conv layer* setelah bergeser

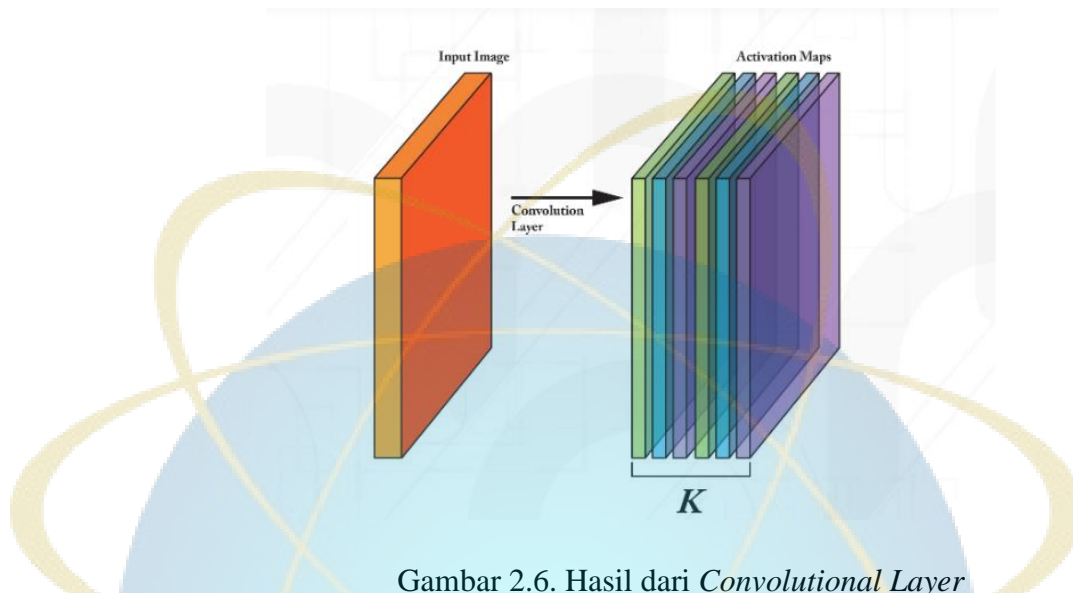
(Medium Richard Dharmadi, 2018)

Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Konvolusi 2 buah fungsi $f(x)$ dan $g(x)$ didefinisikan sebagai berikut :

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a)g(x - a)da$$

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN ((Putra et al., 2016)

Proses pemberian filter dilakukan berulang kali dengan beberapa filter berbeda, hingga menghasilkan “gambar” baru yang merupakan kumpulan dari *activation maps*.



Gambar 2.6. Hasil dari *Convolutional Layer*

(Medium Richard Dharmadi, 2018)

Berdasarkan artikel Richard Dharmadi, ada beberapa hal yang perlu diperhatikan pada *convolutional layer*, yaitu:

- Ukuran ketebalan dari sebuah filter selalu mengikuti ketebalan/volume dari gambar input yang digunakan.
- Tinggi (dan lebar) filter (F) pada umumnya berukuran ganjil. Secara intuisi, filter berukuran ganjil memberikan representasi yang lebih baik karena mencakup bagian kiri dan kanan yang “seimbang”.
- Dalam sebuah *convolutional layer*, filter-filter yang digunakan berukuran sama, untuk kemudahan proses komputasi.
- Jumlah filter (K) yang digunakan dalam sebuah *convolutional layer* adalah kelipatan 2 (powers of 2). Beberapa *library* memiliki *subroutine* khusus untuk komputasi kernel/dimensi berkelipatan 2 yang dapat meningkatkan efisiensi.

- Besaran *zero padding* (P) umumnya menyesuaikan agar ukuran spasial dari *output* yang dihasilkan tetap sama dengan ukuran spasial *input*. ($P = F - 1/2$)

Selain itu operasi konvolusi dapat dilihat sebagai perkalian perkalian matriks antara citra masukan dan kernel dimana keluarannya dihitung dengan *dot product*. Selain itu, penentuan volume output juga dapat ditentukan dari masing-masing lapisan dengan hyperparameters. Hyperparameter yang digunakan pada persamaan di bawah ini digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali output. Perhatikan persamaan berikut:

$$(W - F + 2P)/(S + 1)$$

Keterangan :

W = Ukuran volume gambar

F = Ukuran Filter

P = Nilai Padding yang digunakan

S = Ukuran Pergeseran (Stride)

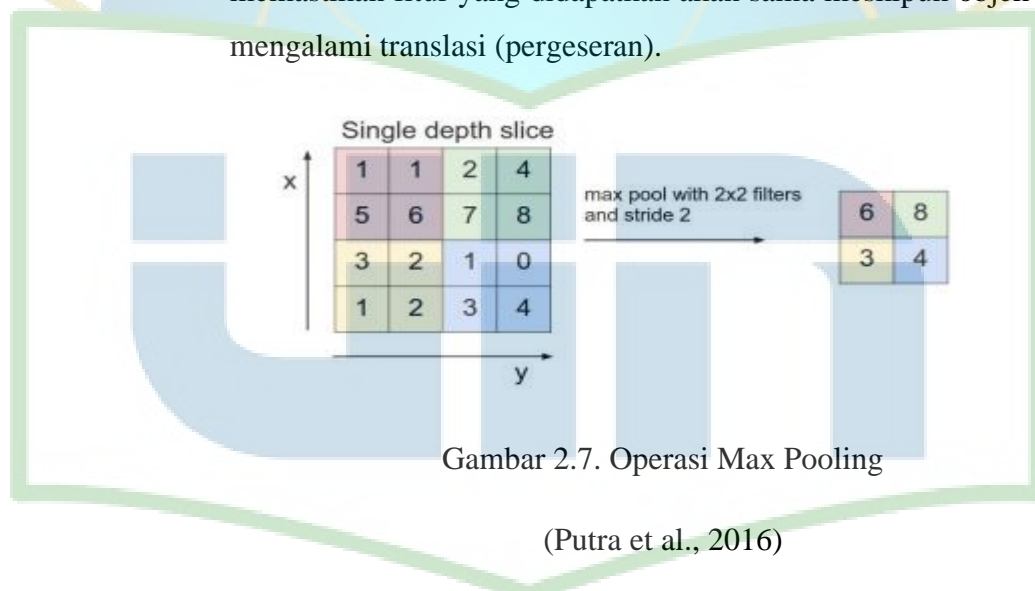
Berdasarkan persamaan di atas, dapat dihitung ukuran spasial dari volume output dimana *hyperparameter* yang dipakai adalah ukuran volume (W), filter (F), *Stride* yang diterapkan (S) dan jumlah padding nol yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser filter melalui input citra dan *Zero Padding* adalah nilai untuk mendapatkan angka nol di sekitar border citra.

2.4.2. Operasi *Pooling*

Setelah melewati proses konvolusi kemudian hasilnya akan melalui tahap *pooling*. Fungsi dari *pooling* ini adalah untuk

mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*. Umumnya, metode *pooling* yang digunakan adalah *max pooling* atau mengambil nilai terbesar dari bagian tersebut.

Berdasarkan penelitian (Putra et al., 2016) *Max Pooling* adalah proses untuk meningkatkan invariansi posisi dari fitur menggunakan operasi *Max*. *Max Pooling* membagi *output* dari *Convolutional Layer* menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti pada gambar 2.7. Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran).



Secara umum, operasi *pooling layer* adalah sebagai berikut :

- Menerima sebuah volume dengan ukuran $W1 \times H1 \times D1$
- Membutuhkan dua hiperparameter yakni:

F = ukuran bidang *receptif*

S = stride

- Menghasilkan sebuah volume dengan ukuran $W2 \times H2 \times D2$, dimana :

$$W2 = (W1 - F)/S + 1$$

$$H2 = (H1 - F)/S + 1$$

$$D2 = D1$$

- Memperkenalkan zero parameter karena menghitung fungsi input yang tetap.
- Pooling layer biasanya tidak memasukkan input menggunakan zero-padding.

Pada praktiknya, hanya ada dua variasi yang terlihat pada max pooling layer yaitu sebuah pooling layer dengan $F=3$, $S=2$ (disebut juga sebagai overlapping pooling) dan lebih umum lagi $F=2$, $S=2$. Ukuran pooling dengan bidang reseptif yang lebih besar sangat merusak.

2.4.3. Fully-Connected Layer

Feature map yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga kita harus melakukan “flatten” atau *reshape feature map* menjadi sebuah vector agar bisa kita gunakan sebagai *input* dari *fully-connected layer*. Neuron pada *Fully Connected Layer* memiliki hubungan yang lengkap pada semua aktivasi dalam layer sebelumnya. Aktivasi tersebut kemudian di komputasi dengan sebuah perkalian matriks diikuti oleh bias *offset*. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

Activation map yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga mau tidak mau

harus dilakukan *reshape activation map* menjadi suatu vektor agar bisa digunakan sebagai input dari *fully-connected layer*. Layer ini memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. Layer ini adalah layer yang biasanya digunakan dalam penerapan *multi layer perceptron* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

Setiap neuron pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully-connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, sedangkan *fully-connected layer* hanya dapat diimplementasikan di akhir jaringan.

Convolution layer dengan ukuran kernel 1×1 melakukan fungsi yang sama dengan *fully-connected layer* namun dengan tetap mempertahankan karakter spasial dari data. Sehingga hal tersebut membuat penggunaan layer ini pada CNN sekarang tidak banyak dipakai.

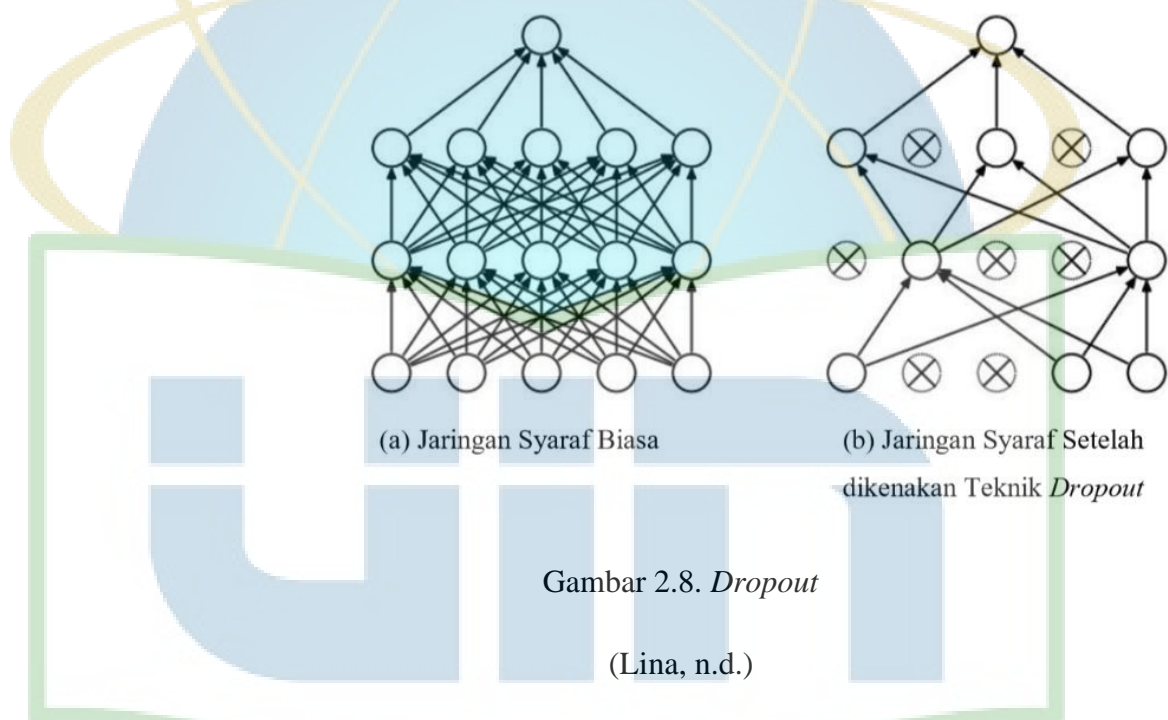
Selain arsitektur yang telah dipaparkan, masih banyak arsitektur lain yang dapat digunakan untuk berbagai karakteristik data maupun arsitektur yang cocok untuk karakteristik/permasalahan tertentu.

