

LAPORAN TUGAS AKHIR

RANCANG BANGUN APLIKASI DETEKSI EMOSI SECARA *REAL TIME* MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID

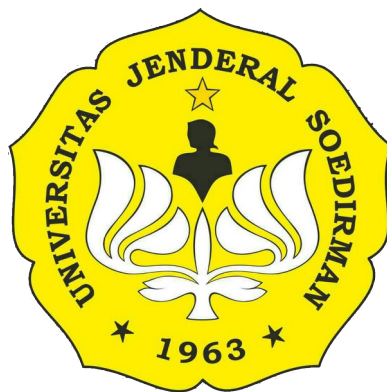
Disusun untuk memenuhi prasyarat memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Ady Septy Widiawati
H1A017004

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2021**



LAPORAN TUGAS AKHIR

RANCANG BANGUN APLIKASI DETEKSI EMOSI SECARA *REAL TIME* MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID

Disusun untuk memenuhi prasyarat memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Ady Septy Widiawati
H1A017004

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2021**

HALAMAN PENGESAHAN

Tugas Akhir dengan Judul:

RANCANG BANGUN KLASIFIKASI EMOSI BERDASARKAN CITRA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID

Disusun oleh:

Ady Septy Widiaawati
H1A017004

Diajukan untuk memenuhi salah satu persyaratan
memperoleh gelar Sarjana Teknik pada
Jurusan/Program Studi Teknik Elektro
Fakultas Teknik
Universitas Jenderal Soedirman

Diterima dan disetujui

Pada Tanggal : _____

Pembimbing I

Pembimbing II

Imron Rosyadi, S.T.,M.Sc.
(NIP : 197909242003121003)

Muhammad Syaiful Aliim, S.T.,M.T.
(NIP : 199009052019031021)

Mengetahui:
Dekan Fakultas Teknik

Prof.Dr.Eng Suroso, S.T., M.Eng.
NIP. 197812242001121002

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Laporan Tugas Akhir dengan judul “**RANCANG BANGUN APLIKASI DETEKSI EMOSI SECARA REAL-TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID**” ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Purbalingga, 4 Februari 2021

[materai sesuai ketentuan uu]
Ttd.

Ady Septy Widiawati
NIM H1A017004

HALAMAN MOTTO DAN PERSEMBAHAN

MOTTO

“Free software” means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”. We sometimes call it “libre software” to show we do not mean it is gratis.

PERSEMBAHAN

Penelitian ini dipersembahkan untuk:

1. Seluruh mahasiswa Teknik Elektro Unsoed,
2. Seluruh mahasiswa Teknik Unsoed, dan
3. Siapapun yang mungkin mendapatkan manfaat dari templat ini.

RINGKASAN

RANCANG BANGUN APLIKASI DETEKSI EMOSI SECARA *REAL TIME* MENGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID

Ady Septy Widiawati

Kepuasan pelanggan terhadap perusahaan suatu produk atau jasa adalah kondisi dimana harapan terhadap produk atau jasa yang diterima oleh pelanggan sesuai dengan kenyataan yang diberikan oleh perusahaan. Perusahaan dapat mengetahui tingkat kepuasan pelanggan dengan melakukan survey yaitu membagikan angket. Selain itu, untuk mengetahui tingkat kepuasan pelanggan yaitu dilihat dari ekspresi wajah setelah pelanggan menggunakan produk atau jasa yang ditawarkan.

Di bidang *computer vision*, sistem pengenalan ekspresi wajah telah dikembangkan. Teknologi pengenalan ekspresi wajah dirancang untuk menganalisis keadaan ekspresi wajah dan secara otomatis mengenali ekspresi wajah manusia. Dalam hal ini penulis membuat penelitian tentang pembuatan aplikasi deteksi emosi secara *real-time* dengan menggunakan metode *Convolutional Neural Network* (CNN) berbasis android. CNN adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis jaringan yang tinggi dan banyak diaplikasikan pada data citra.

Terdapat dua arsitektur CNN yang akan digunakan pada penelitian ini, yaitu *MobileNet* dan VGG-16, arsitektur ini digunakan untuk melatih dan menguji model pada infrastruktur *Google Collaboratory*. Detektor wajah atau muka yang digunakan adalah modul DNN *OpenCV* yaitu *ResNet-10*. Data latih akan disimpan dalam bentuk file *.h5 yang kemudian dikonversi dalam bentuk *TensorFlow Lite* agar mempermudah pembuatan aplikasi di Android. Pembuatan aplikasi dilakukan menggunakan Android Studio.

Kata kunci : deteksi emosi, convolutional neural network, MobileNet, VGG-16, google colaboratory, android

SUMMARY

DESIGN EMOTION DETECTION APPLICATIONS IN REAL TIME USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD ON ANDROID

Ady Septy Widiawati

Customer satisfaction with a company a product or service is a condition in which the expectations of the product or service received by the customer are in accordance with the reality provided by the company. Companies can find out the level of customer satisfaction by conducting a survey, namely distributing questionnaires. In addition, to determine the level of customer satisfaction, it is seen from the facial expressions after the customer uses the product or service offered.

In the field of computer vision, facial expression recognition systems have been developed. Facial expression recognition technology is designed to analyze the state of facial expressions and automatically recognize human facial expressions. In the case, the authors make research on the real-time emotion detection applications using the Android-based Convolutional Neural Network (CNN) method. CNN is a development of the Multilayer Perceptron (MLP) which is designed to process two dimensional data. CNN is a high network type and is widely applied to image data.

*There are two CNN architectures that will be used in the research, namely MobileNet and VGG-16, this architecture is used to train and test model on the Google Collaboratory infrastructure. The face or face detector used is the DNN OpenCV module, namely ResNet-10. The training data will be saved in the form of *.h5 files which are then converted into TensorFlow Lite to make it easier to create applications on Android. Application development is done using Android Studio.*

Keywords : *emotion detection, convolutional neural network, MobileNet, VGG-16, google collaboratory, android*

PRAKATA

Puji syukur kehadiran Allah S.W.T. yang telah melimpahkan berkah dan rahmat-Nya laporan tugas akhir tentang “**RANCANG BANGUN APLIKASI DETEKSI EMOSI SECARA REAL TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID**” ini dapat disusun, adapun maksud dan tujuan penyusunan tugas akhir ini adalah untuk melengkapi salah satu syarat dalam menempuh Pendidikan sarjana pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jenderal Soedirman.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada seluruh pihak yang telah membantu dalam penyusunan laporan tugas akhir ini, yaitu:

1. Allah SWT atas berkat rahmat, karunia serta petunjuk-Nya selama pelaksanaan tugas akhir hingga pembuatan laporan dengan lancar.
2. Bapak, Ibu, Adik, dan saudara-saudara penulis atas limpahan doa, dukungan, dan semangat.
3. Ibu Farida Asriani, S.Si., M.T. selaku kepala jurusan Teknik Elektro Universitas Jenderal Soedirman.
4. Bapak Ari Fadli, S.T., M.Eng. selaku komisi tugas akhir.
5. Bapak Imron Rosyadi, S.T., M.Sc selaku dosen pembimbing I tugas akhir.
6. Bapak Muhammad Syaiful Aliim, S.T., M.T. selaku dosen pembimbing II tugas akhir serta dosen pembimbing akademik.
7. Muhammad Fakhurrozi Sutisna, Mumtaz Prima Rahmaputra, dan Rokhi Iman Sarofi selaku teman seperjuangan bimbingan yang selalu memberikan saran dan menyemangati penulis selama menyelesaikan laporan tugas akhir ini.
8. Ikmal Nurdiansyah, Ahya Suhaila, Mellynia Febrianti, dan teman-teman KTL yang selalu ada untuk menyemangati, berbagi cerita dan juga mendukung penulis selama menyelesaikan laporan tugas akhir ini.

