



TUGAS AKHIR - SM141501

**PENGENALAN INDIVIDU MELALUI
IDENTIFIKASI WAJAH MENGGUNAKAN
*HISTOGRAM OF ORIENTED GRADIENT***

**XENNY ZARVINA LOVIANO
NRP 1213 100 096**

**Dosen Pembimbing :
Dr. Nurul Hidayat M.Kom**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya
2017**



FINAL PROJECT - SM141501

***INDIVIDUAL RECOGNITION FROM FACE
IDENTIFICATION USING HISTOGRAM OF
ORIENTED GRADIENT***

XENNY ZARVINA LOVIANO
NRP 1213 100 096

Supervisor :
Drs. Nurul Hidayat M.Kom

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya
2017

LEMBAR PENGESAHAN

**PENGENALAN INDIVIDU MELALUI IDENTIFIKASI
WAJAH MENGGUNAKAN *HISTOGRAM OF
ORIENTED GRADIENT***

***INDIVIDUAL RECOGNITION FROM FACE
IDENTIFICATION USING HISTOGRAM OF ORIENTED
GRADIENT***

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang minat Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:

XENNY ZARVINA LOVIANO
NRP.1213 100 096

Menyetujui,
Dosen Pembimbing,



Drs. Nurul Hidayat, M.Kom
NIP.19630404 198903 1 002

Mengetahui,
Kepala Departemen Matematika
FEMIPA ITS



Dr. Imam Mukhlash, S.Si, M.T
NIP.19700831 199403 1 003

Surabaya, Juli 2017

**PENGENALAN INDIVIDU MELALUI IDENTIFIKASI
WAJAH MENGGUNAKAN *HISTOGRAM OF ORIENTED
GRADIENT***

Nama Mahasiswa : Kenny Zarvina Loviano
NRP : 1213 100 096
Departemen : Matematika
Dosen Pembimbing : Drs. Nurul Hidayat, M.Kom

Abstrak

Biometrika merupakan cabang matematika terapan yang digunakan untuk pengenalan individu yang ruang lingkungannya adalah mengidentifikasi individu melalui ciri/sifat biologis yang khusus. Bagian individu yang paling umum dan alamiah yang dapat digunakan untuk mengenali seorang individu adalah wajah. Ini disebabkan ciri yang dimiliki wajah bersifat unik untuk setiap individu. Variabilitas wajah ini dapat di-minimalisir dengan pencarian ciri wajah melalui detektor ciri. *Histogram Of Oriented Gradient* (HOG) adalah *descriptor* fitur yang digunakan untuk identifikasi wajah. HOG mendeskripsikan fitur berdasarkan histogram lokal dari orientasi gradien yang diberi bobot dengan *magnitude* gradien. Setiap individu memiliki hasil ekstraksi yang berbeda. Tahapan dari pengenalan individu ini yaitu pra-pemrosesan, proses ekstraksi fitur dengan HOG dan proses klasifikasi dengan *K-Nearest Neighbor* dimana K menunjukkan jumlah tetangga terdekat. Dari 40 citra wajah individu didapatkan hasil pengujian dengan tingkat rata-rata akurasi sebesar 80 % dan hasil terbaik pada penggunaan $K = 1$.

***Kata Kunci* : Pengenalan individu, Identifikasi wajah, Fitur, *Histogram Of Oriented Gradient* (HOG).**

**INDIVIDUAL RECOGNITION FROM FACE
IDENTIFICATION USING HISTOGRAM OF ORIENTED
GRADIENT**

Name of Student : Xenny Zarvina Loviano
NRP : 1213 100 096
Departement : Mathematics
Supervisor : Drs. Nurul Hidayat M.Kom

Abstract

Biometrics is a branch of applied mathematics that is indispensable for the individual that scopes the identification of individual through characteristics of typical biological practice.. Part of the individual that is most common and natural which can be used to recognize the individual is the face. This is caused by the unique characteristics of each face of the individuals. Variability of the face can be minimized by searching feature of the face through the characteristics detector. Histogram of Oriented Gradient (HOG) is a descriptor of the features that are used to define the facial identification. Based on histogram feature described a HOG from the orientation of the gradient are given more weight by the magnitude of the gradient. For each individual, It has a different extraction result. The steps of individual recognition are pre-processing, feature extraction process with the HOG and classification process with the K-Nearest neighbors where as K shows the sum of the nearest neighbor. From 40 human face images, would be obtained a test result with an 80% average accuracy rate and the best result on the use of $K = 1$.

