



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KI141502**

# **IMPLEMENTASI ADABOOST LEARNING MENGUNAKAN FITUR MULTI-SCALE BLOCK LOCAL BINARY PATTERN PADA PENGENALAN WAJAH**

**AFIFAH ASMAR SARI**  
**NRP 05111440000154**

**Dosen Pembimbing I**  
**Dr. Eng. Nanik Suciati, 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 2018**

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - KI141502**

**IMPLEMENTASI ADABOOST LEARNING  
MENGUNAKAN FITUR MULTI-SCALE BLOCK  
LOCAL BINARY PATTERN PADA PENGENALAN  
WAJAH**

**AFIFAH ASMAR SARI  
NRP 0511144000154**

**Dosen Pembimbing I  
Dr. Eng. Nanik Suciati, 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 2018**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESIS - KI141502**

# **ADABOOST LEARNING IMPLEMENTATION USING MULTI-SCALE BLOCK LOCAL BINARY PATTERN FEATURES ON FACE RECOGNITION**

**AFIFAH ASMAR SARI  
NRP 05111440000154**

**Supervisor I  
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Supervisor II  
Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS  
Faculty of Information and Communication Technology  
Insitut Teknologi Sepuluh Nopember  
Surabaya 2018**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### IMPLEMENTASI ADABOOST LEARNING MENGUNAKAN FITUR MULTI-SCALE BLOCK LOCAL BINARY PATTERN PADA PENGENALAN WAJAH

### TUGAS AKHIR

Diajukan Guna 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 :

**AFIFAH ASMAR SARI**  
NRP : 05111440000154

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr. Eng. Nanik Suciati, S.Kom., M.Kom.  
NIP. 19710428 199412 2 001

  
(Pembimbing 1)

Dini Adni Navastara, S.Kom., M.Sc.  
NIP. 19851017 201504 2 001

  
(Pembimbing 2)

**SURABAYA**  
**JUNI 2018**

*[Halaman ini sengaja dikosongkan]*

# IMPLEMENTASI ADABOOST LEARNING MENGUNAKAN FITUR MULTI-SCALE BLOCK LOCAL BINARY PATTERN PADA PENGENALAN WAJAH

Nama Mahasiswa : AFIFAH ASMAR SARI  
NRP : 05111440000154  
Departemen : Informatika FTIK-ITS  
Dosen Pembimbing 1 : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.  
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

## ABSTRAK

*Pengenalan wajah masih menjadi topik yang hangat untuk diteliti sejak diperkenalkan 20 tahun silam. Berbagai teknik pengenalan wajah telah dikembangkan guna mendapatkan sebuah sistem yang tangguh. Salah satu metode ekstraksi fitur wajah yang sering digunakan adalah Local Binary Pattern (LBP). Jika LBP memanfaatkan piksel sebagai fitur, pada pengembangan selanjutnya Multi-scale Block Local Binary Patten (MB-LBP) memanfaatkan area piksel sebagai fitur.*

*Pada Tugas Akhir ini diimplementasikan AdaBoost Learning menggunakan fitur MB-LBP untuk pengenalan wajah. Terdapat empat tahap utama yang dijalankan pada sistem pengenalan wajah untuk Tugas Akhir ini yakni pra-proses, ekstraksi fitur menggunakan metode MB-LBP, seleksi fitur menggunakan metode Statistically Effective MB-LBP (SEMB-LBP), dan klasifikasi menggunakan metode AdaBoost. Data yang digunakan dalam proses uji coba berjumlah 3059 citra wajah frontal yang diperoleh dari Collection of Facial Image : Faces94.*

*Dari hasil uji coba, diperoleh nilai akurasi terbaik sebesar 95,42% untuk MB-LBP dengan seleksi fitur menggunakan skala kombinasi 3x3, 9x9, dan 15x15. Sedangkan nilai akurasi terbaik untuk MB-LBP tanpa seleksi fitur adalah sebesar 98,50% dengan skala 9x9.*

***Kata kunci: Pengenalan Wajah, Multi-scale Local Binary Pattern, Statistically Effective Multi-scale Local Binary Pattern, AdaBoost Learning.***

*[Halaman ini sengaja dikosongkan]*

# ADABOOST LEARNING IMPLEMENTATION USING MULTI-SCALE BLOCK LOCAL BINARY PATTERN FEATURES ON FACE RECOGNITION

**Student's Name** : AFIFAH ASMAR SARI  
**Student's ID** : 0511144000154  
**Department** : Informatics FTIK-ITS  
**First Advisor** : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.  
**Second Advisor** : Dini Adni Navastara, S.Kom., M.Sc.

## ABSTRACT

*Facial recognition is still a hot topic to research since its introduction 20 years ago. Various facial recognition techniques have been developed to obtain a robust system. One of the most commonly used facial feature extraction methods is the Local Binary Pattern (LBP). If LBP utilizes pixels as a feature, the next development Multi-scale Block Local Binary Patten (MB-LBP) utilizes the area as a feature.*

*This Final Project implemented AdaBoost Learning using MB-LBP feature for face recognition. There are four main stages of the facial recognition system in this Final Assignment, pre-processing, feature extraction using MB-LBP method, feature selection using the Statistically Effective MB-LBP (SEMB-LBP) method, and classification using the AdaBoost method. The data used in the trial process amounted to 3059 frontal face images obtained from the Collection of Facial Image: Faces94.*

*From the experimental results, the best accuracy value is 95,42% for MB-LBP with feature selection using 3x3, 9x9, and 15x15 combination scale. While the best accuracy value for MB-LBP without feature selection is 98,50% with 9x9 scale.*

**Kata kunci:** *Face Recognition, Multi-scale Local Binary Pattern, Statistically Effective Multi-scale Local Binary Pattern, AdaBoost Learning.*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji dan syukur bagi Allah SWT, yang telah melimpahkan rahmat, dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

### **IMPLEMENTASI ADABOOST LEARNING MENGUNAKAN FITUR MULTI-SCALE LOCAL BINARY PATTERN PADA PENGENALAN WAJAH**

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang berharga bagi penulis. Dengan pengerjaan Tugas Akhir, penulis dapat memperdalam, meningkatkan, serta menerapkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Departemen Informatika ITS.

Terselesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Dan dalam kesempatan ini penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Allah SWT, karena atas izin-Nya lah penulis dapat menyelesaikan Tugas Akhir dengan baik.
2. Kedua orang tua, seluruh saudara dan keluarga besar atas doa dan berbagai bantuan baik secara moral maupun material selama penulis belajar di Departemen Informatika ITS.
3. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku pembimbing I Tugas Akhir yang telah memberikan banyak waktu untuk berdiskusi dan memberi semangat serta motivasi kepada penulis untuk menyelesaikan Tugas Akhir.
4. Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku pembimbing II Tugas Akhir yang telah memberikan banyak waktu untuk berdiskusi dan memberi dukungan selama penulis menyelesaikan Tugas Akhir.
5. Bapak dan Ibu Dosen di Departemen Informatika yang telah memberikan ilmu selama penulis kuliah di Informatika FTIK-ITS.

6. Sahabat seperjuangan Dini, Datin, Sita, Rani, Mila, Sabila, Delia, Mala, Tion, Tepe, Lucha, Ade, Afiif, Irfan, Galang, Habib, Anwar, Luqman.
7. Adik-adik Admin Komputasi Cerdas Visi (KCV), Nuzul, Vira, Titut, Yoshi, Chasni, Prana, Ocid, Pasha, Dandy, Aldi.
8. Seluruh rekan-rekan TC 2014 yang saya banggakan.

Penulis memohon maaf apabila terdapat kekurangan dalam penulisan Tugas Akhir ini. Kritik dan saran penulis harapkan untuk perbaikan dan pembelajaran di kemudian hari. Semoga Tugas Akhir ini dapat memberikan manfaat yang sebesar besarnya.

Surabaya, Juni 2018

Afifah Asmar Sari

## DAFTAR ISI

LEMBAR PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
DAFTAR KODE SUMBER .....	xxi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi .....	3
1.7. Sistematika Penulisan Laporan Tugas Akhir .....	5
BAB II DASAR TEORI.....	7
2.1. Multi-scale Local Binary Pattern (MB-LBP).....	7
2.2. Statistically Effective Multi-scale Local Binary Pattern (SEMB-LBP).....	8
2.3. AdaBoost Learning.....	9
2.4. Decision Tree .....	13
2.5. Perhitungan Evaluasi.....	14
2.6. K-Fold Cross Validation .....	15
BAB III PERANCANGAN SISTEM .....	17
3.1. Data .....	17
3.2. Perancangan Sistem.....	18
3.3. Perancangan Proses .....	19
3.3.1. Pra-Proses.....	19
3.3.2. Ekstraksi Fitur MB-LBP .....	20
3.3.3. Seleksi Fitur SEMB-LBP .....	22
3.3.4. Klasifikasi AdaBoost.....	25
BAB IV IMPLEMENTASI.....	27

4.1.	Lingkungan Implementasi .....	27
4.2.	Implementasi .....	27
4.2.1.	Implementasi Tahap Pra-proses.....	27
4.2.2.	Implementasi Ekstraksi Fitur MB-LBP .....	28
4.2.3.	Implementasi Seleksi Fitur SEMB-LBP.....	33
4.2.4.	Implementasi AdaBoost .....	34
BAB V PENGUJIAN DAN EVALUASI .....		37
5.1.	Lingkungan Pengujian.....	37
5.2.	Data Uji Coba.....	37
5.3.	Skenario Uji Coba .....	37
5.3.1.	Skenario Uji Coba Ekstraksi Fitur MB-LBP .....	38
5.3.2.	Skenario Uji Coba Seleksi Fitur SEMB-LBP.....	40
5.3.3.	Skenario Uji Coba Klasifikasi AdaBoost .....	41
5.3.4.	Skenario Uji Coba Kombinasi Skala MB-LBP ...	44
5.3.5.	Skenario Uji Coba Perbandingan Metode Ekstraksi 45	
5.4.	Analisa Hasil .....	46
BAB VI KESIMPULAN DAN SARAN .....		49
6.1.	Kesimpulan.....	49
6.2.	Saran.....	50
DAFTAR PUSTAKA.....		51
LAMPIRAN .....		53

## DAFTAR GAMBAR

Gambar 2.1 (a) LBP Original (b) MB-LBP 9x9.....	7
Gambar 2.2 (a) Weak Classifier 1 (b) Penambahan Bobot 1 (c) Weak Classifier 2 (d) Penambahan Bobot 2 (e) Weak Classifier 3 (f) Final Classifier .....	11
Gambar 2.3 Decision Tree.....	13
Gambar 2.4 K-fold Cross Validation .....	15
Gambar 3.1 Collection of Facial Image : Faces94 .....	17
Gambar 3.2 Diagram Alir Sistem Pengenalan Wajah .....	18
Gambar 3.3 (a) Citra Asli (b) Citra <i>Grayscale</i> .....	19
Gambar 3.4 Diagram Alir Ekstraksi Fitur MB-LBP .....	20
Gambar 3.5 Diagram Alir Perbandingan Blok.....	21
Gambar 3.6 (a) Citra <i>grayscale</i> (b) Citra MB-LBP 3x3 (c) Citra MB-LBP 9x9 (d) Citra MB-LBP 15x15 (e) Citra MB-LBP 21x21 .....	22
Gambar 3.7 Diagram Alir Seleksi Fitur SEMB-LBP.....	23
Gambar 3.8 (a) Histogram MB-LBP 3x3 (b) Histogram MB-LBP 9x9 (c) Histogram MB-LBP 15x15 (d) Histogram MB-LBP 21x21 .....	24
Gambar 3.9 (a) Citra <i>Grayscale</i> (b) Citra MB-LBP 3x3.....	24
Gambar 3.10 Diagram Alir Pembuatan Model.....	26
Gambar 5.1 Grafik Akurasi Uji Coba Ekstraksi Fitur MB-LBP .	39
Gambar 5.2 Grafik Waktu Uji Coba Ekstraksi Fitur MB-LBP ...	39
Gambar 5.3 Grafik Akurasi dan Waktu Uji Coba Kedalaman <i>Decision Tree (max_depth)</i> pada <i>AdaBoost</i> .....	42

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 4.1 Spesifikasi Perangkat .....	27
Tabel 5.1 Uji Coba Ekstraksi Fitur .....	38
Tabel 5.2 Uji Coba Seleksi Fitur .....	40
Tabel 5.3 Uji Coba <i>Max_depth</i> .....	41
Tabel 5.4 Uji Coba <i>N Estimators</i> .....	42
Tabel 5.5 Uji Coba <i>Learning Rate</i> .....	43
Tabel 5.6 Uji Coba Kombinasi Skala MB-LBP .....	44
Tabel 5.7 Uji Coba Perbandingan Metode Ekstraksi .....	45

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Pra-proses .....	28
Kode Sumber 4.2 Perbandingan Ketetangaan .....	28
Kode Sumber 4.3 Perhitungan Rata-Rata.....	29
Kode Sumber 4.4 Ekstraksi Fitur MB-LBP .....	33
Kode Sumber 4.5 Seleksi Fitur SEMB-LBP .....	34
Kode Sumber 4.6 Klasifikasi AdaBoost.....	35

*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini, gambaran tugas akhir secara umum dapat dipahami.

### 1.1. Latar Belakang

Pengenalan wajah merupakan sebuah topik yang semakin ramai diteliti serta dikembangkan sejak kali pertama diperkenalkan sekitar 20 tahun lalu. Seperti yang telah kita ketahui bersama, pengenalan wajah memiliki banyak manfaat utamanya dalam bidang keamanan. Hal tersebut menjadi salah satu alasan terbesar dari berbagai penelitian yang dilakukan. Berbagai metode diterapkan guna mendapatkan sistem pengenalan wajah yang *robust* dan akurat. Terdapat metode yang berfokus pada pencarian representasi subbagian wajah optimal seperti *Principal Component Analysis* (PCA) [1] dan *Linear Discriminant Analysis* (LDA) [2]. Ada pula metode berbasis fitur pada sistem pengenalan wajah seperti *Local Feature Analysis* (LFA) [3] dan *Gabor Wavelet-based Features* [4].

Selain metode yang telah disebutkan sebelumnya, terdapat sebuah metode handal pengolahan citra yakni *Local Binary Pattern* (LBP) [5]. Dari sebuah penelitian mengenai pengenalan wajah menggunakan metode LBP, diketahui bahwa metode LBP memberikan hasil yang terbaik bila dibandingkan dengan metode PCA, EBGMM, dan BIC [6]. LBP merupakan sebuah metode sederhana yang mengkalkulasikan piksel pada sebuah citra dengan membandingkan nilai *grayscale*-nya. Nantinya akan didapatkan sebuah nilai baru yang menggantikan nilai asli dari piksel citra tersebut dan selanjutnya menjadi input pada proses klasifikasi. LBP pada umumnya menggunakan area piksel dengan ukuran 3 x 3. Ukuran tersebut membuat LBP tidak memungkinkan untuk menangkap area yang lebih besar, yang mungkin saja justru lebih signifikan untuk proses klasifikasi.