Purbalingga, 2 Februari 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN MOTTO DAN PERSEMBAHAN.....	iv
RINGKASAN.....	v
SUMMARY.....	vi
PRAKATA.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
DAFTAR ISTILAH DAN SINGKATAN.....	xiii
DAFTAR SIMBOL.....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.4.1 Tujuan.....	4
1.4.2 Manfaat.....	4
1.5 Sistematika Penulisan.....	4
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu.....	5
2.2 Ekspresi Wajah dan Emosi.....	6
2.3 Deep Learning.....	7
2.4 <i>Convolutional Neural Networks</i> (CNN).....	7
2.4.1 Input Layer.....	9
2.4.2 Convolutional Layer.....	10
2.4.3 Activation Function.....	11
2.4.4 Pooling Layer.....	11
2.4.5 Fully Connected Layer.....	12

2.5	Arsitektur CNN.....	13
2.5.1	<i>MobileNet</i>	13
2.5.2	Arsitektur VGG-16.....	15
2.6	Pengenalan Wajah (<i>Face Recognition</i>).....	17
2.6.1	Modul DNN <i>OpenCV</i>	18
2.7	Google Colaboratory.....	18
2.8	Android Studio.....	19
2.8.1	Struktur Project.....	20
2.8.2	Antarmuka Pengguna.....	22
2.9	Python.....	23
2.10	<i>Tensorflow</i> dan <i>Keras</i>	24
BAB 3	METODOLOGI PENELITIAN.....	27
3.1	Waktu dan Tempat Penelitian.....	27
3.2	Alat dan Bahan.....	27
3.2.1	Perangkat Keras.....	27
3.2.2	Perangkat Lunak.....	27
3.2.3	Dataset.....	28
3.3	Metode Penelitian.....	28
3.3.1	Tahap Persiapan.....	31
3.3.2	Tahap Persiapan dan <i>Preprocess</i> Dataset.....	31
3.3.3	Tahap Desain Arsitektur.....	31
3.3.4	Tahap Pengujian dan Evaluasi.....	32
3.3.5	Tahap Akhir.....	32
3.4	Waktu dan Jadwal Penelitian.....	33
BAB 4	HASIL DAN PEMBAHASAN.....	34
BAB 5	PENUTUP.....	35
	DAFTAR PUSTAKA.....	36
	LAMPIRAN.....	39
	BIODATA PENULIS.....	41

DAFTAR TABEL

Tabel 2.1 Struktur laporan tugas akhir.....	7
Tabel 3.1 Struktur laporan kerja praktik.....	18
Tabel 4.1 Contoh penyajian tabel.....	28

DAFTAR GAMBAR

Gambar 2.1: Alur Proses CNN.....	8
Gambar 2.2: Proses Input Layer.....	10
Gambar 2.3: Feature Map.....	12
Gambar 2.4: Proses Pooling.....	12
Gambar 2.5: Proses Fully Connected Layer.....	13
Gambar 2.6: Arsitektur MobileNet.....	15
Gambar 2.7: Arsitektur VGG-16.....	17
Gambar 2.8: File Proyek pada Tampilan Android Studio.....	21
Gambar 2.9: Jendela Utama pada Android Studio.....	22
Gambar 2.10: Susunan Perangkat Keras dan Perangkat Lunak Deep Learning....	25
Gambar 3.1: Diagram alir penelitian.....	27
Gambar 3.2: Diagram Alir Sistem.....	28
Gambar 3.3: Desain Arsitektur.....	29

DAFTAR LAMPIRAN

Lampiran 1. Contoh Lampiran 1.....	39
Lampiran 2. Contoh Lampiran 2.....	40

DAFTAR ISTILAH DAN SINGKATAN

CC BY SA : *Creative Common Attribution Share Alike*, merupakan salah satu variasi lisensi *Creative Common*.

Laporan Kerja Praktik : Laporan yang disusun oleh mahasiswa setelah melaksanakan Kerja Praktik sebagai bagian dari studi akhir.

Skripsi : Dokumen laporan tugas akhir yang merupakan syarat kelulusan program Sarjana.

Templat dokumen: Format dasar dokumen (seperti dokumen, surat bisnis, tabel, grafik, dll) yang dapat digunakan berulang kali.

Tabel : Daftar yang berisi ringkasan sejumlah besar data informasi, biasanya berupa kata-kata dan bilangan yang tersusun secara sistematis, urut ke bawah dalam lajur dan deret tertentu dengan garis pembatas sehingga dapat dengan mudah disimak.

Zotero [*zoh-TAIR-oh*] : Perangkat bebas yang mudah digunakan untuk mengumpulkan, mengorganisasikan, mengutip, dan membagikan sumber penelitian.

DAFTAR SIMBOL

E	:	Energi (<i>joule</i>)
m	:	Massa (<i>kg</i>)
c	:	Kecepatan cahaya ($3 \times 10^8 m/s$)
ω	:	Kecepatan sudut (<i>rad/s</i>)
t	:	Waktu (detik)
e	:	Bilangan <i>Euler</i> ($e \approx 2,71828\ 18284\ 59045\ 23536\ 02874\ 71352$)
j	:	Satuan imajiner dengan sifat $j^2 = -1$

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kepuasan pelanggan terhadap perusahaan suatu produk atau jasa adalah kondisi dimana harapan terhadap produk atau jasa yang diterima oleh pelanggan sesuai dengan kenyataan yang diberikan oleh perusahaan. Jika pelayanan yang diberikan perusahaan tidak memenuhi harapan, maka pelanggan akan merasa sedikit kecewa. Suatu perusahaan dapat mengetahui tingkat kepuasan pelanggan dengan melakukan survey yaitu membagikan form kuesioner kepada para pelanggan yang sudah menggunakan produk atau jasanya. Selain itu, tingkat kepuasan pelanggan juga dapat dilihat dari ekspresi wajah setelah pelanggan menggunakan produk atau jasa yang ditawarkan.

Dalam beberapa tahun terakhir, banyak sekali teknologi di bidang kecerdasan buatan dan *computer vision*. Di bidang *computer vision*, sistem pengenalan ekspresi wajah telah dikembangkan. Teknologi pengenalan ekspresi wajah ini dirancang untuk menganalisis keadaan ekspresi wajah dan secara otomatis mengenali ekspresi wajah manusia [1].

Ekspresi wajah adalah salah satu cara yang disebut sebagai komunikasi nonverbal untuk mengungkapkan segala macam emosi, baik yang negatif maupun yang positif. Biasanya orang akan mengenal dengan tepat apakah ekspresi wajah menunjukkan emosi marah, sedih, senang, takut, dan lainnya. Emosi adalah keadaan perasaan yang banyak berpengaruh pada perilaku. Biasanya emosi merupakan reaksi terhadap rangsang dari luar dan dalam diri individu. Emosi

berkaitan dengan perubahan fisiologis dan berbagai pikiran. Emosi dapat diklasifikasikan dengan melihat ekspresi wajah seseorang [2].

Deep learning adalah proses pembelajaran yang menggunakan algoritma yang mengacu pada hukum matematik yang bekerja seperti otak manusia. *Deep learning* dimanfaatkan untuk berbagai macam pekerjaan seperti memprediksi peluang atau kejadian, mengenali objek, dan mendiagnosa penyakit. Salah satu pemanfaatan *deep learning* adalah *image processing*, *image processing* ini digunakan untuk mengenali atau mengklasifikasikan suatu objek dengan cepat, tepat dan efisien [3].

Salah satu metode yang ada di dalam *deep learning* dan sering digunakan untuk *image processing* adalah *Convolutional Neural Network* (CNN). CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis jaringan yang tinggi dan banyak diaplikasikan pada data citra [4].

Perusahaan yang menyediakan produk atau jasa dapat mengetahui tingkat kepuasan pelanggan dengan melihat emosi dari para pelanggan. Hal ini bisa dilakukan dengan bantuan kecerdasan buatan yaitu menggunakan metode CNN untuk mengklasifikasikan emosi pelanggan.

Dilihat dari latar belakang tersebut, penulis akan melakukan penelitian dan perancangan mengenai klasifikasi emosi berbasis android. Sehingga diperoleh judul Tugas Akhir “RANCANG BANGUN APLIKASI DETEKSI EMOSI

SECARA *REAL TIME* MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID”.

1.2 Rumusan Masalah

Sesuai latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana perancangan arsitektur CNN untuk klasifikasi emosi manusia?
2. Bagaimana pelatihan dan pengujian untuk klasifikasi emosi manusia dengan menggunakan metode CNN?
3. Bagaimana pengujian klasifikasi emosi manusia pada aplikasi Android?

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut.

1. Metode *deep learning* yang digunakan adalah CNN (*Convolutional Neural Network*).
2. Program dibuat hanya untuk mengklasifikasikan tiga emosi manusia, yaitu marah, sedih dan senang.
3. Program dibuat hanya untuk mendeteksi satu wajah.
4. Bahasa pemrograman yang digunakan untuk mengklasifikasikan emosi adalah Bahasa Pemrograman *Python* dengan antarmuka dan infrastruktur *Google Colaboratory*.
5. Pembuatan model *deep learning* menggunakan *Framework Keras* dan *TensorFlow*.

6. Aplikasi Android yang digunakan untuk mengklasifikasikan emosi dibuat menggunakan Android Studio.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan pembuatan laporan tugas akhir ini antara lain.

1. Merancang arsitektur CNN untuk klasifikasi emosi dan penerapannya pada aplikasi Android.
2. Melatih arsitektur CNN untuk klasifikasi emosi.
3. Menguji arsitektur CNN klasifikasi emosi yang sudah dilatih sebelumnya.

1.4.2 Manfaat

Manfaat yang diharapkan dalam pelaksanaan tugas akhir ini adalah sebagai berikut.

1. Bagi mahasiswa mampu menerapkan ilmu yang didapat pada mata kuliah yang bersangkutan untuk menyelesaikan tugas akhir.
2. Memudahkan perusahaan dalam melakukan survey tingkat kepuasan pelanggan terhadap produk dan jasa yang ditawarkan.

1.5 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terdiri dari lima bab yang terdiri dari BAB I Pendahuluan, BAB II Tinjauan Pustaka, BAB III Metode Penelitian, BAB IV Hasil dan Pembahasan serta BAB V Penutup.

