



TUGAS AKHIR - IF184802

PENGENALAN EKSPRESI WAJAH MENGUNAKAN WAVELET TRANSFORM DAN CONVOLUTIONAL NEURAL NETWORK

**HENDRY WIRANTO
NRP 0511154000102**

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - IF184802

***PENGENALAN EKSPRESI WAJAH
MENGUNAKAN WAVELET TRANSFORM DAN
CONVOLUTIONAL NEURAL NETWORK***

**HENDRY WIRANTO
NRP 05111540000102**

**Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.**

**Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESIS - IF184802

**FACIAL EXPRESSION RECOGNITION USING
WAVELET TRANSFORM AND
CONVOLUTIONAL NEURAL NETWORK**

**HENDRY WIRANTO
NRP 05111540000102**

First Advisor

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Second Advisor

Dini Adni Navastara, S.Kom., M.Sc.

Department of Informatics

Faculty of Information and Communication Technology

Institut Teknologi Sepuluh Nopember

Surabaya 2019

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

Pengenalan Ekspresi Wajah Menggunakan Wavelet Transform dan Convolutional Neural Network

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

HENDRY WIRANTO
NRP: 05111540000102

Disetujui oleh Pembimbing Tugas Akhir

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
(NIP. 19751220 200112 2 002) (Pembimbing 1)
2. Dini Adni Navastara, S.Kom., M.Sc.
(NIP. 19851017 201504 2 001) (Pembimbing 2)



SURABAYA
Januari, 2019

(Halaman ini sengaja dikosongkan)

PENGENALAN EKSPRESI WAJAH MENGGUNAKAN WAVELET TRANSFORM DAN CONVOLUTIONAL NEURAL NETWORK

Nama Mahasiswa : **Hendry Wiranto**
NRP : **05111540000102**
Jurusan : **Informatika, FTIK-ITS**
Dosen Pembimbing 1 : **Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.**
Dosen Pembimbing 2 : **Dini Adni Navastara, S.Kom., M.Sc.**

ABSTRAK

Machine learning telah menjadi bagian dari kehidupan sehari-hari bagi banyak orang. Salah satu pengaplikasian machine learning adalah pengenalan ekspresi wajah manusia. Pengenalan ekspresi wajah manusia mengkategorikan gambar ekspresi wajah menjadi satu dari banyak kelas ekspresi wajah berdasarkan fitur gambar tersebut. Banyak perusahaan, badan riset dan universitas yang terus mengembangkan machine learning agar mendapat hasil yang lebih akurat dan cepat. Dari situlah lahir algoritma deep learning, yang merupakan bagian dari machine learning. Convolutional Neural Network (CNN) adalah salah satu deep neural network yang cocok digunakan untuk mengolah data yang berbentuk 2 dimensi, seperti gambar dan video.

Pada tugas akhir ini, penulis mengusulkan sebuah algoritma untuk mengubah data gambar menjadi Wavelet Domain dengan menggunakan Wavelet Transform. Tujuannya adalah untuk meningkatkan akurasi pengenalan ekspresi wajah manusia dengan metode Convolutional Neural Network. Data pelatihan dan uji coba diambil dari dataset “Karolinska Directed Emotional Faces” (KDEF) yang berisi foto wajah manusia dengan 7 ekspresi berbeda yang nantinya menjadi tujuan pengenalan ekspresi wajah manusia yang dibuat. Praproses terhadap data antara lain dilakukan perubahan format gambar menjadi grayscale,

perubahan resolusi gambar menjadi 256x256 piksel, dilakukan proses Discrete Wavelet Transform level 1, dan dilakukan proses augmentasi data berupa refleksi horizontal dan perbesaran ukuran gambar. Hasil uji coba terakhir didapatkan nilai akurasi 89,6%.

Kata kunci: *Convolutional Neural Network, Data Gambar, Dataset Karolinska Directed Emotional Faces, Pengenalan ekspresi wajah manusia, Wavelet Transform.*

***FACIAL EXPRESSION RECOGNITION USING WAVELET
TRANSFORM AND CONVOLUTIONAL NEURAL
NETWORK***

Student's Name : Hendry Wiranto
Student's ID : 05111540000102
Department : Informatics, Faculty of ICT-ITS
First Advisor : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Second Advisor : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRACT

Machine learning has become a part of the daily life of people around the world. One of the application of machine learning is human facial expression recognition. Human facial expression recognition categorizes an image of facial expression into one of many facial expression classes based on the features extracted from the image. Many companies, researchers and universities keep improving the machine learning to get a better and faster result. And from those improvements, deep learning algorithm is born. Convolutional Neural Network (CNN) is one of the deep neural network that suitable to process 2 dimensional data like image and video.

In this undergraduate thesis, the images are transformed into Wavelet Domain using Wavelet Transform before being processed into the proposed network. The purpose of this method is to improve the accuracy of the human facial expression recognition using Convolutional Neural Network. The train and test data used in this thesis is taken from "Karolinska Directed Emotional Faces" (KDEF) dataset which contains human facial expression with 7 different expressions which will be the prediction labels of the human facial expression recognition. The preprocessing of the images include changing the image format to grayscale, changing the image resolution to 256x256 pixels, applying level 1 Discrete Wavelet Transform and applying data

augmentation with horizontal reflection and zoom in. The final test accuracy is 89,6%.

Keywords: *Convolutional Neural Network, Human Facial Expression Recognition, Image Data, Karolinska Directed Emotional Faces Dataset, Wavelet Transform.*

KATA PENGANTAR

Puji syukur saya sampaikan kepada Tuhan yang Maha Esa karena berkat rahmat-Nya saya dapat melaksanakan Tugas Akhir yang berjudul:

“Pengenalan Ekspresi Wajah Menggunakan Wavelet Transform dan Convolutional Neural Network”

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Tuhan Yesus Kristus, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Informatika ITS.
2. Kedua orangtua penulis, dan anggota keluarga lainnya yang telah memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. dan Dini Adni Navastara, S.Kom., M.Sc. selaku pembimbing I dan II yang telah membimbing dan memberikan motivasi, nasihat dan bimbingan dalam menyelesaikan Tugas Akhir ini.
4. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Departemen Informatika ITS dan seluruh dosen dan karyawan Departemen Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa kuliah di Informatika ITS.
5. Admin-admin Laboratorium Komputasi Cerdas & Visi (KCV) yang memberikan kesempatan penulis untuk fokus

- mengerjakan Tugas Akhir ini dan menyediakan tempat di laboratorium tersebut.
6. Reinhart Caesar, Ivan Agung dan Joshua Pardosi sebagai teman kontrakan penulis yang selalu memberi semangat dan dukungan moral kepada penulis.
 7. Achmad Ibnu Malik Al Chasni, Cynthia Dewi Tejakusuma dan Pradipta Baskara yang telah menemani dan membantu penulis selama perkuliahan semester 7 dan pengerjaan Tugas Akhir ini.
 8. Sirria Panah Alam, Anisa Putri Diana dan Nuzul Ristyantika yang telah membantu penulis selama proses pengerjaan Tugas Akhir ini.
 9. Seluruh mahasiswa Informatika ITS angkatan 2015 yang telah menjadi teman penulis selama menjalani masa kuliah di Informatika ITS.
 10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Januari 2019

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER	xxi
DAFTAR GAMBAR	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan	2
1.4 Tujuan	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur	3
1.6.3 Implementasi Perangkat Lunak.....	3
1.6.4 Pengujian dan Evaluasi.....	4
1.6.5 Penyusunan Buku	4
1.7 Sistematika Penulisan Laporan	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Pengenalan Ekspresi Wajah Manusia.....	7
2.2 Multi Layer Perceptron	8
2.3 Convolutional Neural Network	10
2.3.1 Convolution Layer	11
2.3.2 Pooling Layer.....	11
2.3.3 Fully Connected Layer.....	12
2.3.4 ReLU Activation Function.....	12
2.3.5 Fungsi Softmax	13
2.3.6 Cross Entropy	13
2.3.7 Stochastic Gradient Descent	14
2.3.8 Adagrad.....	14

2.3.9	RMSProp.....	15
2.3.10	Adam.....	15
2.3.11	Dropout	16
2.3.12	Batch Normalization	16
2.3.13	Zero Padding.....	17
2.3.14	Merge Layer Concatenate	17
2.4	Wavelet Transform.....	18
2.5	Augmentasi Data	20
2.5.1	Refleksi Citra	20
2.5.2	Perbesaran Ukuran Gambar	20
2.6	Akurasi, Precision & Recall	21
2.7	Python.....	22
2.8	Keras.....	22
2.9	TensorFlow.....	22
2.10	OpenCV.....	22
2.11	PyWavelets.....	23
2.12	Numpy	23
2.13	Scikit-learn	23
2.14	Matplotlib	23
BAB III PERANCANGAN SISTEM.....		25
3.1	Perancangan Data	25
3.2	Desain Umum Sistem	26
3.2.1	Tahap Praproses Data.....	28
3.2.2	Tahap Pembangunan Arsitektur.....	31
3.2.3	Tahap Pelatihan dan Pengujian CNN.....	35
BAB IV IMPLEMENTASI.....		37
4.1	Lingkungan Implementasi	37
4.1.1	Perangkat Keras	37
4.1.2	Perangkat Lunak	37
4.2	Implementasi Praproses Data	37
4.2.1	Implementasi Perubahan Ukuran Gambar	38
4.2.2	Implementasi Discrete Wavelet Transform	38
4.3	Implementasi Augmentasi Data	39
4.3.1	Refleksi	39
4.3.2	Perbesar Ukuran Gambar	39

4.4	Implementasi Pembangunan Arsitektur	40
4.5	Implementasi Pelatihan dan Evaluasi CNN	44
BAB V UJI COBA DAN EVALUASI.....		47
5.1	Lingkungan Uji Coba.....	47
5.2	Dataset.....	47
5.3	Hasil Praproses.....	48
5.4	Skenario Uji Coba	50
5.4.1	Uji Coba Pembagian Data.....	51
5.4.2	Uji Coba Penggunaan Wavelet.....	53
5.4.3	Uji Coba Parameter CNN	55
5.4.4	Uji Coba Augmentasi Data	58
5.5	Hasil dan Evaluasi.....	60
BAB VI KESIMPULAN DAN SARAN.....		63
6.1	Kesimpulan.....	63
6.2	Saran.....	64
DAFTAR PUSTAKA		65
LAMPIRAN.....		69
L.1	Hasil Uji Coba Individu F01	69
L.2	Hasil Uji Coba Individu F09	72
L.3	Hasil Uji Coba Individu F15	76
L.4	Hasil Uji Coba Individu F18	79
L.5	Hasil Uji Coba Individu F24	83
L.6	Hasil Uji Coba Individu M21	86
L.7	Hasil Uji Coba Individu M23.....	90
L.8	Hasil Uji Coba Individu M24.....	93
L.9	Hasil Uji Coba Individu M26.....	97
L.10	Hasil Uji Coba Individu M30.....	100
BIODATA PENULIS.....		105

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 <i>Confusion matrix</i>	21
Tabel 3.1 Spesifikasi awal dataset.....	26
Tabel 3.2 Arsitektur Wave Convolution Layers.....	34
Tabel 3.3 Arsitektur Convolution Layers setelah merge	35
Tabel 5.1 Spesifikasi dataset setelah proses DWT	49
Tabel 5.2 Spesifikasi dataset setelah proses augmentasi data	50
Tabel 5.3 Parameter awal yang digunakan dalam arsitektur	51
Tabel 5.4 Perbandingan rata-rata lama waktu pelatihan dan akurasi, <i>precision, recall</i> arsitektur CNN pada uji coba pembagian data.	53
Tabel 5.5 Perbandingan lama waktu pelatihan dan akurasi, <i>precision, recall</i> arsitektur CNN pada uji coba penggunaan data <i>wavelet</i>	55
Tabel 5.6 Perbandingan lama waktu pelatihan dan akurasi terbaik, <i>precision, recall</i> arsitektur CNN pada uji coba penggantian parameter CNN	57
Tabel 5.7 Perbandingan lama waktu pelatihan dan akurasi, <i>precision, recall</i> arsitektur CNN pada uji coba augmentasi data	59
Tabel 5.8 Parameter optimal yang ditetapkan	61

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi perubahan ukuran gambar	38
Kode Sumber 4.2 Fungsi <i>Discrete Wavelet Transform</i>	38
Kode Sumber 4.3 Implementasi augmentasi data	39
Kode Sumber 4.4 Fungsi pembangunan konvolusi <i>wavelet</i>	40
Kode Sumber 4.5 Fungsi pembangunan arsitektur CNN	41
Kode Sumber 4.6 Pemanggilan fungsi pembangunan arsitektur CNN	43
Kode Sumber 4.7 Pembuatan index acak data uji	44
Kode Sumber 4.8 Pembagian data latih dan data uji.....	45
Kode Sumber 4.9 Pelatihan CNN.....	45
Kode Sumber 4.10 Pengujian CNN	46
Kode Sumber 4.11 Evaluasi <i>precision</i> dan <i>recall</i>	46

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Struktur umum sistem FER	7
Gambar 2.2 Ilustrasi neuron manusia dalam model matematika [3]	8
Gambar 2.3 Ilustrasi arsitektur <i>Multi Layer Perceptron</i> [3].....	9
Gambar 2.4 Contoh arsitektur <i>Convolutional Neural Network</i> [7]	10
Gambar 2.5 Ilustrasi cara kerja konvolusi [8]	11
Gambar 2.6 Ilustrasi cara kerja <i>Max Pooling</i> [9]	12
Gambar 2.7 <i>ReLU Activation Function</i> [11].....	13
Gambar 2.8 Ilustrasi <i>neural network</i> dalam mengaplikasikan <i>Dropout</i> [3].....	16
Gambar 2.9 Ilustrasi cara kerja <i>Zero Padding</i> [13].....	17
Gambar 2.10 Ilustrasi cara kerja <i>Merge Layer Concatenate</i>	18
Gambar 2.11 Contoh dekomposisi wavelet level 1 sampai 4 [1]	19
Gambar 3.1 Contoh 7 ekspresi berbeda pada dataset KDEF.....	25
Gambar 3.2 Contoh 5 sudut berbeda pada dataset KDEF	26
Gambar 3.3 Diagram alir sistem yang dibangun	27
Gambar 3.4 Diagram alir praproses data tanpa augmentasi	29
Gambar 3.5 Diagram alir praproses data dengan augmentasi	31
Gambar 3.6 Arsitektur CNN yang digunakan	32
Gambar 3.7 Hasil penggabungan <i>layer</i> konvolusi <i>subband</i> dengan <i>Merge Layer Concatenate</i>	34
Gambar 5.1 Gambar asli dan gambar setelah konversi ke <i>grayscale</i> dan <i>resize</i>	48
Gambar 5.2 Hasil proses DWT	49
Gambar 5.3 Gambar asli dan gambar hasil augmentasi	50
Gambar 5.4 Grafik akurasi arsitektur CNN terbaik pada pembagian data berdasarkan sesi	52
Gambar 5.5 Grafik akurasi arsitektur CNN terbaik pada pembagian data berdasarkan individu.....	52
Gambar 5.6 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data <i>wavelet</i>	54

Gambar 5.7 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data tanpa <i>wavelet</i>	54
Gambar 5.8 Perbedaan akurasi pada arsitektur CNN dalam uji coba penggantian <i>optimizer</i> dan <i>learning rate</i>	56
Gambar 5.9 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data asli.....	58
Gambar 5.10 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data asli dan data augmentasi	59
Gambar 5.11 Contoh tampilan ekspresi wajah yang mengalami salah klasifikasi terbanyak yaitu takut dan sedih.....	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Machine learning telah menjadi bagian dari kehidupan sehari-hari bagi banyak orang di seluruh dunia. Penemuan dan implementasi *machine learning* memungkinkan komputer untuk belajar dan memprediksi pola yang mungkin terjadi dan dapat digunakan untuk membantu manusia melakukan kegiatan sehari-hari. Teknologi ini di zaman modern memungkinkan penyelesaian masalah lama dengan cara yang baru dan efisien. Beberapa pengaplikasian *machine learning* meliputi *fraud detection*, *image classification*, *information retrieval* dan *medical diagnosis* [1].