Keywords : Individual recognition, Face Identification, Histogram of Oriented Gradient (HOG)

KATA PENGANTAR

Alhamdulillah wa syukurillaah penulis panjatkan kepada Allah SWT. karena dengan limpahan ni'mat dan rahmat-Nya yang tiada tara juga izin-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul: **“Pengenalan Individu Melalui Identifikasi Wajah Menggunakan *Histogram of Oriented Gradient*”** yang merupakan salah satu persyaratan dalam menyelesaikan Program Studi S-1 di Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Institut Teknologi Sepuluh Nopember Surabaya.

Tugas akhir ini dapat penulis selesaikan dengan baik walaupun pada awalnya jatuh bangun karena banyak kendala yang menghadang, tapi penulis bisa mendapatkan semangat kembali berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Ketua Departemen Matematika ITS.
2. Bapak Drs. Nurul Hidayat, M.Kom selaku dosen pembimbing yang senantiasa meluangkan waktunya untuk memberikan bimbingan dan dukungan kepada penulis.
3. Bapak Dr. Didik Khusnul Arif, S.Si., M.Si selaku Ketua Program Studi Departemen Matematika ITS.
4. Ibu Soleha, S.Si., M.Si., selaku dosen wali yang senantiasa membimbing dan menasehati penulis.
5. Ibu Alvida Mustika Rukmi, S.Si, M.Si , Bapak Dr. Budi Setiyono, S.Si, MT dan Bapak Drs. Iis Herisman, M.Sc, selaku dosen penguji. Yang telah memberi bimbingan dan nasihat kepada penulis.
6. Seluruh jajaran dosen Matematika ITS yang dengan sabar dan ikhlas mendidik mahasiswa-mahasiswinya dan juga para staf jurusan Matematika ITS yang memberikan pelayanan dengan baik.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga tugas akhir ini bermanfaat bagi semua pihak.

Surabaya, Juni 2017

Penulis

Special Thanks To :

1. Bapak Marzaliano dan Ibu Indriany serta Febrizar Aviliano dan Chappy Marzaliano Junior tercinta yang selalu memberikan dukungan, nasehat, dan doa pada penulis selama ini.
2. Mbak Tuffahatul Ummah yang selalu membantu dan memotivasi penulis untuk menyelesaikan tugas akhir ini.
3. Khusna Amalah dan Marina Nadya Herietta terima kasih atas bantuan, semangat, dan dukungannya.
4. Teman-teman Manis Manja, Amel, Vicky, Desna, Helisyah, Haidar, Prima, Yoga, Fauzia, Satria, Ais, terima kasih atas dukungan dan motivasi kalian.
5. Septia Marga Dartika yang telah memberi dukungan, terima kasih atas dukungannya.
6. Teman-teman Departemen Dagri HIMATIKA periode 2015-2016, terima kasih atas pembelajaran yang sangat berharga bagi penulis.
7. TIM SC Padamu Himatika 2015, terimakasih atas kerjasamanya sehingga penulis mendapat satu tahun yang sangat berharga.
8. Arthur Holi Compton, terima kasih atas dukungan kalian.
9. Teman-teman angkatan 2013 yang sudah meberikan banyak dukungan, motivasi, cerita, bantuan hingga penulis dapat menyelesaikan tugas akhir ini.
10. Teman-teman seperjuangan Wisuda 116, menghabiskan waktu dengan kalian adalah waktu yang sangat berharga.
11. Vina, Aza, Melynda, Metta, Faizah selaku teman-teman satu bimbingan Tugas Akhir yang telah memberikan bantuan dan motivasi.
12. Kelompok Louis Lagrange dan SC mbak Donna yang telah menemani penulis selama menjadi mahasiswa baru.
13. Teman-teman HIMATIKA ITS yang tidak dapat diucapkan satu-satu, terima kasih atas dukungan dan pembelajarannya.
14. Teman-teman LBS, Sari, Aulia, Niken, Audy, Irawan, Adit, Fakhri, Angga, Makki yang telah memberi motivasi dan hiburan pada penulis.