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam beberapa jurnal penelitian terdahulu yang membahas tentang klasifikasi obyek dengan citra dapat dilihat dari beberapa jurnal sebagai berikut :

1. Muhammad Zufar dan Budi Setiyono penelitiannya yang berjudul “*Convolutional Neural Networks* untuk Pengenalan Wajah Secara *Real Time*” membahas mengenai deteksi wajah secara *real time* dengan menggunakan dataset berupa himpunan gambar wajah dalam dua kondisi yaitu gambar wajah dalam kondisi pencahayaan yang minim dan gambar wajah dalam kondisi pencahayaan yang bagus menggunakan 7 lapisan, yaitu *input layer*, *convolutional layer C1*, *pooling layer P2*, *convolutional layer C3*, *pooling layer P4*, *hidden layer H5* dan *output layer O6* [5].
2. Syariful Rijal penelitiannya yang berjudul “Pengenalan Ekspresi Wajah Menggunakan *Deep Learning*” membahas mengenai klasifikasi ekspresi wajah yaitu ekspresi senang, marah dan sedih dengan menggunakan data wajah sebanyak 150 wajah untuk masing-masing ekspresi. Data akan dibagi menjadi 70% data *train* dan 30% data *test* atau data validasi [1].
3. Vidi Fitriansyah Hidarlan penelitiannya yang berjudul “Rancang Bangun Klasifikasi Varietas Beras Berdasarkan Citra Menggunakan Metode *Convolutional Neural Network (CNN)* Berbasis Android” membahas mengenai klasifikasi beras yaitu beras IR 64, beras Basmathi, dan beras

Ketan dengan dataset berjumlah 180 gambar dan menggunakan dua buah arsitektur yaitu arsitektur *MobileNet* dan *VGG16* [6].

4. Jepri penelitiannya yang berjudul “Identifikasi Penyakit Pada Daun Tomat dan Daun Singkong Berdasarkan Citra Daun Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Android” membahas mengenai identifikasi penyakit pada daun tomat dan daun singkong dengan menggunakan tiga buah arsitektur, yaitu arsitektur VGG16, mobileNet, dan Inception V3 [7].

2.2 Ekspresi Wajah dan Emosi

Ekspresi wajah adalah salah satu cara yang disebut komunikasi nonverbal, untuk mengungkapkan segala macam emosi, baik emosi yang positif maupun emosi yang negatif. Biasanya orang akan mengenal dengan tepat apakah ekspresi wajah menunjukkan emosi marah, sedih, senang, takut. Emosi adalah keadaan yang banyak berpengaruh pada perilaku. Biasanya emosi merupakan reaksi terhadap rangsang dari luar dan dalam diri individu. Emosi berkaitan dengan perubahan fisiologis dan berbagai pikiran. Jadi, emosi merupakan salah satu aspek penting dalam kehidupan manusia [2].

Berikut merupakan beberapa bentuk emosi manusia :

1. Marah atau amarah yaitu beringas, mengamuk, benci, marah besar, jengkel, kesal hati, tersinggung, terganggu, bermusuhan, hingga tindakan kekerasan, dan kebencian patologis.

2. Sedih atau kesedihan yaitu pedih, muram, suram, melankolis, mengasihi diri, kesedihan, ditolak, dan depresi berat.
3. Senang yaitu bahagia, gembira, puas, terhibur, bangga, takjub, terpesona, senang sekali, dan manis.

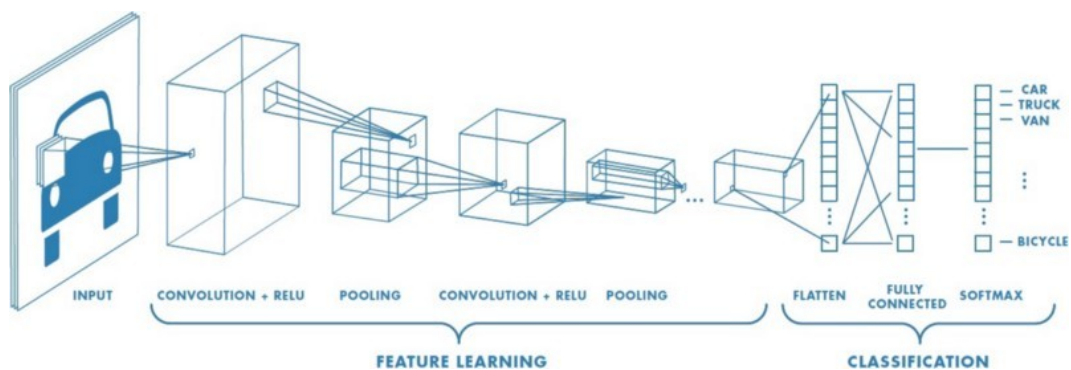
2.3 Deep Learning

Deep learning adalah salah satu cabang *machine learning* (ML) yang menggunakan *Deep Neural Network* (DNN) untuk menyelesaikan permasalahan pada domain ML. *Neural network* sendiri merupakan model yang terinspirasi dari cara kerja neuron di dalam otak manusia. Setiap neuron pada otak manusia saling berhubungan dan informasi mengalir dari setiap neuron. Sederhananya, *Deep Learning* merupakan pembelajaran mendalam dalam pengembangan jaringan saraf tiruan atau neuron yang tentunya memiliki banyak lapisan. Metode ini sangat efektif dan lebih mudah dalam mengidentifikasi pola dari data yang dimasukkan [8].

2.4 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) adalah salah satu jenis model Jaringan Syaraf Tiruan (JST) yang memiliki keunikan dari arsitektur atau susunan *layernya* dibandingkan dengan model JST lainnya. CNN memiliki struktur yang mirip dengan *Artificial Neural Network* (ANN). Dalam hal klasifikasi citra, CNN menerima citra *input* atau masukan kemudian diproses dan diklasifikasi ke kategori tertentu. Yang membedakan CNN dengan ANN adalah CNN memiliki arsitektur tambahan yang dioptimalkan untuk fitur-fitur yang ada pada citra *input* [9].

Gambar 2.1 merupakan alur dari proses CNN dalam mengolah citra masukan untuk mengklasifikasikan citra tersebut ke dalam kategori tertentu berdasarkan nilai keluarannya. CNN memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan [10].



Gambar 2.1: Alur Proses CNN

CNN merupakan *layer* yang memiliki susunan neuron 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer*, sedangkan kedalaman mengacu pada jumlah *layer*. Secara umum jenis *layer* pada CNN dibedakan menjadi dua, yaitu [5]:

1. *Layer* Ekstraksi Fitur Gambar

Layer ini terletak pada awal arsitektur yang tersusun atas beberapa *layer* dan setiap *layer* tersusun atas neuron yang terkoneksi pada daerah lokal (*local region*) *layer* sebelumnya. *Layer* jenis pertama adalah *layer* konvolusi dan *layer* kedua adalah *layer pooling*. Setiap *layer* diberlakukan fungsi aktivasi. Posisinya berselang-seling antara jenis pertama dengan jenis kedua. *Layer* ini menerima

input gambar secara langsung dan memprosesnya hingga menghasilkan keluaran berupa vektor untuk diolah pada *layer* berikutnya [5].

2. *Layer* klasifikasi

Layer ini tersusun atas beberapa *layer* dan setiap *layer* tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan *layer* lainnya. *Layer* ini menerima *input* dari hasil keluaran *layer* ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa skoring kelas untuk klasifikasi [5].

Dengan demikian, CNN merupakan metode untuk mentransformasikan gambar original *layer* per *layer* dari nilai piksel gambar ke dalam nilai skoring kelas untuk klasifikasi. Setiap *layer* ada yang memiliki hyperparameter dan ada yang tidak memiliki parameter (bobot dan bias pada neuron) [5].

2.4.1 *Input Layer*

Input layer menampung nilai piksel dari citra yang menjadi masukan. Untuk citra dengan ukuran 64x64 dengan 3 *channel* warna, RGB(*Red*, *Green*, *Blue*) maka yang menjadi masukan adalah piksel *array* yang berukuran 64x64x3 [9].



Gambar 2.2: Proses Input Layer

2.4.2 Convolutional Layer

Convolutional layer adalah inti dari CNN. *Convolutional layer* menghasilkan citra baru yang menunjukkan fitur dari citra *input*. Dalam proses tersebut, *Convolutional layer* menggunakan filter pada setiap citra yang menjadi masukan. Filter merupakan representasi bidang reseptif dari neuron yang terhubung ke dalam daerah lokal (*local connectivity*) pada citra masukan. Filter pada *layer* ini berbentuk *array* 2 dimensi bisa berukuran 5x5, 3x3 atau 1x1. Proses *convolutional* dengan menggunakan filter pada *layer* ini akan menghasilkan *feature map* yang akan digunakan pada *activation layer* [9].

Convolutional layer memiliki *hyperparameter* dan parameter [5]:

1. *Depth* adalah kedalaman layer atau jumlah layer konvolusi.
2. *Stride* adalah jumlah pergeseran filter pada proses konvolusi.
3. *Zero Padding* adalah jumlah penambahan nilai intensitas nol di daerah sekitar *input* gambar.