Salah satu pengaplikasian *machine learning* yang populer adalah *image classification*. *Image classification* mengkategorikan piksel-piksel di dalam suatu gambar menjadi satu dari banyak kelas gambar berdasarkan fitur yang berhasil diekstrak dari gambar tersebut [1]. Banyak bidang menggunakan *image classification* untuk meningkatkan kualitas produk, seperti bidang bisnis, finansial, kesehatan, riset, teknologi dan lain-lain. Seiring dengan berkembangnya teknologi, banyak perusahaan, badan riset dan universitas yang terus mengembangkan *machine learning* agar mendapat hasil yang lebih akurat, efisien dan cepat. Dari situlah lahir algoritma *deep learning*, yang merupakan bagian dari *machine learning*.

Convolutional Neural Network (CNN) adalah salah satu *deep neural network* yang cocok digunakan untuk mengolah data yang berbentuk 2 dimensi, seperti gambar dan video. Normalnya, CNN diterapkan untuk mengklasifikasikan gambar yang belum diolah (*raw image*). Di dalam tugas akhir ini, penulis mengusulkan sebuah algoritma untuk mengubah data gambar menjadi *Wavelet Domain* dengan menggunakan *Wavelet Transform*. Tujuannya adalah untuk meningkatkan akurasi *image classification* dengan metode *Convolutional Neural Network*. Data pelatihan dan uji coba diambil dari dataset “Karolinska Directed Emotional Faces”

(KDEF) yang merupakan data yang berisi foto wajah manusia dengan 7 ekspresi berbeda yang nantinya menjadi tujuan *image classification* yang dibuat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana praproses pada data gambar ekspresi wajah manusia menggunakan *Wavelet Transform* sebelum diimplementasikan ke *Convolutional Neural Network*?
2. Bagaimana mengimplementasikan *Convolutional Neural Network* dalam mengenali ekspresi wajah manusia pada data gambar?
3. Bagaimana mengevaluasi kinerja *Convolutional Neural Network* yang telah diimplementasikan?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Data gambar diambil dari dataset Karolinska Directed Emotional Faces (KDEF).
2. Implementasi program menggunakan bahasa pemrograman *Python 3*.
3. Pengenalan ekspresi wajah hanya berupa ekspresi yang berasal dari satu wajah setiap gambarnya.
4. Ada 7 ekspresi wajah manusia yang akan dikenali yaitu takut, marah, jijik, senang, netral, sedih, dan kaget.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membangun sebuah sistem klasifikasi gambar dengan menggunakan praproses *Wavelet Transform* dan metode klasifikasi *Convolutional Neural Network* yang dapat mengenali ekspresi wajah manusia pada data gambar.

1.5 Manfaat

Tugas akhir ini diharapkan dapat membantu menambah kemampuan yang ada pada pengenalan wajah manusia dengan menambah pengenalan ekspresi wajah manusia, meningkatkan kualitas interaksi manusia dan komputer dan dapat digunakan untuk sensor dalam bidang robotika atau *Internet of Things* (IoT) serta kebutuhan-kebutuhan lainnya.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir yang berisi pendahuluan, deskripsi dan gagasan metode-metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri dari latar belakang diajukannya Tugas Akhir, rumusan masalah dan batasan masalah yang ditetapkan, serta manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

1.6.2 Studi Literatur

Pada tahap ini dilakukan pencarian literatur berupa jurnal yang digunakan sebagai referensi untuk pengerjaan tugas akhir ini. Literatur yang dipelajari pada pengerjaan tugas akhir ini berasal dari jurnal ilmiah yang diambil dari berbagai sumber di internet, beserta berbagai literatur online tambahan terkait *Wavelet Transform*, *Convolutional Neural Network*, *TensorFlow* dan *Keras*.

1.6.3 Implementasi Perangkat Lunak

Pada tahap ini akan dilaksanakan implementasi metode dan algoritma yang telah direncanakan. Implementasi sistem

menggunakan *Python 3* sebagai bahasa pemrograman, *TensorFlow* dan *Keras* sebagai *framework*, serta *library* pendukung lainnya.

1.6.4 Pengujian dan Evaluasi

Tahap pengujian dan evaluasi dilakukan menggunakan dataset Karolinska Directed Emotional Faces (KDEF) untuk mengetahui hasil dan performa arsitektur yang telah dibangun. Evaluasi dilakukan dengan metode pengukuran akurasi, *precision*, dan *recall*.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

Bab I Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang *Wavelet Transform* dan *Convolutional Neural Network* dan *library* yang digunakan.

Bab III Perancangan Sistem

Bab ini berisi pembahasan mengenai perancangan dari *Wavelet Transform* dan metode *Convolutional Neural*

Network yang digunakan untuk pengenalan ekspresi wajah manusia pada data gambar.

Bab IV Implementasi

Bab ini membahas implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini membahas tahapan uji coba, kemudian hasil uji coba dievaluasi terhadap kinerja dari sistem yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses dan tertulis saat pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

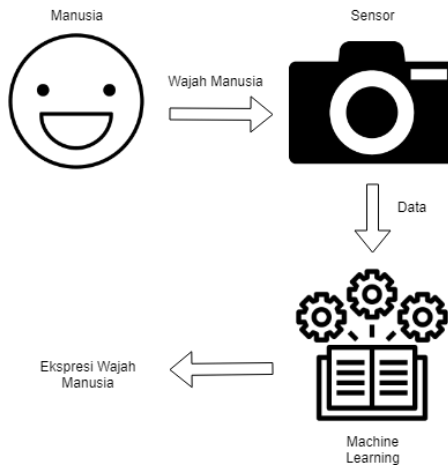
(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai teori-teori dasar yang digunakan dalam Tugas Akhir. Teori-teori tersebut diantaranya adalah *Wavelet Transform* dan *Convolutional Neural Network*, dan beberapa teori lain yang mendukung pembuatan Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran umum dan diharapkan dapat mendukung sistem yang dibangun.

2.1 Pengenalan Ekspresi Wajah Manusia

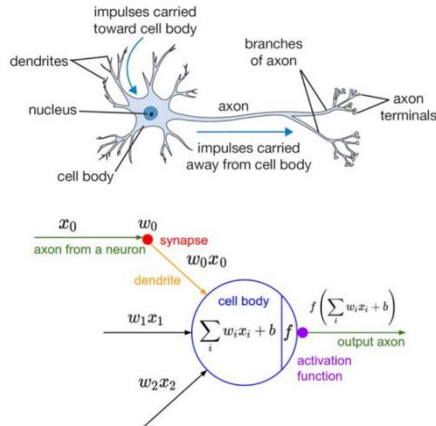
Pengenalan ekspresi wajah manusia atau *Facial Expression Recognition* (FER) adalah kemampuan untuk menafsirkan ekspresi manusia dalam gambar dan menentukan apa ekspresi manusia tersebut, seperti yang terlihat pada Gambar 2.1. Pengenalan ekspresi wajah manusia merupakan bidang penting dalam bidang *Computer Vision* dan penerapannya dapat digunakan di berbagai hal seperti penentuan kesehatan mental, keamanan, deteksi kebohongan, pendeteksi stress, dan lain-lain [2].



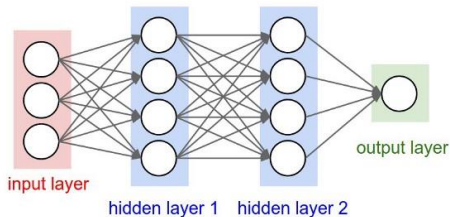
Gambar 2.1 Struktur umum sistem FER

2.2 Multi Layer Perceptron

Multi Layer Perceptron (MLP) atau dikenal dengan *Fully Connected Layer* pada *Convolutional Neural Network* adalah model yang terinspirasi dari bagaimana neuron dalam otak manusia bekerja. Tiap neuron pada otak manusia saling berhubungan dan informasi mengalir dari setiap neuron tersebut. Tiap neuron menerima *input* dan melakukan operasi dot dengan sebuah bobot atau nilai bobot dan menambahkan nilai bias. Hasil dari operasi ini akan dijadikan parameter dari fungsi aktivasi yang akan dijadikan *output* dari neuron tersebut [3]. Ilustrasi dapat dilihat pada Gambar 2.2. MLP terdiri dari *Input Layer*, *Hidden Layer*, *Output Layer*. *Input Layer* menerima masukan (tanpa melalui fungsi aktivasi), kemudian nilai diberikan ke *Hidden Layer*, dimana akan dilakukan perhitungan hasil fungsi aktivasi untuk tiap-tiap neuron, lalu hasilnya diberikan ke *Output Layer*. Ilustrasi arsitektur *Multi Layer Perceptron* terlihat pada Gambar 2.3.



Gambar 2.2 Ilustrasi neuron manusia dalam model matematika [3]



Gambar 2.3 Ilustrasi arsitektur *Multi Layer Perceptron* [3]

Pada tahap pelatihan nilai bobot dan bias pada tiap neuron akan diperbarui terus menerus hingga *output* yang dihasilkan sesuai yang diharapkan. Pada tiap iterasi akan dilakukan proses evaluasi yang biasanya digunakan untuk menentukan kapan harus menghentikan proses pelatihan. Proses pelatihan MLP terdiri dari *Forward* dan *Backward Pass*.

Tahap *Forward Pass* adalah proses dimana data masukan dibawa melewati tiap neuron sampai kepada *Output Layer* yang nantinya akan dihitung nilai *error*. Persamaan (2.1) adalah operasi dot x (*input*) dengan w (bobot) dan ditambah dengan b (bias) yang kemudian menggunakan fungsi aktivasi, misalnya ReLU pada Persamaan (2.2).

$$dot_j = \sum_i^n w_{ij}x_i + b_j \quad (2.1)$$

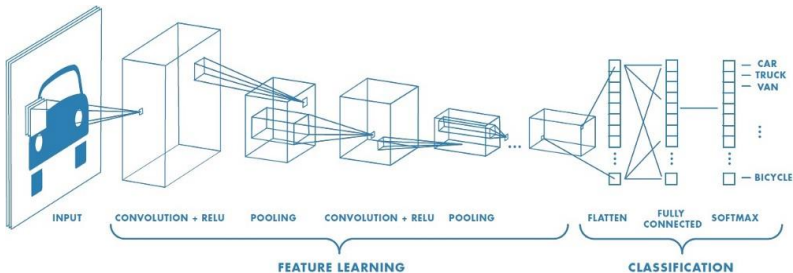
$$h_j = f(dot_j) = \max(0, dot_j) \quad (2.2)$$

Tahap *Backward Pass* bertujuan untuk menyesuaikan kembali tiap bobot dan bias berdasarkan *error* yang didapat pada saat *forward pass*. Proses ini disebut sebagai propagasi balik (*backpropagation*) yaitu tahap pelatihan yang mengubah *weight* neuron-neuron di dalam MLP. Propagasi balik memanfaatkan sebuah *loss function* untuk menghitung *error* dari nilai prediksi dengan nilai sebenarnya. Selanjutnya, digunakan algoritma pengoptimalan untuk memperbarui bobot dan bias dengan tujuan menurunkan nilai *error*, hal ini dilakukan secara iteratif sampai *epoch* tertentu.

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu algoritma dari *deep learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN sering digunakan untuk mengenali citra benda atau pemandangan, melakukan deteksi dan segmentasi objek [4].

Penelitian awal yang mendasari penemuan ini dilakukan oleh Hubel dan Wiesel [5] yang melakukan penelitian visual korteks pada indera penglihatan kucing. Penelitian ini sangat berguna dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan model-model baru. Arsitektur dari CNN dibagi menjadi 2 bagian besar, *Feature Learning / Extraction Layer* dan *Classification Layer* [6], seperti yang dipaparkan pada Gambar 2.4.



Gambar 2.4 Contoh arsitektur *Convolutional Neural Network* [7]

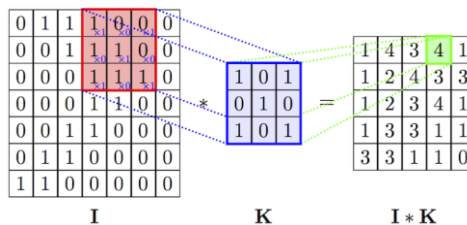
Feature Learning / Extraction Layer adalah bagian dimana terjadi proses penerjemahan dari sebuah citra menjadi *features*. *Features* ini berupa angka-angka yang merepresentasikan citra tersebut, yaitu berupa *feature map*. Proses ini terdiri dari *Convolutional Layer* dan *Pooling Layer*.

Classification Layer adalah dimana *feature map* yang dihasilkan dari *convolutional layers* masih berbentuk array multidimensi, sehingga harus dilakukan pengubahan *feature map* menjadi sebuah *feature vector* agar bisa digunakan sebagai

masuk dari *fully connected layer*. *Fully connected layer* yang dimaksud disini adalah *Multi Layer Perceptron* (MLP) yang memiliki beberapa *hidden layer*, *activation function*, *output layer* dan *loss function*.

2.3.1 Convolution Layer

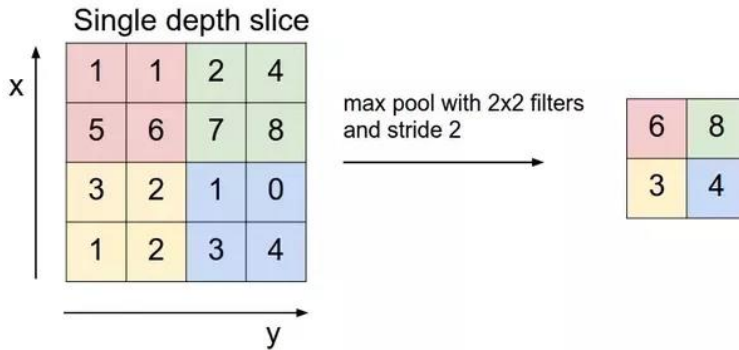
Convolution Layer melakukan operasi konvolusi pada output dari lapisan sebelumnya. Konvolusi adalah istilah matematis yang artinya mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstrak fitur dari citra masukan [4]. Ilustrasi cara kerja konvolusi bisa dilihat pada Gambar 2.5, dimana I adalah citra, K adalah *filter* atau *kernel* yang digunakan, $I * K$ adalah hasil operasi konvolusi.



Gambar 2.5 Ilustrasi cara kerja konvolusi [8]

2.3.2 Pooling Layer

Fungsi dari *Pooling Layer* adalah mereduksi ukuran dari data. Terdapat beberapa tipe *Pooling Layer* diantaranya yaitu *max*, *average*, *sum* dan lainnya. Metode *Pooling* dalam CNN yang biasa digunakan adalah *Max Pooling* & *Average Pooling*. *Max Pooling* membagi *output* dari *Convolution Layer* menjadi beberapa matriks kecil lalu mengambil nilai maksimal dari tiap matriks untuk menyusun matriks citra yang telah direduksi, sedangkan *Average Pooling* akan memilih nilai rata-ratanya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun obyek citra mengalami translasi. Ilustrasi cara kerja *Max Pooling* bisa dilihat pada Gambar 2.6.



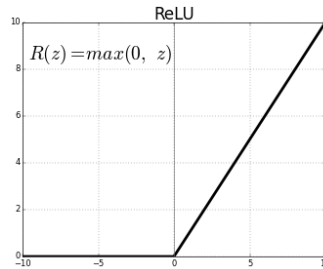
Gambar 2.6 Ilustrasi cara kerja *Max Pooling* [9]

2.3.3 Fully Connected Layer

Fully Connected Layer dalam penerapannya sama dengan *Multi Layer Perceptron* (MLP) yang bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. *Feature map* dari *Convolution Layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu yang disebut *feature vector* sebelum dapat dimasukkan ke dalam sebuah *Fully Connected Layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, *Fully Connected Layer* diimplementasikan di akhir jaringan [10].

2.3.4 ReLU Activation Function

Fungsi aktivasi berfungsi untuk menentukan apakah neuron tersebut harus aktif atau tidak berdasarkan nilai masukan. Salah satu contoh fungsi aktivasi adalah ReLU (Rectified Linear Unit) dimana fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai masukan, dimana seluruh nilai yang kurang dari nol akan dijadikan nol, seperti pada Gambar 2.7.



Gambar 2.7 ReLU Activation Function [11]

2.3.5 Fungsi Softmax

Fungsi *softmax* biasa digunakan dalam klasifikasi banyak kelas. *Softmax* memberikan nilai probabilitas untuk setiap label kelas, dimana jumlah seluruh probabilitas adalah 1. *Softmax* pada dasarnya adalah probabilitas eksponensial yang dinormalisasi dari nilai masukan sejumlah kelas pada model klasifikasi seperti pada Persamaan (2.3).