15. Teman-teman Houcing-Houcing, Dewa, Jasmine, Lina, Gandhes, Fira, Sabrina, Lala, Berti, Risma yang telah memberikan dukungan dan motivasi pada penulis.

Tentu saja masih banyak pihak lain yang turut andil dalam penyelesaian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu. Semoga Allah membalas dengan balasan yang lebih baik bagi semua pihak yang telah membantu penulis. *Aaamiin yaa rabbal'aalamiin.*

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	
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 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Sebelumnya.....	5
2.2 Citra Digital	6
2.3 Pengenalan Wajah.....	7
2.4 Pengolahan Citra.....	7
2.5 Ekstraksi Fitur.....	9
2.6 Klasifikasi Menggunakan Algoritma <i>K-Nearest Neighbor</i>	14
BAB III METODOLOGI PENELITIAN	
3.1 Objek dan Aspek Penelitian	17
3.2 Peralatan	17
3.3 Data Penelitian.....	17
3.4 Tahap Penelitian.....	18
BAB IV ANALISIS DAN PERANCANGAN SISTEM	
4.1 Perancangan Sistem	21
4.2 Format Data	21
4.3 Gambaran Proses Secara Umum	24

BAB V IMPLEMENTASI	
5.1 Implementasi Antarmuka.....	37
5.2 Implementasi Tahap Input Data.....	43
5.3 Implementasi Tahap <i>Preprocessing</i>	43
5.4 Implementasi Tahap Ekstraksi Fitur.....	44
5.5 Implementasi Tahap Klasifikasi.....	44
BAB VI UJI COBA DAN PEMBAHASAN	
6.1 Lingkungan Pengujian Sistem.....	47
6.2 Pengujian Tahap <i>Preprocessing</i>	48
6.3 Pengujian Tahap Ekstraksi Fitur.....	49
6.4 Pengujian Tahap Klasifikasi dengan K-Nearest.....	53
6.5 Pembahasan Hasil Pengujian.....	58
6.6 Pembahasan Penyebab Besar Kecil Akurasi.....	60
BAB VII KESIMPULAN DAN SARAN	
7.1 Kesimpulan.....	63
7.2 Saran.....	63
DAFTAR PUSTAKA.....	65
LAMPIRAN.....	67
BIODATA PENULIS.....	87

DAFTAR GAMBAR

Gambar 2.1	Diagram alir metode <i>Viola-Jones</i>	9
Gambar 2.2	Langkah-langkah ekstraksi HOG	10
Gambar 2.3	Bentuk <i>Cell</i>	13
Gambar 2.4	Klasifikasi Menggunakan <i>K-Nearest Neighbor</i>	14
Gambar 3.1	Blok diagram tahapan penelitian	18
Gambar 4.1	Sebagian Data Training (a)Tampak Depan (b)Tampak Kiri (c)Tampak Kanan (d)Tampak Atas22	
Gambar 4.2	Sebagian Citra <i>Testing</i>	23
Gambar 4.3	Diagram Alir Proses Pengenalan Individu Secara Umum.....	26
Gambar 4.4	Diagram Alir Proses Ekstraksi Fitur dengan HOG.....	28
Gambar 4.5	Diagram alir proses <i>testing K-Nearest Neighbor</i>	35
Gambar 5.1	Halaman Utama.....	37
Gambar 5.2	Antarmuka Menu Utama Program	39
Gambar 5.3	Antarmuka Pengujian Sistem Pengenalan Individu	40
Gambar 5.4	Hasil Running Program Pengujian Sistem Pengenalan Individu	41
Gambar 6.1	Pengujian proses <i>grayscale</i> untuk citra wajah.	487
Gambar 6.2	Pengujian proses deteksi wajah menggunakan <i>Viola-Jones</i>	47
Gambar 6.3	Pengujian proses terhadap citra wajah manusia (a) <i>cropping</i> (b) <i>resizing</i>	48
Gambar 6.4	Pengujian proses komputasi gradien untuk citra wajah (a) Adit (b) Amel (c) Anas	49
Gambar 6.5	Pengujian proses orientasi <i>bin</i> untuk citra wajah (a) Adit (b) Anas (c) Amel.....	50
Gambar 6.6	Pengujian proses ekstraksi 2 individu berbeda (a) citra wajah Adit (b) citra wajah Anas	51
Gambar 6.7	Grafik hasil nilai akurasi sistem	59