2.4.3 Activation Function

Activation layer adalah *layer* dimana *feature map* dimasukkan ke dalam fungsi aktivasi. Fungsi aktivasi digunakan untuk mengubah nilai-nilai *feature map* pada *range* tertentu sesuai dengan fungsi aktivasi yang digunakan. Hal ini bertujuan untuk meneruskan nilai yang menampilkan fitur domain dari citra yang masuk ke *layer* berikutnya [9]. Terdapat beberapa fungsi aktivasi yang sering digunakan yaitu:

1. Fungsi Aktivasi ReLu (*Rectification Linear Unit*)

ReLU merupakan operasi untuk mengenalkan non linearitas dan meningkatkan representasi dari model. Nilai *output* dari neuron bisa dinyatakan sebagai 0 jika *input*-nya adalah negatif. Jika nilai *input* adalah positif, maka *output* dari neuron adalah nilai *input* aktivasi itu sendiri [11].

2. Fungsi Aktivasi Softmax

Fungsi aktivasi *softmax* digunakan untuk mendapatkan hasil klasifikasi. Fungsi aktivasi menghasilkan nilai yang diinterpretasi sebagai probabilitas yang belum dinormalisasi untuk tiap kelas [11].

2.4.4 Pooling Layer

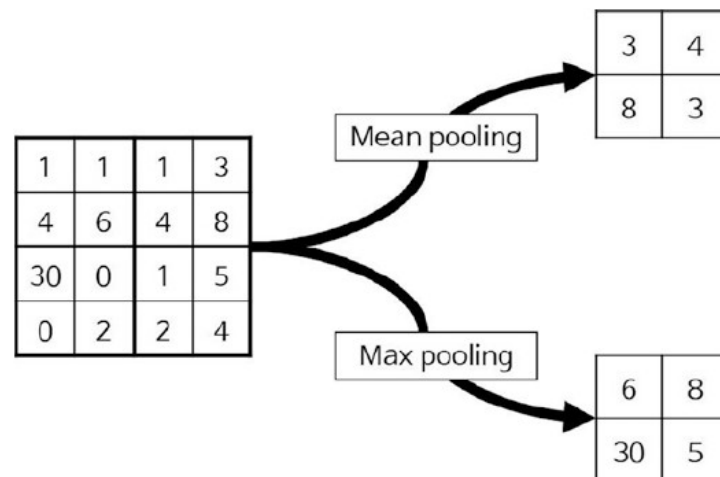
Pooling layer menerima *input* dari *activation layer* kemudian mengurangi jumlah parameternya. *Pooling* juga biasa disebut *subsampling* atau *downsampling* yang mengurangi dimensi dari *feature map* tanpa menghilangkan informasi penting di dalamnya. Proses dalam *pooling layer* cukup sederhana, pertama-tama kita menentukan ukuran *downsampling* yang akan digunakan pada *feature map*, misalnya 2x2. Setelah itu kita akan melakukan proses *pooling* pada

feature map, sebagai contoh kita akan menggunakan *feature map* berukuran 4x4 berikut [9].

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

Gambar 2.3: Feature Map

Setelah itu akan menggunakan matriks 2x2 untuk melakukan proses *pooling*. Proses *pooling* sendiri ada beberapa macam seperti *max pooling*, *mean pooling* dan *sum pooling*.

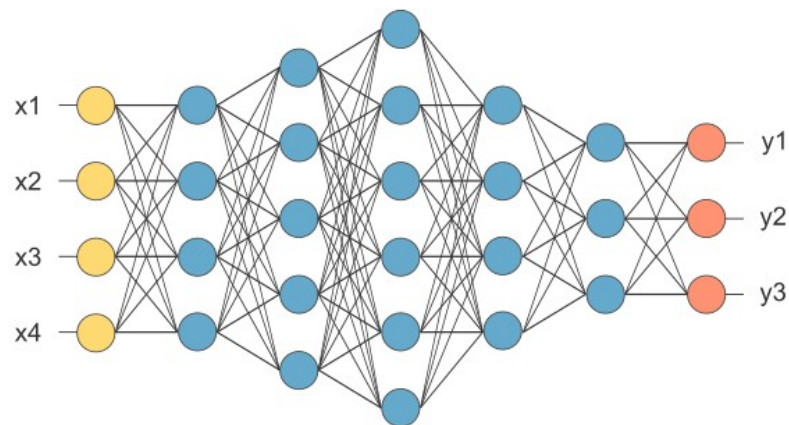


Gambar 2.4: Proses Pooling

2.4.5 Fully Connected Layer

Setelah melewati proses-proses di atas, hasil dari *pooling layer* digunakan menjadi masukan untuk *fully connected layer*. Layer ini memiliki

kesamaan struktur dengan ANN pada umumnya yaitu memiliki *input layer*, *hidden layer*, dan *output layer* yang masing-masing memiliki neuron-neuron yang saling terhubung dengan neuron-neuron di *layer* tetangganya [9].



Gambar 2.5: Proses Fully Connected Layer

Pada gambar 2.5 dapat dilihat sebelum hasil *pooling* digunakan sebagai input, hasil *pooling* terlebih dahulu diubah menjadi vektor (x1, x2, x3, dst) kemudian dari sini diproses ke dalam *fully connected layer*. Pada *layer* terakhir di dalam *fully connected layer* akan digunakan fungsi aktivasi *sigmoid* atau *softmax* untuk menentukan klasifikasi dari citra *input* atau masukan yang dari *input layer* CNN [9].

2.5 Arsitektur CNN

2.5.1 MobileNet

MobileNet adalah arsitektur CNN yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih. *MobileNet* merupakan arsitektur CNN yang dapat digunakan di ponsel. Secara umum, perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN adalah penggunaan lapisan konvolusional atau lapisan dengan ketebalan filter yang sesuai dengan ketebalan

citra masukan. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution* [12].

1. *Depthwise Separable Convolution*

Depthwise Separable Convolution merupakan *novelty* yang dihadirkan pada model *MobileNet V1* [13]. *Depthwise separable convolution* merupakan sebuah blok pada *deep learning* yang terdiri dari *depthwise convolution* dan *pointwise convolution*. Tujuan dari *depthwise separable convolution* ini untuk mereduksi komputasi dan ukuran dari model. *Depthwise separable convolution* sendiri diciptakan pada tahun 2014 sebagai disertasi. *Depthwise convolution* merupakan hasil faktorisasi dari konvolusi standar. Dari N jumlah *input*, *depthwise convolution* melakukan prosesnya untuk setiap kanalnya. Sebagai contoh, *input* dari *layer depthwise convolution* ada 10 kanal, maka akan menghasilkan 10 hasil konvolusi baru. *Pointwise convolution* merupakan kernel dengan ukuran 1×1 yang digunakan untuk menggabungkan seluruh hasil konvolusi dari *depthwise convolution*. Pada konvolusi standar, operasi konvolusi dilakukan oleh setiap filter dengan seluruh kanal. Dengan melakukan faktorisasi ini, *depthwise separable convolution* mereduksi biaya komputasi tersebut.

2. Struktur Jaringan *MobileNet*

Gambar 2.6 merupakan jaringan *MobileNet*. Semua lapisan diikuti oleh *batchnorm* (*backnormalization*) oleh ReLu dengan pengecualian lapisan akhir yang terhubung secara keseluruhan yang tidak memiliki nonlinier dan dimasukkan ke dalam lapisan *softmax* untuk klasifikasi.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1 $3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
	Conv / s1 $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 2.6: Arsitektur MobileNet