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (2.3)$$

Dimana y adalah nilai masukan. Operasi akan menghasilkan nilai probabilitas. Label dari data masukan akan ditentukan berdasarkan kelas dengan nilai probabilitas tertinggi.

2.3.6 Cross Entropy

Loss function merupakan fungsi yang menggambarkan kerugian yang dihasilkan oleh model. *Loss function* dikatakan baik, ketika menghasilkan *error* yang diharapkan paling rendah. Pada permasalahan klasifikasi banyak kelas, *cross entropy* adalah *loss function* yang biasa digunakan. *Cross entropy* akan menghitung *error* antara nilai prediksi S dengan nilai sebenarnya T , seperti pada Persamaan (2.4). Selanjutnya, nilai *error* akhir diambil dari rata-rata hasil *cross entropy*, seperti pada Persamaan (2.4) dan (2.5).

$$D(S_i, T_i) = -\sum_j T_{ij} \log S_{ij} \quad (2.4)$$

$$J(W, b) = \frac{1}{n} \sum_i D(S_i, T_i) \quad (2.5)$$

2.3.7 Stochastic Gradient Descent

Ketika melatih sebuah model, dibutuhkan sebuah *loss function* yang dapat mengukur kualitas dari setiap bobot atau parameter tertentu. *Stochastic Gradient Descent* (SGD) adalah algoritma pengotimalan. Tujuan pengoptimalan adalah untuk menemukan parameter yang dapat meminimalkan nilai *error* dari *loss function*. SGD adalah algoritma yang digunakan untuk memperbarui nilai bobot dan bias pada neuron di *neural network*. Pada dasarnya operasi yang dilakukan hanya mengurangi bobot awal dengan sebagian nilai dari nilai gradien yang sudah kita dapat. Nilai sebagian disini diwakili oleh parameter bernama *learning rate*, seperti yang terlihat pada Persamaan (2.6) dan (2.7).

$$w_{j+1} = w_j - \alpha \frac{\partial}{\partial w_j} J(W, b) \quad (2.6)$$

$$b_{j+1} = w_j - \alpha \frac{\partial}{\partial b_j} J(W, b) \quad (2.7)$$

2.3.8 Adagrad

Adagrad adalah algoritma pengoptimalan berbasis gradien yang memperbarui *learning rate* setiap parameternya. Adagrad adalah algoritma yang digunakan untuk memperbarui nilai bobot dan bias pada neuron di *neural network*. Adagrad menggunakan *adaptive learning rate* untuk setiap parameter. Berbeda dengan *Stochastic Gradient Descent* (SGD) yang selalu menggunakan *learning rate* yang sama, Adagrad memiliki sebuah *learning rate* untuk setiap parameter dan secara terpisah beradaptasi saat proses pelatihan. Pembaruan dilakukan untuk tiap parameter $\theta(\mathbf{j})$ dengan gradien *loss function* $\mathbf{g}(\mathbf{j})$, seperti yang dapat dilihat pada Persamaan (2.8) dan (2.9).

$$g_j = \frac{\partial}{\partial \theta_j} J(\theta_j) \quad (2.8)$$

$$\theta_{j+1} = \theta_j - \frac{\alpha}{\sqrt{\sum_{i=0}^j (g_i)^2}} g_j \quad (2.9)$$

2.3.9 RMSProp

RMSProp (*Root Mean Square*) adalah metode pengoptimalan berbasis *adaptive learning rate* yang diusulkan oleh Geoffrey Hinton [12]. RMSProp memodifikasi Adagrad dengan mengganti akumulasi gradien menjadi rata-rata bergerak gradien yang diberi bobot secara kuadratik, seperti yang dapat dilihat pada Persamaan (2.10) dan (2.11).

$$s_j = \beta \cdot s_{j-1} + (1 - \beta)(g_j)^2 \quad (2.10)$$

$$\theta_{j+1} = \theta_j - \frac{\alpha}{\sqrt{s_j + \epsilon}} g_j \quad (2.11)$$

2.3.10 Adam

Adam (*Adaptive Moment Estimation*) juga adalah algoritma pengoptimalan yang dapat digunakan. Hampir sama dengan Adagrad, Adam memiliki sebuah *learning rate* untuk setiap parameter dan secara terpisah beradaptasi saat proses pelatihan. Adam memperbarui nilai setiap parameter seperti RMSProp [13]. Perbedaannya Adam menggunakan gradien yang telah diperhalus dan semakin mengecil seperti yang dapat dilihat pada Persamaan (2.12). Lalu gradien tersebut akan digunakan untuk memperbarui parameter, seperti yang dapat dilihat pada Persamaan (2.13) dan (2.14).

$$m_j = \beta_1 \cdot m_{j-1} + (1 - \beta_1) \cdot g_j \quad (2.12)$$

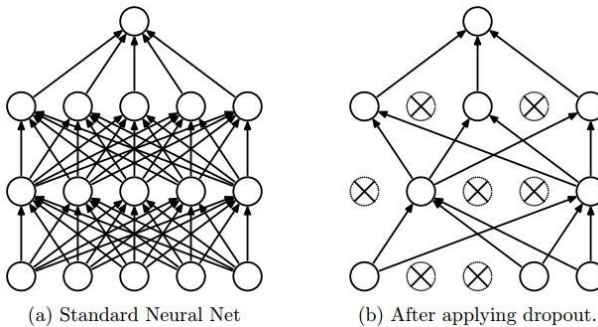
$$s_j = \beta_2 \cdot s_{j-1} + (1 - \beta_2)(g_j)^2 \quad (2.13)$$

$$\theta_{j+1} = \theta_j - \frac{\alpha}{\sqrt{s_j + \epsilon}} m_j \quad (2.14)$$

2.3.11 Dropout

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses learning. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden layer* maupun *visible layer* di dalam jaringan [14]. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak.

Pada Gambar 2.8, (a) neuron tetap utuh pada *neural network* yang belum memakai *Dropout*, dan (b) *neural network* yang sebagian dari neuronnya tidak digunakan setelah diaplikasikan *Dropout*.



Gambar 2.8 Ilustrasi *neural network* dalam mengaplikasikan *Dropout* [3]

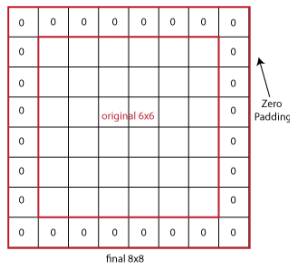
2.3.12 Batch Normalization

Batch Normalization adalah teknik melakukan normalisasi terhadap *batch* atau kumpulan data masukan, seperti yang terlihat pada Persamaan (2.15). Dimana x adalah nilai masukan, μ_b adalah *mean* dari *batch*, σ_b adalah standar deviasi dari *batch*. Normalisasi dilakukan agar data memiliki *mean* mendekati 0 dan standar deviasi mendekati 1.

$$x_{baru} = \frac{x - \mu_b}{\sigma_b} \quad (2.15)$$

2.3.13 Zero Padding

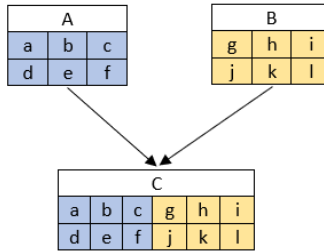
Padding atau *Zero Padding* adalah teknik penambahan nilai 0 di sisi-sisi data masukan. Teknik ini digunakan untuk memanipulasi dimensi *output* dari *Convolution Layer*. Dengan menggunakan *padding*, dimensi *output* dapat tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara signifikan. Sehingga kita bisa memanfaatkan *Convolution Layer* lebih dalam dan lebih banyak *features* yang berhasil didapatkan. Ilustrasi cara kerja *Zero Padding* bisa dilihat pada Gambar 2.9.



Gambar 2.9 Ilustrasi cara kerja *Zero Padding* [13]

2.3.14 Merge Layer Concatenate

Merge layer digunakan untuk menggabungkan dua *layer* atau lebih. *Merge layer* dapat dimanfaatkan untuk menggabungkan beberapa jaringan menjadi satu. Salah satu contoh *merge layer* adalah *Concatenate*. *Concatenate* menyatukan beberapa layer inputnya menjadi satu layer secara serial. Ilustrasi cara kerja *Merge Layer Concatenate* bisa dilihat pada Gambar 2.10.



Gambar 2.10 Ilustrasi cara kerja *Merge Layer Concatenate*

2.4 Wavelet Transform

Wavelet Transform adalah metode transformasi yang mengadopsi metode *Fourier Transform* dan *Short Time Fourier Transform* (STFT). Seperti STFT, *Wavelet Transform* mentransformasi signal dalam domain waktu menjadi signal dalam domain waktu dan frekuensi (yang dalam hal ini dibentuk menjadi domain *translation and scale*) [4].

Discrete Wavelet Transform (DWT) berasal dari dan menyederhanakan *Continuous Wavelet Transform* (CWT). DWT mewakili urutan sampel angka dari fungsi *continuous* [1]. Menerapkan DWT ke gambar digital, terutama pada gambar yang memiliki resolusi besar, menghasilkan berbagai kegunaan untuk praproses dan peningkatan performa. Fungsi matematika yang efisien ini memiliki kerangka kerja yang sangat intuitif untuk karakterisasi dan penyimpanan gambar. DWT membagi (*dekomposisi*) gambar awal menjadi 4 sub gambar baru. Setiap sub gambar berukuran $\frac{1}{4}$ kali dari gambar asli.

Misalkan sebuah gambar memiliki dimensi $M \times N$, maka DWT 2 dimensi bisa ditulis dengan Persamaan (2.16) dan (2.17).

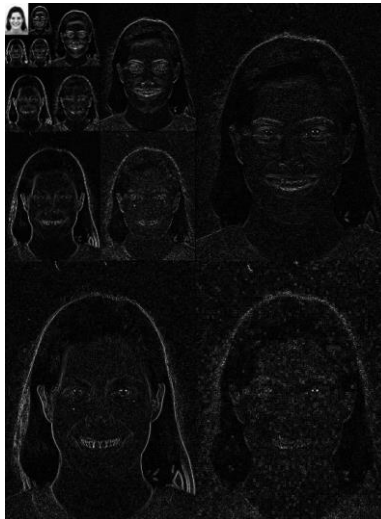
$$W_{\varphi}(j_0, m, n) = \frac{1}{\sqrt{M \cdot N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (2.16)$$

$$W_{\psi}^i(j, m, n) = \frac{1}{\sqrt{M \cdot N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad (2.17)$$

Sedangkan untuk *Inverse Discrete Wavelet Transform* (IDWT) dapat ditulis dengan Persamaan (2.18).

$$f(x, y) = \frac{1}{\sqrt{M \cdot N}} \sum_m \sum_n W_{\varphi}(j_0, m, n) \varphi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{M \cdot N}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_{\psi}^i(j, m, n) \psi_{j, m, n}^i(x, y) \quad (2.18)$$

dimana W_{φ} adalah koefisien aproksimasi, W_{ψ} adalah koefisien detail, m dan n adalah dimensi *subband*, j adalah level resolusi, dan i adalah set *subband* (H, V, D) [1]. Gambar 2.11 menunjukkan contoh dekomposisi *wavelet* pada level 1 sampai 4.



Gambar 2.11 Contoh dekomposisi wavelet level 1 sampai 4 [1]

Sub gambar pada posisi atas kanan (*Low-High* atau LH), bawah kiri (*High-Low* atau HL), dan bawah kanan (*High-High* atau HH) akan tampak seperti versi kasar dari citra asli karena berisi komponen frekuensi tinggi dari citra asli. Sedangkan untuk 1 sub gambar atas kiri (*Low-Low* atau LL) tampak seperti citra asli dan tampak lebih halus karena berisi komponen frekuensi rendah dari citra asli. Sub gambar LL mengandung koefisien aproksimasi sedangkan sub gambar lainnya mengandung koefisien detail.

Sifat independen dari setiap sub gambar membuat sistem klasifikasi gambar dapat melakukan klasifikasi secara optimal pada masing-masing sub gambar jika diperlukan. Setelah proses dekomposisi selesai, IDWT dapat digunakan untuk merekonstruksi gambar.

2.5 Augmentasi Data

Dalam mendapatkan performa yang optimal, *deep learning* seperti *Convolutional Neural Network* membutuhkan data yang lebih banyak dibandingkan dengan algoritma yang lain, untuk itu kita perlu melakukan augmentasi data. Augmentasi data adalah sebuah teknik menambah data dengan cara memanipulasi data yang telah ada dengan pengaturan keragaman tertentu. Untuk data berupa citra, kita bisa lakukan operasi seperti translasi, refleksi, perbesaran ukuran, dan lain-lain.

2.5.1 Refleksi Citra

Refleksi citra atau pencerminan terdapat 2 macam, yaitu horizontal dan vertikal. Refleksi horizontal adalah pencerminan pada sumbu-Y (koordinat kartesian) dari citra A menjadi citra B seperti pada Persamaan (2.19) [16].

$$B[x][y] = A[-x][y] \quad (2.19)$$

2.5.2 Perbesaran Ukuran Gambar

Perbesaran ukuran gambar adalah proses memperbesar ukuran gambar secara acak untuk memperoleh gambar yang lebih

variatif. Gambar diperbesar pada salah satu titiknya untuk dicari keragaman lain dari gambar.

2.6 Akurasi, Precision & Recall

Ketika membangun sebuah model klasifikasi, pertanyaan yang muncul adalah bagaimana mengetahui seberapa baik model tersebut. Mengevaluasi model klasifikasi dilakukan dengan mencari tahu seberapa baik hasil prediksi dari model tersebut. *Recall* di Persamaan (2.20), *precision* di Persamaan (2.21), dan akurasi di Persamaan (2.22) adalah metode pengukuran yang biasa digunakan dalam mengevaluasi model, penjelasan variabel ada pada *confusion matrix* pada Tabel 2.1.

Tabel 2.1 *Confusion matrix*

		Kelas Prediksi	
		Benar	Salah
Kelas sebenarnya	Benar	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
	Salah	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Keterangan :

1. *True Positive* (TP) pada gambar dikenali ekspresi manusia, pada keluaran program mengenali ekspresi tersebut.
2. *True Negative* (TN) pada gambar tidak dikenali ekspresi manusia, pada keluaran program tidak mengenali ekspresi tersebut.
3. *False Positive* (FP) pada gambar dikenali ekspresi manusia, pada keluaran program tidak mengenali ekspresi tersebut.
4. *False Negative* (FN) pada gambar tidak dikenali ekspresi manusia, pada keluaran program mengenali ekspresi tersebut.

$$Recall = TP / (TP + FN) \quad (2.20)$$

$$Precision = TP / (TP + FP) \quad (2.21)$$

$$Akurasi = (TP + TN) / (TP + FP + TN + FN) \quad (2.22)$$

2.7 Python

Python adalah bahasa pemrograman yang populer. *Python* sering dimanfaatkan dalam pengembangan web, perangkat lunak, penelitian, dan *system scripting*. Python dapat digunakan untuk menangani data besar dan melakukan operasi matematika yang kompleks. Python bekerja di berbagai *platform* seperti Windows, Mac, Linux, Raspberry Pi, dan lain-lain. Python dirancang untuk mudah dibaca, yaitu memiliki sintaks yang sederhana dan menggunakan bahasa Inggris [17].

2.8 Keras

Keras adalah *high-level neural networks API*, yang ditulis dalam bahasa pemrograman Python dan mampu berjalan di atas TensorFlow dan Theano. Keras dikembangkan dalam rangka memungkinkan eksperimen dilakukan dengan cepat. Keras dapat berjalan baik di CPU dan GPU. Keras berisi banyak implementasi *neural network* yang umum digunakan, fungsi aktivasi, *optimizer*, dan *tool* lain yang memudahkan dalam pengolahan citra dan data teks [18].

2.9 TensorFlow

TensorFlow adalah *library open source* untuk pembuatan program yang membutuhkan komputasi numerik berkinerja tinggi. TensorFlow dikembangkan oleh tim Google Brain. TensorFlow menyediakan fungsi-fungsi *machine learning* dan *deep learning*, dan dapat dijalankan dalam CPU atau GPU [19].