DAFTAR TABEL

Tabel 4.1	Data Proses	24
Tabel 4.2	Euclidian Distance	34
Tabel 4.3	Tabel urutan dari yang terkecil ke yang terbesar. 34	
Tabel 5.1	Kegunaan Menu Sistem	39
Tabel 5.2	Kegunaan Menu Utama	39
Tabel 6.1	Lingkungan Pengujian Sistem.....	47
Tabel 6.2	Label pada vektor fitur hasil ekstraksi HOG	53
Tabel 6.3	Hasil Proses <i>Testing</i> dengan nilai $K = 1$	55
Tabel 6.4	Hasil Proses <i>Testing</i> dengan nilai $K = 2$	55
Tabel 6.5	Hasil Proses <i>Testing</i> dengan nilai $K = 3$	56
Tabel 6.6	Hasil Proses <i>Testing</i> dengan nilai $K = 4$	56
Tabel 6.7	Hasil Proses <i>Testing</i> dengan nilai $K = 5$	56
Tabel 6.8	Hasil Proses <i>Testing</i> dengan nilai $K = 6$	57
Tabel 6.9	Hasil Proses <i>Testing</i> dengan nilai $K = 7$	57
Tabel 6.10	Hasil Proses <i>Testing</i> dengan nilai $K = 8$	57
Tabel 6.11	Hasil perhitungan akurasi sistem menggunakan data <i>training</i>	59
Tabel 6.12	Hasil perhitungan akurasi sistem menggunakan data <i>testing</i>	59
Tabel 6.13	Hasil Perhitungan Rata-rata Akurasi.....	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Biometrika adalah salah satu cabang matematika terapan yang sangat diperlukan untuk pengenalan individu yang ruang lingkupnyanya adalah mengidentifikasi individu melalui ciri/sifat biologis yang khas/khusus, misalnya sidik jari, selaput retina, telapak tangan, suara ucapan, dan wajah. Pada perkembangannya, biometrika ini dapat juga dikembangkan untuk mengenali obyek apapun asalkan obyek tersebut memiliki ciri khusus, misalnya pengenalan rambu lalu lintas. Pengenalan individu dan/atau obyek, ini akan banyak digunakan dalam membuat peralatan elektronik otomatis (semi otomatis) maupun sistem pengidentifikasi, demi kepentingan usaha dan pengamanan aset yang dimilikinya.

Bagian individu yang paling umum dan alamiah yang dapat digunakan untuk mengenali seorang individu adalah wajah. Ini disebabkan fitur (ciri) yang dimiliki wajah bersifat unik untuk setiap individu. Melalui identifikasi wajah yang dimilikinya, seorang individu dapat dikenali. Contohnya dalam tindak kejahatan yang biasa terjadi, bagian individu yang terekam CCTV atau video dan bisa digunakan untuk identifikasi adalah pola wajah. Sehingga identifikasi wajah sangat berperan dalam sistem keamanan. Namun demikian wajah mempunyai karakteristik yang sangat kompleks, sehingga kerumitan dalam identifikasi wajah kadangkala muncul, yang disebabkan adanya variabilitas wajah. Munculnya variabilitas wajah ini biasanya karena perubahan bentuk wajah, seperti berpose, berekspresi, dan berkacamata. Juga karena pengaruh usia, bulu wajah seperti kumis dan jenggot, penyamaran, dan kualitas pencahayaan. Akan tetapi, variabilitas wajah ini dapat diminimalisir dengan pencarian ciri wajah melalui detektor ciri.