Oleh karena itu, pada tugas akhir ini dilakukan implementasi dari pengembangan LBP menjadi *Multi-scale Block Local Binary Pattern* (MB-LBP) guna meningkatkan performa pengenalan wajah. Perhitungan pada MB-LBP ini nantinya tidak lagi bergantung pada masing-masing piksel, akan tetapi akan memanfaatkan nilai rata-rata pada masing-masing blok dari sebuah bagian pada citra. Dengan adanya pengembangan ini, bentuk *uniform* dari LBP asli tidak dapat digunakan [7], sehingga diterapkan metode bentuk *uniform* baru melalui analisa statistik yang kemudian disebut sebagai *Statistically Effective Multi-scale Block Local Binary Pattern* (SEMB-LBP). SEMB-LBP ini nantinya akan digunakan sebagai metode seleksi fitur. Dan hasil dari SEMB-LBP nantinya akan dijadikan sebagai masukan pada proses klasifikasi menggunakan *AdaBoost Learning*.

## 1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur menggunakan *Multi-scale Block Local Binary Pattern*?
2. Bagaimana melakukan seleksi fitur dari hasil *Multi-scale Block Local Binary Pattern*?
3. Bagaimana melakukan klasifikasi fitur *Multi-scale Local Binary Pattern* dengan menggunakan *AdaBoost Learning*?

## 1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Dataset yang digunakan adalah *Collection of Facial Image : Faces94* yang dikelola oleh Dr.Libor Spacek dari Universitas Essex, Inggris.
2. Algoritma ekstraksi fitur yang diimplementasikan adalah MB-LBP dengan skala berkelipatan 3.
3. Keseluruhan metode pada tugas akhir ini diimplementasikan menggunakan bahasa pemrograman *Python*.

## 1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah mengimplementasikan *AdaBoost Learning* menggunakan fitur MB-LBP dalam sistem pengenalan wajah.

## 1.5. Manfaat

Pengerjaan tugas akhir ini diharapkan dapat menghasilkan sistem pengenalan wajah yang handal dengan akurasi yang tinggi serta nantinya dapat diimplementasikan pada sistem pengenalan wajah berskala besar.

## 1.6. Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

### 1. Penyusunan Proposal Tugas Akhir

Tahap awal yang dilakukan dalam pengerjaan tugas akhir ini adalah penyusunan proposal. Di dalam proposal diajukan suatu gagasan yakni implementasi *AdaBoost Learning* menggunakan fitur *Multi-scale Block Local Binary Pattern* pada pengenalan wajah.

### 2. Studi Literatur

Pada tahap ini dilakukan pencarian dan pemahaman terhadap literatur berupa jurnal, *paper*, buku, serta dokumentasi internet yang digunakan sebagai referensi untuk pengerjaan tugas akhir. Literatur yang digunakan pada tugas akhir ini secara keseluruhan mengacu pada *paper* "*Learning Multi-scale Block Local Binary Pattern for Face Recognition*" [8].

Selain itu, referensi penunjang yang digunakan adalah *paper* serta jurnal yang membahas tentang *Local Binary Pattern* (LBP) yang merupakan metode dasar yang akan dikembangkan sebagai metode ekstraksi fitur pada tugas akhir ini. Digunakan juga berbagai referensi terkait *AdaBoost Learning* yang akan digunakan sebagai metode klasifikasi pada pengenalan wajah.

### 3. Analisis dan Perancangan Sistem

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Masukan untuk sistem yang dibangun berupa citra wajah frontal dengan ukuran 180 x 200. Selanjutnya sistem mengolah citra menggunakan metode MB-LBP sebagai metode ekstraksi fitur. Kemudian hasil dari ekstraksi fitur tersebut akan menjadi masukan pada proses seleksi fitur SEMB-LBP dan dilanjutkan klasifikasi pengenalan wajah menggunakan *AdaBoost*. Keluaran dari sistem berupa hasil pengenalan wajah.

### 4. Implementasi Sistem

Implementasi merupakan tahap membangun rancangan sistem yang telah dibuat. Sistem pengenalan wajah pada tugas akhir ini diimplementasikan menggunakan bahasa pemrograman *Python 3.0*.

### 5. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap sistem yang telah dibuat untuk mengetahui kemampuan algoritma yang dipakai, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul pada sistem yang dibuat. Pengujian dilakukan dengan berbagai ukuran skala dari MB-LBP, jumlah fitur pada tahap seleksi fitur SEMB-LBP, serta pada parameter *AdaBoost classifier* guna mendapatkan hasil klasifikasi terbaik. Adapun parameter evaluasi dari sistem pengenalan wajah tugas akhir ini menggunakan nilai akurasi.

### 6. Penyusunan Buku Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## 1.7. Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

### 1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu rumusan permasalahan, batasan masalah, dan sistematika penulisan juga merupakan bagian dari bab ini.

### 2. Bab II Dasar Teori

Bab ini berisi penjelasan terkait metode-metode yang digunakan dalam tugas akhir ini yakni penjelasan dasar mengenai LBP sebagai metode yang akan dikembangkan, MB-LBP sebagai metode ekstraksi fitur, SEMB-LBP sebagai metode seleksi fitur, serta metode klasifikasi *AdaBoost*.

### 3. Bab III Perancangan Sistem

Bab ini berisi penjelasan mengenai data dan desain dari sistem pengenalan wajah yang dibangun pada tugas akhir ini.

### 4. Bab IV. Implementasi

Bab ini merupakan pembangunan sistem dengan *Python* sesuai permasalahan dan batasan yang telah dijelaskan pada Bab I.

### 5. Bab V. Pengujian dan Evaluasi

Bab ini berisi penjelasan mengenai data hasil percobaan, pengukuran, dan pembahasan mengenai hasil percobaan yang telah dilakukan.

### 6. Bab VI. Kesimpulan dan Saran

Bab ini berupa hasil penelitian, program, dan karya rancangan. Selain itu, pada bab ini diberikan saran-saran yang berisi hal-hal yang masih dapat dikerjakan dengan lebih baik dan dapat dikembangkan lebih lanjut

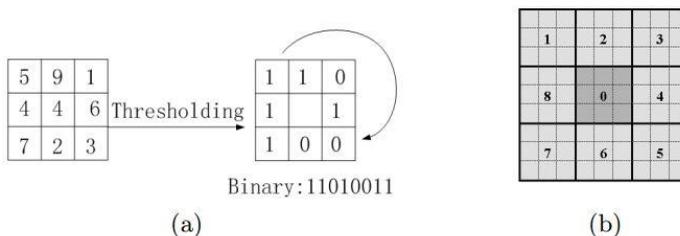
*[Halaman ini sengaja dikosongkan]*

## BAB II DASAR TEORI

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pembuatan sistem pengenalan wajah menggunakan fitur MB-LBP. Penjelasan ini bertujuan untuk memberikan dasar teori yang menjadi pedoman dalam pengembangan sistem.

### 2.1. Multi-scale Local Binary Pattern (MB-LBP)

*Local Binary Pattern* atau biasa disingkat LBP merupakan salah satu algoritma ekstraksi fitur yang umum digunakan pada pengolahan citra. Pada original LBP dilakukan pembagian daerah menjadi sebuah area berukuran 3 x 3. Dengan ukuran 3 x 3 dapat diketahui bahwa area tersebut terdiri dari 9 piksel berbentuk persegi. Selanjutnya, dilakukan perbandingan antara piksel tengah dengan masing-masing piksel tetangganya. Apabila nilai dari piksel tetangganya lebih besar atau sama dengan piksel tengah, maka nilai dari piksel tetangga tersebut adalah 1. Dan apabila nilai dari piksel tetangganya lebih kecil dari piksel tengah, maka nilai yang diberikan adalah 0. Satu persatu piksel tetangga akan dibandingkan dengan pola pergeseran jarum jam, bisa saja searah atau berlawanan arah. Nantinya akan didapatkan 8 bit angka biner yang akan direpresentasikan dalam bentuk desimal untuk menggantikan nilai pada piksel tengah [8].



**Gambar 2.1 (a) LBP Original (b) MB-LBP 9x9**

Jika metode LBP pada Gambar 2.1 (a) melakukan perbandingan terhadap nilai *grayscale* dari masing-masing piksel, sedangkan metode MB-LBP melakukan perbandingan terhadap nilai rata-rata *grayscale* dari sebuah area piksel seperti yang terlihat pada Gambar 2.1 (b). Skala dari MB-LBP sendiri memiliki berbagai variasi dengan ketentuan yakni nilai skala tersebut merupakan kelipatan 3. Skala terkecil dari MB-LBP adalah 3 x 3 yang tak lain merupakan LBP original.

## 2.2. Statistically Effective Multi-scale Local Binary Pattern (SEMB-LBP)

Pada metode LBP original, terdapat proses pemilihan fitur dari hasil ekstraksi operator LBP yang biasa disebut dengan *uniform pattern*. Terdapat 2 keuntungan yang didapat dengan menggunakan *uniform pattern* LBP ini yakni mengurangi *space* dari memori karena berkurangnya ukuran dari vektor fitur dan fitur yang dihasilkan hanya mendeteksi tekstur-tekstur yang penting seperti titik, sudut, dan tepi[6].

Sebuah fitur LBP dianggap memiliki pola *uniform* apabila terdapat paling banyak 2 perubahan transisi antara 0 dan 1 atau sebaliknya [6]. Akan tetapi, syarat *uniform* pada LBP original ini tidak dapat diterapkan pada MB-LBP dikarenakan algoritma dari LBP bekerja pada masing-masing piksel sedangkan MB-LBP berupa sebuah area yang terdiri dari banyak piksel. Oleh karena itu, diperlukan sebuah perhitungan baru untuk melakukan seleksi terhadap fitur MB-LBP yakni *Statistically Effective Multi-scale Block Local Binary Pattern* (SEMB-LBP) [8].

Didapatkan  $f_s(x, y)$  sebagai fitur MB-LBP dengan skala  $s$  pada lokasi  $(x, y)$  dari hasil komputasi citra awal. Maka histogram dari fitur MB-LBP  $f_s(x, y)$  pada sebuah citra  $I(x, y)$  dapat didefinisikan pada Persamaan 2.1 :

$$H_s(l) = 1_{[f_s(x,y)=l]}, \quad \ell = 0, \dots, L - 1 \quad (2.1)$$

Dimana  $1_{(S)}$  adalah indikator himpunan  $S$  dan  $l$  adalah label dari MB-LBP. Karena seluruh fitur MB-LBP berupa 8-bit *string* biner, maka total  $L$  adalah  $2^8$  atau sebanyak 256 label. Dengan demikian, histogram dari MB-LBP memiliki 256 bin. Histogram ini berisi informasi terkait distribusi dari fitur MB-LBP atas keseluruhan citra. Untuk mendapatkan pola *uniform* MB-LBP didefinisikan SEMB-LBP dengan skala  $s$  seperti pada Persamaan 2.2 berikut :

$$SEMB - LBP_s = \{\ell | Rank[H_s(l)] < N\} \quad (2.2)$$

Dimana  $Rank[H_s(l)]$  adalah indeks dari histogram  $H_s(l)$  setelah diurutkan dari besar ke kecil. Dan untuk simbol  $N$  disini adalah jumlah dari pola *uniform*. Nilai  $N$  dapat diberikan secara acak dari 1 sampai 256. Namun perlu diperhatikan, semakin besar nilai  $N$  maka semakin besar pula dimensi dari vektor fitur sehingga akan memperlambat komputasi. Sebaliknya, semakin kecil  $N$  akan mempercepat komputasi karena ukuran vektor fitur yang lebih kecil akan tetapi keragaman fiturnya pun semakin sedikit. Sebagai pertimbangan, untuk MB-LBP dengan skala  $3 \times 3$  direkomendasikan  $N = 58$  dan untuk MB-LBP dengan skala diatas 3, direkomendasikan  $N = 63$  [8]. Tahap selanjutnya adalah melakukan *labelling* pada vektor fitur berdasarkan hasil *ranking* sebelumnya, yang dapat didefinisikan pada Persamaan 2.3 berikut :

$$u_s(x, y) = \begin{cases} Indeks_s[f_s(x, y)], & \text{if } f_s(x, y) \in SEMB - LBP_s \\ N, & \text{otherwise} \end{cases} \quad (2.3)$$

Dimana  $Indeks_s[f_s(x, y)]$  adalah indeks dari  $f_s(x, y)$  pada sekumpulan SEMB-LBP (mulai dari 0).

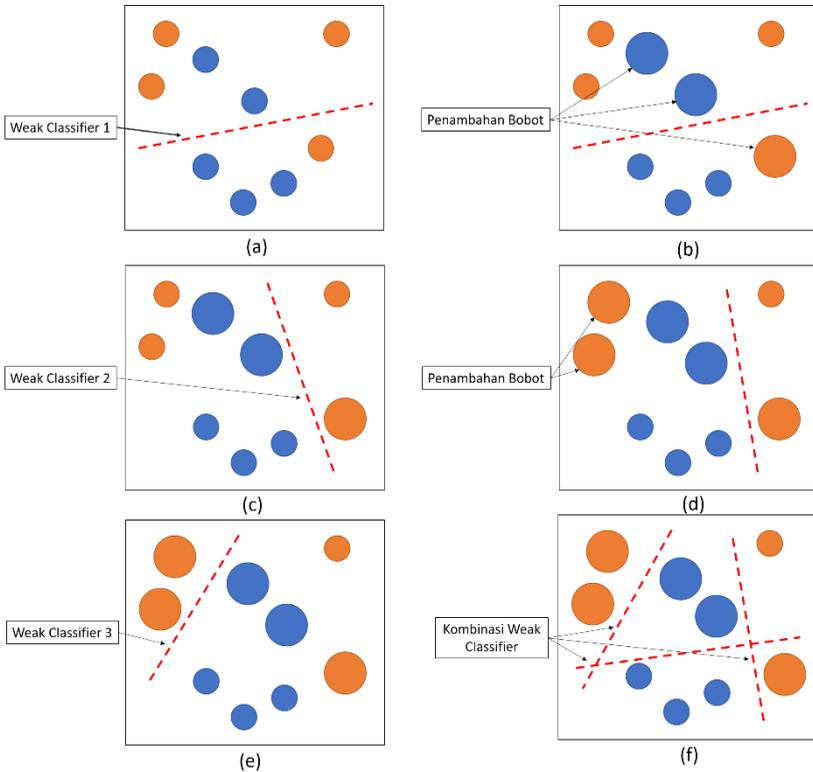
### 2.3. AdaBoost Learning

*Boosting* merupakan metode penggabungan banyak *classifier* guna mendapatkan performa yang lebih baik dibandingkan dengan performa tiap *classifier* dasarnya [9]. Salah satu metode *boosting* yang

sering digunakan adalah *AdaBoost*. Metode ini pertama kali dikenalkan oleh Yoav Freund dan Robert Schapire.