2.10 OpenCV

OpenCV (*Open Source Computer Vision*) adalah *library* yang dimanfaatkan dalam pengolahan citra dinamis secara *real-time*. OpenCV dapat digunakan dalam berbagai bahasa pemrograman seperti Python, C++, Java, atau MATLAB. OpenCV memiliki fitur seperti *Feature & Object Detection*, *Motion Analysis and Object Tracking*, *Image Filtering*, *Image Processing*, dan lain-lain [20].

2.11 PyWavelets

PyWavelets adalah *library wavelet transform open source* yang tersedia untuk bahasa pemrograman *Python*. PyWavelets menggabungkan interface tingkat tinggi yang sederhana dengan bahasa C level rendah dan Cython. PyWavelet dapat digunakan untuk melakukan banyak operasi *wavelet* seperti *Forward and Inverse Discrete Wavelet Transform, Multilevel Discrete Wavelet Transform, Stationary Wavelet transform, Continuous Wavelet Transform* dan lain-lain [21].

2.12 Numpy

Numpy adalah *library Python* yang mendukung pengolahan data pada *array* dan matriks multidimensi yang besar. Numpy menyediakan kumpulan fungsi matematika, seperti aljabar linear, transformasi Fourier, pembuatan angka acak, dan lain-lain. *Numpy* bersifat *open source* sehingga banyak dimanfaatkan dalam pengolahan data penelitian [22].

2.13 Scikit-learn

Scikit-learn adalah *open source machine learning library* untuk bahasa pemrograman *Python*. Scikit-learn menyediakan fitur seperti *classification, regression, clustering*, termasuk juga didalamnya algoritma *support vector machines, random forest, gradient boosting*, dan lain-lain [23].

2.14 Matplotlib

Matplotlib adalah *library Python* yang mendukung pembuatan grafik dua dimensi dalam berbagai format dan dari berbagai jenis data. Matplotlib bersifat *open source* dan banyak digunakan untuk pengolahan data dalam penelitian. Matplotlib dapat membuat plot, histogram, spektrum daya, diagram batang, diagram kesalahan, plot pencah, dan lain-lain [24].

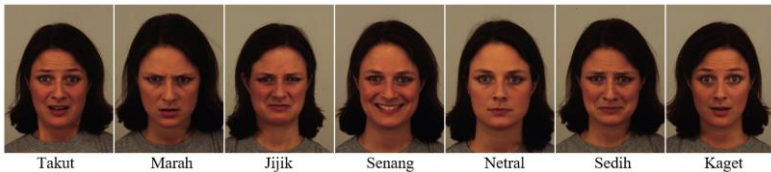
(Halaman ini sengaja dikosongkan)

BAB III PERANCANGAN SISTEM

Bab ini menjelaskan mengenai perancangan data dan sistem pengenalan ekspresi wajah manusia menggunakan *Convolutional Neural Network*. Bab ini juga akan menjelaskan gambaran umum sistem dalam bentuk diagram alir.

3.1 Perancangan Data

Data yang digunakan sebagai masukan awal dari sistem pengenalan ekspresi wajah manusia menggunakan *Convolutional Neural Network* adalah data Karolinska Directed Emotional Faces (KDEF). KDEF adalah dataset yang merupakan 4900 gambar berupa foto 35 orang pria dan 35 orang wanita, masing-masing dengan 7 ekspresi yang berbeda yaitu takut, marah, jijik, senang, netral, sedih, dan kaget. Contoh setiap ekspresi bisa dilihat pada Gambar 3.1.



Gambar 3.1 Contoh 7 ekspresi berbeda pada dataset KDEF

Setiap ekspresi diambil dari 5 sudut pandang yaitu depan, serong kiri, serong kanan, samping kiri dan samping kanan. Contoh setiap sudut gambar bisa dilihat pada Gambar 3.2. Dataset ini juga diambil dalam 2 sesi sehingga setiap individu akan muncul sebanyak 2 kali.



Gambar 3.2 Contoh 5 sudut berbeda pada dataset KDEF

Dari 4900 gambar dataset KDEF ini, akan diambil 3900 data untuk data latih dan 1000 data untuk data uji. Data latih diproses menggunakan *Convolutional Neural Network* untuk membangun model, kemudian model digunakan dalam pengenalan terhadap data uji dan kinerja model diukur dengan akurasi, *precision*, dan *recall* pengujian. Spesifikasi lengkap dataset dapat dilihat pada Tabel 3.1.

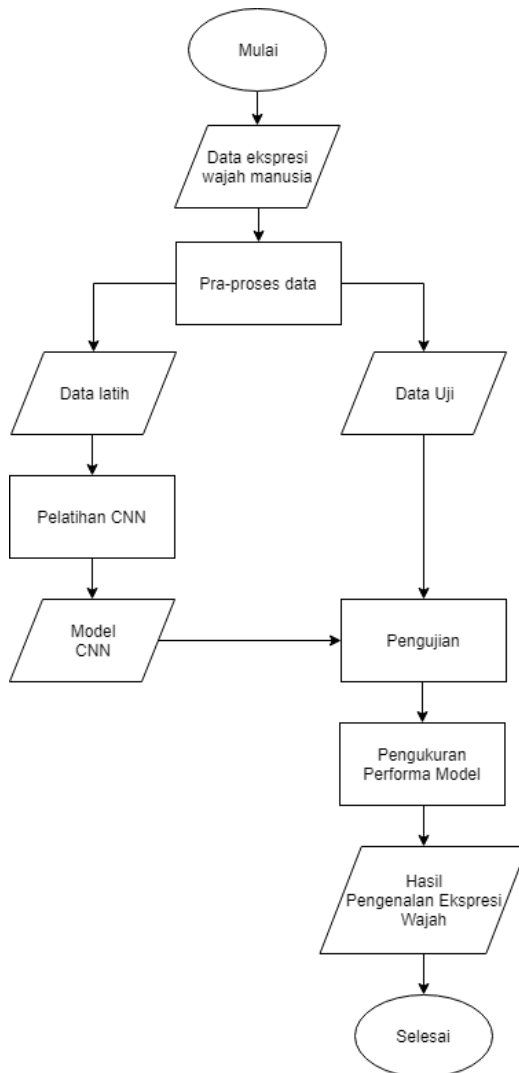
Tabel 3.1 Spesifikasi awal dataset

Keterangan	Spesifikasi
Ukuran resolusi asli	562 x 762 pixels
Ekstensi	.JPG
Jumlah gambar	4900
Jumlah orang	35 Pria, 35 Wanita
Jumlah sesi foto	2 kali
Jumlah kelas	7 kelas
Jumlah gambar per kelas	700
Jumlah sudut pandang per kelas	5 sudut pandang
Ukuran file	80-100 KB
Kanal warna	3 (RGB)

3.2 Desain Umum Sistem

Sistem pengenalan ekspresi manusia yang dibangun memiliki proses utama diantaranya praproses data, pelatihan dan

pengujian *Convolutional Neural Network*. Diagram alir dari sistem ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram alir sistem yang dibangun

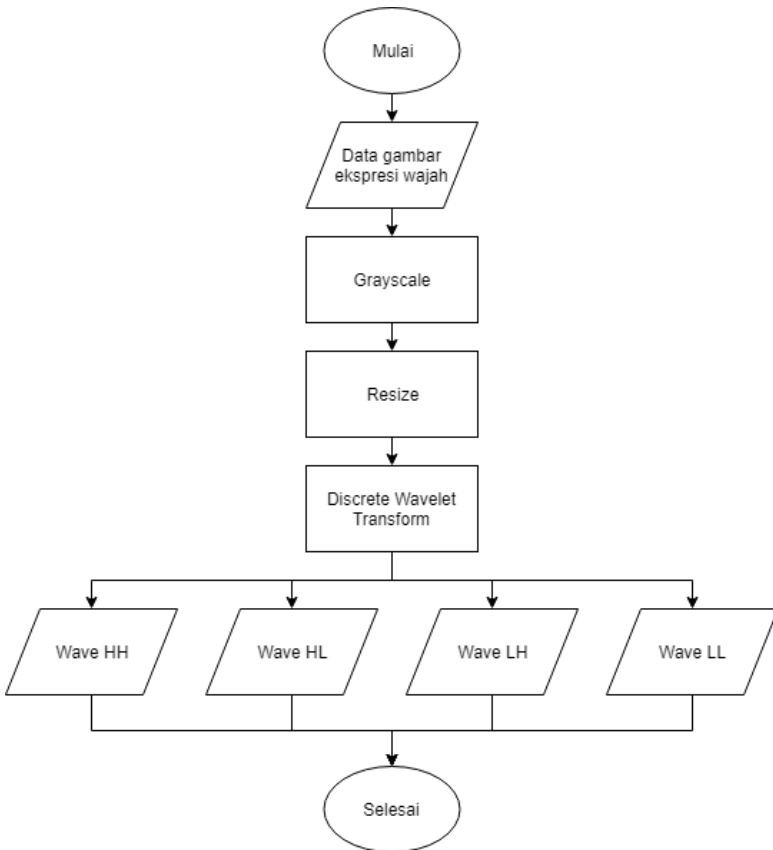
Akan dilakukan praproses data terlebih dahulu sebelum data digunakan sebagai data untuk melatih model. Setiap data akan dijadikan format *grayscale*, lalu diresize menjadi ukuran 256x256 dan dilakukan proses *Discrete Wavelet Transform* (DWT).

Proses pelatihan adalah proses pembuatan model klasifikasi ekspresi wajah manusia. Data latih yang telah dilakukan praproses data akan diekstraksi fiturnya yakni melalui proses konvolusi, *pooling*, dan fungsi-fungsi aktivasi. Selanjutnya, hasil dari proses pelatihan tersebut akan menjadi model untuk proses klasifikasi. Data uji juga akan dilakukan praproses data dahulu sebelum diekstraksi fiturnya untuk menjadi *input* dalam proses pengujian.

Proses pengujian memanfaatkan fungsi *softmax* untuk mengetahui label kelas dengan menghitung nilai probabilitas tertinggi. Selanjutnya prediksi tersebut akan dibandingkan dengan label kelas sebenarnya, maka dapat dievaluasi nilai akurasi, *precision*, dan *recall* dari model tersebut.

3.2.1 Tahap Praproses Data

Pada tugas akhir ini, praproses data yang dilakukan adalah melakukan perubahan format gambar menjadi *grayscale*, perubahan ukuran gambar, lalu melakukan proses *Discrete Wavelet Transform* (DWT) pada gambar. Diagram alir praproses data dapat dilihat pada Gambar 3.4.



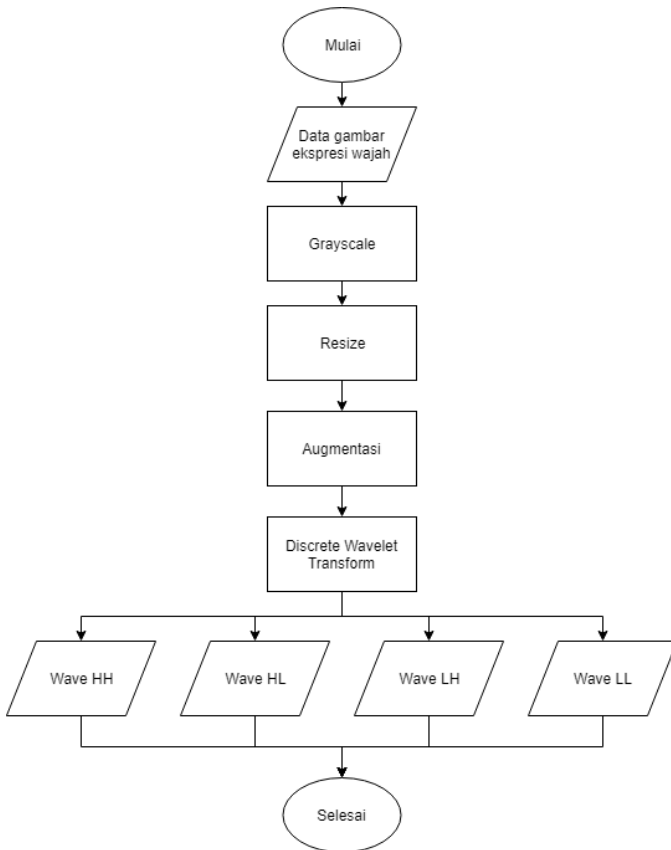
Gambar 3.4 Diagram alir praproses data tanpa augmentasi

Setiap data akan diubah dahulu menjadi format *grayscale*. Lalu data akan diresize menjadi ukuran 256x256. Perubahan ukuran dilakukan untuk mempercepat proses komputasi.

Lalu akan dilakukan proses DWT yang akan mengubah data menjadi *wavelet domain*. Ukuran dari gambar juga akan berkurang menjadi $\frac{1}{4}$ nya yaitu 128x128 setelah proses DWT ini. Proses DWT akan menghasilkan 4 *subband* yang terdiri dari LL, HL, LH,

dan HH yang nantinya keempat *subband* ini akan dijadikan input dari jaringan yang telah dirancang.

Selain praproses yang telah dijelaskan di atas, akan dilakukan juga proses augmentasi data untuk memperbanyak data latih yang dimasukkan ke arsitektur CNN. Proses augmentasi data dilakukan untuk memperbanyak varian data. Banyaknya varian data diharapkan dapat meningkatkan kualitas data dan model yang dibangun dapat memiliki hasil yang lebih baik. Augmentasi data yang dilakukan adalah mengolah data gambar asli menjadi 2 gambar baru. Augmentasi yang dilakukan adalah dengan melakukan refleksi horizontal pada gambar dan perbesaran ukuran gambar. Diagram alir praproses data dengan augmentasi dapat dilihat pada Gambar 3.5.

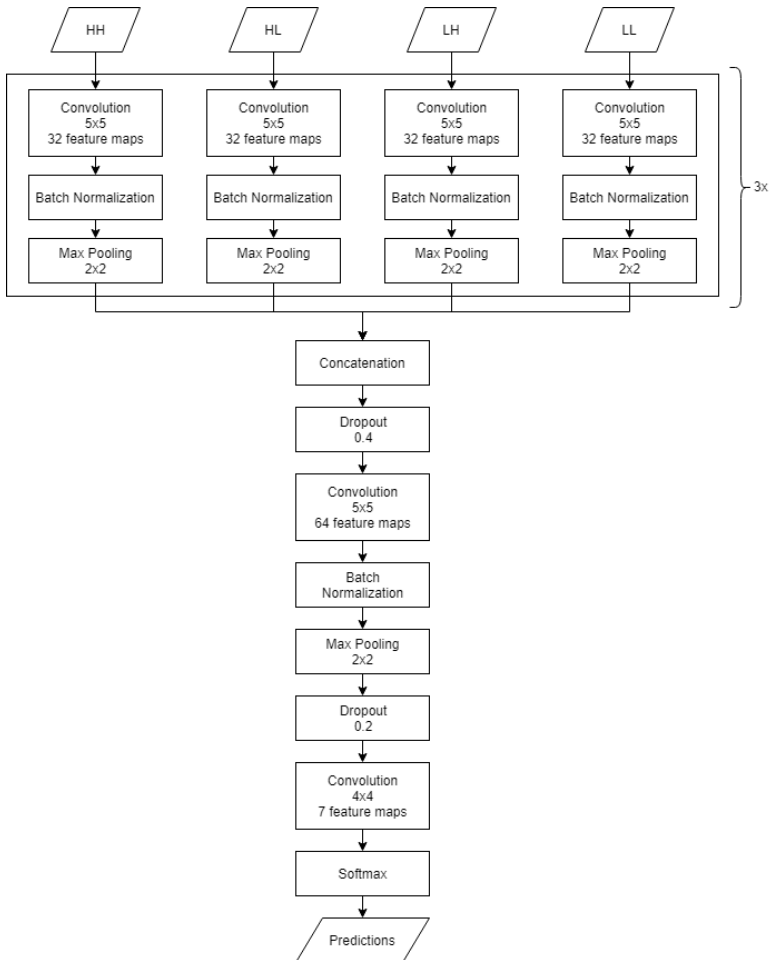


Gambar 3.5 Diagram alir praproses data dengan augmentasi

3.2.2 Tahap Pembangunan Arsitektur

Pembangunan model bertujuan untuk menyiapkan *layer*, fungsi aktivasi, *loss function*, dan parameter apa saja yang dibutuhkan. CNN (*Convolutional Neural Network*) yang digunakan terdiri dari *convolutional layer*, *max pooling layer* dan *fully connected layer*. *Dropout layer* juga digunakan untuk

menghindari *overfitting*. Arsitektur CNN dapat dilihat pada Gambar 3.6.