Salah satu metode yang dapat digunakan untuk mendeteksi individu melalui ciri yang dimilikinya adalah *Histogram of Oriented Gradient*. Metode *Histogram of Oriented Gradient* (HOG) merupakan metode yang pertama kali dikenalkan oleh Dalal dan Triggs (2005) untuk deteksi manusia[1]. Metode ini juga telah digunakan untuk penelitian sebelumnya, misalnya oleh Cahyo Permata dan kawan-kawan tentang deteksi mobil menggunakan metode *Histogram of Oriented Gradient* [4]. Pada penelitian ini dipilih HOG untuk ekstraksi fitur yang memiliki kelebihan daripada metode lain, karena HOG beroperasi pada sel lokal, juga invariant (tidak berubah) untuk transformasi geometrik dan fotometrik[6].

Berdasarkan penjelasan di atas, maka pada Penelitian ini membahas mengenai sistem pengenalan individu melalui identifikasi wajahnya menggunakan *Histogram of Oriented Gradient*.

1.2 Rumusan Masalah

Dari latar belakang yang telah dikemukakan di atas, dapat dirumuskan masalah sebagai berikut:

1. Membuat sistem pengenalan individu melalui identifikasi wajah.
2. Mengaplikasikan metode *histogram of oriented gradient* pada proses ekstraksi fitur wajah.
3. Mengaplikasikan metode *K-Nearest Neighbor* pada proses klasifikasi fitur.

1.3 Batasan Masalah

Batasan-batasan yang ada dalam pembuatan penelitian ini adalah:

1. Pengambilan citra wajah dilakukan menggunakan kamera digital dan/atau CCTV.
2. Input citra hanya menggunakan bagian wajah individu dan tidak menggunakan aksesoris seperti berkacamata atau berkerudung.

3. Data citra yang digunakan adalah data citra dengan format ekstensi *.jpg.
4. Pada citra input hanya terdapat satu wajah manusia..

1.4 Tujuan

Tujuan umum dari penelitian ini adalah untuk mendapatkan program/perangkat lunak yang dapat digunakan untuk mengenali individu.

Sedangkan tujuan khusus dari penelitian ini adalah sebagai berikut:

1. Mendapatkan program untuk *preprocessing* citra individu
2. Mengaplikasikan *Histogram of Oriented Gradient* pada proses ekstraksi fitur.
3. Mengaplikasikan *K-Nearest Neighbor* pada proses klasifikasi fitur.

1.5 Manfaat

Manfaat dari penelitian ini adalah:

1. Dapat digunakan untuk pengenalan individu melalui pengenalan wajah menggunakan metode *Histogram of Oriented Gradient* dengan asumsi dataset identitas wajah sudah dilakukan oleh pengguna dan dapat dimanfaatkan oleh pihak instansi atau lembaga pemerintah, kantor, polisi atau universitas untuk tujuan tertentu.
2. Membangun sistem deteksi dini berbasis pengenalan wajah.

1.6 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini disusun berdasarkan beberapa bab sebagai berikut:

1. BAB I Pendahuluan

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat

dari penelitian yang dilakukan, serta sistematika penulisan Tugas Akhir.

2. BAB II Tinjauan Pustaka
Bab ini menjelaskan tentang studi penelitian sebelumnya dan beberapa teori pendukung yang berkaitan dengan Tugas Akhir ini.
3. BAB III Metodologi Penelitian
Bab ini menjelaskan tentang langkah-langkah yang digunakan untuk menyelesaikan Tugas Akhir ini.
4. BAB IV Analisis dan Perancangan Sistem
Bab ini menjelaskan tentang analisis dan perancangan sistem.
5. BAB V Implementasi
Bab ini menjelaskan tentang implementasi sistem, serta proses pembuatan sistem secara keseluruhan.
6. BAB VI Pengujian dan Pembahasan Hasil
Bab ini menjelaskan tentang hasil uji coba sistem. Kemudian dicatat sebagai bahan untuk merumuskan beberapa kesimpulan dari Tugas Akhir ini.
7. BAB VII Kesimpulan dan Saran
Bab ini berisi tentang kesimpulan dari hasil penelitian dan saran untuk pengembangan dari Tugas Akhir ini.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Tinjauan pustaka yang digunakan dalam penelitian ini adalah beberapa penelitian yang relevan dengan tema yang diambil. Diantaranya adalah penelitian yang dilakukan oleh Cahyo Permata dan kawan-kawan tentang deteksi mobil menggunakan metode *Histogram of Oriented Gradient*. Pada penelitian tersebut didapat bahwa untuk pengujian seluruh sistem dalam dapat ditarik kesimpulan bahwa hasil pengujian tes model menunjukkan tingkat keberhasilan mencapai 99,10% dalam mengenali *image* tes. Pada pengujian *image* tes positif, saat data training ditambah, tingkat keberhasilannya meningkat dari 95.39% menjadi 97,57 % . untuk pengujian *image* tes negatif, hasilnya meningkat dari 99,60% menjadi 99,90% pada pengujian ketiga dan konstan untuk pengujian berikutnya. sedangkan untuk pengujian *image* gabungan (tes *image* positif dan negatif) tingkat keberhasilannya naik 98.23% menjadi 99,10% [4]. Namun dalam penelitian ini menggunakan kenaraan sebagai objek penelitiannya