2.4.4. Dropout Regularization

Dropout adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Neuron-neuron ini dapat dibilang dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak

diterapkan pada neuron pada saat melakukan *backpropagation* (Lina, n.d.).

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan *neuron* yang berupa *hidden* maupun layer yang *visible* di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas yang bernilai antara 0 dan 1 (Lina, n.d.).



Pada gambar diatas jaringan syaraf (a) merupakan jaringan syaraf biasa dengan 2 lapisan tersembunyi. Sedangkan pada bagian (b) jaringan syaraf sudah diaplikasikan teknik regularisasi *dropout* dimana ada beberapa neuron aktivasi yang tidak dipakai lagi. Teknik ini sangat mudah diimplementasikan pada model CNN dan akan berdampak pada performa model dalam melatih serta mengurangi *overfitting* (Lina, n.d.).

2.4.5. *Softmax Layer*

Fungsi *softmax* digunakan untuk metode klasifikasi dengan jumlah kelas yang banyak, seperti regresi logistik multinomial, analisis diskriminan *linear multiclass*, *Naive Bayes Classifier*, dan *Artificial Neural Network* (ANN). *Softmax* adalah sebuah fungsi yang mengubah K-dimensi vektor 'x' yang berupa nilai sebenarnya menjadi vektor dengan bentuk yang sama namun dengan nilai dalam rentang 0-1, yang jumlahnya 1. Fungsi *softmax* digunakan dalam layer yang terdapat pada neural network dan biasanya terdapat pada layer terakhir untuk mendapatkan output. Tidak jauh berbeda dengan neuron pada umumnya *softmax* neuron menerima input lalu melakukan pembobotan dan penambahan bias. Tetapi setelah itu neuron pada *softmax layer* tidak menerapkan fungsi aktivasi melainkan menggunakan fungsi *softmax*. Dapat disimpulkan bahwa *softmax layer*, menentukan probabilitas terbesar untuk hasil kelasnya

2.5. *Artificial Intelligence (AI)*

Kecerdasan Buatan atau *Artificial Intelligence* (AI) adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan. Untuk melakukan hal ini, setidaknya ada tiga metode yang dikembangkan.

- *Fuzzy Logic*(FL).

Teknik ini digunakan oleh mesin untuk mengadaptasi bagaimana makhluk hidup menyesuaikan kondisi dengan memberikan keputusan yang tidak kaku 0 atau 1. Sehingga dimunculkan sistem logika *fuzzy* yang tidak kaku. Penerapan logika *fuzzy* ini salah satunya adalah untuk sistem pengereman kereta api di Jepang.

- *Evolutionary Computing*(EC).

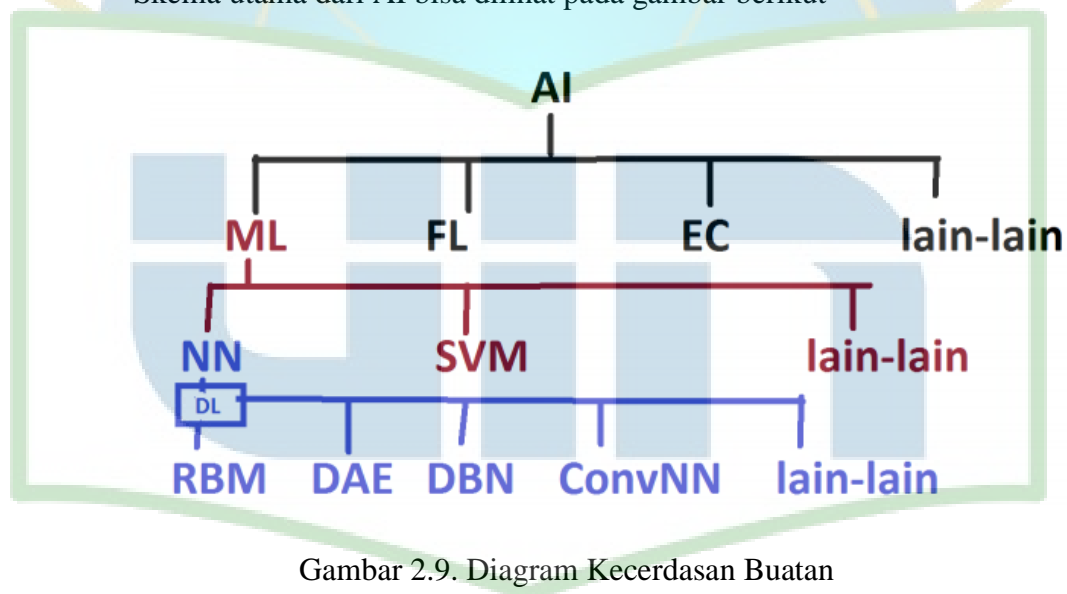
Pendekatan ini menggunakan skema evolusi yang menggunakan jumlah individu yang banyak dan memberikan sebuah ujian untuk

menyeleksi individu terbaik untuk membangkitkan generasi selanjutnya. Seleksi tersebut digunakan untuk mencari solusi dari suatu permasalahan. Contoh dari pendekatan ini adalah Algoritma Genetika yang menggunakan ide mutasi dan kawin silang, *Particle Swarm Optimization* (PSO) yang meniru kumpulan binatang seperti burung dan ikan dalam mencari mangsa, *Simulated Annealing* yang menirukan bagaimana logam ditempa, dan masih banyak lagi.

- *Machine Learning* (ML)

Machine Learning atau pembelajaran mesin merupakan teknik yang paling populer karena banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah. Sesuai namanya ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi (Ahmad, 2017).

Skema utama dari AI bisa dilihat pada gambar berikut



Gambar 2.9. Diagram Kecerdasan Buatan

(Ahmad, 2017)

Berdasarkan diagram pada gambar 2.9. dapat dijelaskan bahwa AI dibagi menjadi beberapa macam algoritma seperti *Machine Learning*, *Fuzzy Logic*, *Evolutionary Computation*, dan lain-lain. *Fuzzy Logic* adalah suatu cabang ilmu *Artificial Intelligence*, yaitu suatu pengetahuan yang membuat komputer dapat

meniru kecerdasan manusia sehingga diharapkan komputer dapat melakukan hal-hal yang apabila dikerjakan manusia memerlukan kecerdasan. *Evolutionary Computation* adalah algoritma untuk optimisasi global yang terinspirasi oleh evolusi biologis, dan subbidang kecerdasan buatan dan komputasi lunak mempelajari algoritma ini. Dalam istilah teknis, mereka adalah keluarga pemecah masalah trial and error berbasis populasi dengan karakter optimisasi metaheuristik atau stokastik.

Dalam metode *Machine Learning* terdapat algoritma NN, SVM, dan lain-lain. *Neural network* atau ANN adalah model yang terinspirasi oleh bagaimana neuron dalam otak manusia bekerja. Tiap neuron pada otak manusia saling berhubungan dan informasi mengalir dari setiap neuron tersebut. Menurut Santoso (2007) *Support vector machine* (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan *Artificial Neural Network* (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*.

Didalam ANN terdapat *Deep Learning* yang mana mesin dilatih lebih mendalam. *Deep Learning* adalah salah satu cabang *Machine Learning* (ML) yang menggunakan *Deep Neural Network* untuk menyelesaikan permasalahan pada domain ML.

2.6. Machine Learning

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. Menurut Mohri dkk (2012) *machine learning* dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat prediksi yang akurat. Definisi pengalaman disini ialah informasi sebelumnya yang telah tersedia dan bisa dijadikan data pembelajar.

Dalam pembelajaran *machine learning*, terdapat beberapa skenario-skenario. Seperti:

- *Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label. Contohnya *Decision tree*, *Nearest - Neighbor Classifier*, *Naive Bayes Classifier*, *Artificial Neural Network*, *Support Vector Machine*, *Fuzzy K-Nearest Neighbor*.

- *Unsupervised Learning*

Penggunaan skenario *Unsupervised Learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui. *K-Means*, *Hierarchical Clustering*, *DBSCAN*, *Fuzzy C-Means*, *Self-Organizing Map*

- *Reinforcement Learning*

Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar.

Saat ini telah banyak pendekatan *machine learning* yang digunakan untuk deteksi *spam*, *Optical character recognition (OCR)*, pengenalan wajah, deteksi penipuan online, *NER (Named Entity Recognition)*, *Part-of-Speech Tagger* (Nurhikmat, 2018).

2.7.DEEP LEARNING

Tidak ada penjelasan konsensual untuk *deep learning* sendiri. *Deep learning* adalah sebuah repertoar dari teknik *machine learning*, yang didasari oleh kecerdasan jaringan syaraf (*artificial neural network*). Kunci aspek dan dasarnya ada pada kata *deep*, yang mana disana terdapat *multiple layers processing multiple levels of abstractions*, sebagai cara untuk mengekspresikan representasi kompleks kedalam representasi yang lebih sederhana (Briot, n.d.).

Deep Learning telah menjadi tantangan untuk didefinisikan bagi banyak orang karena telah berubah bentuk perlahan selama dekade terakhir. Satu definisi yang paling benar menetapkan bahwa *deep learning* berkaitan dengan "jaringan saraf dengan lebih dari dua lapisan." Aspek yang bermasalah dari definisi ini adalah bahwa *deep learning* terdengar seolah-olah telah ada sejak tahun 1980-an. *deep learning* sangat berkaitan dengan "*neural network with more than two layers*." Namun definisi tersebut malah membuat *deep learning* terdengar sudah ada sejak tahun 1980an. *Neural network* atau jaringan saraf lebih baik dari segi arsitektural dibanding jaringan sebelumnya (lebih banyak pada kekuatan *processing*), hal itu dibuktikan pada beberapa tahun belakangan bahwa hasilnya lebih optimal. Berikut ini adalah beberapa aspek dalam perkembangan jaringan saraf:

- Neuron semakin bertambah dari jaringan sebelumnya
- Lebih kompleks dalam menghubungkan layers pada jaringan saraf
- Jumlah daya komputasi yang digunakan untuk proses train sangat besar

- Ekstraksi fitur dilakukan secara otomatis

Deep learning memiliki empat arsitektur mendasar, yakni:

- *Unsupervised pretrained networks*
- *Convolutional neural networks*
- *Recurrent neural networks*
- *Recursive neural networks*

Selain itu terdapat pula beberapa variasi lain, seperti *hybrid convolutional* dan *recurrent neural network*.

Ekstraksi fitur otomatis merupakan keunggulan *deep learning* dibanding dengan algoritma *machine learning* terdahulu. Ekstraksi fitur mengutamakan pemberian label pada dataset yang ada dalam proses jaringan.

Secara historis, pelatihan *machine learning* menghabiskan berbulan-bulan, bertahun-tahun, bahkan puluhan tahun dalam pembuatan set fitur lengkap untuk klasifikasi data secara manual. *Deep learning* merupakan algoritma paling akurat sejauh ini (Patterson & Gibson, 2017).

2.8. Precision dan Recall

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Gunanya fungsi ini agar akurasi model tidak serta merta dipercaya namun cukup dapat dipertanggungjawabkan kebenarannya karena melihat akurasi model dari sisi lain.

2.9. Epoch

Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Netwok sampai dikembalikan ke awal untuk sekali putaran, karena satu Epoch terlalu besar untuk dimasukkan (feeding) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (batches). Didalam *epoch* inilah tiap bobot / *weight* ditambahkan.