Gambar 3.6 Arsitektur CNN yang digunakan

Berikut detail arsitektur CNN:

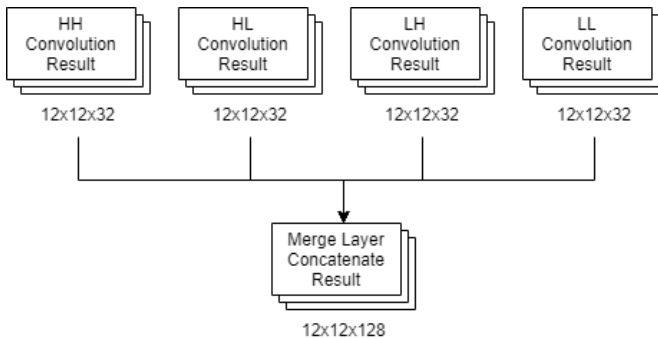
1. Input untuk arsitektur CNN adalah 4 *subband* dari DWT yaitu HH, HL, LH, dan LL.
2. Setiap input *subband* akan melewati 3 *Convolution Layer*, 3 *Batch Normalization* dan 3 *Max Pooling Layer* secara paralel, lalu akan digabungkan dengan *Merge Layer Concatenate*.
3. Semua *Convolution Layer* yang berada sebelum *Merge Layer Concatenate* memiliki filter berjumlah 32 dan berukuran 5x5.
4. Setelah *Merge Layer Concatenate*, data akan melewati 2 *Convolution Layer*, 2 *Dropout layer*, 1 *Batch Normalization* dan 1 *Max Pooling Layer* lalu diikuti dengan sebuah *Softmax output layer*.
5. Setelah *Merge Layer Concatenate*, *Convolution Layer* memiliki filter berjumlah 64 dengan ukuran 5x5 dan 7 dengan ukuran 4x4 secara berurutan.
6. *Dropout layer* memiliki nilai probabilitas 0.4 dan 0.2 secara berurutan.
7. Semua *Max Pooling Layer* memiliki *kernel* berukuran 2x2 dan *stride* berukuran 2x2.
8. CNN dilatih menggunakan *Adam Optimizer* dengan *learning rate* 0,0001.
9. Data yang digunakan memiliki 1 kanal warna (*grayscale*)

Pembangunan arsitektur CNN dapat dibagi menjadi 2 tahap. Tahap pertama yaitu pembangunan arsitektur untuk mengkonvolusi setiap *subband* dari DWT yang nantinya akan digabung menjadi 1 layer dengan *Merge Layer Concatenate*. Detail *layers* dan parameter dapat dilihat pada Tabel 3.2.

Tabel 3.2 Arsitektur Wave Convolution Layers

<i>Layer</i>	<i>Input</i>	<i>Output</i>	<i>Spesifikasi</i>
<i>Input Layer</i>	(1, 128, 128, 1)		-
<i>Convolution Layer 1</i>	(1, 128, 128, 1)	(1, 124, 124, 32)	<i>Filter: 5 x 5 – 32</i> <i>Stride: 1 x 1</i> <i>Fungsi: ReLU</i>
<i>Max Pooling Layer 1</i>	(1, 124, 124, 32)	(1, 62, 62, 32)	<i>Kernel: 2 x 2</i> <i>Stride: 2 x 2</i>
<i>Convolution Layer 2</i>	(1, 62, 62, 32)	(1, 58, 58, 32)	<i>Filter: 5 x 5 – 32</i> <i>Stride: 1 x 1</i> <i>Fungsi: ReLU</i>
<i>Max Pooling Layer 2</i>	(1, 58, 58, 32)	(1, 29, 29, 32)	<i>Kernel: 2 x 2</i> <i>Stride: 2 x 2</i>
<i>Convolution Layer 3</i>	(1, 29, 29, 32)	(1, 25, 25, 32)	<i>Filter: 5 x 5 – 32</i> <i>Stride: 1 x 1</i> <i>Fungsi: ReLU</i>
<i>Max Pooling Layer 3</i>	(1, 25, 25, 32)	(1, 12, 12, 32)	<i>Filter: 2 x 2</i> <i>Stride: 2 x 2</i>
<i>Merge Layer Concatenate</i>	(1, 12, 12, 32) x 4	(1, 12, 12, 128)	-

Lalu setelah digabung dengan *Merge Layer Concatenate*, data akan melewati 2 *Convolution Layer* lagi sebelum nantinya diaktivasi dengan fungsi *Softmax*. Hasil penggabungan setiap *layer* konvolusi *subband* dengan *Merge Layer Concatenate* dapat dilihat pada Gambar 3.7. Detail *layers* dan parameter setelah merge dapat dilihat pada Tabel 3.3.



Gambar 3.7 Hasil penggabungan *layer* konvolusi *subband* dengan *Merge Layer Concatenate*

Tabel 3.3 Arsitektur Convolution Layers setelah merge

<i>Layer</i>	<i>Input</i>	<i>Output</i>	<i>Spesifikasi</i>
<i>Merge Layer Concatenate</i>	(1, 12, 12, 128)		-
<i>Dropout Layer 1</i>	(1, 12, 12, 128)	(1, 12, 12, 128)	Fungsi: <i>Dropout 0.4</i>
<i>Convolution Layer 1</i>	(1, 12, 12, 128)	(1, 8, 8, 64)	<i>Filter: 5 x 5 – 64</i> <i>Stride: 1 x 1</i> Fungsi: ReLU
<i>Max Pooling Layer 1</i>	(1, 8, 8, 64)	(1, 4, 4, 64)	<i>Kernel: 2 x 2</i> <i>Stride: 2 x 2</i>
<i>Dropout Layer 2</i>	(1, 4, 4, 64)	(1, 4, 4, 64)	Fungsi: <i>Dropout 0.2</i>
<i>Convolution Layer 2</i>	(1, 4, 4, 64)	(1, 1, 1, 7)	<i>Filter: 4 x 4 – 7</i> <i>Stride: 1 x 1</i> Fungsi: ReLU
<i>Fully Connected Layer</i>	(1, 1, 1, 7)	Jumlah kelas	Fungsi: <i>Softmax</i>

3.2.3 Tahap Pelatihan dan Pengujian CNN

Setelah melakukan praproses terhadap data dan menyiapkan arsitektur. Data akan dibagi menjadi 2, yaitu data latih dan data uji dengan jumlah data latih sebanyak 3900 gambar dan jumlah data uji sebanyak 1000 gambar. Dikarenakan data gambar setiap orang akan muncul 2 kali dalam dataset, maka pembagian dilakukan dengan cara memasukkan semua data gambar sesi pertama (sesi A) ke data pelatihan, lalu data sesi kedua (sesi B) akan diambil secara acak sebanyak 1000 gambar untuk dimasukkan ke data uji dan sisanya akan masuk ke data latih. Pembagian ini dilakukan untuk memastikan semua individu dan semua sudut pandang gambar yang ada di dataset pernah masuk ke model setidaknya 1 kali sehingga data terbagi dengan seimbang. Juga akan dicoba skenario pembagian data dengan membagi data berdasarkan individu yang difoto.

Proses pelatihan memanfaatkan data latih untuk membangun model CNN. Pelatihan menggunakan Adam *Optimizer* dengan *learning rate* yang sudah ditentukan sebelumnya. Proses pelatihan akan dijalankan pada *batch size* 39 dan jumlah *epoch* menyesuaikan skenario uji coba. Dalam setiap akhir *epoch* terdapat

proses pengujian model terhadap data uji untuk mengetahui seberapa baik model dilatih. Lalu pada akhir pelatihan akan dilakukan pengujian untuk mendapatkan nilai akurasi, *precision* dan *recall*.

BAB IV IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi perangkat lunak dari rancangan sistem yang telah dibahas pada Bab 3 meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Dalam mengimplementasikan aplikasi pengenalan ekspresi manusia diperlukan beberapa perangkat pendukung sebagai berikut.

4.1.1 Perangkat Keras

Implementasi tugas akhir ini menggunakan desktop *personal computer* (PC) HP-580-011d. Sistem operasi yang digunakan adalah Windows 10 64-bit. PC yang digunakan memiliki spesifikasi Intel Core i7-7700 dengan kecepatan 3,6 GHz, *Random Access Memory* (RAM) sebesar 16 GB, dan mempunyai *Graphics Processing Unit* (GPU) yaitu NVIDIA GeForce GTX 1060 sebesar 3 GB.

4.1.2 Perangkat Lunak

PC dari sisi perangkat lunak memiliki spesifikasi antara lain menggunakan bahasa pemrograman Python 3.6, dilengkapi dengan *library* antara lain OpenCV, Tensorflow, Keras, PyWavelets, Numpy, Matplotlib dan Scikit-learn.

4.2 Implementasi Praproses Data

Pada subbab ini akan dijabarkan implementasi pada tahap praproses data, yaitu perubahan ukuran gambar dan *Discrete Wavelet Transform* (DWT).

4.2.1 Implementasi Perubahan Ukuran Gambar

Proses perubahan ukuran gambar diimplementasikan pada Kode Sumber 4.1. Proses dilakukan pada gambar dengan memanfaatkan fungsi *resize* dari OpenCV. Dimana parameter yang digunakan adalah *img* sebagai input gambar yang akan diresize dan *size* sebagai ukuran gambar setelah diubah berupa *tuple* beranggota 2 angka yang mewakili panjang dan lebar, seperti (256, 256).

```
1. size = (256, 256)
2. resized_img = cv2.resize(img, size)
```

Kode Sumber 4.1 Fungsi perubahan ukuran gambar

Data diresize menjadi 256x256 agar nantinya setelah dilakukan proses DWT ukuran data akan menjadi $\frac{1}{4}$ nya yaitu 128x128 sesuai dengan ukuran input arsitektur CNN.

4.2.2 Implementasi Discrete Wavelet Transform

Proses perubahan gambar dari *raw image* ke *Discrete Wavelet Transform* (DWT) diimplementasikan pada Kode Sumber 4.2. Proses dilakukan pada gambar dengan memanfaatkan fungsi *wavedec2* dari PyWavelets. Dimana parameter yang digunakan adalah *resized_img* sebagai input gambar yang akan dilakukan proses DWT, *wavelet* sebagai jenis DWT yang akan digunakan yaitu *haar wavelet*, dan *level* sebagai kedalaman *wavelet* yang digunakan.

```
1. wp = pywt.wavedec2(data = resized_img, wavelet = 'haar', level = 1)
```

Kode Sumber 4.2 Fungsi *Discrete Wavelet Transform*

4.3 Implementasi Augmentasi Data

Pada subbab ini akan dijelaskan proses augmentasi data yaitu refleksi dan memperbesar ukuran gambar.

4.3.1 Refleksi

Refleksi yang dilakukan disini adalah refleksi horizontal. Refleksi diimplementasikan pada Kode Sumber 4.3. Proses refleksi horizontal dilakukan dengan menggunakan fungsi *ImageDataGenerator* dari library Keras. Parameter *horizontal_flip=True* menandakan refleksi horizontal akan diimplementasikan pada data gambar. Lalu setelah objek *ImageDataGenerator* dibuat, akan dipanggil fungsi *flow* untuk menerapkan pengaturan augmentasi yang telah dibuat.

4.3.2 Perbesar Ukuran Gambar

Perbesaran ukuran gambar dilakukan untuk mencari keragaman yang lebih variatif dari data. Perbesaran ukuran gambar diimplementasikan pada Kode Sumber 4.3. Proses perbesaran ukuran gambar dilakukan dengan menggunakan fungsi *ImageDataGenerator* dari library Keras. Parameter *zoom_range=0.1* menandakan perbesaran ukuran gambar sebesar 10% akan diimplementasikan pada data gambar secara acak. Lalu setelah objek *ImageDataGenerator* dibuat, akan dipanggil fungsi *flow* untuk menerapkan pengaturan perbesaran ukuran gambar yang telah dibuat.

```
1. datagen = ImageDataGenerator(horizontal_flip=True,
    zoom_range=0.1, fill_mode='nearest')
2. datagen.flow(image, save_to_dir=save_dir,
    save_prefix=fileName, save_format='JPG')
```

Kode Sumber 4.3 Implementasi augmentasi data

4.4 Implementasi Pembangunan Arsitektur

Pada subbab ini akan dijabarkan implementasi fungsi-fungsi pada tahap pembangunan model. Arsitektur CNN dimulai dengan mengimplementasi desain *convolution layer* keempat *subband*. Implementasi tersebut tercantum pada Kode Sumber 4.4 dengan fungsi *wave_convolution_layer* dengan layer-layer sesuai dengan spesifikasi yang dijelaskan pada Bab 3. Fungsi ini akan mengkonvolusi semua *subband* yang menjadi masukan model. Lalu, implementasi dilanjutkan dengan menggabungkan 4 hasil konvolusi *subband* tersebut dengan sebuah *Concatenation Layer* yang tercantum pada Kode Sumber 4.5. Lalu pada tahap terakhir, hasil *Concatenation Layer* akan dikonvolusi lagi dengan 2 *convolutional layer* sesuai yang dijelaskan pada Bab 3 sebelum akhirnya akan dimasukkan ke fungsi aktivasi *Softmax*. Pembangunan arsitektur memanfaatkan Keras dan Tensorflow.

```
1. def wave_convolution_layer(inputs, size):
2.     conv_layer = Conv2D(32, (5,5), strides=(1,1),
3.         activation='relu',
4.         kernel_initializer='he_normal')(inputs)
5.     conv_layer = BatchNormalization()(conv_layer)
6.     conv_layer = MaxPooling2D((2, 2))(conv_layer)
7.     conv_layer = Conv2D(32, (5,5), strides=(1,1),
8.         activation='relu',
9.         kernel_initializer='he_normal')(conv_layer)
10.    conv_layer = BatchNormalization()(conv_layer)
11.    conv_layer = MaxPooling2D((2, 2))(conv_layer)
12.    return conv_layer
```

Kode Sumber 4.4 Fungsi pembangunan konvolusi *wavelet*

Pada baris 1, parameter *inputs* adalah *input layer* dan parameter *size* adalah dimensi data masukan yakni 128x128 (*image*)

length, image width). Baris 2 adalah inialisasi arsitektur yang menggabungkan *input layer* tersebut dengan *layer* yang lain. Baris 2-10 masing-masing adalah penambahan *layer* yang dibutuhkan.

```

1. def model_build():
2.     inputs1 = Input(shape=(127, 127, 1))
3.     conv_layer1 =
4.         wave_convolution_layer(inputs1, 127)
5.     inputs2 = Input(shape=(128, 128, 1))
6.     conv_layer2 =
7.         wave_convolution_layer(inputs2, 128)
8.     inputs3 = Input(shape=(129, 129, 1))
9.     conv_layer3 =
10.        wave_convolution_layer(inputs3, 129)
11.     inputs4 = Input(shape=(130, 130, 1))
12.     conv_layer4 =
13.        wave_convolution_layer(inputs4, 130)
14.     concat =
15.        keras.layers.Concatenate()([conv_layer1,
16.        conv_layer2, conv_layer3, conv_layer4])
17.     conv_layer = Dropout(0.4)(concat)
18.     conv_layer = Conv2D(64, (5,5),
19.        strides=(1,1), activation='relu',
20.        kernel_initializer='he_normal')(conv_layer)
21.     print(conv_layer._keras_shape)
22.     conv_layer =
23.        BatchNormalization()(conv_layer)
24.     conv_layer = Activation('relu')(conv_layer)
25.     conv_layer = MaxPooling2D((2,2))(conv_layer)
26.     conv_layer = Dropout(0.2)(conv_layer)
27.     conv_layer = Conv2D(7, (4,4), strides=(1,1),
28.        activation='relu',
29.        kernel_initializer='he_normal')(conv_layer)
30.     flatten = Flatten()(conv_layer)
31.     outputs = Dense(7,
32.        activation='softmax')(flatten)
33.     model = Model(inputs=[inputs1, inputs2,
34.        inputs3, inputs4], outputs=outputs)
35.     return model

```

Kode Sumber 4.5 Fungsi pembangunan arsitektur CNN

Pada baris 2-9 dilakukan inisialisasi *input layer* untuk setiap *subband* dengan menggunakan fungsi *wave_convolution_layer* yang telah disebutkan pada Kode Program 4.4. Baris 10 adalah penggabungan 4 *convolution layer* setiap *subband* dengan menggunakan *Concatenation Layer*. Baris 11-21 masing-masing adalah penambahan *layer* yang dibutuhkan.