2.5.2 Arsitektur VGG-16

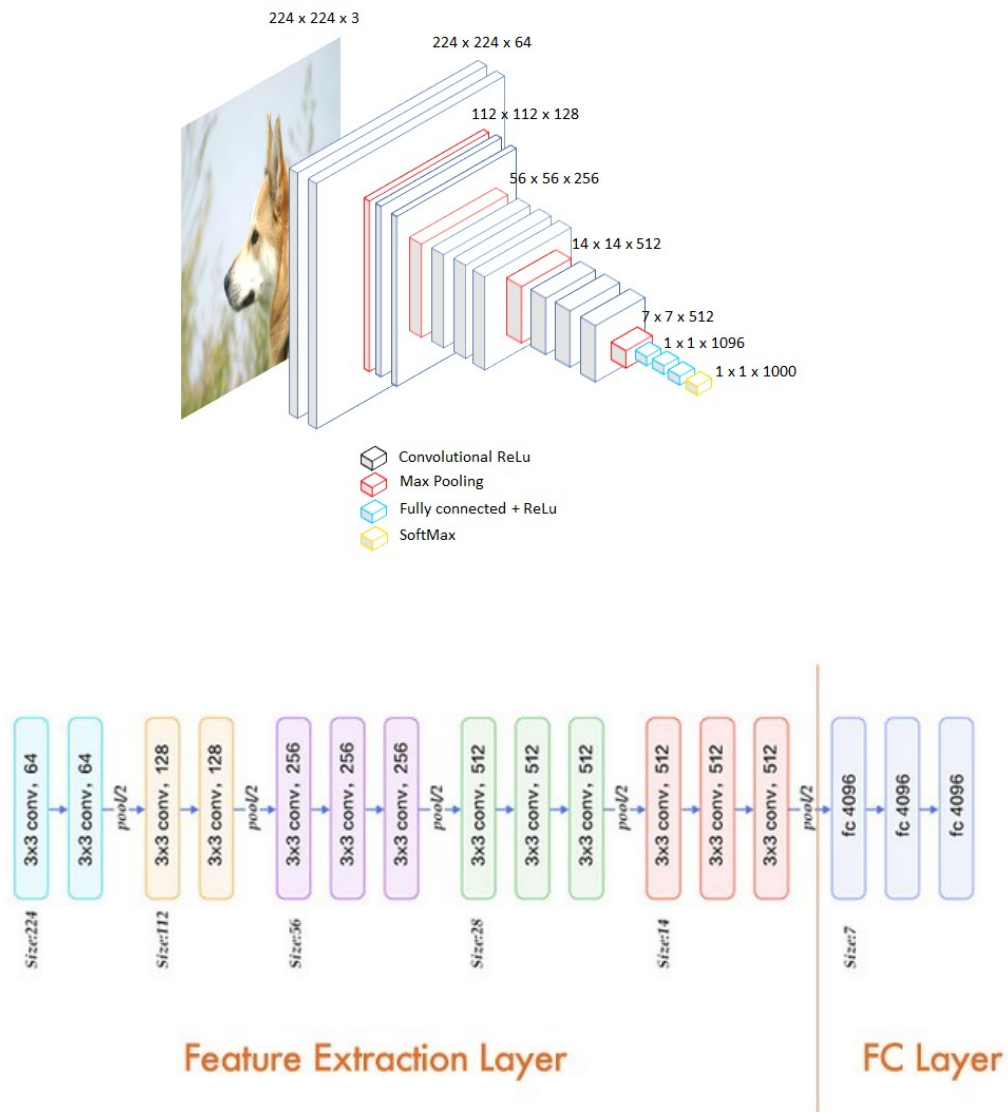
Arsitektur VGG-16 pertama kali diperkenalkan oleh Simonyan dan Zisserman pada tahun 2014 dengan makalahnya yang berjudul “*Very Deep Convolutional Networks of Large Scale Image Recognition*”. Arsitektur VGG 16 berhasil mencapai 92,7% akurasi pengujian top-5 di *ImageNet*, yang mana terdiri dari 14 juta gambar yang berasal dari 1000 kelas yang berbeda [14].

Arsitektur VGG-16 adalah arsitektur CNN dengan *input* yang digunakan berupa RGB *image* berukuran 224×224 piksel. *Convolutional layer* yang digunakan dalam arsitektur ini ada 2 jenis, yaitu *convolutional layer* dengan ukuran filter 3×3 (*conv3*) dan ukuran filter 1×1 (*conv1*) [14].

Gambar dilewatkan melalui tumpukan lapisan konvolusional (*conv*), dimana filter digunakan dengan bidang reseptif yang sangat kecil sebesar 3x3. Dalam salah satu konfigurasi, ia juga menggunakan filter konvolusi 1x1 yang dapat dilihat sebagai transformasi linear dari saluran masukan (diikuti oleh non linearitas). Langkah (*stride*) konvolusi ditetapkan pada 1 piksel, *spasial padding* dari masukan lapisan konvolusi sedemikian rupa sehingga resolusi *spasial*/ukuran tersebut dipertahankan setelah konvolusi, yaitu *padding* sebesar 1 piksel untuk 3x3 lapisan konvolusi. *Spasial pooling* dilakukan oleh lima lapisan *max-pooling*, yang mengikuti beberapa lapisan konvolusi (tidak semua lapisan konvolusi diikuti oleh *max-pooling*). *Max-pooling* dilakukan dengan ukuran 2x2 piksel dan 2 *stride* [14].

Tiga lapisan *Fully Connected* (FC) mengikuti tumpukan lapisan konvolusional (yang memiliki kedalaman berbeda dalam arsitektur yang berbeda), yaitu dua yang pertama memiliki ukuran masing-masing 4096 kanal, yang ketiga melakukan klasifikasi ILSVRC 1000 keluaran dan dengan demikian memuat 1000 saluran (satu untuk setiap kelas). Lapisan terakhir adalah lapisan *soft-max*. Konfigurasi *fully connected layers* adalah sama di semua jaringan [14].

Semua *hidden layers* dilengkapi dengan fungsi aktivasi ReLu. Juga dicatat bahwa tidak ada jaringan (kecuali satu) yang berisi Normalisasi Respon Lokal (LRN), normalisasi seperti itu tidak meningkatkan kinerja pada dataset ILSVRC, tetapi mengarah pada peningkatan konsumsi memori dan waktu komputasi [14].



Gambar 2.7: Arsitektur VGG-16

2.6 Pengenalan Wajah (Face Recognition)

Pengenalan wajah adalah proses mengidentifikasi atau memverifikasi sebuah citra wajah yang tidak diketahui dengan algoritma komputasi, dan membandingkannya dengan data wajah yang ada [15]. Operasi ini dapat dilakukan dengan membandingkan wajah yang tidak diketahui dengan wajah yang tersimpan

di dalam database. Dalam proses pengenalan wajah, gambar dapat diambil dari jarak jauh tanpa menyentuh orang yang sedang diidentifikasi. Proses pengenalan wajah memiliki tiga tahap, yaitu lokasi deteksi wajah, ekstraksi fitur, dan klasifikasi citra wajah.

2.6.1 Modul DNN OpenCV

OpenCV adalah *library open source* yang dikhususkan untuk melakukan *image processing*. Secara resmi *OpenCV* versi 3.3 dirilis pada Agustus 2017 dengan modul *Deep Neural Network* (DNN) yang telah ditingkatkan. Modul ini mendukung sejumlah *framework deep learning*, termasuk *Caffe*, *TensorFlow*, dan *Torch/PyTorch*. Modul DNN *OpenCV* ini merupakan model *Caffe* yang didasarkan pada *Single Shot-Multibox Detector* (SSD) dan menggunakan arsitektur *ResNet-10* sebagai tulang punggungnya [16].

Untuk menggunakan modul DNN *OpenCV* dengan model *Caffe*, yang harus dilakukan adalah mengunduh file *Caffe* dari repositori *OpenCV* terlebih dahulu, file *.prototxt* mendefinisikan arsitektur jaringan dan file *.caffemodel* yang berisi bobot untuk lapisan [16].

2.7 Google Colaboratory

Google Colaboratory atau *Google Colab* adalah produk *Google* yang berbasis *cloud* dan gratis untuk tujuan penelitian. *Google Colab* dibuat dengan *environment Jupyter Notebook* dan mendukung hampir semua *library* yang dibutuhkan dalam pengembangan *Artificial Intelligence* (AI) [17]. Berikut adalah beberapa kelebihan dari *Google Colab*.

1. Spesifikasi bagus

Ketika menggunakan *Google Colab* maka akan diberikan akses *cloud* komputer dengan spesifikasi GPU Nvidia K80s, T4s, P4s dan P100s, RAM 13 GB dan *disk* 130GB.

2. Tidak memerlukan konfigurasi

Tidak ada konfigurasi yang diperlukan untuk menggunakan *Google Colab*, tetapi ketika ingin menambahkan *library* baru, maka perlu menginstall *library package*.

3. Berbagi dengan mudah

Google Colab dapat diintegrasikan dengan *google drive* pengguna dan kemudian menyimpan *script* ke proyek *Github*, selain itu juga dapat berbagi kode sumber dengan orang lain secara *online*.

2.8 Android Studio

Android studio adalah lingkungan pengembangan terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi android berdasarkan *Intellij IDEA* [18]. Selain sebagai editor kode dan fitur *developer Intellij* yang hebat, Android Studio juga menyediakan banyak fitur untuk membuat aplikasi Android, seperti :

1. Sistem *build* berbasis *Gradle* yang fleksibel.
2. Emulator yang cepat dan kaya fitur.

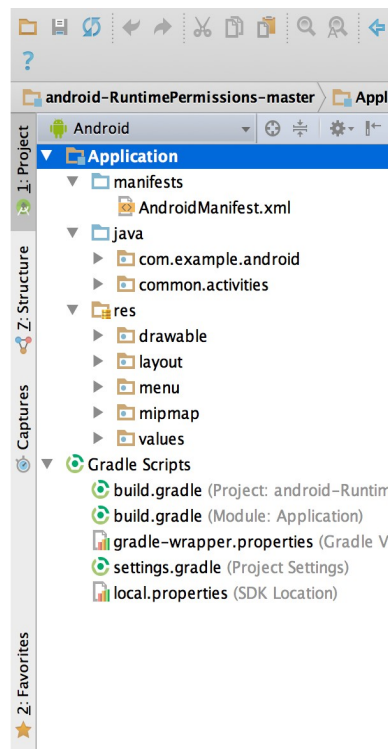
3. Lingkungan terpadu untuk mengembangkan aplikasi untuk semua perangkat android.
4. Terapkan perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
5. Template kode dan integrasi *Github* untuk membantu pembuatan fitur aplikasi umum dan mengimpor kode sampel.
6. *Framework* dan fitur pengujian yang lengkap.
7. Fitur *lint* untuk merekam performa, kegunaan, kompatibilitas versi dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk *Google Cloud Platform* yang memudahkan integrasi *Google Cloud Messaging* dan *APP Engine*.