2.10. Python

Python merupakan salah satu contoh bahasa tingkat tinggi. Contoh lain bahasa tingkat tinggi adalah Pascal, C++, Pert, Java, dan sebagainya. Sedangkan bahasa tingkat rendah merupakan bahasa mesin atau bahasa assembly. Secara sederhana, sebuah komputer hanya dapat mengeksekusi program yang ditulis dalam bentuk bahasa mesin. Oleh karena itu, jika suatu program ditulis dalam bentuk bahasa tingkat tinggi, maka program tersebut harus diproses dulu sebelum bisa dijalankan dalam komputer. Hal ini merupakan salah satu kekurangan bahasa tingkat tinggi yang memerlukan waktu untuk memproses suatu program sebelum program tersebut dijalankan. Akan tetapi, bahasa tingkat tinggi mempunyai banyak sekali keuntungan. Bahasa tingkat tinggi mudah dipelajari, mudah ditulis, mudah dibaca, dan tentu saja mudah dicari kesalahannya. Bahasa tingkat tinggi juga mudah diubah portabel untuk disesuaikan dengan mesin yang menjalankannya. Hal ini berbeda dengan bahasa mesin yang hanya dapat digunakan untuk mesin tersebut. Dengan

berbagai kelebihan ini, maka banyak aplikasi ditulis menggunakan bahasa tingkat tinggi. Proses mengubah bentuk bahasa tingkat tinggi ke tingkat rendah dalam bahasa pemrograman ada dua tipe, yakni interpreter dan compiler. (Utami, 2004)

2.11. Keras

Keras merupakan *framework* yang dibuat untuk mempermudah pembelajaran terhadap komputer. Keras juga merupakan sebuah perangkat lunak yang berjalan diatas *platform* Python untuk mendukung proses komputasi cerdas diantaranya adalah jaringan Konvolusi. Keras diciptakan oleh Francois Chollet. (Ilmiah et al., 2018)

Keras adalah API *neural network* berlevel tinggi yang ditulis dengan bahasa pemrograman Python. Keras dapat berjalan diatas Tensorflow, CNTK, atau Theano sebagai *backend*. Keras dapat digunakan dalam penelitian *deep learning* yang memerlukan pembuatan prototipe yang mudah dan cepat dan *user friendly*. Selain itu Keras juga mendukung komputasi pada *Convolutional Neural Network* dan *Reccurent Neural Network* ataupun kombinasi dari keduanya. Keras dapat berjalan dengan baik pada CPU maupun GPU. Keras kompatibel dengan Python versi 2.7 sampai versi 3.6 (Keras.io.).

2.12. Tensorflow

TensorFlow adalah perpustakaan perangkat lunak, yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. TensorFlow kemudian menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan.

Fitur utamanya meliputi:

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan array multi dimensi (tensors).
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin

3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. Tensorflow bisa menulis kode yang sama dan menjalankannya baik di CPU atau GPU. Lebih khusus lagi, TensorFlow akan mengetahui bagian perhitungan mana yang harus dipindahkan ke GPU.
4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

2.13. Open CV

OpenCV adalah sebuah library (perpustakaan) yang digunakan untuk mengolah gambar dan video hingga kita mampu meng-ekstrak informasi didalamnya. OpenCV dapat berjalan di berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan juga support diberbagai platform seperti Windows, Linux, Mac OS, iOS dan Android. Salah satu contoh sederhana dalam penggunaan OpenCV adalah bagaimana kita dengan mudah bisa mendeteksi wajah dalam sebuah gambar.

2.14. Google Colaboratory

Google colab atau *Google colab* merupakan sebuah *tools* yang dikeluarkan oleh *Google*. *Tools* ini memberikan fasilitas kepada para peneliti atau orang yang ingin mempelajari dan mengolah data menggunakan *machine learning* maupun *deep learning*, namun memiliki keterbatasan perangkat untuk melakukan komputasi. *Google Colab* menyediakan layanan GPU gratis sebagai *backend* komputasi yang dapat digunakan selama 12 jam.

Google Colab dibuat diatas *environment Jupyter* sehingga mirip dengan *Jupyter Notebook*. Penggunaannya pun hampir sama dengan *Jupyter Notebook* hanya saja berbeda dalam hal media penyimpanannya. Media penyimpanan pada *Google Colab* adalah *Google Drive* dan *tools* ini berjalan pada sistem *Cloud*. Dengan menggunakan *Google Colab*, kita dapat membuat suatu aplikasi berbasis *deep learning* dengan berbagai *library* seperti TensorFlow, Keras, Pytorch maupun OpenCV (Digimi, 2018).

2.15. Flask

Flask adalah *microframework* untuk Python yang dibuat dengan *toolkit* WSGI dan Jinja2. Flask dibuat dan dimaintain oleh Armin Ronacher. Pertama kali dirilis 4 tahun yang lalu. Flask saat ini berada di versi 0.10.1 dan dilisensikan dengan BSD license.

Aplikasi web yang dibuat dengan Flask disimpan dalam satu berkas .py. Flask ingin menjadi web *framework* yang sederhana namun dapat diperluas dengan beragam pustaka tambahan yang sesuai dengan kebutuhan penggunanya. Flask memang belum mencapai versi 1.0 namun dokumentasi yang dimilikinya sangat-sangat lengkap

2.16. Partitur

Penulisan notasi musik menghasilkan tulisan musik atau disebut partitur. Partitur merupakan istilah umum yang mendeskripsikan tangan kertas musik atau notasi musik. Dalam notasi musik, nada dilambangkan dengan not. Not terdiri dari not balok dan not angka. Notasi musik Standar saat ini yaitu not balok yang didasarkan pada paranada dengan lambang setiap nada menunjukkan ketinggian nada dan durasi nada tersebut. Di Indonesia masih banyak menggunakan not angka karena penulisan yang berupa angka lebih mudah dipelajari (Hildayanti et al., 2019).

2.17. Notasi Balok

Dalam notasi musik, nada dilambangkan oleh not. Setiap nada memiliki frekuensi yang berbeda. Masing-masing nada digambarkan pada posisi yang berbeda pula. Notasi balok adalah standar yang digunakan dalam penulisan notasi musik (Tedjokusuma & Alamveta, 2009). Menurut Purwanto didalam Notasi balok dalam bahasa Indonesia berasal dari pengaruh bahasa Belanda: *noten balk*, yaitu notasi musik yang menggunakan lima garis horizontal untuk menempatkan titi nada. Titi - titi nada digambarkan dengan lambang yang disebut not. Dalam notasi balok, nada dilambangkan oleh not yang berbentuk telur yang terdapat tangkai dan bendera not yang mempunyai nilai tertentu dan diletakkan pada garis dan kolom yang disebut “ Great Staff ” (Wijayama,

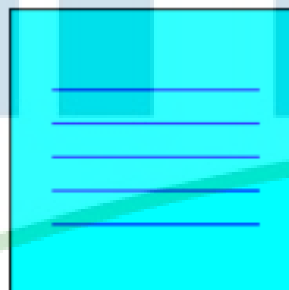
2006:7). Menurut Banoe (2003:299), notasi balok adalah tulisan musik dengan lima garis datar, guna menunjukkan tinggi rendah suatu nada. Sedangkan menurut Martinus (2001:404), notasi balok adalah tanda tulis yang memiliki titik nada dan tanda nada. Martinus juga mengartikan notasi sebagai proses membuat tanda nada.

Notasi balok merupakan sistem penulisan karya musik yang telah menjadi standar penulisan notasi musik di seluruh dunia sampai saat ini. Notasi balok didasarkan pada lima garis horizontal tempat not ditulis yang dikenal dengan nama paranada. Untuk menguasai notasi balok butuh proses pembelajaran yang sangat banyak dan panjang. Di Indonesia, notasi angka lebih cenderung dipahami dan dipakai dibandingkan dengan not balok. Hal ini dikarenakan notasi angka lebih mudah dipahami (Haumahu, 2019).

Berdasarkan penelitian (Tedjokusuma & Alamveta, 2009) ada beberapa hal yang menjadi bagian dalam sebuah notasi music, diantaranya:

2.17.1. Paranada (Musical Staff)

Garis Paranada menggambarkan urutan waktu dari kejadian musik. Paranada dibentuk oleh 5 buah garis yang dipisahkan oleh 4 buah spasi, seperti terlihat pada Gambar 2.10.



Gambar 2.10. Paranada (Musical Staff)

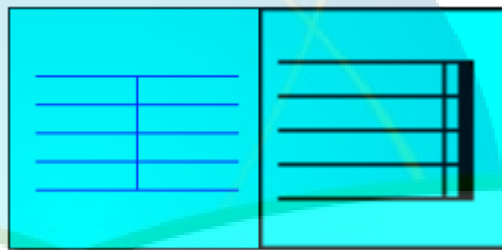
(Tedjokusuma & Alamveta, 2009)

Not yang terletak di garis atau spasi lebih tinggi berarti memiliki tinggi nada yang lebih tinggi. Not pada

paranada dibaca dari kiri ke kanan. Apabila penulisan membutuhkan posisi not yang melebihi paranada, dapat menggunakan garis tambahan.

2.17.2. Garis Bar (Barlines)

Garis Bar diletakkan pada paranada untuk memisahkan kumpulan ketukan sesuai dengan tanda birama. Jarak antara garis bar disebut dengan measure atau bar. Terdapat pula tanda garis bar ganda, yang menandakan bahwa akhir dari suatu notasi sudah tercapai. Garis bar dan bar ganda dapat dilihat pada gambar berikut



Gambar 2.11. Garis Bar/Bar Lines (kiri), garis bar ganda/Double Bar Line (kanan)

(Tedjokusuma & Alamveta, 2009)

2.17.3. Tanda Kunci (Clef Symbol);

Tanda kunci menunjukkan rentang not dari karya musik yang dimainkan. Tanda kunci yang umum digunakan adalah tanda kunci treble dan tanda kunci bass, seperti pada Gambar 2.12.

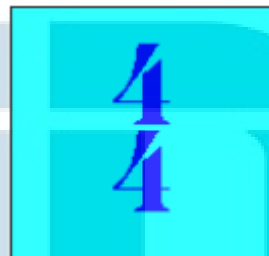


Gambar 2.12. Tanda kunci G/Treble (kiri), tanda kunci F/Bass(kanan)

(Tedjokusuma & Alamveta, 2009)

2.17.4. Tanda Birama (Time Signature)

Angka di atas pada tanda birama menunjukkan banyaknya ketukan pada sebuah bar, sedangkan angka di bawah pada tanda birama menunjukkan nilai not yang dianggap sebagai satu ketuk. Tanda birama diletakkan setelah tanda kunci dengan tulisan seperti pada Gambar 2.13.



Gambar 2.13. Tanda Birama (Time Signature)

(Tedjokusuma & Alamveta, 2009)

Terdapat bermacam-macam tanda birama, seperti simple duple (2/4), simple triple (3/4), simple quadruple (4/4), dan lain sebagainya.

2.17.5. Not (*Note*)

Not adalah representasi dari nada di dalam notasi musik. Not memiliki informasi berupa tinggi nada dan

durasinya. Gambar 2.14. menunjukkan posisi dari tiap-tiap not dalam tanda kunci treble dan bass.









Gambar 2.14. Posisi Not dalam Tanda Kunci Treble dan Bass

(Tedjokusuma & Alamveta, 2009)

2.17.6. Durasi Not

Untuk durasi dari suatu not digambarkan dengan simbol not yang berbeda, seperti terdapat pada Gambar 2.15.