Arsitektur yang dibangun terdiri dari 6 macam *layer* antara lain Convolution2D, MaxPooling2D, BatchNormalization, Flatten, Dropout, dan Dense.

Convolution2D adalah implementasi *Convolution Layer* dalam bentuk 2 dimensi. Convolution2D menggunakan 6 parameter. Berikut penjelasan parameter-parameter tersebut:

1. Parameter *filters* adalah jumlah *filter* atau *kernel*.
2. Parameter *kernel_size* adalah ukuran *filter* atau *kernel*.
3. Parameter *strides* adalah ukuran *stride* yang digunakan.
4. Parameter *activation* adalah fungsi aktivasi apa yang digunakan, terdapat banyak pilihan fungsi aktivasi yang disediakan oleh Keras, salah satunya adalah 'relu' yakni fungsi ReLU.
5. Parameter *kernel_initializer* untuk menentukan weights awal matrix.
6. Parameter *padding* menentukan apakah *output* diaplikasikan *Zero Padding*. Jika *padding* bernilai 'valid' maka *output* tidak diaplikasikan *Zero Padding* dan jika bernilai 'same' maka bentuk keluaran akan diaplikasikan *Zero Padding* untuk menjaga keluaran sama dengan *input*. Nilai default *padding* jika tidak dicantumkan adalah 'valid'.

MaxPooling2D adalah implementasi *Max Pooling Layer* dalam bentuk 2 dimensi. MaxPooling2D menggunakan 2 parameter. Berikut penjelasan parameter-parameter tersebut:

1. Parameter *pool_size* adalah ukuran *kernel Pooling Layer*.

2. Parameter *strides* adalah ukuran *stride* yang digunakan. Nilai default *strides* jika tidak dicantumkan adalah *None* yang berarti ukuran *stride* akan disamakan dengan *pool_size*.

Flatten biasa digunakan oleh arsitektur CNN pada Keras ketika *Convolution* dan *Max Pooling Layer* akan memasuki *Fully Connected Layer*. Hal ini karena Flatten memiliki fungsi untuk merubah keluaran sebelumnya yang berupa *feature map* menjadi *feature vector*, sehingga dapat dijadikan *input* pada *Fully Connected Layer*.

Dropout adalah implementasi teknik *dropout* yang berfungsi membuat beberapa neuron yang dipilih secara acak untuk tidak dipakai selama proses pelatihan. Hal ini dilakukan untuk mengurangi *overfitting*. Dropout memiliki parameter *rate* yakni nilai probabilitas yang dipakai dalam menentukan secara acak neuron.

Dense adalah implementasi *Fully Connected Layer*. Dense menggunakan 2 parameter. Berikut penjelasan parameter-parameter tersebut:

1. Parameter *units* adalah jumlah neuron.
2. Parameter *activation* adalah fungsi aktivasi apa yang digunakan. Terdapat banyak pilihan fungsi aktivasi yang disediakan oleh Keras, ada *'relu'* atau fungsi ReLU dan *'softmax'* atau fungsi *softmax* yang biasa digunakan pada akhir arsitektur untuk mengklasifikasikan label kelas.

Pemanggilan fungsi pembangunan CNN diimplementasikan pada Kode Sumber 4.6.

```
1. model = model_build()
```

Kode Sumber 4.6 Pemanggilan fungsi pembangunan arsitektur CNN

4.5 Implementasi Pelatihan dan Evaluasi CNN

Setelah pembangunan arsitektur *Convolutional Neural Network*, dilakukan proses pelatihan dan pengujian. Data akan dibagi menjadi 2, yaitu data latih dan data uji dengan jumlah data latih sebanyak 3900 gambar dan jumlah data uji sebanyak 1000 gambar. Dikarenakan data gambar setiap orang akan muncul 2 kali dalam dataset, maka pembagian dilakukan dengan cara memasukkan semua data gambar sesi pertama (sesi A) ke data pelatihan, lalu data gambar sesi kedua (sesi B) akan diambil secara acak sebanyak 1000 gambar untuk dimasukkan ke data uji dan sisanya akan masuk ke data latih. Pembagian ini dilakukan untuk memastikan semua orang yang ada di dataset pernah masuk ke model setidaknya 1 kali sehingga data terbagi dengan seimbang.

Pembagian data dilakukan dengan cara menggunakan seluruh gambar dari sesi A yaitu gambar dari index 0 sampai 2449. Lalu sesi B akan diambil datanya secara acak yaitu antara index 2450 sampai 4899 seperti yang diimplementasikan pada Kode Sumber 4.7.

```
1. def random_index(start,end):
2.     x = [i for i in range(start,end)]
3.     random.shuffle(x)
4.     return x
5.
6. random_index = random_index(2450,4900)
```

Kode Sumber 4.7 Pembuatan index acak data uji

Lalu setelah index acak sesi B dibuat, akan diambil seluruh gambar dari sesi A dan 1450 gambar dari sesi B berdasarkan index acak tersebut seperti yang diimplemtasikan pada Kode Sumber 4.8.

```

1. for i in range(2450):
2.     train_x.append(image[i][0])
3.     train_y.append(image[i][1])
4.
5. for i in range(len(random_index)):
6.     if i < 1450:
7.         train_x.append(image[random_index[i]][0])
8.         train_y.append(image[random_index[i]][1])
9.     else:
10.        test_x.append(image[random_index[i]][0])
11.        test_y.append(image[random_index[i]][1])

```

Kode Sumber 4.8 Pembagian data latihan dan data uji

Proses pelatihan memanfaatkan data latihan untuk membangun model CNN. Pelatihan menggunakan *loss function* berupa *Cross Entropy* dan *optimizer* berupa *Adam Optimizer* dengan *learning rate* yang telah ditentukan sebelumnya. Hal ini diimplementasikan pada Kode Sumber 4.9 di baris 1 dan 2. Proses pelatihan akan dijalankan pada *batch size* 39 dan jumlah *epoch* 500. Dalam setiap akhir *epoch* terdapat proses pengujian model terhadap data uji dan didapatkan nilai akurasi, *precision* dan *recall*. Pada baris 3 dilakukan proses pelatihan dengan memanggil fungsi *fit* dari Keras. Dimana *batch_size* adalah ukuran *batch*, *epochs* adalah jumlah *epoch* yang akan dijalankan, *validation_data* adalah data uji dan label kelasnya.

```

1. adam = Adam(lr=0.0001)
2. model.compile(optimizer=adam, loss=
   'categorical_crossentropy', metrics=['accuracy'])
3. model.fit(train_x, train_y, batch_size=39, epochs=
   500, verbose=1, validation_data=(test_x, test_y))

```

Kode Sumber 4.9 Pelatihan CNN

Setelah proses pelatihan selesai, model akan diuji dengan menggunakan data uji. Hal ini diimplementasikan pada Kode Sumber 4.10. Fungsi *evaluate* akan mengevaluasi model dengan data uji dan labelnya.

```
1. print(model.evaluate(test_x, test_y))
```

Kode Sumber 4.10 Pengujian CNN

Pada fungsi *evaluate*, didapatkan nilai akurasi model terhadap data uji. Namun library Keras ini tidak memiliki fitur untuk menghitung *precision* dan *recall*. Sehingga perlu dilakukan cara alternatif untuk melakukan evaluasi terhadap model agar mendapatkan nilai *precision* dan *recall*. Dengan memanfaatkan fungsi *predict_classes* dari library Keras terhadap *test_x* (data uji). Hal ini diimplementasikan pada Kode Sumber 4.11. Pada baris 7, fungsi *predict_classes* digunakan untuk mendapatkan nilai prediksi untuk masing-masing kelas dan disimpan pada variabel *pred_y*. Pada baris 1-5, dibuat sebuah fungsi *return_to_label* untuk mengembalikan bentuk data prediksi dari Keras yang masih berupa tipe kelas kategorikal menjadi tipe kelas yang numerik. Lalu pada baris 9-10, fungsi *return_to_label* digunakan untuk mengembalikan *pred_y* dan *test_y* ke tipe kelas numerik. Baris 12 memanfaatkan fungsi *classification_report* dari library Scikit-learn untuk mengevaluasi nilai prediksi (*pred*) terhadap nilai sesungguhnya (*test*) untuk masing-masing kelas sehingga didapatkan nilai *precision* dan *recall*.

```
1. def return_to_label(y):
2.     label = []
3.     for i in range(len(y)):
4.         label.append(np.argmax(y[i]))
5.     return label
6.
7. pred_y = model.predict(test_x)
8.
9. test = return_to_label(test_y)
10. pred = return_to_label(pred_y)
11.
12. print(classification_report(test, pred))
```

Kode Sumber 4.11 Evaluasi *precision* dan *recall*

BAB V

UJI COBA DAN EVALUASI

Bab ini akan membahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

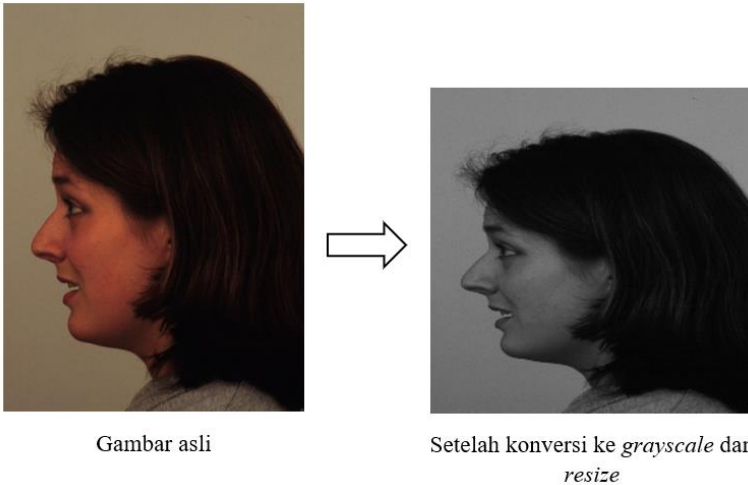
Lingkungan uji coba pada tugas akhir ini adalah sebuah desktop *personal computer* (PC) HP-580-011d. Sistem operasi yang digunakan adalah Windows 10 64-bit. PC yang digunakan memiliki spesifikasi perangkat keras Intel Core i7-7700 dengan kecepatan 3,6 GHz, *Random Access Memory* (RAM) sebesar 16 GB, dan mempunyai *Graphics Processing Unit* (GPU) yaitu NVIDIA GeForce GTX 1060 sebesar 3 GB. Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan bahasa pemrograman Python 3.6 dilengkapi dengan *library* antara lain Keras, Tensorflow, PyWavelets, OpenCV, Numpy, Matplotlib dan Scikit-learn.

5.2 Dataset

Pada tugas akhir ini, data yang digunakan adalah data Karolinska Directed Emotional Faces (KDEF). Dataset KDEF mengambil gambar setiap individu sebanyak 2 kali. Pengambilan sesi pertama (sesi A) dan kedua (sesi B) diambil dengan interval waktu 1 jam. Dikarenakan data gambar setiap individu akan muncul 2 kali dalam dataset, maka pembagian dilakukan dengan cara memasukkan semua data gambar sesi A ke data pelatihan, lalu data sesi B akan diambil secara acak sebanyak 1000 gambar untuk dimasukkan ke data uji dan sisanya akan masuk ke data latih. Pembagian ini dilakukan untuk memastikan semua individu yang ada di dataset pernah masuk ke model setidaknya 1 kali sehingga data terbagi dengan seimbang.

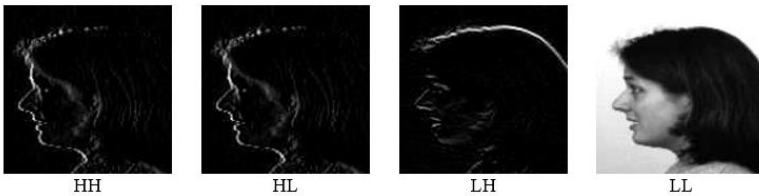
5.3 Hasil Praproses

Sebelum memasuki proses pelatihan dengan arsitektur CNN yang telah dirancang sebelumnya, akan dilakukan praproses berupa perubahan data menjadi format *grayscale* dan *resize* data menjadi ukuran 256x256. Hasil konversi *grayscale* dan *resize* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Gambar asli dan gambar setelah konversi ke *grayscale* dan *resize*

Setelah itu akan dilakukan proses *Discrete Wavelet Transform* (DWT) terhadap data. DWT akan menghasilkan 4 *subband* yang nantinya akan menjadi input dari arsitektur CNN. Hasil proses DWT dapat dilihat pada Gambar 5.2. Spesifikasi dataset setelah proses DWT data dapat dilihat pada Tabel 5.1.

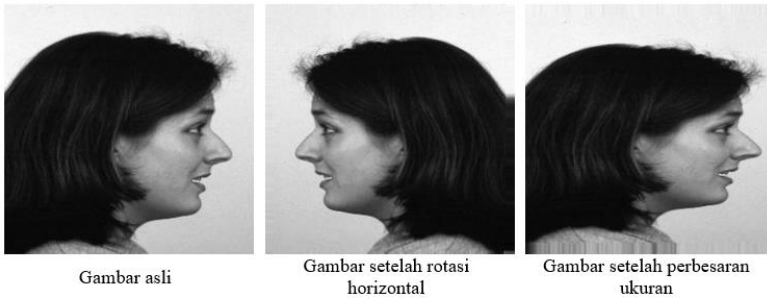


Gambar 5.2 Hasil proses DWT

Tabel 5.1 Spesifikasi dataset setelah proses DWT

Keterangan	Spesifikasi
Ukuran resolusi	128 x 128
Ekstensi	.JPG
Jumlah gambar	4900
Jumlah orang	35 Pria, 35 Wanita
Jumlah sesi foto	2 kali
Jumlah kelas	7 kelas
Jumlah gambar per kelas	700
Jumlah sudut pandang per kelas	5 sudut pandang
Ukuran file	2-3 KB
Kanal warna	1 (<i>grayscale</i>)

Setelah itu akan dilakukan proses augmentasi data dengan melakukan refleksi horizontal pada gambar dan perbesaran ukuran gambar. Augmentasi akan menghasilkan 2 gambar baru yang nantinya akan digabung dengan gambar asli untuk dijadikan data latih dari arsitektur CNN. Hasil augmentasi data dapat dilihat pada Gambar 5.3. Spesifikasi dataset setelah proses augmentasi data dapat dilihat pada Tabel 5.2.



Gambar 5.3 Gambar asli dan gambar hasil augmentasi

Tabel 5.2 Spesifikasi dataset setelah proses augmentasi data

Keterangan	Spesifikasi
Ukuran resolusi	128 x 128
Ekstensi	.JPG
Jumlah gambar	11700
Jumlah orang	35 Pria, 35 Wanita
Jumlah sesi foto	2 kali
Jumlah kelas	7 kelas
Jumlah gambar per kelas	2100
Jumlah sudut pandang per kelas	5 sudut pandang
Ukuran file	2-3 KB
Kanal warna	1 (<i>grayscale</i>)

5.4 Skenario Uji Coba

Proses uji coba berguna untuk menemukan parameter yang menghasilkan performa model yang paling optimal. Parameter yang tepat akan memberikan hasil yang lebih baik pada saat proses uji coba. Hasil terbaik dari suatu skenario uji coba akan digunakan untuk skenario uji coba berikutnya. Ada 4 macam skenario uji coba dan semuanya akan dicoba pada arsitektur CNN yang telah dirancang. Skenario uji coba yang akan dilakukan yaitu:

1. Uji Coba Pembagian Data
2. Uji Coba Penggunaan *Wavelet*
3. Uji Coba Parameter CNN
4. Uji Coba Augmentasi Data

Tabel 5.3 berisi parameter-parameter awal arsitektur CNN yang digunakan dan dapat berubah di setiap uji coba yang dilakukan. Pada setiap skenario uji coba akan ditetapkan nilai parameter yang dapat meningkatkan kinerja arsitektur.