*AdaBoost* mampu beradaptasi dengan persebaran data dan mendapatkan hasil klasifikasi yang optimal dengan error yang kecil. Pada algoritma *AdaBoost*, masing-masing *classifier* akan diberikan bobot tertentu, dimana bobot tersebut bergantung dari performa *classifier* sebelumnya. Ketika seluruh *classifier* dasar telah dilatih, semuanya akan dikombinasikan untuk menghasilkan sebuah *classifier* akhir yang terbaik. Pada Gambar 2.3 (a), dilakukan proses pelatihan *weak classifier* 1 terhadap data yang ada. Selanjutnya, pada Gambar 2.3 (b) diketahui terdapat tiga buah titik yang salah diklasifikasikan, sehingga ketiga titik tersebut diberikan kenaikan bobot. Kenaikan bobot tersebut kemudian digunakan sebagai pembelajaran pada proses pelatihan *weak classifier* 2 yang terlihat pada Gambar 2.3 (c). Kembali dilakukan tahap pengecekan *error* dan perubahan bobot seperti pada Gambar 2.3 (d). Selanjutnya *weak classifier* 3 belajar untuk mengklasifikasikan data berdasarkan bobot baru dari *weak classifier* 2. Sehingga diperoleh klasifikasi baru seperti pada Gambar 2.3 (e). Dan terakhir, dilakukan proses penggabungan seluruh *weak classifier* yang ada untuk didapatkan sebuah *classifier* akhir yang tangguh seperti pada Gambar 2.3 (f).

*AdaBoost* pada awal diperkenalkan merupakan klasifikasi yang dikhususkan untuk data yang memiliki dua kelas saja. Namun, pada realita yang ada, data pada umumnya memiliki kelas yang lebih dari dua, terlebih untuk sistem pengenalan wajah. Oleh karena itu dikembangkan *AdaBoost* yang mampu menangani kelas dengan jumlah lebih dari dua [9]. Tidak jauh beda dengan *AdaBoost* dua kelas, *AdaBoost* multi kelas ini juga melakukan kombinasi beberapa *weak classifier*.



**Gambar 2.2 (a) Weak Classifier 1 (b) Penambahan Bobot 1 (c) Weak Classifier 2 (d) Penambahan Bobot 2 (e) Weak Classifier 3 (f) Final Classifier**

Hal pertama yang dilakukan pada proses klasifikasi menggunakan *AdaBoost* adalah dengan menginisialisasi bobot untuk masing-masing data seperti pada Persamaan 2.4

$$w_i = \frac{1}{n}, i = 1, 2, \dots, n \quad (2.4)$$

$N$  pada Persamaan 2.4 diatas menyatakan jumlah dari keseluruhan data. Selanjutnya masuk pada tahap iterasi. Di masing-masing iterasi

akan dilakukan lima proses, mulai dari proses pembelajaran hingga normalisasi bobot. Pada proses pembelajaran, terdapat sebuah *weak classifier*  $T(x)$  yang digunakan untuk membuat model berdasarkan bobot yang telah didefinisikan sebelumnya. Kemudian dihitung nilai *error* dari masing-masing *weak classifier* dengan Persamaan 2.5

$$err^{(m)} = \sum_{i=1}^n w_i (c_i \neq T^{(m)}(x_i)) / \sum_{i=1}^n w_i \quad (2.5)$$

$err^{(m)}$  merupakan nilai *error* dari *weak classifier* ke- $m$ . Variabel  $c_i$  merupakan kelas aktual dari data  $i$ . Dari Persamaan 2.5 tersebut dapat diketahui terjadi perhitungan jumlah dari perkalian bobot dengan kesalahan klasifikasi yang kemudian dibagi dengan jumlah keseluruhan bobot. Selanjutnya dilakukan perhitungan *stage value*  $\alpha^{(m)}$  yang dinyatakan pada Persamaan 2.6

$$\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1) \quad (2.6)$$

Semakin besar nilai dari *stage value*, maka semakin akurat model yang dihasilkan dan memberikan kontribusi yang cukup besar terhadap prediksi akhir. Setelah didapatkan nilai *stage value*, selanjutnya proses perhitungan ulang bobot dengan Persamaan 2.7

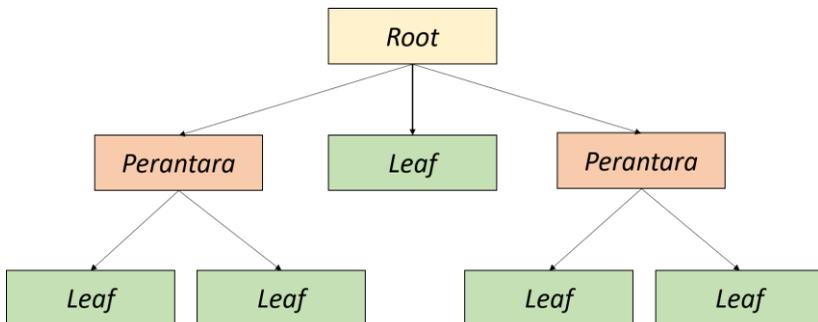
$$w_i \leftarrow w_i \cdot \exp\left(\alpha^{(m)} (c_i \neq T^{(m)}(x_i))\right), i = 1, 2, \dots, n \quad (2.7)$$

Perhitungan ulang bobot ini dilakukan untuk memberikan bobot yang lebih pada data yang salah diprediksi dan memberikan bobot yang lebih sedikit untuk data yang telah diprediksi dengan benar. Barulah dilakukan tahap kelima yang merupakan tahap terakhir pada iterasi, yakni proses normalisasi bobot. Setelah seluruh iterasi selesai dilakukan, barulah dilakukan proses penggabungan *weak classifier* melalui Persamaan 2.8

$$C(x) = \arg \max_k \sum_{m=1}^M \alpha^{(m)} \cdot (T^{(m)}(x) = k) \quad (2.8)$$

## 2.4. Decision Tree

*Decision Tree* merupakan salah satu metode klasifikasi yang cukup populer digunakan karena kemudahannya untuk diinterpretasikan oleh manusia. *Decision Tree* terbentuk dari 3 buah elemen simpul yakni *root*, simpul perantara, dan *leaf*. Simpul *root* adalah titik awal pembangunan *Tree* dimana ia hanya memiliki sebuah *edge* keluar. Simpul perantara merupakan simpul yang berhubungan dengan suatu pengujian atau pertanyaan dimana ia memiliki sebuah *edge* masuk dan sebuah *edge* keluar. Sedangkan *leaf* adalah simpul yang memuat suatu keputusan akhir atau kelas target pada *Tree* dimana ia hanya memiliki sebuah *edge* masuk.



Gambar 2.3 Decision Tree

Untuk menentukan penggunaan atribut untuk *root*, simpul perantara atau *leaf*, digunakan sebuah ukuran *impurity* dari masing-masing atribut yang ada. Ukuran *impurity* tersebut dapat dihitung dengan berbagai metode yakni *Information Gain*, *Gain Ratio*, *Gini Index*, dan lain sebagainya [10]. Atribut-atribut yang digunakan haruslah bertipe kategorik, apabila atribut data berupa data kontinu maka perlu didiskritkan terlebih dahulu.

Pada sub bab ini, hanya akan dijelaskan metode *Gini Index* yang akan digunakan pada Tugas Akhir ini. *Gini Index* memiliki *range* nilai antara 0 hingga 1. Semakin rendah nilai *Gini Index* maka semakin besar ukuran kesamaannya. Persamaan untuk *Gini Index* dituliskan pada Persamaan 2.9 berikut

$$Gini(t) = 1 - \sum_{i=1}^m p_i^2 \quad (2.9)$$

Dimana  $t$  adalah atribut,  $m$  menyatakan banyaknya kelas dan  $p$  menyatakan probabilitas. Nantinya setelah *Gini Index* telah dihitung untuk masing-masing atribut, diambil atribut dengan nilai terbesar untuk dijadikan simpul. Begitu seterusnya hingga didapatkan simpul *leaf*.

## 2.5. Perhitungan Evaluasi

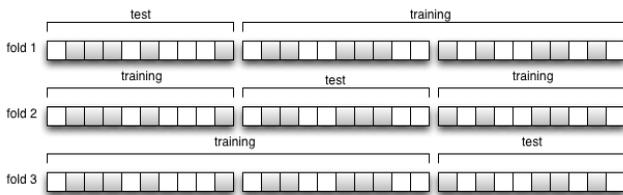
Terdapat berbagai parameter evaluasi yang digunakan untuk menilai apakah suatu sistem yang dibangun sudah memiliki hasil yang optimal. Salah satunya adalah menggunakan perhitungan nilai Akurasi. Akurasi dari suatu sistem adalah tingkat kedekatan pengukuran kuantitas terhadap nilai yang sebenarnya [11]. Perhitungan akurasi dapat dihitung melalui Persamaan 2.10

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

Dimana  $TP$  atau *True Positive* adalah data-data prediksi yang tepat diprediksikan pada kelas sesungguhnya. Sedangkan  $TN$  atau *True Negative* adalah data-data prediksi yang benar diprediksikan pada kelas yang memang bukan kelasnya.

## 2.6. K-Fold Cross Validation

Merupakan salah satu teknik validasi model dengan melakukan pembagian data menjadi beberapa bagian dan melakukannya sebanyak pengulangan tertentu. Data akan dibagi menjadi  $k$  sub bagian [12]. Sub bagian pertama akan menjadi data uji dan sub-sub bagian lainnya akan menjadi data latih pada iterasi pertama. Pada iterasi kedua, sub bagian kedua menjadi data uji dan sub bagian lainnya menjadi data latih. Begitu seterusnya hingga memenuhi iterasi sebanyak  $k$ . Nantinya nilai dari akurasi akan didapatkan dari hasil rata-rata akurasi masing-masing iterasi yang telah dilakukan.



**Gambar 2.4 K-fold Cross Validation**

*[Halaman ini sengaja dikosongkan]*

## BAB III PERANCANGAN SISTEM

Pada bab ini dijelaskan mengenai rancangan sistem perangkat lunak yang akan dibuat. Perancangan yang dijelaskan meliputi data dan proses. Data yang dimaksud adalah data yang akan diolah dalam sistem baik digunakan sebagai pembelajaran maupun pengujian sehingga tujuan tugas akhir ini bisa tercapai. Proses yaitu tahap-tahap yang ada dalam sistem sebagai pengolah data meliputi ekstraksi fitur menggunakan MB-LBP, seleksi fitur menggunakan SEMB-LBP dan klasifikasi menggunakan *AdaBoost*.

### 3.1. Data

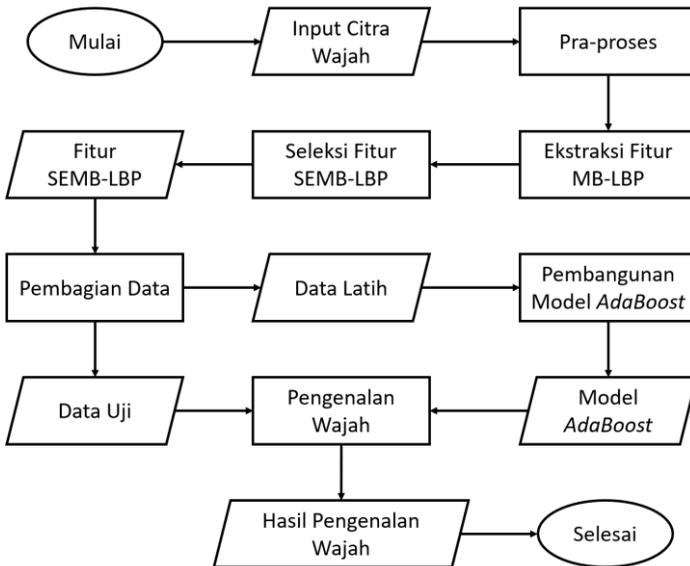
Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan sistem untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan. Dataset yang digunakan pada tugas akhir ini adalah *Collection of Facial Image : Faces94* yang dikelola oleh Dr.Libor Spacek dari Universitas Essex, Inggris [13]. Terdiri dari 153 individu yang berbeda yakni 20 wanita dan 133 laki-laki. Masing-masing individu diwakili oleh kurang lebih 20 citra dengan ukuran 180 x 200. Total keseluruhan dari citra yang digunakan pada tugas akhir ini adalah sebanyak 3059. Karena data yang digunakan sudah berupa citra wajah, sehingga tidak lagi diperlukan proses penentuan *Region of Interest* untuk kemudian dilakukan pemotongan gambar. Data citra ini nantinya hanya akan melalui proses perubahan dari citra RGB menjadi citra *grayscale*. Karena pada proses ekstraksi, perbandingan yang dilakukan menggunakan nilai *grayscale*-nya.



**Gambar 3.1 Collection of Facial Image : Faces94**

### 3.2. Perancangan Sistem

Pada sub bab ini akan dijelaskan secara runtut tahapan demi tahapan proses yang dilakukan pada sistem pengenalan wajah tugas akhir ini. Adapun proses yang dijalankan pada tugas akhir ini terdiri dari 4 tahap utama, yakni pra-proses, ekstraksi fitur, seleksi fitur, dan terakhir adalah pengenalan wajah atau klasifikasi. Diagram alir sistem dapat dilihat pada Gambar 3.2



**Gambar 3.2 Diagram Alir Sistem Pengenalan Wajah**

Proses pertama yang dilakukan adalah pra-proses yakni merubah citra wajah ke dalam bentuk *grayscale*. Selanjutnya masuk pada tahap kedua yakni ekstraksi fitur MB-LBP. Masing-masing citra nantinya akan dibagi menjadi beberapa bagian bergantung dari ukuran skala  $s$  yang digunakan. Dari bagian tersebut perlu dilakukan pembagian lagi menjadi beberapa blok. Barulah akan dilakukan perbandingan dari nilai masing-masing blok. Setelah perbandingan dan konversi selesai dilakukan, dibuatlah histogram dari fitur MB-

LBP. Tahap ketiga yakni implementasi SEMB-LBP sebagai metode seleksi fitur. Pada sebuah citra wajah, terdapat kemungkinan adanya hasil yang redundan. Oleh karena itu diperlukan metode seleksi fitur untuk menghilangkannya. Setelah histogram MB-LBP dibuat, selanjutnya dilakukan pengurutan berdasarkan jumlah frekuensinya secara *descending*. Pada tahap ini akan dipilih sebanyak N fitur dengan nilai frekuensi tertinggi. Tahap berikutnya adalah melakukan pembagian data dari hasil seleksi fitur menjadi data training dan testing menggunakan metode *K-fold Cross Validation*. Data latih akan digunakan untuk membuat model. Dan data uji akan digunakan sebagai pengujian model yang telah dibangun dari data latih pada tahap keempat yakni pengenalan wajah atau klasifikasi. Output yang dihasilkan pada tahap pengenalan wajah berupa label output nama dari citra wajah pada data uji. Evaluasi dari proses klasifikasi berupa nilai akurasi. Selain meninjau akurasi dari metode MB-LBP, dilakukan juga perbandingan dengan akurasi dari LBP original. Dengan begitu nantinya akan dapat diketahui keberhasilan dari metode yang ditawarkan.

### 3.3. Perancangan Proses

Pada sub-bab ini dijelaskan secara lebih detail terkait empat proses utama dalam sistem pengenalan wajah pada tugas akhir ini.

#### 3.3.1. Pra-Proses

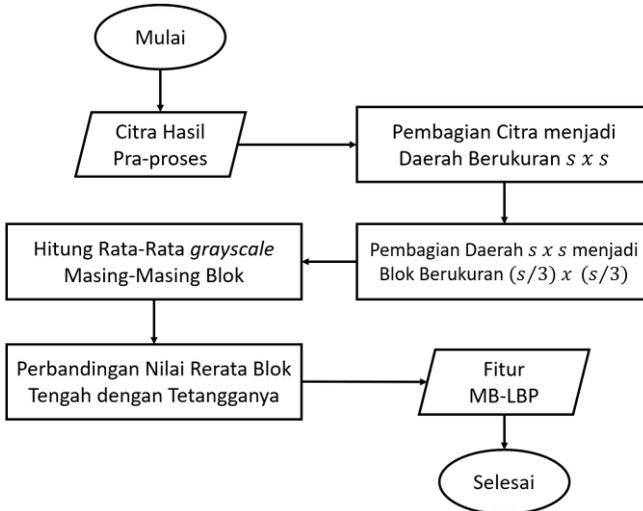
Pada tahap ini, proses yang dilakukan hanyalah merubah citra wajah yang semula RGB menjadi citra *grayscale*. Sebanyak 3059 citra wajah akan menjadi input pada pra-proses ini.