Not	Nilai	Not	Nilai
	Not Penuh (1)		Not Setengah (1/2)
	Not Seperempat (1/4)		Not Seperdelapan (1/8)
	Not Seperenam belas (1/16)		Not Sepertigadua (1/32)

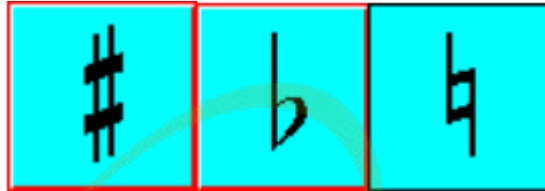
Gambar 2.15. Jenis-jenis Nilai Not

(Tedjokusuma & Alamveta, 2009)

2.17.7. *Accidental*

Accidental adalah simbol yang digunakan untuk menaikkan atau menurunkan tinggi nada dari suatu not. Simbol ini diletakkan di sebelah kiri suatu not. Terdapat 3 simbol *accidental*, yaitu *sharp* (menaikkan nada sebesar 1 *semitone*), *flat* (menurunkan nada sebesar 1 *semitone*), dan *natural* (membatalkan semua *accidental* pada nada tersebut).

Sebagai contoh, untuk menghasilkan not yang menunjukkan nada C#, dapat digunakan simbol sharp sebelum nada C. Simbol *accidental* yang lengkap terlihat pada Gambar 2.16.

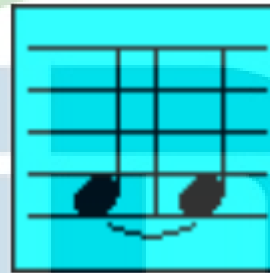


Gambar 2.16. Tanda Kres/Sharp (kiri), Tanda Mol/ Flat (tengah), Tanda Pugar/Natural(kanan)

(Tedjokusuma & Alamveta, 2009)

2.17.8. *Tie*

Tie menghubungkan nilai suatu not dengan not yang lain, lihat Gambar 2.17. Biasanya digunakan untuk not yang nilainya melebihi nilai dari suatu bar.



Gambar 2.17. Tanda Tie

(Tedjokusuma & Alamveta, 2009)

2.18. Flask

Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa

Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur behaviour suatu web dengan lebih mudah (Irsyad, 2018).

Flask termasuk pada jenis microframework karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi form, database, dan sebagainya tidak terpasang secara default di Flask. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga dan Flask dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh Flask sendiri (Irsyad, 2018).

Selain itu, meskipun Flask disebut sebagai microframework, bukan berarti Flask mempunyai kekurangan dalam hal fungsionalitas. Microframework disini berarti bahwa Flask bermaksud untuk membuat core dari aplikasi ini sesederhana mungkin tapi tetap dapat dengan mudah ditambahkan. Dengan begitu, fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan framework lainnya (Irsyad, 2018). Dikarenakan system ini menggunakan Bahasa python sebagai pembuatan modelnya, maka pembuatan interface menggunakan Flask. Menurut (Ramdani, Firmansyah, Info, Coefficient, & Tunggal, 2018) Flask menggunakan python sebagai bahasa pemrograman yang menyediakan konten web yang interaktif.

Flask merupakan *framework* sederhana dengan standar paling kecil atau biasa disebut dengan "*micro-framework*", dan cukup sederhana sehingga seluruh *source codenya* mudah dibaca, dan dipahami oleh *coder*. Namun menjadi *source code* yang sederhana bukan berarti kinerjanya tidak sebagus *framework* lainnya, flask dirancang sebagai *framework* yang dapat dikembangkan seluruhnya (Grinberg, 2018).

Flask memiliki tiga dependensi utama. *Routing, debugging, and web server interface* (WSGI) system tambahan dari Werkzeug; dukungan template disediakan oleh Jinja2; dan integrasi baris perintah berasal dari Click. sSmua dependensi ini ditulis oleh Armin Rocacher, penulis Flask. Flask tidak memiliki dukungan asli

untuk mengakses database, memvalidasi formulir web, mengotentikasi user, atau tugas tingkat tinggi lainnya (Grinberg, 2018).



BAB III

METODE PENELITIAN

Pada proses perancangan sistem, dibuatlah kerangka kerja dalam bentuk skema. Metodologi penelitian dijadikan panduan untuk menentukan tahap-tahap yang harus dilakukan. Tahap-tahap tersebut meliputi:

3.1. Metode Pengumpulan Data

3.1.1. Studi Pustaka

Salah satu metode pengumpulan data yang penulis lakukan pada pembuatan sistem ini adalah studi pustaka. Studi pustaka dilakukan dengan cara membaca, mengutip, mengolah informasi yang berkaitan dengan pembuatan sistem, serta membuat catatan yang bersumber pada bahan-bahan pustaka yang mendukung dan berkaitan dengan pembuatan sistem ini. Sumber-sumber yang penulis kumpulkan dan pelajari berupa data dan informasi definisi yang terdapat pada beberapa jurnal dan buku yang terkait dengan penelitian dan pembuatan sistem ini serta dari beberapa situs online yang terpercaya. Sumber-sumber tersebut diperlukan penulis untuk membantu penyelesaian pada bab pendahuluan, landasan teori dan metode penelitian.

3.1.2. Observasi

Penulis melakukan observasi untuk mengumpulkan data not balok berupa gambar di internet. Tahap awal adalah pencarian gambar utuh sebuah lagu dalam bentuk notasi balok. Kemudian gambar tersebut penulis potong-potong menjadi satu notasi dan dikelompokkan sesuai dengan kelas yang sudah ditentukan. Data-data tersebut penulis kumpulkan dari pencarian internet dan juga ada yang penulis buat sendiri melalui aplikasi photoshop maupun paint yang nantinya akan menjadi dataset untuk training dan testing pada sistem.

3.2. Metode Simulasi

Dalam penelitian ini, penulis menggunakan metode simulasi. Berikut ini merupakan langkah-langkah metode simulasi :

3.2.1. Formulasi Masalah (*Problem Formulation*)

Pada tahap ini, penulis melakukan identifikasi masalah berdasarkan penelitian-penelitian yang berkaitan dengan metode *Convolutional Neural Network* dan *Image Classification* pada *deep learning*. Pada penelitian yang sudah ada belum pernah ada yang membahas permasalahan sistem pengenalan notasi balok pada musik menggunakan *deep learning*.

3.2.2. Model Pengkonsepkan (*Conceptual Model*)

Pada tahap ini, penulis merancang konsep model pada penelitian ini. Dalam penelitian ini akan melakukan pemodelan untuk pengenalan pola notasi balok pada partitur musik dengan arsitektur model *Convolutional Neural Network* pada proses klasifikasinya. Pada proses pembuatan sistem ini nantinya terdapat proses ekstraksi fitur pada dataset gambar, mendapatkan nilai akurasi tertinggi, membuat model sistem untuk mengenali notasi, lalu menghubungkan model dengan *interface* yang telah dibuat. Setelah itu mendapat keluaran hasil sistem berupa musik yang telah diinput dan dikenali oleh sistem.

3.2.3. Pengumpulan Data Masukan/Keluaran (*Collection Input/Output Data*)

Pada tahap ini, penulis melakukan pengumpulan data masukan yang dibutuhkan untuk membuat model sistem *generate music* ini. Data yang dikumpulkan berupa gambar notasi balok berdasarkan kelas yang penulis batasi pada penelitian ini. Data tersebut penulis kumpulkan dari sumber internet maupun penulis buat sendiri melalui photoshop dan paint. Data awal berupa sebuah partiture lagu not balok yang kemudian penulis potong-potong menjadi satu notasi balok yang diperlukan sesuai kelas yang telah ditentukan.

Setelah seluruh data gambar berhasil dikumpulkan dan dilakukan pemilihan dari gambar yang berkualitas terbaik, data gambar tersebut nantinya akan dibagi menjadi data latih, dan data uji pada proses pembuatan model untuk mencapai nilai akurasi yang diinginkan, setelah data dilatih dan diuji, dan telah mendapatkan nilai akurasi yang sesuai kemudian dibuatkan model dengan metode *Convolutional Recurrent Neural Network*.

3.2.4. *Preprocessing*

Data yang sudah ada kemudian masuk ke tahap preprocessing. Data – data yang ada akan diresize, untuk menyamakan ukuran dari gambar. Data yang telah diresize kemudian melalui proses segmentasi citra dimana data diproses untuk menentukan deteksi tepi, yang bertujuan untuk mengenali objek yang ada dalam data.

3.2.5. *Pemodelan (Modelling)*

Pada tahap ini penulis melakukan pembuatan model dari dataset yang ada untuk mengenali gambar dan proses klasifikasi menggunakan algoritma atau metode *Convolutional Neural Network* agar sistem dapat mengenali citra notasi balok.

3.2.6. *Pelatihan dan Pengujian*

Proses pelatihan dan pengujian menggunakan metode *Convolutional Neural Network*. Pada tahap ini, data yang dikumpulkan dalam bentuk citra dibagi menjadi data latih dan data uji. Untuk mendapatkan model yang ingin dibuat dibutuhkan suatu analisis dari data latih. Setelah data dilatih, dilakukan tahap selanjutnya dengan menghitung nilai loss dan akurasi dari model-model yang dihasilkan dari data uji. Data uji berguna untuk mengetahui nilai akurasi yang dihasilkan dari model yang telah dilatih. Penentuan nilai dari *epoch* dimana suatu perulangan dari proses pengujian pada satu kali sesi lewatkan guna mendapatkan error terkecil dan menghasilkan kinerja yang baik untuk model yang dibuat.

3.2.7. Menghubungkan Model ke API

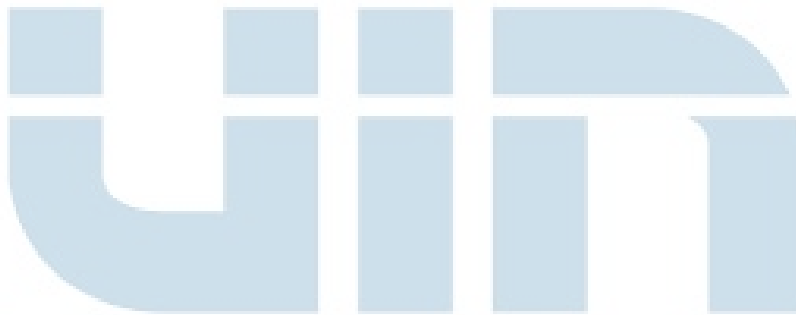
Pada tahap ini, penulis menghubungkan model sistem yang dibuat tadi dengan *interface* berupa API agar dapat melakukan simulasi dengan tampilan *interface* yang ada. Penulis menggunakan Flask sebagai framework websitenya