Tabel 5.3 Parameter awal yang digunakan dalam arsitektur

Keterangan	Parameter
Jumlah <i>epoch</i>	500
Ukuran <i>batch</i>	39
<i>Optimizer</i>	<i>Adam Optimizer</i>
<i>Loss function</i>	<i>Categorical Cross Entropy</i>
<i>Learning rate</i>	0,0001

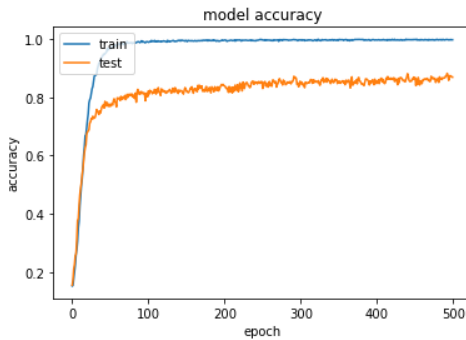
5.4.1 Uji Coba Pembagian Data

Uji coba pertama adalah uji coba pembagian data. Uji coba ini dilakukan untuk menguji kemampuan model saat wajah seseorang belum pernah memasuki model sebelumnya. Akan ada 2 jenis pembagian data, yaitu:

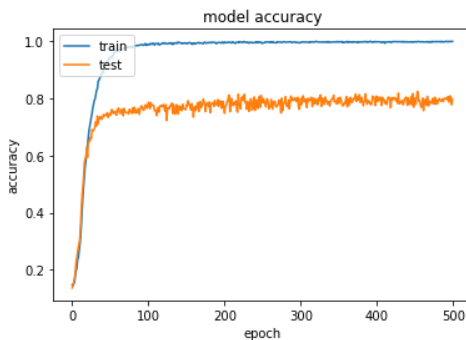
1. Pembagian data dengan membagi data berdasarkan individu yang menjadi model dataset. Dari 35 pria dan 35 wanita yang menjadi model dataset, akan diambil 29 individu dari setiap jenis kelamin secara acak untuk dijadikan data latih dan sisanya akan dijadikan data uji. Dengan demikian, individu data uji tidak pernah melatih model.
2. Pembagian data dengan memasukkan semua dataset sesi A yang mana berarti semua individu pernah masuk ke dalam model lalu mengambil secara acak dataset sesi B untuk digunakan sebagai data uji. Dataset sesi B yang tidak digunakan sebagai data uji akan dijadikan data latih.

Pengambilan data secara acak pada kedua jenis pembagian data akan dilakukan sebanyak 5 kali. Rata-rata hasil akan dijadikan perbandingan untuk menentukan pembagian data yang paling baik. Hasil pembagian data yang paling baik akan dibawa ke skenario berikutnya untuk dijadikan parameter masukan.

Uji coba pembagian data masing-masing menghasilkan grafik akurasi pengujian yang dapat dilihat pada Gambar 5.4 dan Gambar 5.5. Perbandingan nilai akurasi, *precision*, *recall* pengujian serta lama waktu pelatihan dapat dilihat pada Tabel 5.4.



Gambar 5.4 Grafik akurasi arsitektur CNN terbaik pada pembagian data berdasarkan sesi



Gambar 5.5 Grafik akurasi arsitektur CNN terbaik pada pembagian data berdasarkan individu

Tabel 5.4 Perbandingan rata-rata lama waktu pelatihan dan akurasi, *precision*, *recall* arsitektur CNN pada uji coba pembagian data

Pembagian Data	Lama waktu pelatihan	Akurasi	Kelas	Data uji	<i>Precision</i>	<i>Recall</i>
Berdasarkan Sesi	8901 detik	84,68%	<i>Afraid</i>	121	77%	66%
			<i>Angry</i>	143	89%	83%
			<i>Disgust</i>	155	85%	83%
			<i>Happy</i>	147	94%	97%
			<i>Neutral</i>	135	84%	95%
			<i>Sad</i>	149	77%	75%
			<i>Surprised</i>	150	84%	92%
Berdasarkan Individu	8917 detik	74,74%	<i>Afraid</i>	140	63%	55%
			<i>Angry</i>	140	79%	67%
			<i>Disgust</i>	140	82%	75%
			<i>Happy</i>	140	91%	92%
			<i>Neutral</i>	140	66%	85%
			<i>Sad</i>	140	68%	63%
			<i>Surprised</i>	140	77%	84%

5.4.2 Uji Coba Penggunaan Wavelet

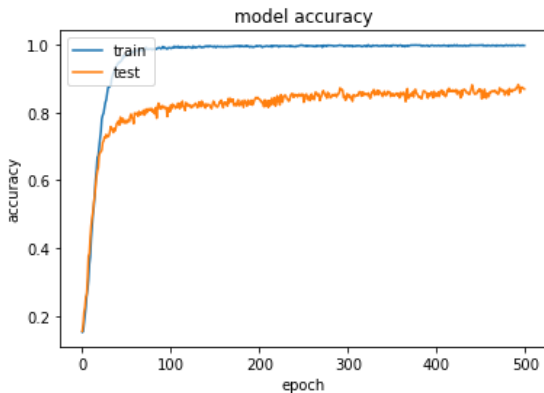
Uji coba terhadap penggunaan proses *Discrete Wavelet Transform* (DWT) dilakukan untuk mengetahui apakah penggunaan DWT ini berhasil menambah performa model CNN yang dirancang. Uji coba akan dilakukan dengan 2 macam data, yaitu:

1. Data tanpa proses DWT
2. Data dengan proses DWT

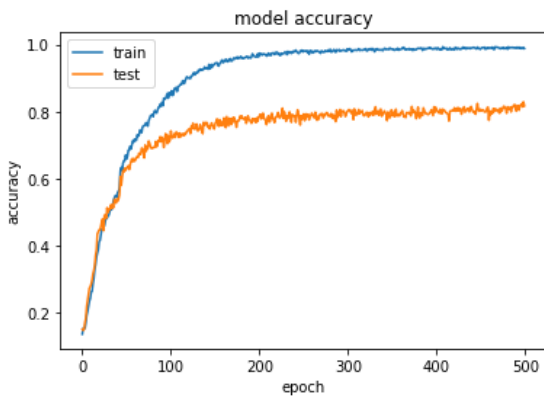
Pada uji coba pembagian data diperoleh hasil optimal pada pembagian berdasarkan sesi. Pembagian data tersebut kemudian digunakan untuk mengetahui hasil yang paling optimal untuk skenario uji coba penggunaan *wavelet*.

Uji coba penggunaan *wavelet* masing-masing menghasilkan grafik akurasi pengujian yang dapat dilihat pada Gambar 5.6 dan

Gambar 5.7. Perbandingan nilai akurasi, *precision*, *recall* pengujian serta lama waktu pelatihan dapat dilihat pada Tabel 5.5.



Gambar 5.6 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data *wavelet*



Gambar 5.7 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data tanpa *wavelet*

Tabel 5.5 Perbandingan lama waktu pelatihan dan akurasi, *precision*, *recall* arsitektur CNN pada uji coba penggunaan data *wavelet*

Wavelet	Lama waktu pelatihan	Akurasi	Kelas	Data uji	<i>Precision</i>	<i>Recall</i>
Tanpa <i>Wavelet</i>	2500 detik	81,6%	<i>Afraid</i>	121	77%	60%
			<i>Angry</i>	143	87%	76%
			<i>Disgust</i>	155	82%	85%
			<i>Happy</i>	147	92%	95%
			<i>Neutral</i>	135	77%	88%
			<i>Sad</i>	149	71%	76%
			<i>Surprised</i>	150	85%	88%
Dengan <i>Wavelet</i>	8806 detik	86,8%	<i>Afraid</i>	121	83%	70%
			<i>Angry</i>	143	91%	81%
			<i>Disgust</i>	155	88%	85%
			<i>Happy</i>	147	98%	99%
			<i>Neutral</i>	135	85%	93%
			<i>Sad</i>	149	78%	83%
			<i>Surprised</i>	150	85%	95%

5.4.3 Uji Coba Parameter CNN

Uji coba penggantian parameter CNN digunakan untuk mengetahui parameter mana saja yang menghasilkan performa model terbaik. Parameter CNN yang akan dicoba untuk divariasikan antara lain:

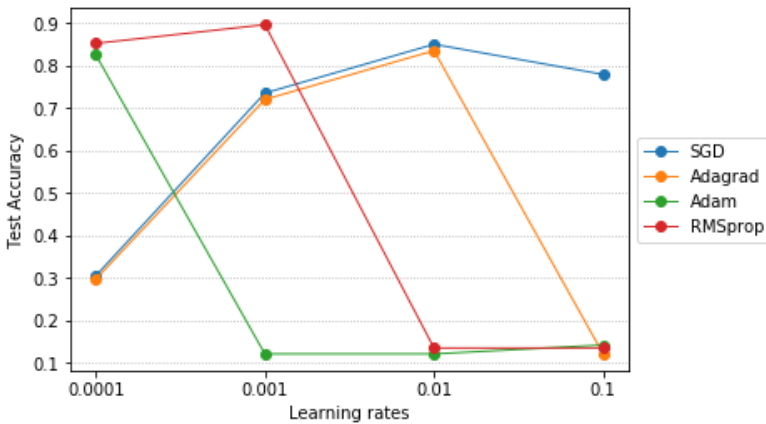
1. *Optimizer*
2. *Learning rate*

Pada uji coba penggunaan *wavelet* diperoleh hasil optimal pada penggunaan *wavelet*. Data yang telah diproses menggunakan *Discrete Wavelet Transform* (DWT) tersebut kemudian digunakan untuk mengetahui hasil yang paling optimal untuk skenario uji coba penggantian parameter CNN.

Optimizer yang telah digunakan untuk uji coba sebelumnya adalah Adam *Optimizer* dengan *learning rate* 0,0001. Skenario uji coba penggantian parameter akan dilakukan dengan 4 penggantian *optimizer* antara lain dengan menggunakan SGD, Adagrad, Adam dan RMSprop.

Learning rate juga akan dicoba untuk divariasikan dengan harapan mendapatkan performa model yang lebih baik dari sebelumnya. Akan dicoba beberapa variasi *learning rate* yaitu 0.1, 0.01, 0.001 dan 0.0001.

Uji coba penggantian *optimizer* dan *learning rate* pada arsitektur CNN menghasilkan grafik akurasi pengujian yang dapat dilihat pada Gambar 5.8 dan perbandingan nilai akurasi, *precision*, *recall* pengujian serta lama waktu pelatihan dapat dilihat pada Tabel 5.6.



Gambar 5.8 Perbedaan akurasi pada arsitektur CNN dalam uji coba penggantian *optimizer* dan *learning rate*

Tabel 5.6 Perbandingan lama waktu pelatihan dan akurasi terbaik, *precision, recall* arsitektur CNN pada uji coba penggantian parameter CNN

Optimizer	Learning Rate	Lama waktu pelatihan	Akurasi	Kelas	Data uji	Precision	Recall
SGD	0,01	8685 detik	85%	<i>Afraid</i>	121	77%	67%
				<i>Angry</i>	143	87%	86%
				<i>Disgust</i>	155	89%	80%
				<i>Happy</i>	147	95%	97%
				<i>Neutral</i>	135	82%	96%
				<i>Sad</i>	149	79%	77%
				<i>Surprised</i>	150	85%	92%
Adagrad	0,01	8817 detik	83,5%	<i>Afraid</i>	121	71%	70%
				<i>Angry</i>	143	93%	73%
				<i>Disgust</i>	155	91%	83%
				<i>Happy</i>	147	93%	98%
				<i>Neutral</i>	135	77%	95%
				<i>Sad</i>	149	73%	74%
				<i>Surprised</i>	150	87%	91%
Adam	0,0001	8806 detik	86,8%	<i>Afraid</i>	121	83%	70%
				<i>Angry</i>	143	91%	81%
				<i>Disgust</i>	155	88%	85%
				<i>Happy</i>	147	98%	99%
				<i>Neutral</i>	135	85%	93%
				<i>Sad</i>	149	78%	83%
				<i>Surprised</i>	150	85%	95%
RMSprop	0,001	8846 detik	89,6%	<i>Afraid</i>	121	76%	85%
				<i>Angry</i>	143	91%	85%
				<i>Disgust</i>	155	89%	88%
				<i>Happy</i>	147	97%	99%
				<i>Neutral</i>	135	97%	96%
				<i>Sad</i>	149	83%	85%
				<i>Surprised</i>	150	94%	89%

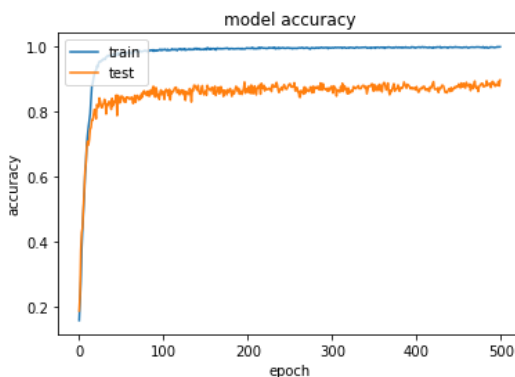
5.4.4 Uji Coba Augmentasi Data

Uji coba terhadap penggunaan data augmentasi dilakukan untuk menambah akurasi model. Augmentasi data juga diharapkan dapat mengurangi *overfit* karena augmentasi data memvariasikan data dengan refleksi horizontal dan perbesaran ukuran gambar. Skenario uji coba penggunaan data augmentasi dilakukan pada 2 kelompok data yaitu:

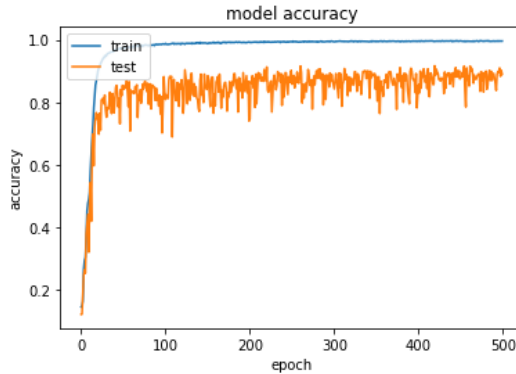
1. Data asli saja
2. Data asli & data augmentasi refleksi horizontal dan perbesaran ukuran gambar

Pada uji coba penggantian parameter CNN diperoleh hasil optimal pada penggunaan *optimizer* RMSprop dengan *learning rate* 0,001. Parameter yang telah dicoba tersebut kemudian digunakan untuk mengetahui hasil yang paling optimal untuk skenario uji coba augmentasi data.

Uji coba penggunaan augmentasi data masing-masing menghasilkan grafik akurasi pengujian yang dapat dilihat pada Gambar 5.9 dan Gambar 5.10. Perbandingan nilai akurasi, *precision*, *recall* pengujian serta lama waktu pelatihan dapat dilihat pada Tabel 5.7.



Gambar 5.9 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data asli



Gambar 5.10 Grafik akurasi pada arsitektur CNN dalam uji coba penggunaan data asli dan data augmentasi

Tabel 5.7 Perbandingan lama waktu pelatihan dan akurasi, *precision*, *recall* arsitektur CNN pada uji coba augmentasi data

Data	Lama waktu pelatihan	Akurasi	Kelas	Data uji	<i>Precision</i>	<i>Recall</i>
Data asli	8846 detik	89,6%	<i>Afraid</i>	121	76%	85%
			<i>Angry</i>	143	91%	85%
			<i>Disgust</i>	155	89%	88%
			<i>Happy</i>	147	97%	99%
			<i>Neutral</i>	135	97%	96%
			<i>Sad</i>	149	83%	85%
			<i>Surprised</i>	150	94%	89%
Data asli dan augmentasi	25025 detik	89%	<i>Afraid</i>	121	75%	83%
			<i>Angry</i>	143	98%	87%
			<i>Disgust</i>	155	86%	90%
			<i>Happy</i>	147	94%	99%
			<i>Neutral</i>	135	93%	93%
			<i>Sad</i>	149	83%	87%
			<i>Surprised</i>	150	95%	83%

5.5 Hasil dan Evaluasi

Pada uji coba pembagian data, dapat diketahui bahwa pembagian data berdasarkan sesi pengambilan gambar dapat menghasilkan akurasi yang lebih tinggi dibandingkan dengan pembagian data berdasarkan individu yang difoto yaitu sebesar 84,68%. Hal ini dapat terjadi karena pembagian data berdasarkan sesi membagi data secara lebih merata sehingga semua individu yang menjadi model dataset dan semua sudut pandang kelas ekspresi wajah pernah dimasukkan ke model. Kelas ekspresi wajah yang mendapatkan nilai *precision* dan *recall* tertinggi adalah kelas ekspresi senang. Hal ini dapat terjadi karena secara visual, ekspresi senang tidak mirip dengan ekspresi kelas yang lain sehingga dapat dibedakan dengan baik. Kelas ekspresi yang mengalami salah klasifikasi terbanyak adalah pada kelas ekspresi takut dan sedih, terlihat dari nilai *precision* dan *recall* yang paling rendah diantara kelas ekspresi yang lain. Hal ini disebabkan karena secara visual, kedua kelas ini terlihat mirip dan lebih sulit untuk dibedakan, seperti yang dapat dilihat pada Gambar 5.11.