2.8.1 Struktur Project

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file *resource* [18]. Jenis modul meliputi :

1. Modul aplikasi Android
2. Modul *library*
3. Modul *Google App Engine*

Secara *default*, Android Studio menampilkan file proyek Anda dalam tampilan proyek Android, seperti ditunjukkan pada Gambar 2.7. Tampilan diatur dalam modul untuk memberikan akses cepat ke file sumber utama proyek Anda.



Gambar 2.8: File Proyek pada Tampilan Android Studio

Semua file *build* terlihat di tingkat atas bagian *Script Gradle* dan setiap modul aplikasi berisi folder berikut :

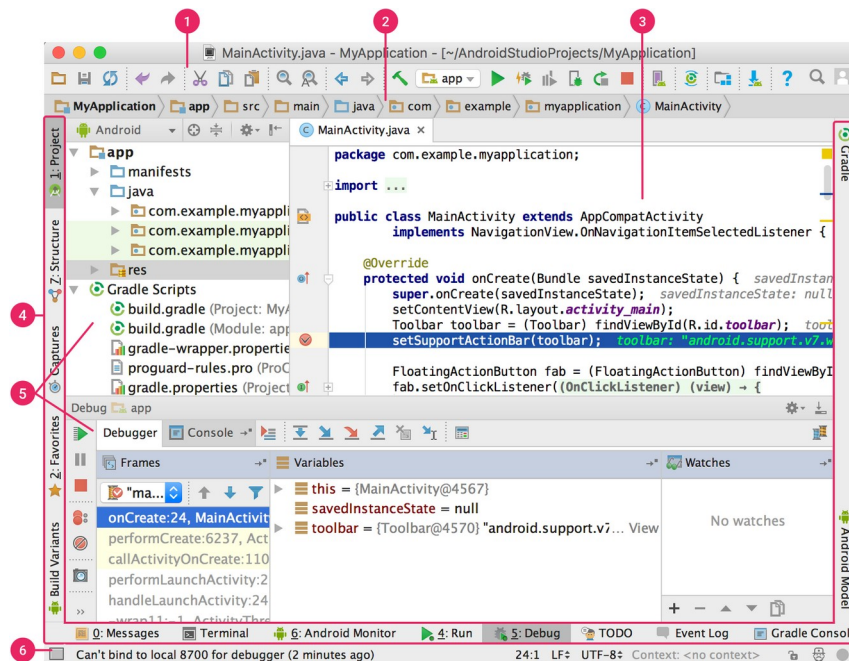
- a) *Manifests* : berisi file *AndroidManifest.xml*.
- b) *Java* : berisi file kode sumber *Java*, termasuk kode pengujian *Junit*.
- c) *Res* : berisi semua *resource* non-kode, seperti tata letak *XML*, *string* UI dan gambar *bitmap*.

Anda juga dapat menyesuaikan tampilan *file* proyek untuk fokus pada aspek tertentu dari pengembangan aplikasi. Misalnya, memilih tampilan “*problems*” pada proyek akan menampilkan link file sumber yang berisi kesalahan

coding dan sintaks yang dikenali, seperti tag penutup elemen XML yang tidak ada di file tata letak.

2.8.2 Antarmuka Pengguna

Jendela utama Android Studio berisi beberapa area logos, seperti yang ditunjukkan pada Gambar 2.9



Gambar 2.9: Jendela Utama pada Android Studio

1. *Toolbar* memungkinkan Anda melakukan berbagai operasi, termasuk meluncurkan aplikasi dan menjalankan fungsi Android.
2. Menu navigasi membantu Anda menavigasi proyek dan membuka file untuk diedit. Menu ini memberikan tampilan yang lebih ringkas dari struktur yang terlihat di jendela *Project*.

3. Jendela editor adalah tempat Anda membuat dan mengubah kode. Menurut tipe file yang ada, editor ini bisa diubah. Misalnya, saat melihat file tata letak, editor akan menampilkan *Layout Editor*.
4. Panel jendela fitur terletak di luar jendela IDE dan berisi tombol untuk membuka atau menutup setiap jendela fungsi.
5. Jendela fitur memungkinkan Anda mengakses tugas tertentu seperti pengelolaan *project*, penelusuran, kontrol versi dan banyak lagi. Anda dapat memperbesar dan memperkecil jendela ini.
6. Status bar menampilkan status project dan IDE itu sendiri, serta semua peringatan atau pesan.

2.9 Python

Python merupakan bahasa pemrograman model skrip (*scriping language*) yang berorientasi obyek [19]. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di banyak sekali *platform* sistem operasi, seperti *Linux*, *Microsoft Windows*, *Mac Os*, *Android*, *Symbian OS*, *Amiga*, *Palm*, dan lain-lain. *Python* merupakan bahasa pemrograman perangkat lunak gratis atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinan atau distribusinya. Lengkap dengan *source code*-nya, *debugger*, dan *profiler*, dan berisi antarmuka layanan, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya.

Beberapa fitur yang dimiliki oleh *python* adalah sebagai berikut [19]:

1. Memiliki kepustakaan yang luas, dalam distribusi *python* telah disediakan modul-modul.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali, dan penulisan ulang kode sumber.
4. Berorientasi obyek.
5. Dapat dibangun dengan bahasa *python* maupun C/C++.

2.10 TensorFlow dan Keras

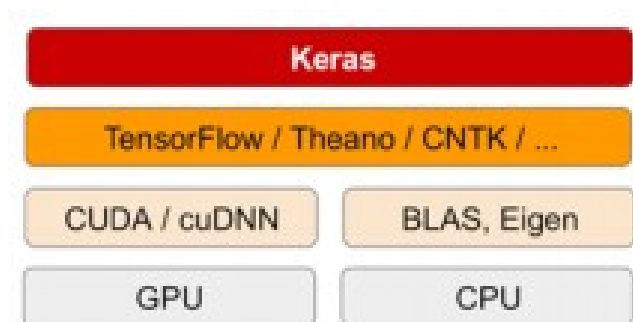
TensorFlow adalah *framework* komputasi yang digunakan untuk membuat model *machine learning*. *TensorFlow* menyediakan berbagai *toolkit* yang memungkinkan Anda membuat model pada level abstraksi yang diperlukan. Anda dapat menggunakan API tingkat rendah untuk membangun model dengan menentukan serangkaian operasi matematika [20].

Keras adalah *interface library* yang dirancang untuk menyederhanakan implementasi algoritma *deep learning* di atas *TensorFlow* [21]. Tidak hanya *TensorFlow* yang didukung oleh *Keras*, tetapi kita dapat mengubah *backend* yang akan digunakan. Saat ini kita dapat menggunakan *TensorFlow*, *Theano* dan CNTK sebagai *backend Keras*.

Berikut merupakan beberapa fitur utama dari *Keras* [22]:

1. Kemampuan untuk menjalankan kode sumber yang sama dengan lancar menggunakan CPU atau GPU.
2. API yang *user-friendly* memudahkan pengguna untuk membangun prototipe model *deep learning*.
3. Dukungan built in untuk CNN, RNN (*Recurrent Neural Network*) untuk pemrosesan urutan dan kombinasi keduanya.
4. Ini dapat digunakan untuk hampir semua jenis model pembelajaran mendalam.

Keras didistribusikan di bawah lisensi MIT, yang berarti dapat digunakan secara bebas dalam proyek komersial. *Keras* kompatibel dengan semua *Python* dari 2.7 hingga 3.6.



Gambar 2.10: Susunan Perangkat Keras dan Perangkat Lunak Deep Learning

Gambar 2.9 merupakan susunan perangkat keras dan perangkat lunak *Deep Learning*. Melalui *TensorFlow* (atau *theano*, atau *CNTK*), *Keras* dapat berjalan mulus di CPU dan GPU. Saat berjalan di CPU, *TensorFlow* sendiri

menggabungkan *library* level rendah untuk operasi tensor yang disebut *Eigen*. Pada GPU, *TensorFlow* menggabungkan pustaka operasi deep learning yang dioptimalkan dengan baik yang disebut *NVIDIA CUDA Deep Neural Network library* (cuDNN).

BAB 3

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dilaksanakan dimulai pada bulan Januari 2021 sampai bulan April 2021. Bertempat di lingkungan kampus Fakultas Teknik Universitas Jenderal Soedirman, Kabupaten Purbalingga.

3.2 Alat dan Bahan

Dalam penelitian tugas akhir ini, daftar alat dan bahan yang digunakan selama penelitian adalah sebagai berikut.