**Gambar 3.3 (a) Citra Asli (b) Citra *Grayscale***

### 3.3.2. Ekstraksi Fitur MB-LBP

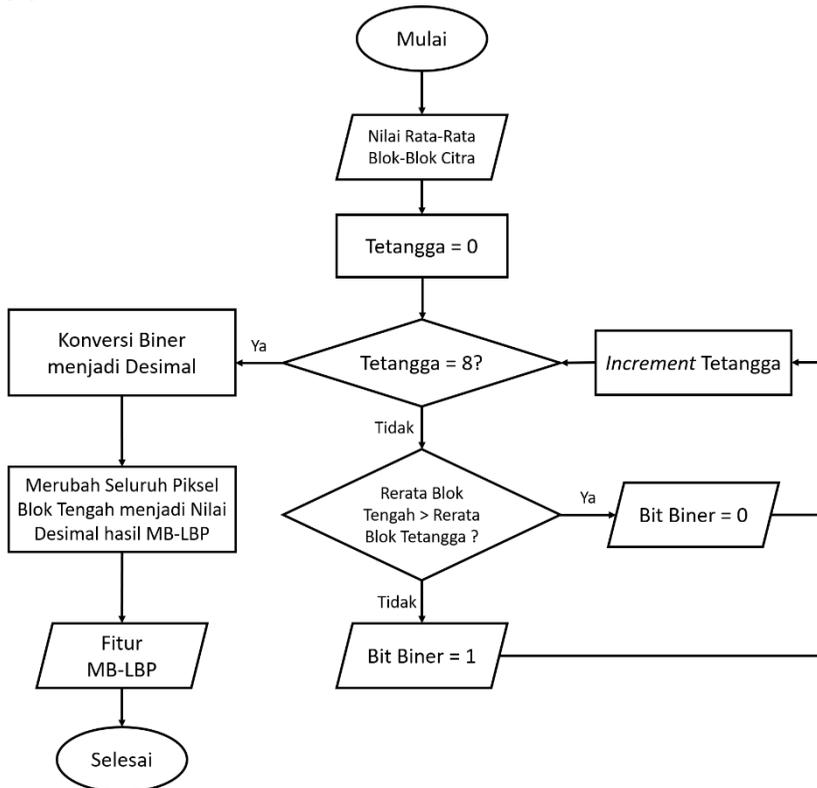
Ekstraksi fitur MB-LBP terdiri dari beberapa proses utama yang dapat diamati dari diagram alir pada Gambar 3.4 berikut



**Gambar 3.4 Diagram Alir Ekstraksi Fitur MB-LBP**

Setelah citra dirubah dalam bentuk *grayscale* pada pra-proses, selanjutnya citra tersebut menjadi masukan pada tahap ekstraksi fitur. Seperti yang telah dijelaskan pada Bab II, MB-LBP memiliki ukuran skala  $s$  yang beragam dengan ketentuan nilainya merupakan kelipatan dari 3. Dari skala tersebut, citra dibagi menjadi beberapa blok dengan ukuran skala  $s / 3$ . Misalkan skala yang digunakan adalah  $15 \times 15$ , maka luas masing-masing blok adalah  $5 \times 5$ . Selanjutnya akan dilakukan perhitungan rata-rata dari masing-masing blok. Inilah yang membedakan MB-LBP dengan LBP original yakni melakukan perbandingan terhadap nilai rata-rata *grayscale* blok bukan nilai *grayscale* dari masing-masing piksel. Setelah rata-rata dari masing-masing blok dihitung, tahap berikutnya yang dilakukan adalah melakukan perbandingan nilai rerata masing-masing blok tersebut.

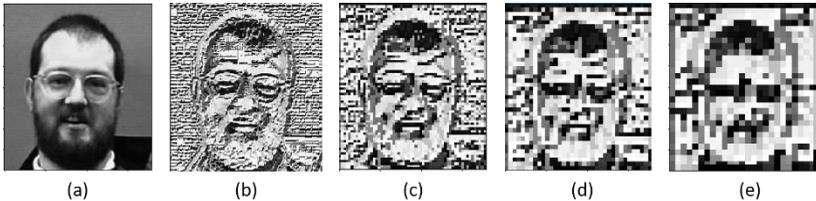
Adapun detail proses dari perbandingan blok dapat dilihat pada Gambar 3.5 berikut :



**Gambar 3.5 Diagram Alir Perbandingan Blok**

Sama seperti LBP original, perbandingan dilakukan terhadap 8 tetangga yakni atas, bawah, kanan, kiri dan diagonal dari sebuah blok tengah. Apabila nilai rerata blok tengah lebih besar dari nilai rerata blok tetangganya, maka nilai biner yang didapat adalah 0. Dan sebaliknya, apabila nilai rerata blok tengah lebih kecil dari atau sama dengan nilai rerata blok tetangganya, maka nilai biner yang didapat adalah 1. Perbandingan dilakukan searah jarum jam dimulai dari tetangga pojok kiri atas. Apabila blok tengah merupakan tepi atau

dapat disimpulkan bahwa blok tengah tidak memiliki tetangga di beberapa posisi tertentu, maka nilai rerata blok tetangganya dianggap nol. Setelah seluruh perbandingan terhadap 8 tetangganya telah selesai dilakukan, maka langkah berikutnya adalah melakukan konversi nilai biner yang didapatkan dari hasil perbandingan. Hasil dari perbandingan nilai *grayscale* nantinya akan berupa 8bit nilai biner. Nilai biner tersebut dikonversikan ke dalam desimal untuk selanjutnya akan menggantikan nilai *grayscale* dari blok tengah. Hasil akhir dari proses ekstraksi fitur ini berupa fitur-fitur MB-LBP yang selanjutnya akan menjadi input pada tahap seleksi fitur.



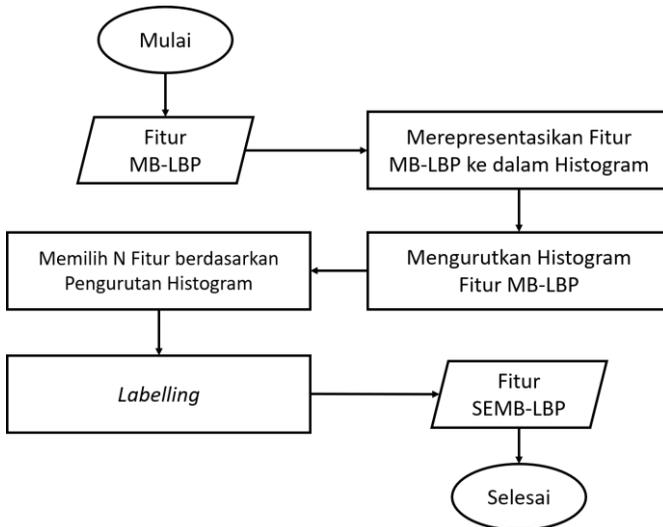
**Gambar 3.6 (a) Citra *grayscale* (b) Citra MB-LBP 3x3 (c) Citra MB-LBP 9x9 (d) Citra MB-LBP 15x15 (e) Citra MB-LBP 21x21**

Dapat diamati pada Gambar 3.6, semakin besar skala dari MB-LBP maka akan semakin berkurang *noise*-nya dan membuat hasil representasi lebih mampu menangani kesalahan dalam klasifikasi nantinya. Akan tetapi perlu diketahui juga, bahwa semakin besar skala maka akan semakin besar pula informasi pembeda yang hilang karena pada area yang besar memiliki kesamaan nilai. Oleh karena itu, skala dari MB-LBP perlu diperhatikan karena akan sangat berpengaruh pada proses pengenalan wajah nantinya.

### 3.3.3. Seleksi Fitur SEMB-LBP

Seperti yang telah dijelaskan sebelumnya, LBP memiliki sebuah metode seleksi fitur yang disebut pola *uniform*. Pola *uniform* pada LBP original mengacu pada masing-masing piksel sehingga tidak dapat diterapkan pada MB-LBP yang berbasis blok yang terdiri dari bukan hanya satu piksel melainkan banyak piksel. Oleh karena itu

muncul sebuah metode SEMB-LBP yang digunakan sebagai seleksi fitur pada MB-LBP. Berikut adalah tahapan SEMB-LBP yang representasikan ke dalam diagram alir pada Gambar 3.7

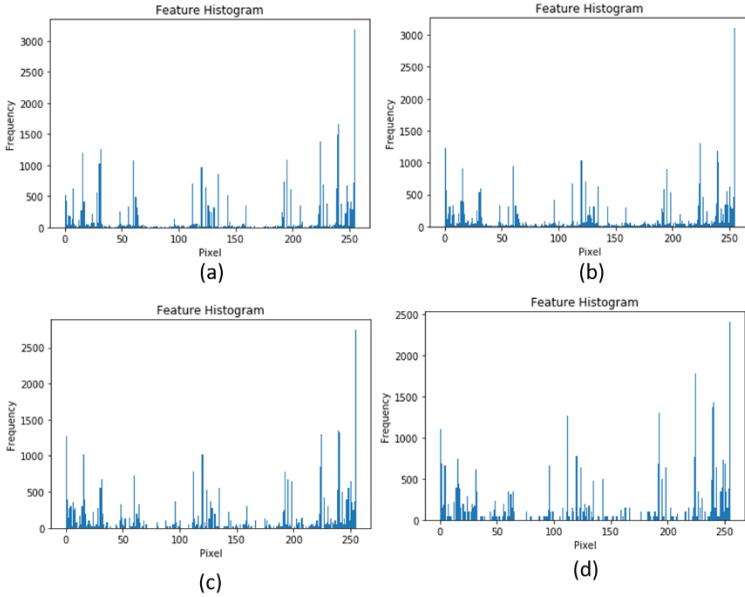


**Gambar 3.7 Diagram Alir Seleksi Fitur SEMB-LBP**

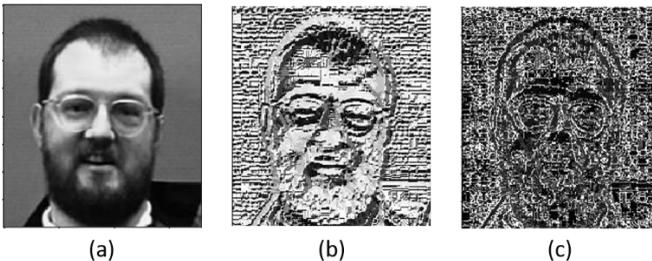
Fitur MB-LBP dari hasil ekstraksi fitur akan menjadi masukan pada tahap seleksi fitur SEMB-LBP ini. Tahap pertama yang dilakukan adalah merepresentasikan fitur MB-LBP menjadi histogram dengan bin sebanyak 256. Hal tersebut dikarenakan nilai *grayscale* memiliki *range* mulai dari 0 hingga 255. Bentuk histogram dari fitur MB-LBP dalam beberapa skala  $s$  dapat dilihat pada Gambar 3.7

Tahap berikutnya adalah mengurutkan histogram berdasarkan nilai frekuensi terbesar. Dari histogram yang telah diurutkan, dipilih  $N$  fitur teratas. Nilai  $N$  disini dapat dirubah sesuai keinginan dan nantinya akan diuji coba guna mendapatkan hasil klasifikasi yang optimal. Setelah diperoleh  $N$  fitur teratas, kemudian dilakukan proses *labelling* pada fitur MB-LBP. Misalkan saja suatu nilai pada vektor fitur MB-LBP memiliki nilai 128. Dan diketahui bahwa piksel 128 memiliki peringkat 2 dari histogram yang telah diurutkan, maka pada

fitur MB-LBP setiap nilai 128 dirubah menjadi 2. Apabila sebuah nilai tidak termasuk pada peringkat N teratas, maka rubah nilainya menjadi N. Contoh dari citra hasil labelling dapat dilihat pada Gambar 3.9



**Gambar 3.8 (a) Histogram MB-LBP 3x3 (b) Histogram MB-LBP 9x9  
 (c) Histogram MB-LBP 15x15 (d) Histogram MB-LBP 21x21**

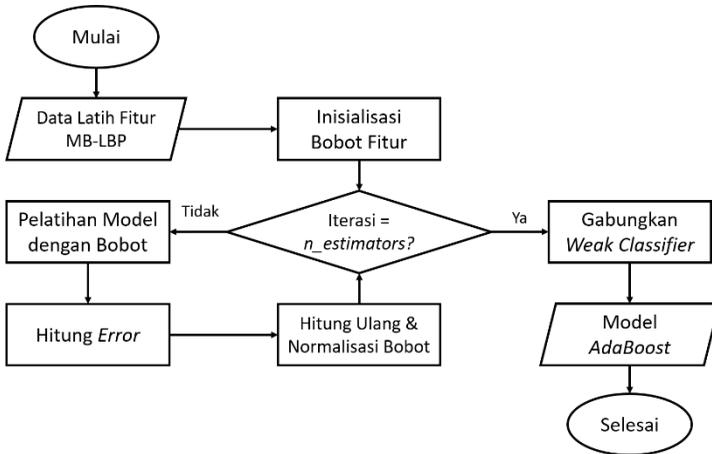


**Gambar 3.9 (a) Citra *Grayscale* (b) Citra MB-LBP 3x3  
 (c) Citra SEMB-LBP 3x3**

Walaupun MB-LBP 3x3 secara teknis melakukan perbandingan piksel sama halnya dengan LBP original, namun ia menggunakan metode seleksi fitur SEMB-LBP yang dapat dilihat pada Gambar 3.9(c).

### 3.3.4. Klasifikasi AdaBoost

Metode yang digunakan pada klasifikasi tugas akhir ini adalah *AdaBoost*. Diagram alir dari *AdaBoost classifier* ditunjukkan pada Gambar 3.10. Dari hasil SEMB-LBP didapatkan vektor fitur baru yang berbeda dengan fitur MB-LBP. Selanjutnya fitur tersebut dirubah kedalam bentuk histogram untuk menjadi masukan pada proses pembagian data. Setelah data terbagi menjadi data latih dan data uji, barulah data latih ini menjadi input pada tahap pembuatan model. *Classifier* dasar (*base estimator*) yang digunakan pada *AdaBoost* adalah *Decision Tree*. Kedalaman dari *tree*, banyaknya *weak classifier* ( $n\_estimators$ ) yang digunakan, dan nilai *learning rate* juga akan diuji coba agar mendapatkan model dengan akurasi terbaik. Proses pelatihan akan dilakukan sebanyak iterasi yang ditentukan. Bobot untuk masing-masing data terlebih dulu diinisialisasi sebelum memasuki tahap iterasi. Kemudian dilakukan proses pelatihan dan dihitung nilai *error* dari *weak classifier*. *Weak classifier* akan belajar dari nilai *error* yang dihasilkan pelatihan *weak classifier* pada iterasi sebelumnya. Kemudian dilakukan perhitungan bobot dan normalisasi nilai bobot. Setelah iterasi selesai, seluruh model yang telah dilatih kemudian digabungkan sehingga didapatkan model terbaik.