3.2.8. Eksperimentasi (*Eksperimentation*)

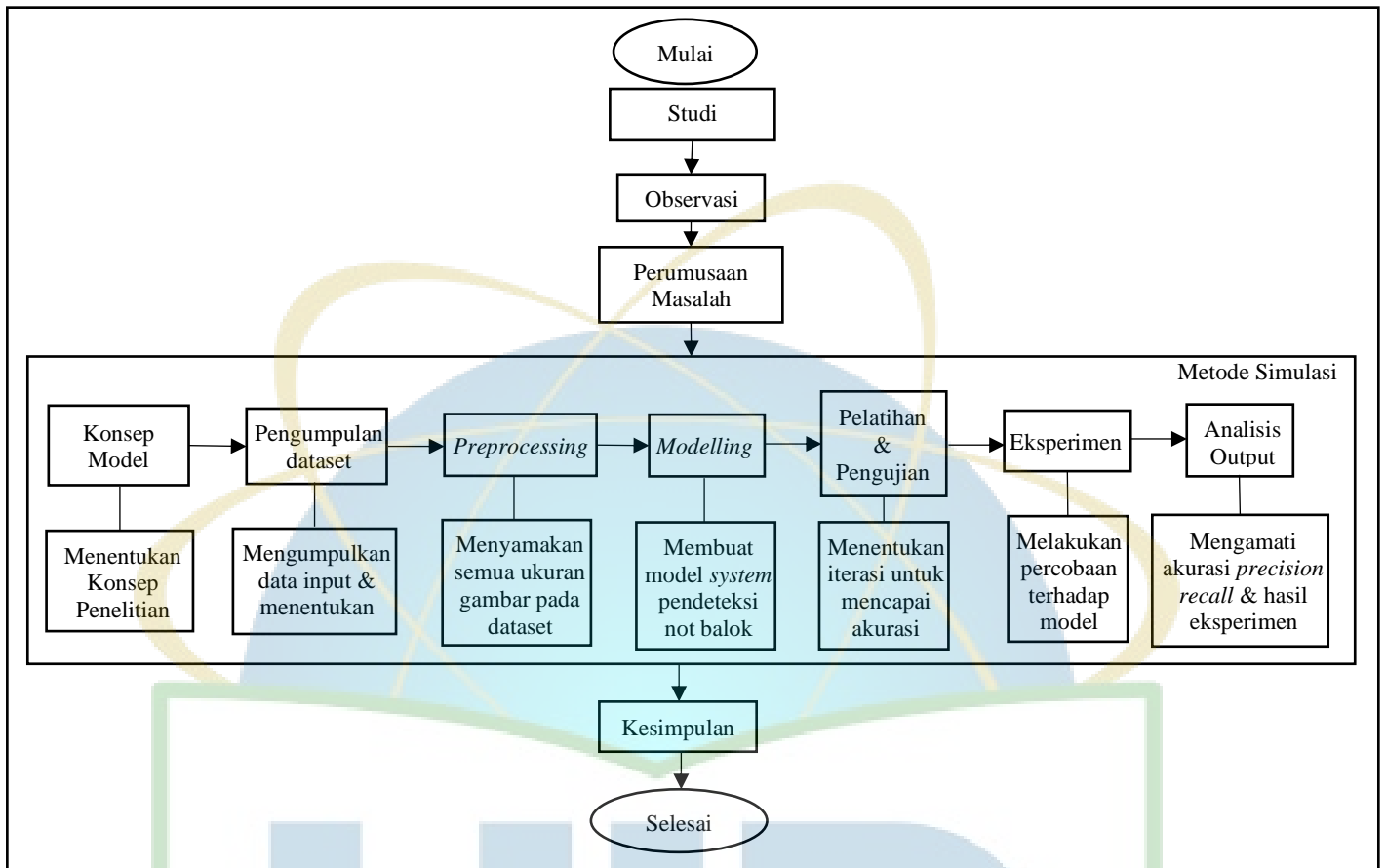
Setelah model dan API telah terhubung dengan baik, selanjutnya penulis melakukan tahap eksperimentasi yaitu mencoba keberhasilan model yang dibuat. Penulis menginput dokumen lagu notasi balok berformat jpg yang telah penulis siapkan, lalu sistem akan memproses inputan tersebut. Setelah diproses melalui model yang dibuat, keluarlah hasil dari inputan tersebut berupa kelas yang telah ditentukan.

3.2.9. Analisis Output (*Output Analysis*)

Penulis melakukan analisa terhadap output atau keluaran berdasarkan skenario yang dilakukan apakah sesuai dengan inputan. Selain itu juga pada tahap Zini akan ditampilkan hasil keluaran dengan inputan dokumen.



3.3. Kerangka Berfikir Penelitian



Gambar 3.1. Kerangka Berfikir Penelitian

BAB IV

IMPLEMENTASI

Pada bab ini akan dilakukan implementasi dan pembahasan dalam perancangan sistem secara bertahap sesuai dengan metode penelitian yang digunakan selama perancangan pembuatan sistem, yaitu metode simulasi,

4.1. Pengumpulan Data Notasi Balok

Tahap pertama yang dilakukan penulis dalam perancangan sistem adalah mengumpulkan data notasi balok, yang nantinya akan digunakan sebagai data latih dan data uji. Pengumpulan data diambil dari sumber internet dan beberapa data yang dimodifikasi dan *editing*. Berdasarkan batasan yang ada, data yang dikumpulkan hanya notasi-notasi balok yang ada pada kunci *Treble*, dari not C hingga C', yang mana nantinya akan menjadi 40 kelas. Dalam satu not akan menjadi 5 kelas, dan dalam satu kelas berisi data kurang lebih 50 gambar berformat .jpg.

Berikut adalah tahap-tahap bagaimana data dibuat dan dimasukkan ke berbagai kelas yang sesuai

4.1.1. Data dari Sebuah Lagu Berbentuk Notasi Balok

Pada pengumpulan data ini, penulis mencari gambar-gambar partitur lagu berbentuk notasi balok dari internet, kemudian dilakukan *editing* yaitu pemotongan gambar agar menjadi gambar sebuah notasi balok.

Contoh Dokumen Lagu / Partitur dalam bentuk notasi balok

Setelah mengumpulkan partitur lagu diinternet, kemudian gambar tersebut akan dicrop menjadi satuan notasi seperti berikut

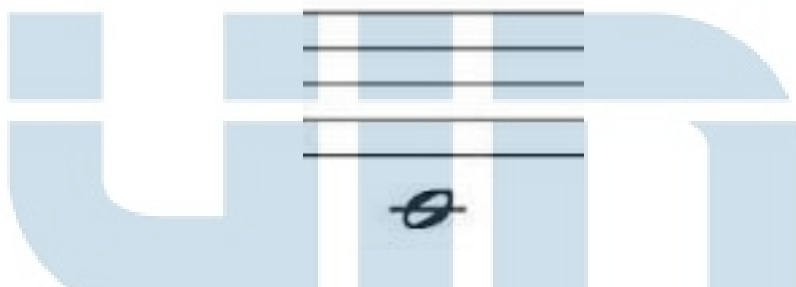


Gambar 4.1. Contoh data gambar yang telah dicrop

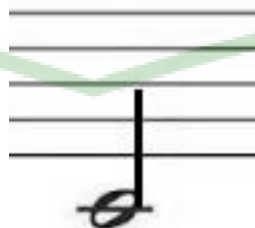
4.1.2. Modifikasi data yang telah dikumpulkan

Pada pengumpulan data yang dimodifikasi, data yang ada akan melalui proses editing menggunakan Photosop atau Paint. Cara ini bertujuan agar data yang dikumpulkan mencapai target yang diinginkan, karena data yang dicari melalui internet saja sangat tidak cukup dan tidak lengkap.

Berikut cara pemodifikasian data agar mendapatkan data baru



Gambar 4.2. Contoh Gambar yang belum di modifikasi



Gambar 4.3. Contoh Gambar Setelah di Modifikasi

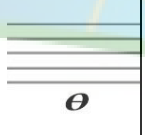

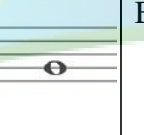
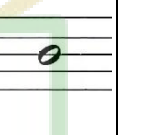

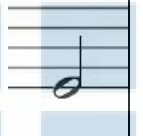

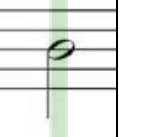

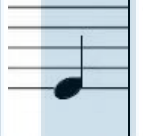
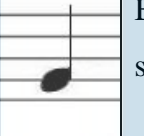



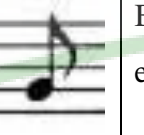
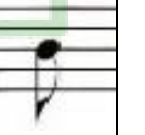


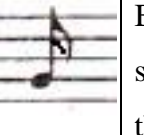

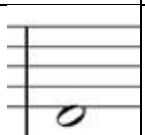

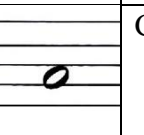
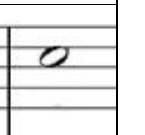
Pada contoh gambar yang diberikan tadi, penulis menambahkan garis pada gambar 4.3. agar gambar notasi tersebut dapat menjadi gambar










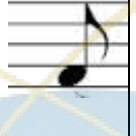



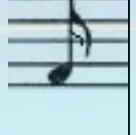
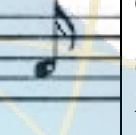

yang berbeda kelas seperti pada gambar 4.4. Hal tersebut menjadikan penambahan data pada kelas lain.

Selain penambahan garis pada notasi, modifikasi lain juga dapat berupa penghitaman pada notasi, pemindahan garis bar pada gambar, pemberian garis seperti bendera agar menjadi kelas seper4 atau seper 8, dan modifikasi lainnya.

Kelas yang telah ditentukan adalah 40 kelas yang mana kelas tersebut terdiri dari not C hingga C' dengan nilai penuh (*full*), setengah (*half*), seperempat (*quarter*), seperdelapan (*eighth*), dan seperenam belas (*sixteenth*). Berikut adalah contoh gambar perkelas yang dikumpulkan:

Tabel 4.1. Daftar Gambar Berdasarkan Kelas

Kelas	Gambar	Kelas	Gambar	Kelas	Gambar	Kelas	Gambar
C full		E full		G full		B full	
C half		E half		G half		B half	
C quarter		E seper4		G seper4		B seper4	
C eighth		E eighth		G eighth		B eighth	
C sixteen th		E sixteen th		G sixteen th		B sixteen th	
D full		F full		A full		C' full	

D half		F half		A half		C' half	
D quarter		F quarter		A quarter		C' quarter	
D eighth		F eighth		A eighth		C' eighth	
D sixteen th		F sixteen th		A sixteen th		C' sixteen th	

Setelah mendapatkan data sebanyak-banyaknya, kemudian dilakukan pemilihan gambar yang berkualitas terbaik dan sesuai format yang diinginkan. Setelah dilakukan pemilihan gambar, data yang berhasil dikumpulkan adalah sebanyak 2405 gambar dari 40 kelas dimana masing-masing kelas berisi kurang lebih 50 gambar berformat .jpg.

4.2. Pembuatan Model menggunakan *Google Colab*

Hal pertama yang harus dilakukan untuk menjalankan proses pemodelan pada *Google Colab* adalah menghubungkan *Google Colab* ke *Google Drive* untuk mengakses dataset yang telah diupload sebelumnya kedalam *GDrive*.

```
from google.colab import drive
drive.mount('/content/drive')
```

Source code diatas berguna untuk mengkoneksi dataset pada *GDrive* ke *Google Colab*. Setelah perintah tersebut dijalankan, kemudian diberikan url yang berisi *authorization code*. Setelah dataset terkoneksi, kemudian masukkan fungsi library yang akan digunakan agar perintah yang akan dijalankan dan pembuatan model berjalan lancar dan dapat dieksekusi. Berikut *Source Code* yang digunakan


```

import cv2
import math
import matplotlib.pyplot as plt
import pandas as pd
from keras.preprocessing import image
import numpy as np
from keras.utils import np_utils
from skimage.transform import resize
import glob
import os
import keras
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers import Dense, InputLayer, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
from sklearn.metrics import confusion_matrix
from keras.optimizers import SGD, Adam
from keras.callbacks import EarlyStopping

```

4.3. Preprocessing dan Labeling

Setelah data notasi berhasil dikumpulkan, kemudian data tersebut dilanjutkan ke tahap *preprocessing*. Semua data tersebut akan diubah ukurannya atau diresize untuk menyamakan ukuran dari gambar. Seluruh ukuran gambar diubah menjadi 32 x 32 pixels. Setelah diubah ukurannya, kemudian data diberikan label sesuai dengan folder yang telah dibuat. Selanjutnya label tersebut diubah menjadi kategori 0 hingga 39 (berdasarkan jumlah kelas yang ditentukan) dengan menggunakan fungsi `dummy_y = np_utils.to_categorical(y)`. Pada proses inilah

data sudah masuk sesuai dengan kategori atau klasifikasi yang telah ditentukan. Berikut adalah *code* yang dijalankan.

```
pics, labels = [], []
i=0
hitung = 1
for dir_path in glob.glob('/content/drive/My Drive/Data
musik/*'):
    for pic in glob.glob(os.path.join(dir_path,"*.jpg")):
        temp = cv2.imread(pic)
        if hitung % 100 ==0:
            print(hitung)
        hitung = hitung + 1
        temp = cv2.resize (temp, (32,32))
        pics.append(temp)
        labels.append(i)

    i=i+1
X = np.array(pics)
y = np.array(labels)
X.shape

dummy_y = np_utils.to_categorical(y)
dummy_y
```