3.2.1 Perangkat Keras

Perangkat keras yang digunakan adalah sebagai berikut.

1. *Laptop* Asus X441M dengan spesifikasi *processor* Intel Celeron N4000 dan RAM 4 GB.
2. *Smartphone Android* Realme C12 dengan spesifikasi *processor Eight Core* dan RAM 3 GB.

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan untuk melakukan penelitian ini adalah sebagai berikut.

1. Sistem Operasi *Windows* 10 64 bit.
2. *Google Colaboratory (Jupyter Notebook versi cloud)*.
3. Web browser *Google Chrome* versi 86.0.4240.111

4. Layanan *Repository Web Development* pada platform *Github*.

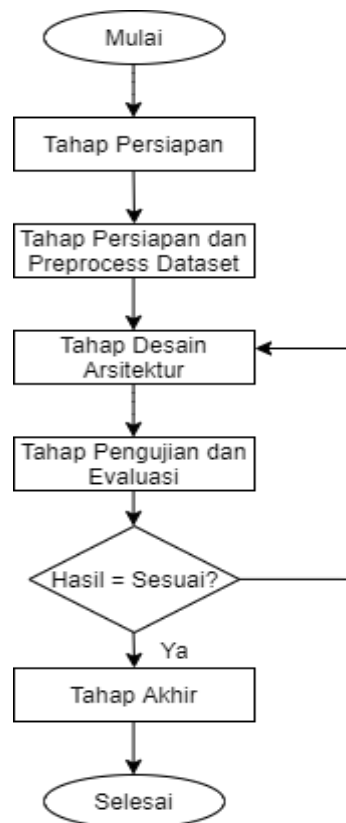
5. Android Studio versi 4.1.

3.2.3 Dataset

Dataset *training* dan *testing* yang digunakan pada penelitian ini diperoleh dari *Google* serta beberapa citra yang diambil dengan menggunakan smartphone oleh penulis. Dataset akan di-*process* dan dibagi dalam dua kelompok, yaitu dataset validasi sebesar 20% dan dataset pelatihan sebesar 80%.

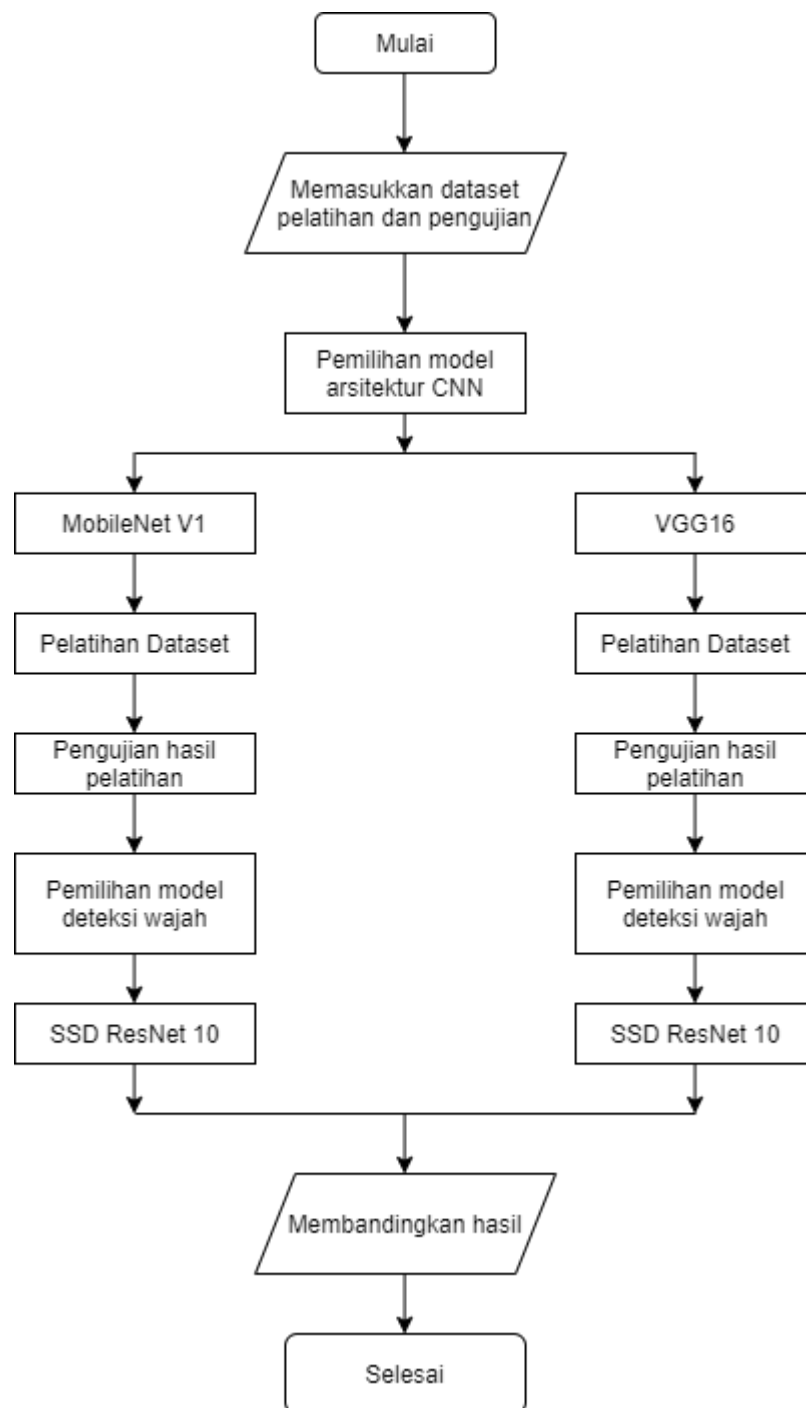
3.3 Metode Penelitian

Pada penelitian ini terdapat beberapa tahapan yaitu tahap persiapan, tahap persiapan dan *preprocess* dataset, tahap desain arsitektur, tahap pengujian dan evaluasi sistem serta tahap akhir. Pelaksanaan penelitian ini dapat digambarkan dalam bentuk diagram seperti pada Gambar 3.1 berikut.



Gambar 3.1: Diagram alir penelitian

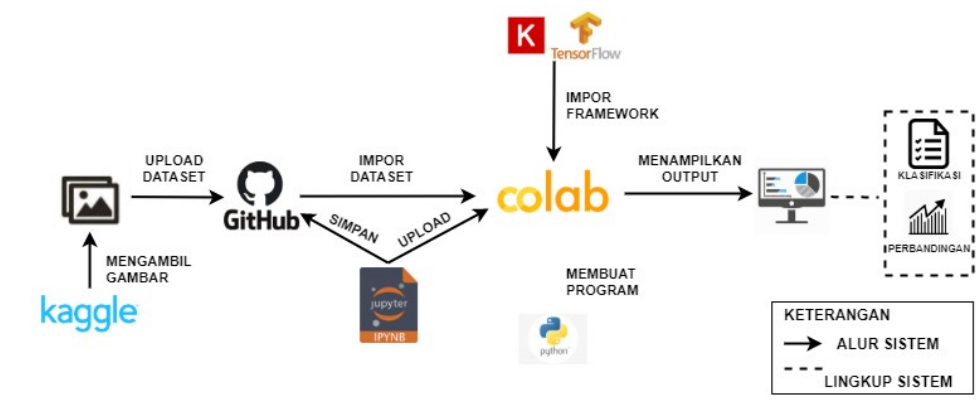
Sedangkan untuk diagram alir dari sistem yang dirancang pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2: Diagram Alir Sistem

Adapun desain arsitektur dari sistem yang akan dibuat dapat dilihat pada

Gambar 3.3.



Gambar 3.3: Desain Arsitektur

3.3.1 Tahap Persiapan

Pada tahap ini penulis melakukan studi pustaka dengan mengumpulkan informasi yang sesuai dengan topik penelitian. Informasi diperoleh dari berbagai sumber, seperti jurnal, buku, website, makalah dan melakukan bimbingan secara langsung dengan dosen pembimbing.

3.3.2 Tahap Persiapan dan *Preprocess* Dataset

Pada tahap ini penulis mengumpulkan dan menyiapkan dataset yang berisi tiga jenis emosi dalam bentuk citra. Penulis memperoleh dataset emosi dari *Google* serta terdapat beberapa citra yang diambil sendiri dengan menggunakan *smartphone*. Dataset dari *Google* akan di-*download* dan kemudian di-*cropping*, begitu juga untuk dataset yang diambil dengan *smartphone*. Setelah itu dataset tersebut akan dimasukkan ke dalam *Google Colaboratory*.