**Gambar 3.10 Diagram Alir Pembuatan Model**

## BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan sistem yang telah dijelaskan pada bab sebelumnya. Implementasi kode program dilakukan sepenuhnya menggunakan bahasa pemrograman *Python*.

### 4.1. Lingkungan Implementasi

Spesifikasi perangkat keras dan perangkat lunak yang digunakan ditampilkan pada Tabel 4.1.

**Tabel 4.1 Spesifikasi Perangkat**

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i7-4510U CPU @ 2.00 GHz Memori: 12.00 GB
Perangkat lunak	Sistem Operasi: Windows 10 Perangkat Pengembang: Python Perangkat Pembantu: Spyder

### 4.2. Implementasi

Sub-bab implementasi ini menjelaskan tentang implementasi proses yang sudah dijelaskan pada Bab III Perancangan Sistem.

#### 4.2.1. Implementasi Tahap Pra-proses

Sub bab ini menjelaskan implementasi dari pra-proses yang ditunjukkan pada Kode Sumber 4.1. Seperti yang telah dijelaskan pada Bab III Perancangan Sistem, tahap pre-proses yang dilakukan hanyalah merubah citra wajah ke dalam bentuk *grayscale*. Hal pertama yang dilakukan adalah melakukan *load* data dengan fungsi *cv2.imread()*. Setelah citra wajah berhasil dimuat, selanjutnya citra dirubah ke dalam bentuk *grayscale* menggunakan fungsi *cv2.cvtColor()* dengan menggunakan parameter *cv2.COLOR\_BGR2GRAY*.

Baris ke 4 dijalankan fungsi *shape()* untuk mengetahui ukuran dari citra wajah. Pada baris ke 5 dan 6 dilakukan proses preparasi data kelas. Variabel *file* berisi *datapath* beserta nama dari *file* yang dimuat. Sedangkan nama dari sebuah *file* sendiri berisi nomor urut data diikuti dengan nama kelas dari citra wajah tersebut. Misalkan saja *faces/1\_tony.jpeg*, maka untuk mendapatkan nama kelas *tony* diperlukan proses pemisahan *string* menggunakan fungsi *split()*. Setelah tahap pra-proses dijalankan, diperoleh tiga keluaran yakni nama kelas dalam variabel *cls*, citra *grayscale* dalam variabel *gray* dan ukuran dari citra dalam variabel *size*.

1.	<code>def preprocess(file) :</code>
2.	<code>img = cv2.imread(file)</code>
3.	<code>gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)</code>
4.	<code>size = gray.shape</code>
5.	<code>cls = file.split('\ ')[1]</code>
6.	<code>cls = cls.split('.')[0]</code>
7.	<code>return cls, gray, size</code>

**Kode Sumber 4.1 Pra-proses**

#### 4.2.2. Implementasi Ekstraksi Fitur MB-LBP

Pada sub bab ini akan dijelaskan tentang implementasi dari ekstraksi fitur MB-LBP yang terdiri dari 3 buah fungsi yakni fungsi untuk melakukan perbandingan nilai, fungsi untuk menghitung nilai rata-rata pada suatu blok, dan fungsi utama ekstraksi fitur. Fungsi pertama adalah *check()* yang ditunjukkan pada Kode Sumber 4.2. Hal yang dilakukan pada fungsi ini adalah membandingkan nilai *grayscale* dari blok tengah dengan blok tetangganya. Keluaran dari fungsi ini berupa angka biner 1 atau 0.

1.	<code>def check(center, neighbour):</code>
2.	<code>if center &gt; neighbour:</code>
3.	<code>return 0</code>
4.	<code>else :</code>
5.	<code>return 1</code>

**Kode Sumber 4.2 Perbandingan Ketetanggaan**

Fungsi kedua adalah fungsi perhitungan rata-rata nilai *grayscale* pada suatu blok yang ditunjukkan pada Kode Sumber 4.3. Terdapat delapan macam kondisi perhitungan rata-rata berdasarkan posisi blok tetangga terhadap blok tengah. Delapan kondisi tersebut terdiri dari *top(t)*, *diagonal right top(drt)*, *right(r)*, *diagonal right bottom(drb)*, *bottom(b)*, *diagonal left bottom(dlb)*, *left(l)*, dan *diagonal left top(dlt)*.

1.	<code>def count_mean(i,j,s,img,types):</code>
2.	<code>    if types == 't':</code>
3.	<code>        return int(np.mean(img[i-s:i,j:j+s]))</code>
4.	<code>    elif types == 'drt':</code>
5.	<code>        return int(np.mean(img[i-s:i,j+s:j+(2*s)]))</code>
6.	<code>    elif types == 'r':</code>
7.	<code>        return int(np.mean(img[i:i+s,j+s:j+(2*s)]))</code>
8.	<code>    elif types == 'drb':</code>
9.	<code>        return int(np.mean(img[i+s:i+(2*s),j+s:j+(2*s)]))</code>
10.	<code>    elif types == 'b':</code>
11.	<code>        return int(np.mean(img[i+s:i+(2*s),j:j+s]))</code>
12.	<code>    elif types == 'dlb':</code>
13.	<code>        return int(np.mean(img[i+s:i+(2*s),j-s:j]))</code>
14.	<code>    elif types == 'l':</code>
15.	<code>        return int(np.mean(img[i:i+s,j-s:j]))</code>
16.	<code>    else :</code>
17.	<code>        return int(np.mean(img[i-s:i,j-s:j]))</code>

**Kode Sumber 4.3 Perhitungan Rata-Rata**

Fungsi ketiga adalah fungsi utama ekstraksi fitur MB-LBP yang ditunjukkan oleh Kode Sumber 4.4. Implementasi dari ekstraksi fitur MB-LBP dibedakan menjadi dua kondisi utama yakni apabila skalanya adalah 3 dan skala dengan nilai lebih dari 3. Hal tersebut dikarenakan MB-LBP berskala 3x3 berjalan dengan kaidah perbandingan piksel sama seperti LBP original sedangkan skala

dengan nilai yg lebih besar dari 3 berjalan dengan kaidah perbandingan rata-rata nilai blok.

MB-LBP dengan skala tiga ditunjukkan pada baris 3 hingga 27. Pada kode tersebut, terdapat sembilan kondisi berdasarkan posisi dari piksel. Delapan kondisi dari sembilan kondisi yang ada, dibutuhkan untuk menangani piksel yang terletak pada bagian tepi citra dimana bisa dipastikan ia tidak memiliki tetangga pada posisi tertentu. Jika hal tersebut terjadi, maka nilai piksel dari tetangganya dianggap nol sehingga nantinya nilai biner yang didapat juga 0. Misalkan saja, kondisi pertama pada baris 9 dikhususkan untuk piksel yang terletak pada posisi (0,0) atau pojok kiri atas dari citra. Hal tersebut membuat ia tidak memiliki tetangga bagian kiri atas, atas, kanan atas, kiri, dan kiri bawah, sehingga pada posisi-posisi tersebut otomatis bernilai biner 0. Satu kondisi dari sembilan kondisi yang ada dilakukan untuk posisi piksel yang bukan tepian. Selanjutnya, pada baris 28 hingga 30 diimplementasikan konversi 8 bit nilai biner yang telah dihasilkan dari proses ekstraksi fitur MB-LBP berskala 3. Pada baris 31 dilakukan penyimpanan nilai desimal hasil konversi menggantikan nilai piksel asli dari citra.

MB-LBP dengan nilai skala lebih dari 3 ditunjukkan pada baris 32 hingga 67. Pada kode tersebut, terdapat 14 kondisi berdasarkan posisi blok. Sama halnya dengan MB-LBP skala 3x3, pada MB-LBP skala besar juga memiliki kondisi untuk posisi area yang berada pada tepi sebanyak delapan kondisi. Sedangkan lima kondisi lainnya digunakan untuk melakukan teknik redundansi. Hal tersebut diperlukan karena pembagian blok dengan ukuran (skala/3) tidak selalu pas dengan ukuran citra. Sehingga pada bagian tepi kanan, tepi bawah, pojok kanan atas, pojok kanan bawah dan pojok kiri bawah diperlukan penanganan khusus untuk melakukan redundansi apabila ukuran blok tidak cukup. Dan satu kondisi digunakan untuk menangani blok selain tepi. Selanjutnya pada baris 68 hingga 70 dilakukan konversi biner menjadi desimal. Dan pada baris 71 hingga 73 dilakukan penyimpanan nilai desimal hasil konversi menggantikan nilai piksel-piksel yang berada pada blok. Karena sistem ekstraksi fitur memiliki batasan untuk nilai skala, yakni harus berkelipatan 3, maka

diperlukan *error handling* apabila nilai skala yang diinputkan bukan kelipatan dari 3 yang ditunjukkan pada baris 75.

1.	<code>def extraction(img, scale, size):</code>
2.	<code>    extract_img = np.full((size[0], size[1]), -1)</code>
3.	<code>    if scale == 3:</code>
4.	<code>        for i in range(size[0]):</code>
5.	<code>            for j in range(size[1]):</code>
6.	<code>                center = img[i][j]</code>
7.	<code>                binary = []</code>
8.	<code>                if i == 0 :</code>
9.	<code>                    if j == 0 :</code>
10.	<code>                        binary = [0,0,0,</code>
11.	<code>                            check(center, img[0][1]),</code>
12.	<code>                            check(center, img[1][0]),</code>
13.	<code>                            check(center, img[1][1]),</code>
14.	<code>                                0,0]</code>
15.	<code>                                ...</code>
16.	<code>                                ...</code>
17.	<code>                                ...</code>
18.	<code>                    else :</code>
19.	<code>                        binary =</code>
20.	<code>                            [check(center, img[i-1][j-1]),</code>
21.	<code>                            check(center, img[i-1][j]),</code>
22.	<code>                            check(center, img[i-1][j+1]),</code>
23.	<code>                            check(center, img[i][j+1]),</code>
24.	<code>                            check(center, img[i+1][j+1]),</code>
25.	<code>                            check(center, img[i+1][j]),</code>
26.	<code>                            check(center, img[i+1][j-1]),</code>
27.	<code>                            check(center, img[i][j-1])]</code>
28.	<code>                        value = 0</code>

29.	<code>for k in range(len(binary)):</code>
30.	<code>value += (binary[k] * (2**(7-k)))</code>
31.	<code>extract_img[i][j] = value</code>
32.	<code>elif scale &gt; 3 and scale % 3 == 0 :</code>
33.	<code>s = int(scale/3)</code>
34.	<code>for i in range(size[0]):</code>
35.	<code>for j in range(size[1]):</code>
36.	<code>binary = []</code>
37.	<code>center = int(np.mean(img[i:i+s,j:j+s]))</code>
38.	<code>if extract_img[i][j] != -1:</code>
39.	<code>if j == (size[1]-s):</code>
40.	<code>if i == 0:</code>
41.	<code>binary = [0,0,0,0,0</code>
42.	<code>check(center,</code>
43.	<code>mean(i,j,s,img,'b')),</code>
44.	<code>check(center,</code>
45.	<code>mean(i,j,s,img,'dlb')),</code>
46.	<code>check(center,</code>
47.	<code>mean(i,j,s,img,'l'))]</code>
48.	<code>...</code>
49.	<code>...</code>
50.	<code>...</code>
51.	<code>else :</code>
52.	<code>binary = check(center,</code>
53.	<code>mean(i,j,s,img,'dlt')),</code>
54.	<code>check(center,</code>
55.	<code>mean(i,j,s,img,'t')),</code>
56.	<code>check(center,</code>
57.	<code>mean(i,j,s,img,'drt')),</code>
58.	<code>check(center,</code>

59.	<code>mean(i, j, s, img, 'r'),</code>
60.	<code>check(center,</code>
61.	<code>mean(i, j, s, img, 'drb'),</code>
62.	<code>check(center,</code>
63.	<code>mean(i, j, s, img, 'b'),</code>
64.	<code>check(center,</code>
65.	<code>mean(i, j, s, img, 'dlb'),</code>
66.	<code>check(center,</code>
67.	<code>mean(i, j, s, img, 'l')]]</code>
68.	<code>value = 0</code>
69.	<code>for k in range(len(binary)):</code>
70.	<code>value += (binary[k] * (2**(7-k)))</code>
71.	<code>for k in range(s):</code>
72.	<code>for l in range(s):</code>
73.	<code>extract_img[i+k][j+l] = value</code>
74.	<code>else :</code>
75.	<code>sys.exit()</code>
76.	<code>return extract_img</code>

**Kode Sumber 4.4 Ekstraksi Fitur MB-LBP**

### 4.2.3. Implementasi Seleksi Fitur SEMB-LBP

Pada sub bab ini akan dijelaskan mengenai implementasi dari seleksi fitur SEMB-LBP yang ditunjukkan pada Kode Sumber 4.5. Proses pertama yang dilakukan pada proses seleksi fitur adalah merepresentasikan fitur MB-LBP kedalam histogram. Hal tersebut dilakukan pada baris 4. Selanjutnya, histogram tersebut akan diurutkan berdasarkan nilai frekuensi tertinggi yang diimplementasikan pada baris 6 hingga 9. Tahap pemilihan N fitur dengan nilai frekuensi tertinggi sekaligus proses *labelling* dilakukan pada baris 10 hingga 19. Dari proses *labelling* akan dihasilkan fitur SEMB-LBP yang kemudian direpresentasikan ke dalam bentuk histogram (baris 20-21) untuk menjadi masukan pada klasifikasi.

1. <code>def selection(extract_img, N, size):</code>
2. <code>select_img = np.full((size[0],size[1],-1)</code>
3. <code>flatten_extract = extract_img.flatten()</code>
4. <code>histogram = np.histogram(flatten_extract,256)</code>
5. <code>hist = []</code>
6. <code>for i in range(256):</code>
7. <code>hist.append([i,int(histogram[0][i]))]</code>
8. <code>desc_hist = sorted(hist,key=lambda x:x[1],</code>
9. <code>reverse=True)</code>
10. <code>for i in range(int N):</code>
11. <code>for j in range(size[0]):</code>
12. <code>for k in range(size[1]):</code>
13. <code>if extract_img[0][1] ==</code>
14. <code>(desc_hist[i])[0]:</code>
15. <code>select_img[j][k] = i</code>
16. <code>for i in range(size[0]):</code>
17. <code>for j in range(size[1]):</code>
18. <code>if select_img[i][j] == -1:</code>
19. <code>select_img[i][j] = int(N)</code>
20. <code>flatten_select = select_img.flatten()</code>
21. <code>histogram = np.histogram(flatten_select,N)</code>
22. <code>return histogram[0]</code>

**Kode Sumber 4.5 Seleksi Fitur SEMB-LBP**

#### 4.2.4. Implementasi AdaBoost

Pada sub bab ini dijelaskan terkait implementasi dari klasifikasi menggunakan *AdaBoost* sebagai *classifier*. Implementasi *AdaBoost* dilakukan menggunakan *library scikit learn*. Pada baris 1 dan 2 dilakukan proses pemuatan data yang selanjutnya akan dibagi menggunakan *K-fold Cross Validation* dengan *k* sebanyak 10. Proses deklarasi fungsi *K-fold* dilakukan pada baris 3 dan 4. Selanjutnya,

pada baris 5 dilakukan deklarasi *classifier* yang digunakan untuk *boosting* pada *AdaBoost* yakni menggunakan *Decision Tree*. Deklarasi untuk *AdaBoost* dilakukan pada baris 7. Proses klasifikasi dilakukan pada baris 11 hingga 21 menggunakan metode *K-fold Cross Validation*. Hasil akhir dari klasifikasi adalah nilai rata-rata akurasi dari iterasi yang telah dilakukan serta waktu rata-rata yang dibutuhkan pada setiap iterasi.