Pada *source code* tersebut mengambil dataset dari folder Data Musik yang berformat .jpg atau gambar pada My Drive. Lalu gambar tersebut dibaca menggunakan `cv2.imread`. Selanjutnya seluruh gambar *diresize* menjadi ukuran 32 x 32 pixel. Kemudian diinisialisasi X menjadi gambar dan y menjadi labelnya. Lalu diberi inisial `dummy_y` untuk gambar yang telah diberi proses pelabelan

4.4. Pemodelan

Pada tahap ini, semua data yang telah di samakan seluruh ukurannya dan telah diberi label sesuai kelasnya akan melalui proses pemodelan menggunakan

algoritma atau metode *Convolutional Neural Network*. Berikut ini adalah *source code* yang digunakan.

```
base_model = Sequential()
base_model.add(Conv2D(32, kernel_size=(3, 3),
                      activation='relu', input_shape=(32, 32, 3)))
base_model.add(MaxPooling2D(pool_size=(2, 2)))
base_model.add(Conv2D(32, (3, 3), activation='relu'))
base_model.add(MaxPooling2D(pool_size=(2, 2)))
base_model.add(Dropout(0.5))
base_model.add(Flatten())
base_model.add(Dense(1024, activation='relu'))
base_model.add(Dense(512, activation='relu'))
base_model.add(Dropout(0.5))
base_model.add(Dense(40, activation='softmax'))
```

4.5. Pelatihan dan Pengujian

Proses selanjutnya adalah pelatihan dan pengujian data. Pada proses pelatihan, data yang telah dikumpulkan nantinya akan dibagi menjadi data latih dan data uji. Untuk mendapatkan model yang ingin dibuat dibutuhkan suatu analisis dari data latih. Dalam proses pelatihan data digunakan test size sebesar 0.25 dan random statenya 42. Setelah data dilatih, dilakukan tahap selanjutnya dengan menghitung nilai loss dan akurasi dari model-model yang dihasilkan dari data uji.

Penentuan nilai dari epoch untuk pengujian data adalah 100 yang dimana suatu

```
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(X,
dummy_y, test_size=0.25, random_state=42)

base_model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])

history = base_model.fit(X_train, y_train, epochs=100,
verbose=1, validation_data=(X_valid, y_valid))
```

perulangan dari proses pelatihan pada satu kali sesi lewatan sebanyak 100 kali guna mendapatkan error terkecil dan menghasilkan kinerja yang baik untuk model yang dibuat. Sedangkan untuk kategori loss yang digunakan adalah 'categorical_crossentropy', dan untuk optimizer menggunakan 'adam'. Berikut ini adalah *source code* yang digunakan.

4.6. Membuat *base model* untuk dihubungkan ke *Interface*

Pada tahap selanjutnya hasil yang ada akan dijadikan dalam bentuk base model .h5 untuk mengenali gambar agar sistem dapat mengenali citra digital notasi balok sesuai dengan kelas yang telah ditentukan. Pada pemodelan untuk sistem yang dibuat menggunakan module keras. Yang kemudian model nantinya dapat dihubungkan dengan *back-end* dan *front-end*.

1.1.1. Menghubungkan Model ke API

Pada tahap ini, penulis menghubungkan model sistem yang dibuat tadi dengan *interface* berupa API agar dapat melakukan simulasi dengan tampilan *interface* yang ada.

1.1.2. Eksperimentasi (*Eksperimentation*)

Setelah model dan API telah terhubung dengan baik, selanjutnya penulis melakukan taham eksperimentasi yaitu mencoba keberhasilan model yang dibuat. Penulis menginput citra digital notasi balok berformat jpg yang telah penulis siapkan, lalu sistem akan memproses inputan tersebut. Setelah diproses melalui model yang dibuat, keluarlah hasil dari inputan tersebut berupa kelas yang telah ditentukan.

BAB V

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari penelitian ini, penulis melakukan klasifikasi 40 kelas not balok yakni not C full-sixteenth hingga C' full-sixteenth atau yang biasa kita kenali nada do hingga do tinggi. Klasifikasi tersebut menggunakan algoritma Convolutional Neural Network (CNN). Proses utama dalam pembuatan model ini diawali dengan proses training data. Proses ini bertujuan untuk pembentukan model yang akan digunakan untuk pengujian data testing. Parameter untuk mengukur tingkat keberhasilan model adalah nilai akurasi. Nilai akurasi model dapat ditentukan dengan melakukan pengujian menggunakan data testing. Proses training menggunakan packages Keras pada python dengan back-end Flask. Keras merupakan salah satu modul yang dibuat oleh Google untuk mempermudah dalam research mengenai neural network dan mampu berjalan diatas Flask, tensorflow, theano, MXNet.

5.1. Pelabelan

Setelah dataset telah terkumpul dan dirapikan dalam folder sesuai kelas dan telah melalui tahap preprocessing, selanjutnya dataset diberi label sesuai kelas/folder/kategori. Pelabelan dimulai dari label 0 hingga label 39 dengan dtype=float32. Sehingga menghasilkan label dan kelas sebagai berikut :

Tabel 5.1. Pembagian Kelas berdasarkan Label

Kelas	Label	Kelas	Label
0	C' sixteenth	20	A eighth
1	C' half	21	A half
2	C' quarter	22	A sixteenth
3	D quarter	23	A quarter
4	D sixteenth	24	A full
5	D half	25	E full
6	C' full	26	E sixteenth
7	D eighth	27	E quarter
8	D full	28	E eighth
9	C' eighth	29	F half
10	B half	30	E half
11	B quarter	31	F eighth
12	B eighth	32	F quarter
13	C sixteenth	33	F full
14	C full	34	G full
15	B full	35	F sixteenth
16	C half	36	G half
17	B sixteenth	37	G eighth
18	C quarter	38	G quarter
19	C eighth	39	G sixteenth

5.2. Membagi dataset untuk dilatih dan diuji dari dataset

Tahap selanjutnya adalah dataset yang telah ada akan dibagi menjadi data latih dan data uji menggunakan model sklearn yang sudah ada pada *library* python. Pada penelitian ini digunakan *library* sklearn karena *library* ini memudahkan pemrograman karena lebih sedikit baris kode dibanding *library* lainnya seperti tensorflow atau numpy. *Library* sklearn juga biasa digunakan untuk clustering atau

mengelompokkan data dengan karakteristik yang sama ke suatu 'wilayah' yang sama dan data dengan karakteristik yang berbeda ke 'wilayah' yang lain.

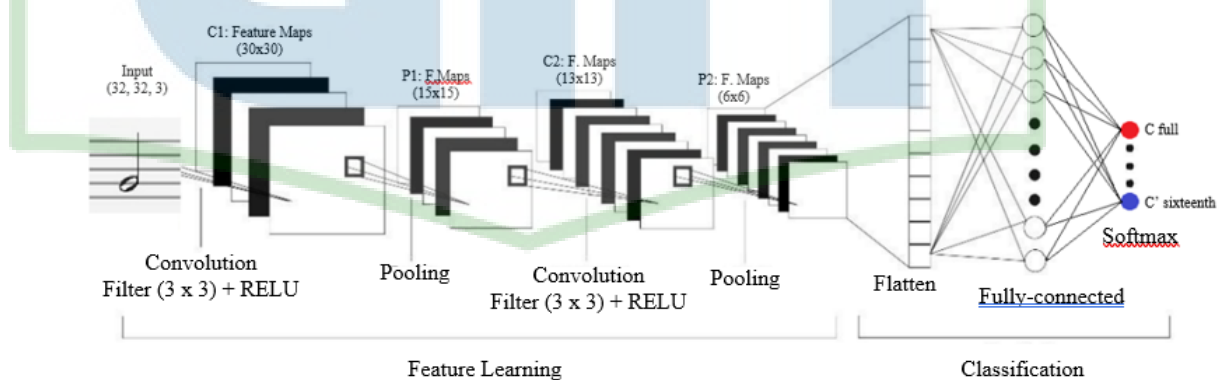
Dimasukkan fungsi `train_test_split` pada program guna memisahkan array atau matriks menjadi data latih dan data test secara acak. Pada proses ini digunakan `test_size` sebesar 0.25 dan `random_state` sebanyak 42.

Tabel 5.2. Pembagian Shape

X_train.shape	(1803, 32, 32, 3)
X_valid.shape	(602, 32, 32, 3)
y_train.shape	(1803, 40)
y_valid.shape	(602, 40)

5.3. Arsitektur Jaringan

Setelah melalui tahap palabelan dan pengambilan data training dari dataset, selanjutnya dataset akan dilatih menggunakan algoritma *Convolutional Neural Network* (CNN). Dalam algoritma *Convolutional Neural Network* (CNN) pembentukan arsitektur jaringan dapat mempengaruhi hasil dari akurasi model.



Gambar 5. 1. Arsitektur Jaringan

Gambar 5.1. merupakan arsitektur jaringan pada proses training untuk menghasilkan model yang optimal. Penelitian ini menggunakan input gambar dengan ukuran $32 \times 32 \times 3$, tujuannya adalah untuk membandingkan nilai akurasi berdasarkan ukuran gambarnya. Arsitektur diatas dapat dijelaskan seperti penjelasan dibawah ini :

1. Proses Konvolusi pertama digunakan kernel berukuran 3×3 dan jumlah filter sebanyak 32 filter , proses konvolusi ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Setelah proses konvolusi, maka ditambahkan sebuah aktivasi fungsi yaitu RELU (*Retrified Linear Unit*). Fungsi aktivasi ini bertujuan untuk mengubah nilai negative menjadi nol (menghilangkan nilai negative dalam sebuah matriks hasil konvolusi). Hasil konvolusi ini memiliki ukuran baru yakni 30×30 .
2. Proses *pooling* pertama. Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Penelitian ini menggunakan *maxpooling* untuk mendapatkan nilai matriks yang baru hasil dari proses *pooling*. Berdasarkan hasil pooling menghasilkan matriks baru berukuran 15×15 dari input hasil konvolusi pertama yang berukuran 30×30 . Proses ini menggunakan kernel *pooling* 2×2 . Cara kerja *max-pooling* adalah mengambil nilai paling maksimum berdasarkan pergeseran kernelnya sebanyak nilai *stridenya* yaitu 2.
3. Proses Kovolusi kedua yaitu meneruskan hasil dari proses pooling pertama yakni dengan input matriks gambar sebesar 15×15 dengan jumlah filter yang sama dengan konvolusi pertama yakni sebanyak 32 filter dan dengan ukuran kernel 3×3 . Proses konvolusi kedua ini sama-sama menggunakan fungsi aktivasi RELU. Proses ini menghasilkan gambar berukuran 13×13 .
4. Proses selanjutnya masuk ke proses *pooling* yang kedua, proses ini hampir sama dengan proses *pooling* yang pertama, namun ada perbedaan pada nilai

output akhir dari matriksnya . Output yang dihasilkan memiliki ukuran gambar 6x6.

5. Selanjutnya *Flatten* dan *fully connected*. Pada tahap ini digunakan hanya satu hidden layer pada jaringan MLP (*Multi Layer Perceptron*). *Flatten* disini mengubah *output pooling* layer menjadi sebuah vector. Sebelum melakukan proses klasifikasi atau memprediksi gambar, pada proses ini digunakan nilai *Dropout*. *Dropout* adalah sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dibuang secara acak. Tujuan dari proses ini yaitu mengurangi *overfitting* pada saat proses *training*. Nilai *Dropout* yang digunakan sebesar 0,5 dengan mengeluarkan hasil yang sama 6x6.
6. Proses terakhir adalah menggunakan aktivasi fungsi *Softmax*. Fungsi ini secara spesifiknya biasa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*.