Gambar 5.11 Contoh tampilan ekspresi wajah yang mengalami salah klasifikasi terbanyak yaitu takut dan sedih

Pada uji coba penggunaan *wavelet*, diperoleh akurasi yang lebih baik pada data yang menggunakan *wavelet* dibanding dengan

data asli tanpa *wavelet* yaitu 86,8%. Lama waktu pelatihan berbeda cukup signifikan yaitu 2500 detik untuk data tanpa *wavelet* dan 8806 detik untuk data dengan *wavelet*. Hal ini terjadi karena *wavelet* menggunakan 4 gambar yaitu gambar *subband* HH, HL, LH, dan LL sebagai input model yang dibangun sedangkan data asli hanya menggunakan 1 gambar sebagai input.

Pada uji coba penggantian parameter CNN, diperoleh hasil akurasi yang paling baik pada penggunaan RMSprop *optimizer* dengan *learning rate* sebesar 0,001 yaitu sebesar 89,6%. RMSProp *optimizer* tepat digunakan untuk arsitektur CNN yang telah dibangun karena menggunakan *adaptive learning rate* yang cenderung menghasilkan performa yang lebih baik dibandingkan dengan *optimizer* yang menggunakan *learning rate* statis seperti SGD.

Pada uji coba augmentasi data, akurasi yang hampir sama didapatkan untuk kedua jenis data yaitu 89,6% untuk data asli dan 89% untuk data asli dan augmentasi. Penambahan variasi data menggunakan augmentasi menghasilkan akurasi yang tidak jauh berbeda dibandingkan dengan data tanpa augmentasi walaupun lama waktu pelatihan bertambah secara signifikan yaitu sekitar 3 kali lipat. Hal ini terjadi karena data gambar asli telah mewakili semua kelas yang ada dengan baik sehingga tidak diperlukan data augmentasi lagi.

Dari keempat hasil uji coba yang telah dilakukan, pada Tabel 5.8 ditetapkan parameter optimal dari seluruh uji coba tersebut.

Tabel 5.8 Parameter optimal yang ditetapkan

Keterangan	Parameter optimal
Pembagian data	Berdasarkan sesi
Penggunaan <i>wavelet</i>	Menggunakan DWT
<i>Optimizer</i> dan <i>Learning rate</i>	RMSprop dengan <i>learning rate</i> 0.001
Augmentasi data	Data asli

(Halaman ini sengaja dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Dalam pengerjaan Tugas Akhir ini setelah melalui tahap perancangan aplikasi, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut:

1. Pembagian data berdasarkan sesi menghasilkan akurasi yang lebih baik dibanding pembagian data berdasarkan individu. Akurasi model bertambah karena model pernah dilatih dengan data wajah semua individu yang menjadi model dataset. Maka dipilihlah pembagian data berdasarkan sesi yang menghasilkan rata-rata akurasi sebesar 84,68%.
2. Data yang menggunakan *Discrete Wavelet Transform* (DWT) pada tahap praproses menghasilkan akurasi yang lebih baik dibandingkan dengan data yang tidak menggunakan proses DWT yaitu sebesar 86,8%.
3. Augmentasi data berupa refleksi gambar secara horizontal dan perbesaran ukuran gambar kurang efektif untuk menambah akurasi model. Hal ini terbukti dengan akurasi model yang dilatih dengan data asli yaitu sebesar 89,6% yang tidak jauh berbeda dengan akurasi model yang dilatih dengan data yang telah dilakukan proses augmentasi yaitu sebesar 89%.
4. Sistem pengenalan ekspresi wajah manusia telah berhasil diimplementasikan dengan akurasi tertinggi 89,6% yang didapatkan dari uji coba menggunakan pembagian data berdasarkan sesi, praproses *Discrete Wavelet Transform*

(DWT), RMSProp *optimizer* dengan *learning rate* 0,001 dan tanpa augmentasi data.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem pengenalan ekspresi wajah manusia menggunakan *Convolutional Neural Network* pada data gambar, yaitu:

1. Pengembangan sistem yang dapat mengenali lebih banyak ekspresi wajah manusia.
2. Pengembangan sistem yang dapat mengenali banyak ekspresi wajah manusia dalam suatu gambar.
3. Pengembangan sistem dengan arsitektur yang menghasilkan performa yang lebih baik.
4. Pengembangan sistem yang dapat mengenali ekspresi wajah manusia secara *real-time*.
5. Melakukan eksplorasi parameter selain *optimizer* dan *learning rate* yang dapat menambah performa arsitektur seperti *activation function*, jumlah dan ukuran *filter* konvolusi, ukuran *max pooling layer* dan level *Discrete Wavelet Transform* (DWT).

DAFTAR PUSTAKA

- [1] T. William dan R. Li, "An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification," *Journal of Software Engineering and Applications*, no. 11, pp. 69-88, 2018.
- [2] K. Pooja, J. Kumari dan R. Rajesh, "Facial expression recognition: A survey," *Procedia Computer Science*, vol. 58, pp. 486-491, 2015.
- [3] A. Karpathy, "Convolutional Neural Networks for Visual Recognition," Stanford University, [Online]. Available: <http://cs231n.github.io/>. [Diakses 30 November 2018].
- [4] S. Fadillah, "Penerapan Pengolahan Citra menggunakan Metode Deep Learning untuk Mendeteksi Kecacatan Permukaan Buah Manggis," Yogyakarta, 2017.
- [5] D. Hubel dan T. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *Journal of Physiology*, vol. 195, p. 215–243, 1968.
- [6] M. Zufar dan B. Setiyono, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time," *Jurnal Sains dan Seni ITS*, vol. 5, pp. 2337-3520, 2016.
- [7] "Convolutional Neural Network," MathWorks, [Online]. Available: <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>. [Diakses 29 November 2018].
- [8] "Deep learning for complete beginners: convolutional neural networks with keras," Cambridgespark, 20 March 2017. [Online]. Available: <https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html>. [Diakses 29 November 2018].













- [9] “An Intuitive Explanation of Convolutional Neural Networks,” Ujjwalkarn, 11 August 2016. [Online]. Available: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets>. [Diakses 29 November 2018].
- [10] I. W. Suartika, A. Y. Wijaya dan R. Soelaiman, “Klasifikasi Citra Menggunakan Convolutional pada Caltech 101,” *JURNAL TEKNIK ITS*, vol. 5, 2016.
- [11] S. Sena, “Pengenalan Deep Learning Neural Network,” 28 October 2017. [Online]. Available: <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>. [Diakses 30 November 2018].
- [12] G. Hinton, *Neural Networks for Machine Learning*.
- [13] S. Ruder, “Ruder.io,” 19 January 2016. [Online]. Available: <http://ruder.io/optimizing-gradient-descent/index.html#rmsprop>. [Diakses 23 December 2018].
- [14] A. Budhiraja, “Dropout in (Deep) Machine Learning,” [Online]. Available: <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>. [Diakses 11 12 2018].
- [15] “Introduction to Convolution Neural Networks,” 4 April 2016. [Online]. Available: <https://www.analyticsvidhya.com/blog/2016/04/deep-learning-computer-vision-introduction-convolution-neural-networks/>. [Diakses 29 November 2018].
- [16] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Bandung: Informatika ITB.
- [17] “About Python,” Python, [Online]. Available: <https://www.python.org/about/>. [Diakses 30 November 2018].
- [18] “Keras: The Python Deep Learning library,” Keras, [Online]. Available: <https://keras.io/>. [Diakses 30 November 2018].













- [19] “TensorFlow,” TensorFlow, [Online]. Available: <https://www.tensorflow.org/>. [Diakses 30 November 2018].
- [20] “OpenCV,” [Online]. Available: <https://opencv.org/>. [Diakses 30 November 2018].
- [21] “PyWavelets,” PyWavelets, [Online]. Available: <https://pywavelets.readthedocs.io/en/latest/>. [Diakses 30 November 2018].
- [22] “NumPy,” NumPy, [Online]. Available: <http://www.numpy.org/>. [Diakses 30 November 2018].
- [23] “Scikit-learn,” Scikit-learn, [Online]. Available: <http://scikit-learn.org/stable/index.html>. [Diakses 30 November 2018].
- [24] “Matplotlib,” Matplotlib, [Online]. Available: <https://matplotlib.org/index.html>. [Diakses 30 November 2018].













(Halaman ini sengaja dikosongkan)







LAMPIRAN

L.1 Hasil Uji Coba Individu F01







1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				













3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				













5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				
6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				













7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.2 Hasil Uji Coba Individu F09

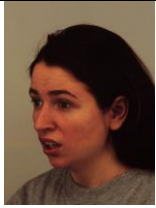





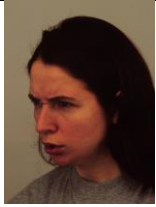





1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				


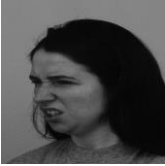










2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				
3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				





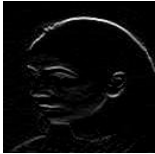

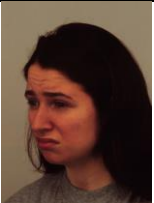





4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				
5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				

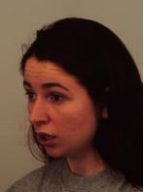





6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				
7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.3 Hasil Uji Coba Individu F15





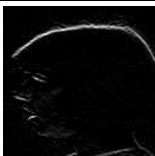

1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				

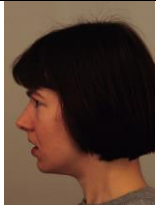



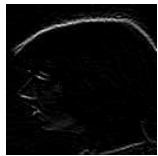


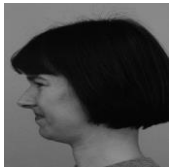


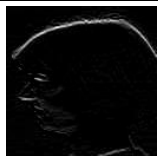

3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				





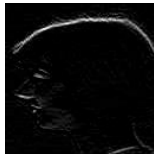

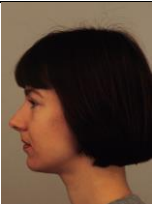
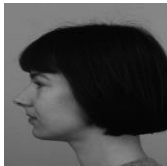




5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				
6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				

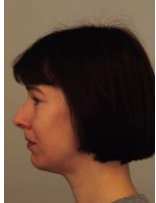



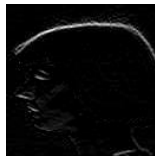

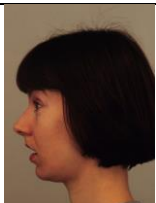
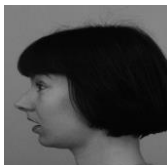

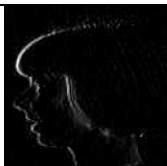
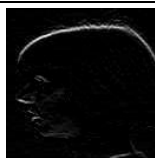

7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.4 Hasil Uji Coba Individu F18













1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				




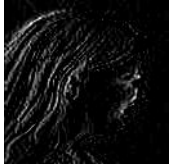








2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				













4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				
5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				







6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				
7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.5 Hasil Uji Coba Individu F24







1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				

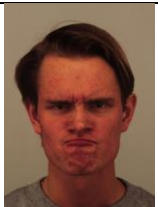











3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				













5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				
6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				







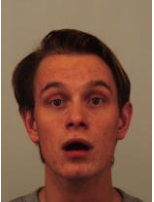
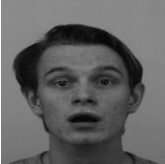




7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.6 Hasil Uji Coba Individu M21

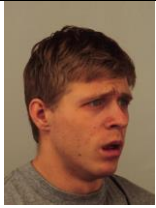





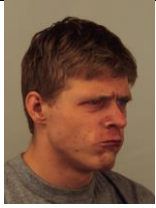





1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				













2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				
3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				







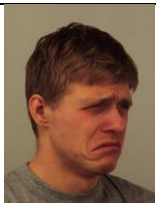
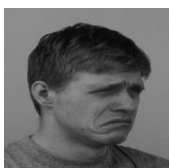




4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				
5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				

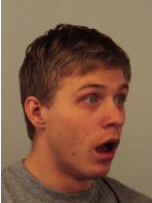





6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				
7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.7 Hasil Uji Coba Individu M23

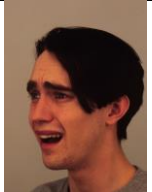





1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				













3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				

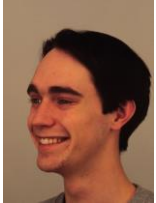





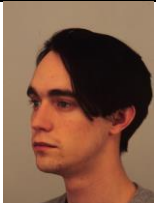





5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				
6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				

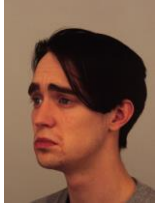





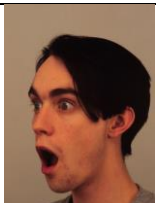
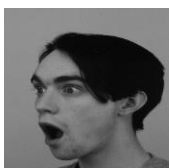




	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
7			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.8 Hasil Uji Coba Individu M24







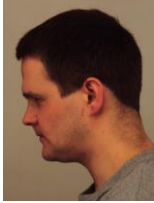





	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
1			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				













2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				








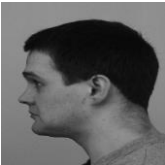




4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				
5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				

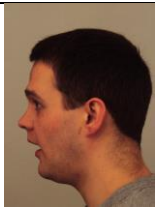





6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				
7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.9 Hasil Uji Coba Individu M26







1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				
2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				













3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Jijik
	Hasil Wavelet			
	HH	HL	LH	LL
				
4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				













5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				
6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				













7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

L.10 Hasil Uji Coba Individu M30

1	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Takut	Takut
	Hasil Wavelet			
	HH	HL	LH	LL
				

2	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Marah	Marah
	Hasil Wavelet			
	HH	HL	LH	LL
				
3	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Jijik	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				

4	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Senang	Senang
	Hasil Wavelet			
	HH	HL	LH	LL
				
5	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Netral	Netral
	Hasil Wavelet			
	HH	HL	LH	LL
				

6	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Sedih	Sedih
	Hasil Wavelet			
	HH	HL	LH	LL
				
7	Citra Asli	Hasil resize dan grayscale	Kelas Aktual	Kelas Prediksi
			Kaget	Kaget
	Hasil Wavelet			
	HH	HL	LH	LL
				

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Hendry Wiranto, lahir di Ponorogo pada tanggal 24 Oktober 1996. Penulis menempuh pendidikan mulai dari TK Santa Melania Ponorogo (2002-2003), SD Katolik Santa Maria Ponorogo (2003-2009), SMP Katolik Slamet Riyadi Ponorogo (2009-2012), SMA Katolik St. Louis 1 Surabaya (2012-2015), dan sekarang sedang menjalani pendidikan S1 Informatika di ITS. Penulis aktif dalam organisasi dan kepanitiaan Himpunan Mahasiswa Teknik Computer (HMTC) dan Schematics. Diantaranya adalah menjadi staff Departemen Kewirausahaan HMTC ITS 2016-2017, staff ahli Departemen Kewirausahaan HMTC ITS 2017-2018, staff Departemen Website dan Kesekretariatan Schematics ITS 2016 dan staff ahli Departemen Website dan Kesekretariatan Schematics ITS 2017. Penulis juga merupakan salah satu penerima beasiswa dari Bank Indonesia pada tahun 2017. Komunikasi dengan penulis dapat melalui telepon: +6285334449875 dan *email*: **hendrywiranto24@gmail.com**.