1. data = np.load("result.npy")
2. cls = np.load("class.npy")
3. split = 10
4. skf = KFold(n_splits=split)
5. dtree = DecisionTreeClassifier(criterion='gini',
6.       max_depth=15)
7. adabst = AdaBoostClassifier(base_estimator=dtree,
8.       n_estimators=500,learning_rate=0.1)
9. acc = []
10. times = []
11. for train_data, test_data in skf.split(data,cls):
12.     start_time = time.time()
13.     adabst.fit(data[train_data], cls[train_data])
14.     Y_actual = cls[test_data]
15.     Y_pred = adabst.predict(data[test_data])
16.     accuracy = accuracy_score.(cls[test_data],
17.         Y_pred)
18.     accuracy_average = np.mean(accuracy)
19.     acc.append(accuracy_average)
20.     elapsed_time = time.time() - start_time
21.     times.append(elapsed_time)
22. acc_final = np.mean(acc)
23. times_final = np.mean(times)

**Kode Sumber 4.6 Klasifikasi AdaBoost**

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Pada bab ini akan dijelaskan hasil uji coba dan evaluasi program yang telah selesai diimplementasikan.

#### **5.1. Lingkungan Pengujian**

Lingkungan uji coba yang akan digunakan adalah,

1. Perangkat Keras  
Prosesor: Intel® Core™ i7-4510U CPU @ 2.00 GHz Memori: 12.00 GB.  
Sistem Operasi: 64-bit .
2. Perangkat Lunak  
Sistem Operasi: Windows 10  
Perangkat Pengembang: Python  
Perangkat Pembantu: Spyder

#### **5.2. Data Uji Coba**

Data yang digunakan untuk uji coba pada sistem pengenalan wajah tugas akhir ini diperoleh dari *Collection of Facial Image : Faces94* yang dikelola oleh Dr.Libor Spacek dari Universitas Essex, Inggris. Terdiri dari 153 individu yang diwakili oleh kurang lebih 20 citra per individu dengan ukuran 180 x 200. Sehingga total dari citra yang digunakan pada tugas akhir ini adalah sebanyak 3059. Seluruh citra wajah tersebut akan menjalani pra-proses hingga seleksi fitur. Sehingga citra wajah nantinya akan berupa histogram untuk kemudian menjadi masukan pada tahap klasifikasi. Pembagian data menjadi data latih dan data uji terjadi saat memasuki tahap klasifikasi dengan memanfaatkan metode *K-fold Cross Validation* dengan  $K = 10$ .

#### **5.3. Skenario Uji Coba**

Pada sub bab ini akan dijelaskan mengenai skenario uji coba yang dilakukan untuk membangun sistem pengenalan wajah dengan performa yang optimal. Pada tahap ekstraksi fitur MB-LBP, dilakukan uji coba untuk masing-masing skala guna mengukur performa masing-

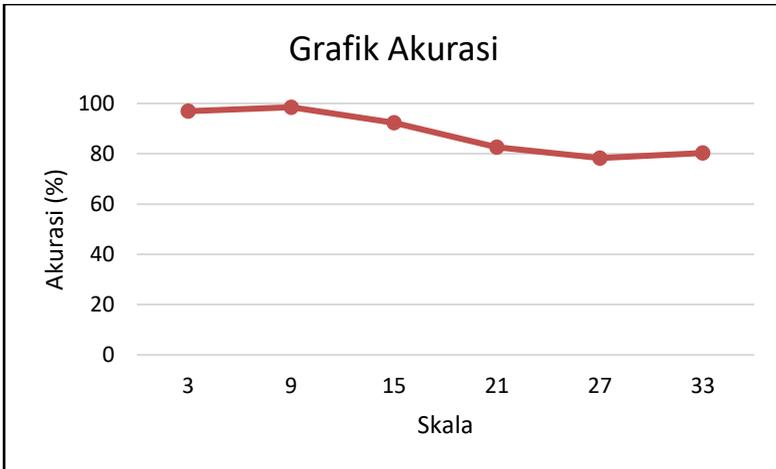
masing skala. Pada tahap seleksi fitur SEMB-LBP dilakukan uji coba menentukan nilai  $N$  yang optimal untuk masing-masing skala. Pada tahap klasifikasi dilakukan uji coba nilai parameter-parameter untuk mendapatkan *classifier* terbaik. Mulai dari kedalaman *Decision Tree*, jumlah *weak classifier*, dan *learning rate*. Kemudian dilakukan uji coba mengkombinasikan skala. Dan terakhir uji coba untuk membandingkan hasil LBP biasa dengan MB-LBP. Nilai akurasi yang ditampilkan berupa nilai rata-rata dari 10-fold yang dijalankan. Khusus untuk Sub Bab 5.3.5 akan ditampilkan nilai akurasi masing-masing fold disertai rata-ratanya.

### 5.3.1. Skenario Uji Coba Ekstraksi Fitur MB-LBP

Pada sub bab ini akan dijelaskan mengenai skenario uji coba yang dilakukan secara khusus pada proses ekstraksi MB-LBP. Uji coba ini dilakukan dengan tujuan untuk mengukur peforma dari masing-masing skala yang nantinya akan menjadi pertimbangan juga pada uji coba kombinasi skala. Input dari uji coba ini berupa citra hasil pra-proses dengan ukuran 180 x 200. Citra kemudian diolah menggunakan ekstraksi fitur dengan variasi skala 3x3, 9x9, 15x15, 21x21, 27x27, dan 33x33. Hasil dari ekstraksi fitur ini berupa fitur MB-LBP sebanyak 256 fitur. Selanjutnya fitur-fitur tersebut akan direpresentasikan ke dalam bentuk histogram untuk diuji akurasinya menggunakan *AdaBoost* dengan parameter  $max\_depth = 15$ ,  $n\_estimators = 500$ , dan  $learning\_rate = 0.1$ . Hasil dari pengujian ekstraksi fitur ini dapat diamati pada Tabel 5.1

Tabel 5.1 Uji Coba Ekstraksi Fitur

Skala	Akurasi Rata-Rata (%)	Waktu Training (detik)	Waktu Testing (detik)
3x3	96,89	336,63	0,54
<b>9x9</b>	<b>98,50</b>	<b>249,89</b>	<b>0,53</b>
15x15	92,32	149,06	0,50
21x21	82,64	157,71	0,49
27x27	78,29	115,20	0,49
33x33	80,32	116,43	0,49



**Gambar 5.1 Grafik Akurasi Uji Coba Ekstraksi Fitur MB-LBP**



**Gambar 5.2 Grafik Waktu Uji Coba Ekstraksi Fitur MB-LBP**

Dari grafik akurasi pada Gambar 5.1 dan grafik waktu pada Gambar 5.2, dapat diketahui bahwa semakin besar skala yang digunakan akan semakin menurun akurasi yang dihasilkan namun semakin sedikit atau cepat waktu yang diperlukan.

### 5.3.2. Skenario Uji Coba Seleksi Fitur SEMB-LBP

Pada sub bab ini akan dipaparkan uji coba yang dilakukan pada tahap seleksi fitur SEMB-LBP dengan tujuan untuk menentukan nilai  $N$  yang tepat untuk masing-masing skala. Seperti yang telah dijelaskan pada Bab II Dasar Teori dimana nilai  $N$  yang menyatakan banyaknya fitur terpilih, ditentukan secara acak dari 1 hingga 256. Output dari seleksi fitur SEMB-LBP ini berupa  $N$  fitur SEMB-LBP yang diuji akurasiya menggunakan *AdaBoost* dengan parameter  $max\_depth = 15$ ,  $n\_estimators = 500$ , dan  $learning\_rate = 0.1$ . Hasil uji coba nilai  $N$  untuk masing-masing skala dapat diamati pada Tabel 5.2.

**Tabel 5.2 Uji Coba Seleksi Fitur**

<b>Skala</b>	<b><math>N</math></b>	<b>Akurasi Rata-Rata (%)</b>	<b>Waktu (detik)</b>
3x3	30	86,64	95,79
	45	88,59	127,48
	<b>58</b>	<b>89,73</b>	<b>153,47</b>
	75	89,41	195,93
	63	90,32	191,77
9x9	30	63,45	77,91
	45	67,31	100,69
	<b>63</b>	<b>69,63</b>	<b>124,22</b>
	75	69,89	134,51
	90	70,74	145,13
15x15	30	33,61	49,71
	45	38,12	60,91
	63	41,22	72,72
	<b>75</b>	<b>41,71</b>	<b>77,16</b>
	90	42,11	93,93
21x21	30	24,29	74,03
	45	26,51	80,93
	<b>63</b>	<b>25,63</b>	<b>79,28</b>
	75	26,18	82,47
	90	26,87	88,81
27x27	30	16,15	46,48

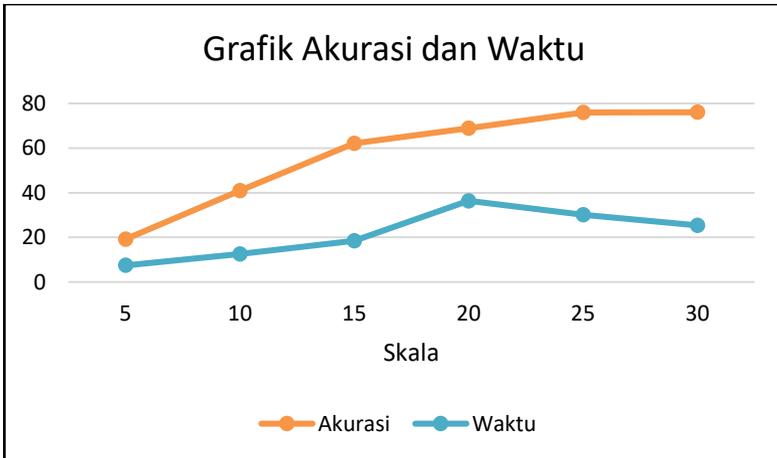
	45	16,41	50,79
	<b>63</b>	<b>16,90</b>	<b>53,84</b>
	75	16,51	56,27
	90	16,51	57,99
33x33	30	14,65	48,77
	45	14,81	54,91
	<b>63</b>	<b>15,73</b>	<b>65,32</b>
	75	15,69	66,54
	90	15,79	62,91

### 5.3.3. Skenario Uji Coba Klasifikasi AdaBoost

Pada sub bab ini akan dijelaskan hasil uji coba parameter-parameter yang berpengaruh pada tahap klasifikasi. Uji coba ini bertujuan untuk membangun *classifier* terbaik dalam sistem pengenalan wajah. Input dari uji coba kali ini adalah fitur MB-LBP dengan skala gabungan 3x3, 9x9, 15x15. Sebanyak 196 fitur untuk masing-masing citra wajah menjadi input pada uji coba klasifikasi ini. 196 fitur tersebut terdiri dari 58 fitur dengan skala 3x3, 63 fitur dengan skala 9x9, dan 75 fitur dengan skala 15x15. Uji coba yang pertama kali dilakukan adalah mencari nilai kedalaman maksimal pada *classifier* dasar yang digunakan yakni *Decision Tree*. Secara *default*, nilai dari kedalaman *tree* atau *max\_depth* ini adalah 5. Pada uji coba ini, hanya dilakukan dengan 6 nilai saja, yakni mulai dari 5 hingga 30 dengan interval 5. Tabel 5.3 merepresentasikan hasil uji coba pada *Decision Tree*

**Tabel 5.3 Uji Coba *Max\_depth***

<i>max_depth</i>	Akurasi Rata-Rata (%)	Waktu (detik)
5	19,19	7,45
10	40,96	12,57
<b>15</b>	<b>62,08</b>	<b>18,44</b>
<b>20</b>	<b>68,91</b>	<b>23,62</b>
<b>25</b>	<b>76,00</b>	<b>30,19</b>
<b>30</b>	<b>76,04</b>	<b>25,34</b>



**Gambar 5.3 Grafik Akurasi dan Waktu Uji Coba Kedalaman *Decision Tree (max\_depth)* pada *AdaBoost***

Pada grafik akurasi dan waktu Gambar 5.3, diketahui bahwa semakin besar nilai dari *max\_depth* membuatnya memiliki akurasi yang semakin baik dan juga waktu yang lebih lama. Hal ini juga yang menjadi alasan penulis untuk membatasi nilai hanya sampai nilai 30 saja. Karena nantinya akan diuji kembali dengan parameter lainnya.

Dari hasil uji coba diatas selanjutnya dipilih 4 nilai *max\_depth* dengan akurasi terbaik yakni 15, 20, 25, dan 30 untuk kemudian dikombinasikan nilainya dengan parameter *n\_estimators* dan *learning\_rate*. Uji coba berikutnya adalah menentukan nilai dari parameter *n\_estimators* yang ditampilkan pada Tabel 5.4. Nilai *default* dari *n\_estimators* adalah 50. Selanjutnya akan diuji coba dengan 4 nilai yakni 100, 300, 500, dan 1000.

**Tabel 5.4 Uji Coba *N Estimators***

<i>max depth</i>	<i>n estimators</i>	Akurasi Rata-Rata (%)	Waktu (detik)
15	100	63,22	37,36
	300	67,01	106,56
	<b>500</b>	<b>70,22</b>	<b>177,55</b>
	1000	71,69	344,77

20	100	71,23	46,52
	300	74,83	135,91
	<b>500</b>	<b>77,99</b>	<b>237,43</b>
	1000	79,79	455,38
25	100	77,08	57,86
	300	81,20	168,37
	<b>500</b>	<b>82,28</b>	<b>280,23</b>
	1000	82,74	561,62
30	<b>100</b>	<b>80,78</b>	<b>61,74</b>
	300	78,52	142,32
	500	77,08	200,39
	1000	81,53	543,10

Dari hasil uji coba diatas, didapatkan nilai  $n_{estimators}$  yang optimal dari segi akurasi maupun waktu untuk masing-masing -  $max\_depth$ . Selanjutnya dilakukan uji coba nilai  $learning\_rate$  yang dapat diamati dari Tabel 5.5. Secara *default* nilai dari  $learning\_rate$  adalah 1. *Range* dari  $learning\_rate$  berkisar antara 0 hingga 1. Semakin besar nilainya, maka akan semakin cepat proses pelatihan model akan tetapi tidak mampu mencapai keadaan optimal yakni pada saat dicapai nilai *error* yang paling minimal. Oleh karena itu, uji coba dengan nilai  $learning\_rate$  yang lebih kecil yakni 0.8, 0.5, 0.3, 0.1, dan 0.05.