3.3.3 Tahap Desain Arsitektur

Pada tahap ini, hal pertama yang penulis lakukan adalah mengimpor dataset dari *Github* ke penyimpanan sementara *Google Colaboratory*. Selanjutnya membuat kode sumber untuk program CNN. Bahasa yang digunakan adalah

bahasa *python* dengan *Framework Keras* dan *Tensorflow* serta dibuat menggunakan infrastruktur *Google Colaboratory*. Dataset dibagi menjadi dua, yaitu dataset pelatihan (*train*) dan dataset pengujian (*test*). Dataset akan diproses dan dibagi dalam dua kelompok, yaitu dataset validasi sebesar 20% dan pelatihan sebesar 80%. Selanjutnya akan membuat model dari jaringan CNN yang sudah dipelajari sebelumnya, dalam penelitian ini, penulis menggunakan arsitektur *MobileNet* dan *VGG-16* untuk model jaringan CNN. Setelah model terbentuk, dataset akan dilatih untuk dapat mengklasifikasikan emosi dan mendapatkan data latih dalam bentuk file *.h5. Data pelatihan tersebut akan dikonversi ke dalam *TensorFlow Lite*, hal ini bertujuan agar mempermudah pembuatan model aplikasi di Android. Pembuatan aplikasi android dilakukan menggunakan Android Studio.

3.3.4 Tahap Pengujian dan Evaluasi

Pada tahap ini sistem akan diuji secara keseluruhan, apakah sistem bekerja sesuai dengan yang diinginkan atau tidak. Apabila sistem kurang sesuai maka penulis akan melakukan perbaikan dan pengkajian ulang. Sistem dianggap tepat atau baik ketika nilai akurasi mendekati angka 1 dan nilai kesalahan atau *loss* mendekati angka 0.

3.3.5 Tahap Akhir

Pada tahap ini penulis akan menulis laporan yang berjudul **“Rancang Bangun Aplikasi Deteksi Emosi Secara *Real Time* Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Android”**. Laporan tersebut akan memuat hasil penelitian yang telah dilakukan.

3.4 Waktu dan Jadwal Penelitian

Penelitian dilaksanakan selama 16 minggu atau 4 bulan, dimulai pada bulan Januari 2020 sampai April 2021. Rincian jadwal kegiatan disajikan dalam Tabel berikut.

[illegible]

BAB 4

HASIL DAN PEMBAHASAN

BAB 5
PENUTUP

DAFTAR PUSTAKA

- [1] S. Rijal, T. Informatika, and F. Teknik, “Pengenalan Ekspresi Wajah Menggunakan Deep,” 1945.
- [2] J. E. Prawitasllri, “Mengenai Emosi Melalui Komunikasi Nonverbal,” *Bul. Psikol.*, vol. 3, no. 1, pp. 27–43, 2016, doi: 10.22146/bpsi.13384.
- [3] F. F. Maulana and N. Rochmawati, “Klasifikasi Citra Buah Menggunakan Convolutional Neural Network,” *J. Informatics Comput. Sci.*, vol. 01, pp. 104–108, 2019.
- [4] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [5] M. Zufar and B. Setiyono, “Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time,” *J. Sains dan Seni ITS*, vol. 5, no. 2, p. 128862, 2016, doi: 10.12962/j23373520.v5i2.18854.
- [6] V. F. Hidarlan, “Rancang Bangun Klasifikasi Varietas Beras Berdasarkan Citra Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android,” pp. 1–17, 2020.
- [7] Jepri, “BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID DISEASES IDENTIFICATION OF TOMATO LEAVES AND CASSAVA LEAVES IMAGE-BASED USING CONVOLUTIONAL ARTIFICIAL NEURAL NETWORK (CNN) METHOD ON ANDROID.”
- [8] A. Safitri, “Deep Learning dan Manfaatnya Bagi Perkembangan AI,” 31 juli, 2018. [Online]. Available: <https://medium.com/iykra/deep-learning-dan-manfaatnya-bagi-perkembangan-ai-cab94e20c19a>. [Accessed: 02-Nov-2020].
- [9] S. Tandungan, “Pengenalan Convolutional Neural Network-Part 1,” 16 Maret, 2020. [Online]. Available: <http://sofyantandungan.com/pengenalan-convolutional-neural-network-part-1/>. [Accessed: 04-Nov-2020].
- [10] Q. Lina, “Apa Itu Convolutional Neural Network?,” 2 januari, 2019. [Online]. Available: <https://medium.com/@16611110/apa-itu->

convolutional-neural-network-836f70b193a4#:~:text=Dropout adalah teknik regularisasi jaringan,dan tidak dipakai selama pelatihan.&text=Dropout merupakan proses mencegah terjadinya overfitting dan juga mempercepat prose. [Accessed: 24-Jan-2021].

- [11] E. N. Arrofiqoh and H. Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *Geomatika*, vol. 24, no. 2, p. 61, 2018, doi: 10.24895/jig.2018.24-2.810.
- [12] R. O. Ekoputris, "MobileNet : Deteksi Objek pada Platform Mobile," 9 *Maret*, 2018. [Online]. Available: <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>. [Accessed: 06-Nov-2020].
- [13] H. Abdurrohman, R. Dini, and A. P. Muharram, "Evaluasi Performa Metode Deep Learning Untuk Klasifikasi Citra Lesi Kulit the Ham10000," pp. 63–68, 2019, doi: 10.5614/sniko.2018.10.
- [14] Neurohive, "VGG16-Convolutional Network for Classification and Detection," 20 November, 2019. [Online]. Available: <https://neurohive.io/en/popular-networks/vgg16/>. [Accessed: 02-Feb-2021].
- [15] Derisma, "Faktor-Faktor yang Mempengaruhi Sistem Pengenalan Wajah Menggunakan Metode Eigenface pada Perangkat Mobile Berbasis Android," *J. Politek. Caltex Riau*, vol. 2, no. 2, pp. 127–136, 2016.
- [16] A. Rosebrock, "Face detection with OpenCV and deep learning," 26 *Februari*, 2018. [Online]. Available: <https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/>. [Accessed: 02-Feb-2021].
- [17] D. Brahma, "Cara Menggunakan Google Colaboratory," *Februari*, 2020. [Online]. Available: <https://medium.com/@dede.brahma2/cara-menggunakan-google-colaboratory-5f5e4393ac2f>.
- [18] S. Android, "Mengenal Android Studio." [Online]. Available: <https://developer.android.com/studio/intro?hl=id>. [Accessed: 05-Nov-2020].
- [19] T. R. Perkasa, H. Widyantera, and P. Susanto, "Rancang Bangun Pendeteksi Gerak Menggunakan Metode Image Substraction Pada Single Board Computer (SBC)," *J. Control Netw. Syst.*, vol. 3, no. 2, pp. 90–97, 2014.

- [20] R. D. Ramadhani, “Memahami Deep Learning Dengan R dan TensorFlow,” *23 Juni*, 2020. [Online]. Available: <https://medium.com/@16611129/memahami-deep-learning-dengan-r-dan-tensorflow-e73ea9088245>. [Accessed: 06-Nov-2020].
- [21] D. Data, “Implementasi Deep Learning Sederhana Menggunakan Keras,” *Oktober*, 2, 2018. [Online]. Available: <https://medium.com/@danau.data/implementasi-deep-learning-sederhana-menggunakan-keras-3f5726f007e7>. [Accessed: 06-Nov-2020].
- [22] E. Tjio, “Klasifikasi Gambar menggunakan Keras,” *24 Maret*, 2020. [Online]. Available: <https://rpubs.com/enlik/keras>. [Accessed: 06-Nov-2020].

LAMPIRAN

Lampiran 1. Contoh Lampiran 1

Ini adalah isi dari lampiran 1

Lampiran 2. Contoh Lampiran 2

Ini adalah isi dari lampiran 2

BIODATA PENULIS

Pas photo ukuran
3 x 4 cm
berjas

A. Identitas

Nama : Ady Septy Widiawati
NIM : H1A017004
Tempat, tanggal lahir : Batang, 15 September 1999
Alamat : Desa Cluwuk, RT 03/01, Kec.Tulis, Kab.Batang
No. Telp. : 085329656552
Alamat e-mail : adysept15@gmail.com

B. Riwayat Pendidikan Akademik

Periode	Jenjang	Institusi
2017-2021	S1	Teknik Elektro Universitas Jenderal Soedirman
2014 – 2017	SMA	SMAN 1 Batang
2011 – 2014	SMP	SMPN 1 Tulis

C. Riwayat Pendidikan Non Formal (jika ada)

Tahun	Keahlian	Penyelenggara	Kota
2014	Bahasa Inggris Tingkat Mahir	Lembaga Kursus xxxxxx	Purwokerto
2013	Kemanan Jaringan Mikrotik Tingkat Mahir	Lembaga xxxxxxxx	Jakarta

D. Prestasi

Tahun	Tingkat	Prestasi
2014	Nasional	Juara 1 lomba penulisan karya ilmiah, Yogyakarta
2013	Internasional	Medali emas olimpiade sains internasional, Dakka, India

E. Keahlian (tuliskan secara diskriptif)

Memiliki minat di bidang pengembangan perangkat tertanam. Mampu merancang sistem embedded berbasis mikro kontroler atmega, arduino dan ESP8266. Terlibat secara aktif dalam kegiatan asistem Laboratorium Sistem Telekomunikasi dan Informasi sebagai asisten praktikum Algoritma dan Struktur Data, Jaringan Komputer, dan Dasar Pemrograman.