Berdasarkan uraian penjelasan dari arsitektur jaringan diatas, arsitektur tersebut digunakan untuk proses *training*. Sehingga dari proses *training* didapatkan model dari arsitektur tersebut. Berikut model yang terbentuk :

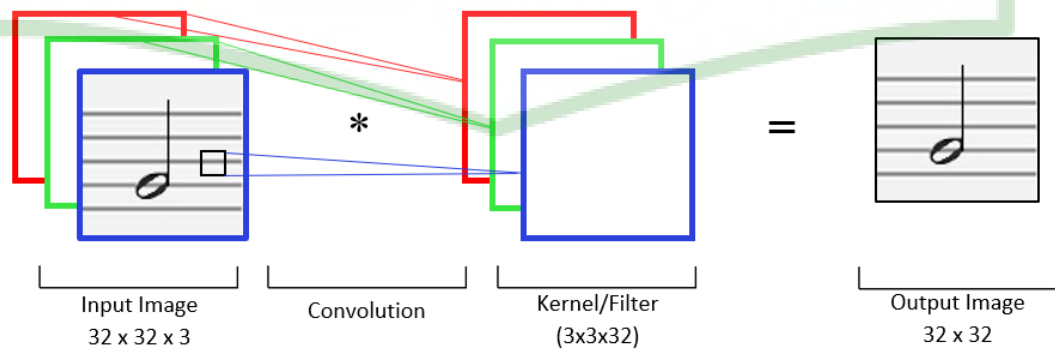
Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_7 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_8 (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 32)	0
dropout_7 (Dropout)	(None, 6, 6, 32)	0
flatten_4 (Flatten)	(None, 1152)	0
dense_10 (Dense)	(None, 1024)	1180672
dense_11 (Dense)	(None, 512)	524800
dropout_8 (Dropout)	(None, 512)	0
dense_12 (Dense)	(None, 40)	20520
Total params: 1,736,136		
Trainable params: 1,736,136		
Non-trainable params: 0		

Gambar 5.2. Model CNN

Gambar 5.2. merupakan model yang terbentuk dari hasil *training*. Total parameter yang terbentuk dari model sebanyak 1.736.136 neuron.

5.3.1. Proses Convolution Layer

Selanjutnya akan dibahas proses konvolusi yang terjadi berdasarkan arsitektur jaringan yang telah dibahas diatas.



Gambar 5. 3. Proses Konvolusi

Konvolusi merupakan proses menggabungkan dua deret angka pada suatu gambar sehingga menghasilkan deret angka yang baru, dengan kata lain gambar tersebut dimanipulasi dengan menggunakan *external mask / subwindows* untuk menghasilkan citra yang baru, angka konvolusi pada citra tersebut berbentuk matriks *array*. Berdasarkan gambar 5.3. digunakan input gambar dengan ukuran pixel 32x32x3, yang menjelaskan ukuran tinggi dan lebar pixel gambar sebesar 32 pixel dan gambar input tersebut memiliki 3 *channel* yaitu *red*, *green*, dan *blue* atau yang biasa disebut dengan RGB.

Setiap *channel* piksel memiliki nilai matriks yang berbeda-beda. Input akan di konvolusi dengan nilai filter yang sudah ditentukan. Filter merupakan blok lain atau kubus dengan tinggi dan lebar yang lebih kecil namun kedalaman yang sama yang tersapu di atas gambar dasar atau gambar asli. Filter digunakan untuk menentukan pola apa yang akan dideteksi yang selanjutnya dikonvolusi atau dikalikan dengan nilai pada matriks input, nilai pada masingmasing kolom dan baris pada matriks sangat bergantung pada jenis pola yang akan dideteksi. Jumlah filter pada konvo ini sebanyak 32 dengan ukuran kernel (3x3), ini artinya gambar yang dihasilkan dari hasil konvolusi akan sebanyak 32 fitur map.

Berdasarkan arsitektur jaringan diatas digunakan aktivasi ReLU (Rectified Linear Unit) setelah proses konvolusi. Fungsi dari aktivasi ini adalah melakukan “threshod” dari 0 hingga infinity. Nilai yang ada pada hasil konvolusi yang bernilai negative akan diubah dengan aktivasi ini menjadi nol dan yang lainnya sampai infinity.

5.3.2. Proses Pooling

Pooling atau penggabungan merupakan proses dari CNN dimana matriks input disederhanakan menjadi matriks yang baru. Pada penelitian ini digunakan metode max pooling pada proses poolingnya dimana diambil nilai terbesar dari matriks *input* dan menghasilkan matriks baru sebagai *outputnya*. Berdasarkan penelitian (Urbánek et al., 2010) berjudul “*Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*” menunjukkan bahwa max pooling lebih efektif dan menghasilkan hasil yang lebih baik dibanding dengan

metode lain seperti subsampling untuk data berupa gambar. Itu sebabnya penelitian ini menggunakan max pooling pada proses poolingnya.

5.3.3. Proses *Fully Connected*

Proses terakhir adalah fully-connected. Hasil dari proses terakhir dari model yang ada pada *feature extraction layer* masih berbentuk multidimensional array, sehingga harus dilakukan “*flatten*” atau *reshape feature map* menjadi sebuah vector agar bisa digunakan sebagai input dari *fully-connected layer*. Setelah itu diberi fungsi dense. Dense adalah fungsi untuk menambahkan layer pada *fully connected*.

```
base_model.add(Dense(1024, activation='relu'))
base_model.add(Dense(512, activation='relu'))
```

Gambar 5.4. Fully Connected Layer

Pada gambar tersebut, digunakan dense sebanyak 1024 dan 512 yang berarti ditambahkan layer sebanyak 1024 unit lalu disederhanakan lagi menjadi 512 unit. Unit tersebut artinya jumlah node yang harus ada di hidden layer, nilainya antara jumlah input node dan output node. Sedangkan fungsi aktivasinya digunakan fungsi relu. Fungsi dari fully connected layer ini adalah untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

```
base_model.add(Dense(40, activation='softmax'))
```

Gambar 5.5. Fully Connected Layer akhir

Kemudian di inialisasi layer outputnya, yang berupa 40 node, karena klasifikasi ini terdapat 40 kelas.

5.4. Akurasi

Setelah melewati tahap pemodelan, kemudian model tersebut diuji nilai akurasi dan lossnya dengan menggunakan adam optimizer

```
base_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = base_model.fit(X_train, y_train, epochs=100, verbose=1, validation_data=(X_valid, y_valid))
```

Gambar 5. 6. Uji Akurasi

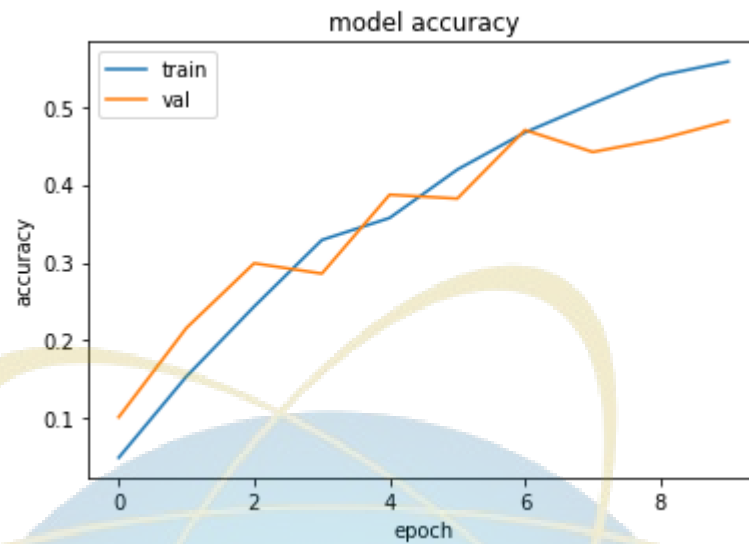
Dengan menggunakan iterasi epoch sebanyak 100 jaku selanjutnya didapatkan hasil akurasi validasi sebagai berikut

```
loss: 0.0981 - acc: 0.9734 - val_loss: 5.4337 - val_acc: 0.6628
```

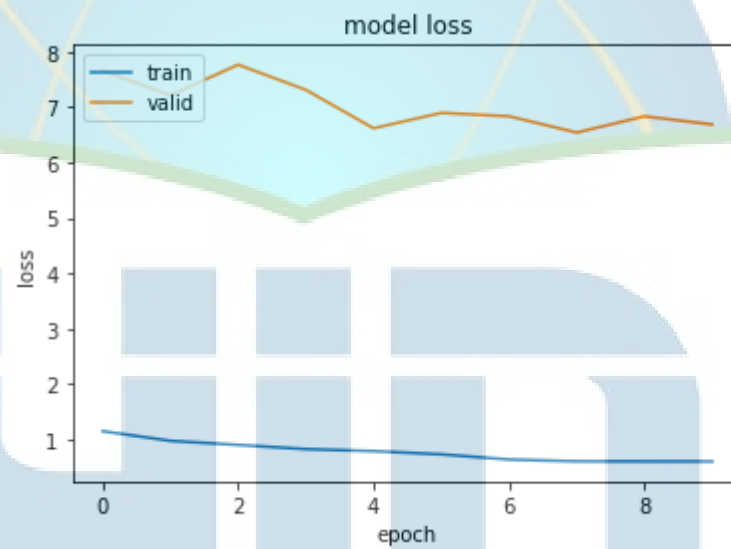
Gambar 5. 7. Hasil Akurasi

Berdasarkan gambar 5.7. yang menunjukkan hasil akurasi model, dapat diketahui besar loss 0,9% dengan akurasi yang tinggi sebesar 97,34%. Sedangkan untuk data validasi, nilai loss cukup tinggi yaitu sebesar 543% dan akurasi sebesar 66,28%. Yang berarti menunjukkan akurasi model sebesar 66,28%. Untuk klasifikasi dengan multiclass atau kelas yang banyak yaitu 40 kelas, nilai akurasi sebesar tersebut dapat dikatakan sudah bagus untuk dijalankan, dan sistem sudah cukup pintar untuk mendeteksi not balok sesuai dengan kelas yang ada.