**Tabel 5.5 Uji Coba Learning Rate**

<i>max depth</i>	<i>n estimators</i>	<i>learning rate</i>	Akurasi Rata-Rata (%)	Waktu (detik)
15	500	0,8	63,75	188,66
		0,5	72,44	216,61
		0,3	82,67	232,52
		<b>0,1</b>	<b>95,49</b>	<b>313,59</b>
		0,05	95,35	314,55
20	500	0,8	67,08	261,63
		0,5	81,01	292,58
		0,3	90,06	321,69
		0,1	92,38	277,07
		0,05	86,92	168,14

25	500	0,8	73,91	314,42
		0,5	86,59	365,86
		0,3	80,78	290,83
		0,1	80,74	251,45
		0,05	83,39	480,67
30	100	0,8	75,06	54,74
		0,5	67,79	95,01
		0,3	67,21	29,83
		0,1	67,15	102,04
		0,05	66,72	264,59

### 5.3.4. Skenario Uji Coba Kombinasi Skala MB-LBP

Pada sub bab ini akan dijelaskan hasil uji coba kombinasi skala MB-LBP. Berdasarkan Sub Bab 5.3.2 yakni skenario uji coba seleksi fitur SEMB-LBP, semakin besar skala yang digunakan, hasil akurasi menurun secara drastis. Oleh karena itu, agar akurasi yang dihasilkan meningkat, dilakukan penggabungan kombinasi skala MB-LBP. Hasil uji coba kombinasi skala MB-LBP dapat dilihat pada Tabel 5.6 dibawah ini.

**Tabel 5.6 Uji Coba Kombinasi Skala MB-LBP**

Skala	Akurasi Rata-Rata (%)	Waktu Training (detik)	Waktu Testing (detik)
3-9	94,77	254,05	0,50
3-15	93,66	212,97	0,53
9-15	82,71	180,95	0,52
<b>3-9-15</b>	<b>95,42</b>	<b>304,05</b>	<b>0,53</b>
3-9-15-21	95,91	334,04	0,52
3-9-15-21-27	95,88	358,50	0,53
3-9-15-21-27-33	95,62	378,59	0,53

Seperti yang telah disampaikan pada Bab II Dasar Teori, fitur citra *grayscale* pada umumnya adalah sebanyak 256. Dengan pertimbangan ini, dibuatlah batasan jumlah fitur setelah penggabungan dari MB-LBP yang tidak lebih dari atau sama dengan 256 fitur. Oleh

karena itu dipilih skala gabungan 3x3, 9x9, dan 15x15 yang memiliki jumlah fitur sebanyak 196. Walaupun secara akurasi, skala gabungan 3x3, 9x9, 15x15, dan 21x21 memiliki nilai yang lebih besar, akan tetapi jumlah fiturnya sebanyak 259. Dimana jumlah tersebut melampaui batas yang telah ditentukan.

### 5.3.5. Skenario Uji Coba Perbandingan Metode Ekstraksi

Pada sub bab ini akan dipaparkan hasil uji coba sistem pengenalan wajah dengan LBP biasa, MB-LBP dengan seleksi fitur menggunakan gabungan skala 3x3, 9x9, 15x15, dan MB-LBP tanpa seleksi fitur menggunakan skala 9x9. Uji coba perbandingan metode ekstraksi ini dilakukan menggunakan *AdaBoost* dengan parameter  $max\_depth = 15$ ,  $n\_estimators = 500$ , dan  $learning\_rate = 0.1$ . Ketiganya dijelaskan secara detil pada setiap *fold*. Hasil dari skenario uji ini dapat diamati pada Tabel 5.7 berikut

**Tabel 5.7 Uji Coba Perbandingan Metode Ekstraksi**

Metode	<i>fold</i>	Akurasi (%)	Waktu Training (detik)	Waktu Testing (detik)
LBP	1	96,73	331,67	0,52
	2	99,02	342,86	0,54
	3	97,71	343,14	0,56
	4	98,37	332,24	0,58
	5	96,73	342,31	0,55
	6	95,75	343,14	0,53
	7	95,10	339,50	0,53
	8	97,06	328,86	0,55
	9	96,73	333,19	0,53
	10	95,74	329,38	0,53
Rata-Rata		<b>96,89</b>	<b>336,63</b>	<b>0,54</b>
MB-LBP (3-9-15)	1	95,75	299,70	0,53
	2	96,73	299,72	0,50
	3	97,06	300,06	0,52
	4	97,39	303,55	0,52

	5	94,77	307,75	0,53
	6	92,16	304,54	0,50
	7	93,14	294,65	0,55
	8	96,73	315,32	0,61
	9	94,77	312,85	0,53
	10	95,74	302,38	0,52
Rata-Rata		<b>95,42</b>	<b>304,05</b>	<b>0,53</b>
MB-LBP (9x9)	1	98,04	252,45	0,52
	2	99,35	240,15	0,52
	3	99,35	250,86	0,52
	4	99,67	242,37	0,52
	5	98,69	246,31	0,50
	6	97,06	240,33	0,52
	7	97,71	263,13	0,50
	8	97,71	252,28	0,56
	9	98,69	254,55	0,50
	10	98,69	256,45	0,66
Rata-Rata		<b>98,50</b>	<b>249,89</b>	<b>0,53</b>

#### 5.4. Analisa Hasil

Setelah seluruh skenario uji dilakukan, selanjutnya pada sub bab ini akan dibahas hasil yang diperoleh dari masing-masing skenario uji coba yang dilakukan. Pertama-tama, pada skenario uji coba ekstraksi fitur MB-LBP diketahui bahwa skala yang menghasilkan akurasi terbaik adalah 9x9 sebesar 98,50%, diikuti 3x3 sebesar 96,89% dan selanjutnya 15x15 sebesar 92,32%. Hal tersebut membuktikan bahwa benar adanya dengan ukuran area yang lebih besar berpengaruh signifikan dalam proses klasifikasi.

Selanjutnya, dari hasil uji coba seleksi fitur SEMB-LBP, diperoleh nilai  $N$  untuk masing-masing skala. Untuk skala 3x3 digunakan nilai  $N = 58$ . Untuk skala 15x15 digunakan nilai  $N = 75$ . Sedangkan untuk skala 9x9, 21x21, 27x27, dan 33x33 digunakan nilai  $N = 63$ . Dari hasil uji coba MB-LBP diketahui bahwa semakin besar skala maka semakin menurun akurasi. Secara otomatis, apabila skala yang besar dikurangi jumlah fiturnya, maka akan semakin

menurun secara drastis nilai dari akurasi. Hal tersebut terbukti melalui skenario uji coba pada seleksi fitur SEMB-LBP ini.

Pada uji coba klasifikasi, diperoleh parameter-parameter yang berpengaruh signifikan dalam klasifikasi yakni  $max\_depth = 15$ ,  $n\_estimators = 500$ , dan  $learning\_rate = 0,1$ . Selanjutnya dilakukan uji coba mengkombinasikan skala. Hal tersebut dikarenakan dari hasil uji coba seleksi fitur, didapatkan akurasi yang rendah sehingga diperlukan penggabungan skala untuk meningkatkan akurasi. Akan tetapi dalam penggabungan skala ini diberikan batasan fitur sebesar 256. Nilai 256 tersebut merupakan jumlah fitur normal suatu citra *grayscale*. Oleh itu, baiknya jumlah fitur dalam uji coba penggabungan skala ini tidak boleh lebih dari 256. Dari hasil uji kombinasi, didapatkan skala gabungan yang optimal yakni 3x3, 9x9, dan 15x15. Kombinasi skala tersebut diambil dengan pertimbangan akurasi yang optimal dan jumlah fitur yang lebih sedikit bila dibandingkan dengan citra biasa yakni sebanyak 196 fitur. Dan akurasi yang dihasilkan adalah sebesar 95,42%.

Kemudian dilakukan skenario uji yang terakhir yakni perbandingan metode ekstraksi. Diperoleh hasil akurasi dari fitur LBP 3x3 sebesar 96,89%, MB-LBP gabungan 3-9-15 sebesar 95,42% dan akurasi dari fitur MB-LBP 9x9 sebesar 98,50%. Dari sini dapat diketahui bahwa fitur MB-LBP mampu meningkatkan akurasi walaupun tidak terlalu signifikan. Terbukti dengan lebih besarnya akurasi dari MB-LBP 9x9 dibandingkan dengan LBP original. Selain itu, fitur MB-LBP memberikan variasi fitur-fitur utama dari berbagai ukuran area, tidak hanya bergantung dengan piksel saja. Berdasarkan pengamatan penulis, fitur MB-LBP gabungan tidak lebih baik dari LBP dikarenakan prinsip area yang digunakan memberikan dampak berkurangnya pembeda antar fitur dimana hal tersebut secara otomatis menghilangkan variasi ciri dari sebuah citra.

Disamping keinginan untuk mendapatkan fitur yang signifikan dalam klasifikasi, mau tidak mau efek samping dari MB-LBP yakni berkurangnya variasi ciri citra juga harus dihadapi. Terbukti pada Tabel 5.2 yang memperlihatkan fenomena penurunan akurasi secara drastis ketika skala yang digunakan semakin besar. Oleh karena itu, hal tersebut dianggap wajar karena pada sebuah area yang besar nantinya

akan memiliki nilai yang sama. Selain itu, ukuran dari sebuah citra juga menjadi pertimbangan dalam pemilihan skala. Apabila sebuah citra tidak memiliki ukuran citra yang luas, maka tidak disarankan menggunakan skala yang besar. Karena hal tersebut akan semakin mengurangi variasi ciri yang memang pada citra aslinya sudah terbatas. Penggunaan skala ini akan sangat berpengaruh dan menghasilkan akurasi yang berbeda apabila digunakan dengan citra yang berbeda ukuran pula.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dijelaskan mengenai kesimpulan dari proses dan uji coba dari sistem pengenalan wajah serta saran untuk pengembangan dari sistem itu sendiri.

#### **6.1. Kesimpulan**

Dari hasil uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Banyaknya fitur yang optimal pada tahap seleksi menggunakan SEMB-LBP untuk skala 3x3 adalah 58 fitur. Sedangkan untuk skala 15x15 adalah 75 fitur. Dan untuk skala 9x9, 21x21, 27x27, dan 33x33 adalah 63 fitur.
2. Parameter klasifikasi *AdaBoost* yang paling optimal dengan menggunakan  $max\_depth = 15$ ,  $n\_estimators = 500$ , dan  $learning\_rate = 0.1$
3. Skala terbaik untuk MB-LBP adalah 9x9 dengan akurasi sebesar 98,50%. Dan dengan ini dapat dibuktikan bahwa fitur berbasis area cukup berpengaruh bila dibandingkan dengan fitur berbasis piksel yakni metode LBP orgininal dengan akurasi sebesar 96,89%
4. Kombinasi skala MB-LBP terbaik ialah menggunakan skala 3x3, 9x9, dan 15x15 yang menghasilkan akurasi sebesar 95,42%. Walaupun akurasi dari MB-LBP gabungan tidak lebih baik dari LBP original, akan tetapi pada MB-LBP gabungan terdapat variasi fitur yang lebih banyak bila dibandingkan LBP original.

## 6.2. Saran

Saran yang diberikan untuk pengembangan sistem ini adalah sebagai berikut :

1. Semakin besar skala yang digunakan sebagai operator MB-LBP menyebabkan semakin berkurang juga akurasi yang dihasilkan. Ukuran citra sangat berpengaruh juga dalam penentuan skala. Penggunaan skala besar hanya ketika memang citra memiliki ukuran yang cukup besar juga. Apabila ukuran citra terbatas, tidak dianjurkan untuk menggunakan skala yang terlalu besar karena hal tersebut akan menurunkan performa MB-LBP secara drastis.
2. Perhitungan fitur MB-LBP dapat dikembangkan dengan parameter radius atau jarak ketetangaan dan sudut seperti halnya metode LBP yang telah berkembang.

## DAFTAR PUSTAKA

- [1] M. A. Turk, A. P. Pentland. 1991. *Eigenfaces for Recognition*, *Journal of Cognitive Neuroscience*, p. 71-86.
- [2] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman. 1997. *Eigenfaces vs Fisherfaces : Recognition using Class Specific Linear Projection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 711-720.
- [3] P. Penev, J. Atick. 1996. *Local Feature Analysis : A General Statistical Theory of Object Representation*, *Neural System*, p. 477-500.
- [4] L. Wiskott, J. Fellous, N. Kruger, C. V. D. Malsburg. 1997. *Face Recognition by Elastic Bunch Graph Matching*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 775-779.
- [5] T. Ojala, M. Pietikainen, Harwood. 1996. *A Comparative Study of Texture Measures with Classification based on Feature Distributions*, *Pattern Recognition* 29(1), 51-59.
- [6] T. Ahonen, A. Hadid, M. Pietikainen. 2004. *Face Recognition with Local Binary Patterns*, p. 469-481.
- [7] T. Ojala, M. Pietikainen, M. Maenpaa. 2002. *Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Pattern*, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, p. 971-987.
- [8] S. Liao, X. Zhu, Lei, L. Zhang, S. Z. Li. 2007. *Learning Multi-scale Block Local Binary Patterns for Face Recognition*, *Biometric Fuzzy Extraction Made Practical : A Proposal Based on Finger Codes*, p. 828-837.

- [9] Z. Ji, R. Saharon, Z. Hui, H. Trevor. 2006. *Multi-class AdaBoost, Statistics and Its Interface*, p.349-360.
- [10] Saxena Rahul. 2017. *How Decision Tree Algorithm*. [Online]. Available : <http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/>. [Diakses 31 Mei 2018].
- [11] Wikipedia. 2018. *Confusion Matrix*. [Online]. Available : [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix). [Diakses 31 Mei 2018].
- [12] Wibowo Antoni. 2017. *10 Fold-Cross Validation*. [Online]. Available : <https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/>. [Diakses 19 April 2018].
- [13] Spacek Libor. 2007. *Collection of Facial Images : Faces94*. [Online]. Available : <http://cswww.essex.ac.uk/mv/allfaces/faces94.html>. [Diakses 18 April 2018].

## LAMPIRAN

### A. Hasil Prediksi MB-LBP (Label Kelas)

No	Aktual	Prediksi	
		MB-LBP (9x9)	MB-LBP Gabungan (3-9-15)
1	yfhsie	yfhsie	yfhsie
2	9326871	9326871	9326871
3	9332898	9332898	9332898
4	9336923	9336923	9336923
5	9338446	9338446	9338446
6	9338454	9338454	9338454
7	9338462	9338462	9338462
8	9338489	9338489	<b>lyond</b>
9	9338497	9338497	9338497
10	9338519	9338519	9338519
11	9338527	9338527	9338527
12	9338535	9338535	9338535
13	9338543	9338543	9338543
14	9414649	9414649	9414649
15	9416994	9416994	9416994
16	admars	admars	admars
17	ahodki	ahodki	ahodki
18	ajflem	ajflem	ajflem
19	ajones	ajones	ajones
20	ajsega	ajsega	ajsega
21	akatsi	akatsi	akatsi
22	ambarw	ambarw	ambarw
23	anonym	anonym	anonym
24	anonym1	anonym1	anonym1
25	anonym2	anonym2	anonym2
26	anpage	anpage	anpage
27	asamma	asamma	asamma
28	asewil	asewil	asewil
29	asheal	asheal	asheal
30	astefa	astefa	astefa
31	bplyce	bplyce	bplyce
32	cchris	cchris	cchris
33	ccjame	ccjame	ccjame

34	cferdo	cferdo	cferdo
35	cgboyc	cgboyc	cgboyc
36	cjcarr	cjcarr	cjcarr
37	cjdenn	cjdenn	cjdenn
38	cjsake	cjsake	cjsake
39	cmkirk	cmkirk	cmkirk
40	csanch	csanch	csanch
41	cshubb	cshubb	cshubb
42	cwang	cwang	cwang
43	cwchoi	cwchoi	cwchoi
44	dagran	dagran	dagran
45	dakram	dakram	dakram
46	dcbowe	dcbowe	dcbowe
47	dioann	dioann	dioann
48	djbirc	djbirc	djbirc
49	djhugh	djhugh	djhugh
50	djmart	djmart	djmart
51	dmwest	dmwest	dmwest
52	doraj	doraj	doraj
53	drbost	drbost	drbost
54	ekavaz	ekavaz	ekavaz
55	elduns	elduns	elduns
56	fordj	fordj	fordj
57	gdhatc	gdhatc	gdhatc
58	ggeorg	ggeorg	ggeorg
59	ggrego	ggrego	ggrego
60	gjhero	gjhero	gjhero
61	gjnorm	gjnorm	gjnorm
62	gmwate	gmwate	gmwate
63	gpapaz	gpapaz	gpapaz
64	gpsmit	gpsmit	gpsmit
65	gsreas	gsreas	gsreas
66	hartb	hartb	hartb
67	hensm	hensm	hensm
68	ieorf	ieorf	ieorf
69	irdrew	irdrew	irdrew
70	jabins	jabins	jabins
71	jagrif	jagrif	jagrif
72	jcarte	jcarte	jcarte

73	jdbenm	jdbenm	jdbenm
74	jgloma	jgloma	jgloma
75	jlemon	jlemon	jlemon
76	jmedin	jmedin	jmedin
77	jrtobi	jrtobi	jrtobi
78	kaatki	kaatki	kaatki
79	kaknig	kaknig	kaknig
80	kdjone	kdjone	kdjone
81	khchan	khchan	khchan
82	khughe	khughe	khughe
83	kjwith	kjwith	kjwith
84	klclar	klclar	klclar
85	ksunth	ksunth	ksunth
86	lejno	lejno	lejno
87	lfso	lfso	lfso
88	lyond	<b>njmoor</b>	lyond
89	maasht	maasht	maasht
90	macci	macci	macci
91	martin	martin	martin
92	mberdo	mberdo	mberdo
93	mbutle	mbutle	mbutle
94	mdpove	mdpove	<b>djbirc</b>
95	mefait	mefait	mefait
96	mhwil	mhwil	mhwil
97	miaduc	miaduc	miaduc
98	michael	michael	michael
99	mjhans	mjhans	mjhans
100	moors	moors	moors
101	mpetti	mpetti	mpetti
102	muthay	muthay	muthay
103	nahaig	nahaig	nahaig
104	namull	namull	namull
105	ndbank	ndbank	ndbank
106	ndhagu	ndhagu	ndhagu
107	nhrams	nhrams	nhrams
108	njmoor	njmoor	njmoor
109	npbour	npbour	npbour
110	npmitc	npmitc	npmitc
111	nrclar	nrclar	nrclar

112	nrrbar	nrrbar	nrrbar
113	nwilli	nwilli	nwilli
114	obeidn	obeidn	obeidn
115	ohpark	ohpark	ohpark
116	pacole	pacole	pacole
117	phughe	phughe	phughe
118	pmives	pmives	pmives
119	pshurr	pshurr	pshurr
120	pspliu	<b>ndbank</b>	<b>anonym1</b>
121	ptnich	ptnich	ptnich
122	rarobi	rarobi	rarobi
123	rgharr	rgharr	rgharr
124	rgspru	rgspru	rgspru
125	rjlabr	rjlabr	rjlabr
126	rlocke	rlocke	rlocke
127	rmcoll	rmcoll	rmcoll
128	rmpugh	rmpugh	rmpugh
129	rnpwil	rnpwil	rnpwil
130	robin	robin	robin
131	rrowle	rrowle	<b>vstros</b>
132	rsanti	rsanti	rsanti
133	saduah	saduah	saduah
134	saedwa	saedwa	saedwa
135	sandm	sandm	sandm
136	sbains	sbains	sbains
137	sidick	sidick	sidick
138	sjbeck	sjbeck	sjbeck
139	skumar	skumar	skumar
140	slbirc	slbirc	slbirc
141	smrobb	smrobb	smrobb
142	spacl	spacl	spacl
143	spletc	spletc	spletc
144	svkriz	svkriz	svkriz
145	swewin	swewin	swewin
146	swsmit	swsmit	swsmit
147	tony	tony	<b>rmpugh</b>
148	voudcx	voudcx	voudcx
149	vpsavo	vpsavo	vpsavo
150	vstros	vstros	vstros

151	whussa	whussa	<b>phughe</b>
152	wjalbe	wjalbe	wjalbe
153	yfhsie	yfhsie	yfhsie
154	9326871	9326871	9326871
155	9332898	9332898	9332898
156	9336923	9336923	9336923
157	9338446	9338446	9338446
158	9338454	9338454	9338454
159	9338462	9338462	9338462
160	9338489	9338489	9338489
161	9338497	9338497	9338497
162	9338519	9338519	9338519
163	9338527	9338527	9338527
164	9338535	9338535	9338535
165	9338543	9338543	9338543
166	9414649	9414649	9414649
167	9416994	9416994	9416994
168	admars	admars	admars
169	ahodki	ahodki	ahodki
170	ajflem	ajflem	ajflem
171	ajones	ajones	ajones
172	ajsega	ajsega	ajsega
173	akatsi	akatsi	akatsi
174	ambarw	ambarw	ambarw
175	anonym	anonym	anonym
176	anonym1	anonym1	anonym1
177	anonym2	anonym2	anonym2
178	anpage	anpage	anpage
179	asamma	asamma	asamma
180	asewil	asewil	asewil
181	asheal	asheal	asheal
182	astefa	astefa	astefa
183	bplyce	bplyce	bplyce
184	cchris	cchris	cchris
185	ccjame	ccjame	ccjame
186	cferdo	cferdo	cferdo
187	cgboyc	cgboyc	cgboyc
188	cjcarr	cjcarr	cjcarr
189	cjdenn	cjdenn	cjdenn

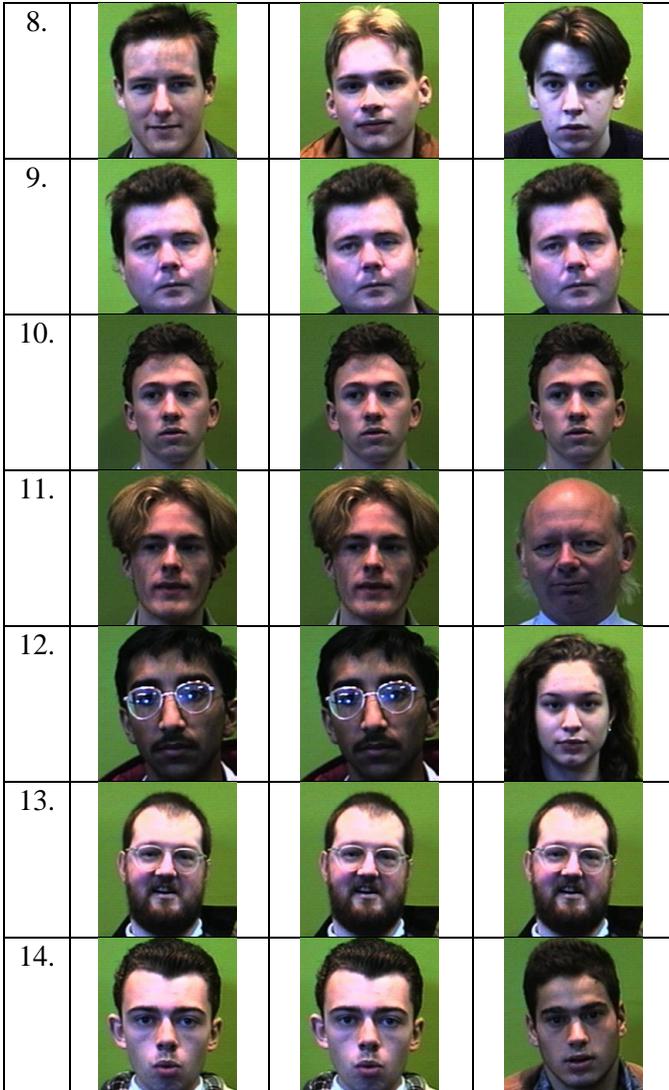
190	cjsake	<b>kaatki</b>	cjsake
191	cmkirk	cmkirk	cmkirk
192	csanch	csanch	csanch
193	cshubb	cshubb	cshubb
194	cwang	cwang	cwang
195	cwchoi	cwchoi	cwchoi
196	dagran	dagran	dagran
197	dakram	dakram	dakram
198	dcbowe	dcbowe	dcbowe
199	dioann	dioann	dioann
200	djbirc	djbirc	djbirc
201	djhugh	djhugh	djhugh
202	djmart	djmart	djmart
203	dmwest	dmwest	dmwest
204	doraj	doraj	doraj
205	drbost	drbost	drbost
206	ekavaz	ekavaz	ekavaz
207	elduns	elduns	elduns
208	fordj	fordj	fordj
209	gdhatc	gdhatc	gdhatc
210	ggeorg	ggeorg	ggeorg
211	ggrego	ggrego	ggrego
212	gjhero	gjhero	gjhero
213	gjnorm	gjnorm	gjnorm
214	gmwate	gmwate	gmwate
215	gpapaz	gpapaz	gpapaz
216	gpsmit	gpsmit	gpsmit
217	gsreas	gsreas	<b>9338543</b>
218	hartb	hartb	hartb
219	hensm	hensm	hensm
220	ieorf	ieorf	<b>cferdo</b>
221	irdrew	irdrew	irdrew
222	jabins	jabins	jabins
223	jagrif	jagrif	jagrif
224	jcarte	jcarte	jcarte
225	jdbenm	jdbenm	jdbenm
226	jgloma	jgloma	jgloma
227	jlemon	jlemon	jlemon
228	jmedin	jmedin	jmedin

229	jrtobi	jrtobi	jrtobi
230	kaatki	kaatki	kaatki
231	kaknig	kaknig	kaknig
232	kdjone	kdjone	kdjone
233	khchan	khchan	khchan
234	khughe	khughe	khughe
235	kjwith	kjwith	kjwith
236	klclar	klclar	klclar
237	ksunth	ksunth	ksunth
238	lejno	lejno	lejno
239	lfso	lfso	lfso
240	lyond	lyond	lyond
241	maasht	maasht	maasht
242	macci	macci	macci
243	martin	martin	martin
244	mberdo	<b>vpsavo</b>	<b>9414649</b>
245	mbutle	mbutle	mbutle
246	mdpove	mdpove	mdpove
247	mefait	mefait	mefait
248	mhwil	mhwil	mhwil
249	miaduc	miaduc	miaduc
250	michael	michael	michael
251	mjhans	mjhans	mjhans
252	moors	moors	moors
253	mpetti	mpetti	mpetti
254	muthay	muthay	muthay
255	nahaig	nahaig	nahaig
256	namull	namull	namull
257	ndbank	ndbank	<b>9326871</b>
258	ndhagu	ndhagu	ndhagu
259	nhrams	nhrams	nhrams
260	njmoor	njmoor	njmoor
261	npbour	npbour	npbour
262	npmitc	npmitc	npmitc
263	nrclar	nrclar	nrclar
264	nrrbar	nrrbar	nrrbar
265	nwilli	nwilli	nwilli
266	obeidn	obeidn	obeidn
267	ohpark	ohpark	ohpark

268	pacole	pacole	pacole
269	phughe	phughe	phughe
270	pmives	pmives	<b>rgspru</b>
271	pshurr	pshurr	pshurr
272	pspliu	pspliu	pspliu
273	ptnich	ptnich	ptnich
274	rarobi	rarobi	rarobi
275	rgharr	rgharr	rgharr
276	rgspru	rgspru	rgspru
277	rjlabr	rjlabr	mhwil
278	rlocke	rlocke	rlocke
279	rmcoll	rmcoll	rmcoll
280	rmpugh	rmpugh	rmpugh
281	rnpwil	rnpwil	<b>michael</b>
282	robin	robin	robin
283	rrowle	rrowle	rrowle
284	rsanti	rsanti	rsanti
285	saduah	saduah	saduah
286	saedwa	saedwa	saedwa
287	sandm	sandm	sandm
288	sbains	sbains	sbains
289	sidick	sidick	sidick
290	sjbeck	sjbeck	sjbeck
291	skumar	skumar	skumar
292	slbirc	slbirc	slbirc
293	smrobb	smrobb	smrobb
294	spacl	spacl	spacl
295	spletc	spletc	spletc
296	svkriz	svkriz	svkriz
297	swewin	swewin	swewin
298	swsmit	swsmit	swsmit
299	tony	tony	tony
300	voudcx	voudcx	voudcx
301	vpsavo	vpsavo	vpsavo
302	vstros	vstros	vstros
303	whussa	whussa	<b>phughe</b>
304	wjalbe	wjalbe	wjalbe
305	yfhsie	yfhsie	yfhsie

### B. Hasil Prediksi MB-LBP (Citra)

No	Citra Asli	Prediksi	
		9x9	3-9-15
1.			
2.			
3.			
4.			
5.			
6.			
7.			



## BIODATA PENULIS



Afifah Asmar Sari lahir di Surakarta pada tanggal 9 Juni 1996 dan lama dibesarkan di Mojokerto, Jawa Timur. Penulis telah menempuh pendidikan dari SDN 1 Sooko Mojokerto, SMPN 1 Mojokerto, SMAN 1 Sooko Mojokerto dan terakhir menempuh pendidikan perguruan tinggi di Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi (FTIK) Institut Teknologi Sepuluh Nopember Surabaya.

Selama menempuh pendidikan di Departemen Informatika ITS, penulis aktif dalam berbagai organisasi dan kepanitiaan. Pada tahun kedua, penulis menjadi staf Kesejahteraan Mahasiswa di Himpunan Mahasiswa Teknik-Computer Informatika (HMTC), staf Al-Afifah di Keluarga Muslim Informatika, staf Acara di event FTif Festival, dan staf National Programming Contest (NPC) Schematics 2015. Pada tahun ketiga, penulis menjabat sebagai Kepala Administrasi dan Keuangan di HMTC dan menjadi staf ahli Web Kestari di Schematics 2016. Selain itu, penulis juga mengikuti kegiatan Latihan Keterampilan dan Manajemen Mahasiswa (LKMM) mulai dari tingkat pra-dasar hingga pelatihan untuk menjadi pemandu LKMM. Disamping kesibukan organisasi, penulis juga mengemban amanah sebagai administrator laboratorium Komputasi Cerdas dan Visi (KCV).

Selain menjadi admin laboratorium KCV, penulis juga tertarik dengan bidang KCV tersebut hingga memutuskan untuk mengambil Tugas Akhir dengan peminatan KCV yang berfokus pada pengolahan citra digital. Penulis dapat dihubungi melalui email : [afifahasmarsari@gmail.com](mailto:afifahasmarsari@gmail.com)