Berikut ini akan dipaparkan gambar grafik akurasi dan loss dari model training dan validasi:

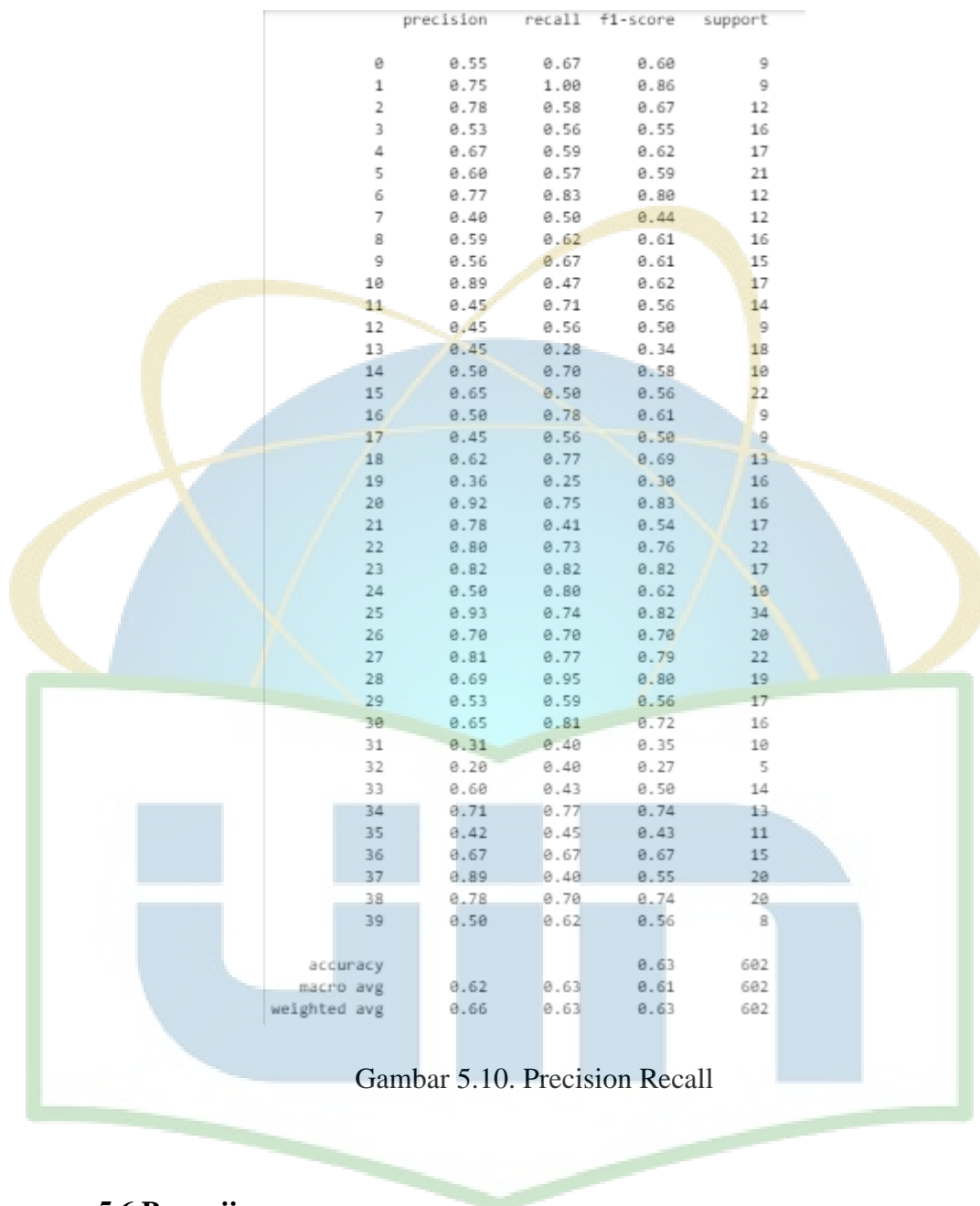


Gambar 5.8. Grafik Model Accuracy



Gambar 5.9. Grafik Model Loss

5.5. Precision Recall



Gambar 5.10. Precision Recall

5.6. Pengujian

Setelah pemodelan telah selesai dan tahap finalisasi, kemudian model tersebut dihubungkan pada back-end dan interface agar dapat dijalankan, digunakan, dan diuji dengan data baru.

Berikut ini adalah tampilan sederhana setelah model sudah dibuatkan interface.

Masukan Gambar Not Balok

Upload Gambar:

No file chosen

Gambar 5.11. Interface Sistem

Gambar diinput pada tombol “Choose File” lalu kemudian klik “Predict” agar dapat diproses pada sistem. Gambar yang dapat diproses hanyalah gambar dengan format .jpg.

Selanjutnya setelah sistem telah selesai dibuat dengan tampilan sederhana tersebut, dilakukan pengujian menggunakan data gabungan, yaitu data yang dipakai untuk dataset dan data baru yang tidak termasuk pada dataset. Jumlah data yang digunakan untuk pengujian sebanyak 10 dimasing-masing kelas dengan total data 400 gambar. Hasil akhir menunjukkan sistem rata-rata menebak gambar dengan parameter keberhasilan sebesar 69%.

Tabel 5.3. Hasil Uji Coba Sistem

Label	Jumlah Data	Berhasil	Gagal	%
C' sixteenth	10	8	2	80%
C' half	10	8	2	80%
C' quarter	10	7	3	70%
D quarter	10	8	2	80%
D sixteenth	10	7	3	70%
D half	10	8	2	80%
C' full	10	8	2	80%
D eighth	10	7	3	70%
C' eighth	10	6	4	60%
B half	10	7	3	70%
B quarter	10	7	3	70%
B eighth	10	8	2	80%

C sixteenth	10	7	3	70%
C full	10	8	2	80%
B full	10	8	2	80%
C half	10	7	3	70%
B sixteenth	10	6	4	60%
C quarter	10	6	4	60%
C eighth	10	8	3	80%
A eighth	10	7	3	60%
A half	10	7	3	70%
A sixteenth	10	8	2	80%
A quarter	10	8	2	80%
A full	10	7	3	70%
E full	10	9	1	90%
E sixteenth	10	7	3	70%
E quarter	10	6	4	60%
E eighth	10	7	3	70%
F half	10	7	3	70%
E half	10	8	2	80%
F eighth	10	7	3	70%
F quarter	10	6	4	60%
F full	10	7	3	70%
G full	10	5	5	50%
F sixteenth	10	6	4	60%
G half	10	6	4	60%
G eighth	10	7	3	70%
G quarter	10	6	4	60%
G sixteenth	10	7	3	70%
Rata-Rata % berhasil				69%

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan pembahasan yang sudah diuraikan, maka dapat ditarik kesimpulan sebagai berikut:

1. Model CNN pada penelitian ini menggunakan input shape berukuran 32x32x3, ukuran filter 3x3, Jumlah Epoch sebanyak 100. Data yang digunakan untuk proses training model sebanyak 2405. Menghasilkan tingkat akurasi training dan testing dalam melakukan deteksi not balok sebesar 97,34% untuk accuracy dan 66,28 % untuk validasi *accuracy*.
2. Semakin banyak kelas dengan jumlah dataset yang sama mempengaruhi besar akurasi. Saat diuji dengan setengah jumlah kelas yaitu 20, nilai akurasi hanya mendapat 50,12%
3. Penelitian ini menggunakan data testing baru sebanyak 400 gambar dimana per kelas terdapat 10 gambar untuk diujikan kedalam model yang telah dibuat. Hasil testing menghasilkan tingkat akurasi baru dalam melakukan pendeteksi not balok sebesar 69 %.

6.2. Saran

Adapun saran yang diberikan pada penelitian ini sebagai berikut :

1. Penelitian selanjutnya diharapkan dapat meningkatkan nilai akurasi dan menambahkan jumlah kelas klasifikasi dari seluruh notasi balok pada treble key.
2. Diharapkan penelitian ini dapat dikembangkan dalam keterkaitan dengan MIR (Music Information Retrieval) dimana nantinya output yang dihasilkan berupa suara musik tersebut.
3. Menambahkan parameter seperti perbandingan input gambar yang lebih besar ukuran pikselnya, nilai dropout, fungsi aktivasi,

penggunaan optimizer. Sehingga dengan menghasilkan model dengan penggunaan hyperparameter terbaik.

4. Penelitian ini dapat di kembangkan kedalam sebuah aplikasi yang digabungkan dengan smartphone.



DAFTAR PUSTAKA

- Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Jurnal Teknologi Indonesia*, (October), 3.
- Arrofiqoh, E. N., & Harintaka, H. (2018). Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi. *Geomatika*, 24(2), 61.
- Briot, J. (n.d.). Deep Learning Techniques for Music Generation – A Survey. 2017.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. USA: O'Reilly Media, Inc.
- Hakim, L. (2017). Menentukan kualitas musik menggunakan metode neural fuzzy.
- Haumahu, J. P. (2019). Implementasi Jaringan Syaraf Tiruan Untuk Pengenalan Pola Notasi Balok Menggunakan Metode Backpropagation. 6(3), 255–259.
- Hildayanti, K. I., Darmanto, T., & Informatika, T. (2019). Perancangan Aplikasi Pembuat Partitur Musik dengan Not Angka. 1–11.
- Ilmiah, P., Nurfita, R. D., Informatika, P. S., Komunikasi, F., Informatika, D. A. N., & Surakarta, U. M. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari.
- Irsyad, R. (2018). Penggunaan Python Web Framework Flask Untuk Pemula. *Laboratorium Telematika, Sekolah Teknik Elektro & Informatika*, 1–4.
- Kartika, A., Irawan, B., & Triyanto, D. (2016). Prediksi Wilayah Rawan Kebakaran Hutan dengan Metode Jaringan Syaraf Tiruan Propagasi Balik (Study Kasus : Daerah Kabupaten Kuburaya). *Coding Jurnal Komputer Dan Aplikasi Untan*, 4(2), 66–75.
- Kusumaningsih, D., Adhie, S., & Pramudita, P. (2016). Penggunaan Metode

Backpropagation Artificial Neural Network Dalam Sistem Pengenalan Notasi Balok Menjadi MidI. 8(2), 146–152.

Lina, Q. (n.d.). Apa itu Convolutional Neural Network? Retrieved from Medium website: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>

Maesaroh, S. (2017). *Sistem Prediksi Produktifitas Pertanian Padi Menggunakan Data Mining*. 7(2), 25–30.

Noviana, D., Susanti, Y., & Susanto, I. (2019). Analisis Rekomendasi Penerima Beasiswa Menggunakan *K-Nearest Neighbor* (K-Nn) Dan Algoritma C4.5.

Nugroho, R. A., Tarno, & Prahutama, A. (2017). Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode Smooth Support Vector Machine (Ssvm). *Gaussian*, 6, 439–448.

Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari. *Jurnal Emitter*, 18(01), 22–27.

Nurhikmat, T. (2018). Implementasi *Deep Learning* untuk *image classification* menggunakan algoritma *convolutional neural network* (CNN) pada citra wayang golek.

Patterson, J., & Gibson, A. (2017). *Deep Learning*. Gravenstein Highway North, Sebastopol.: O'Reilly Media, Inc.

Putra, I. W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101 Image Classification Using Convolution Neural Network (Cnn) on Caltech 101*.

Putri, R. K. S. C. (2018). *Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network Untuk Klasifikasi Gambar*. 121.

Ramdani, A. L., Firmansyah, H. B., Info, A., Coeficient, S., & Tunggal, U. K. (2018). *Clustering Application for UKT Determination Using Pillar K- Means Clustering Algorithm and Flask Web Framework*. 1(2), 53–59.

- Rokhana, R., Priambodo, J., Karlita, T., Sunarya, I. M. G., & Yuniarno, E. M. (2019). *Convolutional Neural Network untuk Pendeteksian Patah Tulang Femur pada Citra Ultrasonik B – Mode*. 8(1).
- Salsabila. (2018). *Penerapan Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Wayang Punakawan*. 121.
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21.
- Soeharto, M. (1992). *Kamus Musik*. Jakarta: PT Gramedia Widiasarana Indonesia.
- Sugiarto, W., Kristian, Y., & Setyaningsih, E. R. (2018). Estimasi Arah Tatapan Mata Menggunakan Ensemble Convolutional Neural Network. *Teknika*, 7(2), 94–101.
- Syamsina, R. N. (2016). *Meningkatkan Kemampuan Membaca Notasi Balok Siswa Les Keyboard Privat melalui Media Flashcard*. 1–9.
- Tedjokusuma, S., & Alamveta, R. (2009). *Aplikasi Penulisan Notasi Balok Dari File Midi Monophonic*. 101–106.
- Tombeng, M. T., Maramis, F. S., Alfred, F., & Puasa, J. (2018). *Perancangan Aplikasi Pembelajaran Notasi Balok Dan Ritme Bagi Mahasiswa Filsafat Universitas Klabat Berbasis Android*.
- Urbánek, J., Brabec, K., Dušek, L., Holoubek, I., Hřebíček, J., & Kubásek, M. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6354(PART 3), 483–488. <https://doi.org/10.1007/978-3-642-15825-4>