Penelitian lain adalah Implementasi Pengenalan Wajah Menggunakan PCA (*Principal Component Analysis*) yang dilakukan oleh Dian Esti Pratiwi dan Agus Harjoko. Pada penelitian tersebut Dian dan Agus menggunakan nearest neighbor untuk mengklasifikasikan fitur wajah yang didapat dari ekstraksi fitur menggunakan metode PCA. Pada uji pengenalan yang dilakukan dalam penelitian tersebut, hasil pengujian menunjukkan dari 16 sampel yang diujikan sebanyak 64 kali pengenalan menghasilkan 53 pengenalan benar dan 11 pengenalan salah. Berdasarkan hasil pengujian, data wajah sejumlah 160 wajah dari delapan orang memerlukan waktu kurang lebih dua menit pada setiap satu kali proses pengenalan. Prosentase keberhasilan pengenalan wajah pada sampel yang dilakukan adalah 82,81% [5].

Namun dalam penelitian ini penulis menggunakan PCA sebagai pengekstraksi fitur.

2.2 Citra Digital

Citra adalah representasi, kemiripan atau imitasi dari suatu objek atau benda[9]. Secara matematis, citra dinyatakan sebagai suatu fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Citra dibedakan menjadi dua yaitu citra kontinu diperoleh dari sistem optik yang menerima sinyal analog(mata manusia dan kamera analog) dan citra diskrit(digital) dihasilkan melalui proses digitalisasi terhadap citra kontinu. Proses digitalisasi pada citra digital dibagi menjadi dua proses yakni sampling dan kuantisasi. Proses sampling merupakan proses pengambilan nilai diskrit koordinat bidang(x, y) secara periodik dengan periode sampling T . Proses kuantisasi merupakan proses pengelompokkan nilai tingkat keabuan citra kontinu kedalam beberapa level atau merupakan proses membagi skala keabuan $(0, L)$ menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat(integer), dinyatakan *sebagai* $G = 2^m$, dengan G adalah derajat keabuan dan m adalah bilangan bulat positif.

Dengan demikian citra digital dapat didefinisikan suatu matriks A berukuran M (baris) x N (kolom) seperti pada Persamaan 2.1 dengan indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya menyatakan tingkat keabuan pada titik tersebut.

$$A = \begin{bmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ a_{1,0} & \cdots & a_{1,N-1} \\ \vdots & \ddots & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (2.1)$$

2.3 Pengenalan Wajah

Wajah merupakan bagian dari tubuh manusia yang menjadi fokus perhatian di dalam interaksi sosial, wajah memainkan peranan vital dengan menunjukkan identitas dan emosi. Kemampuan manusia untuk mengetahui seseorang dari wajahnya sangat luar biasa. Kita dapat mengenali ribuan wajah karena frekuensi interaksi yang sangat sering ataupun hanya sekilas bahkan dalam rentang waktu yang sangat lama. Bahkan kita mampu mengenali seseorang walaupun terjadi perubahan pada orang tersebut karena bertambahnya usia atau pemakaian kacamata atau perubahan gaya rambut. Oleh karena itu wajah digunakan sebagai organ dari tubuh manusia yang dijadikan indikasi pengenalan seseorang atau face recognition [11].

2.4 Pengolahan Citra

Pengolahan citra merupakan pemrosesan citra, khususnya dengan menggunakan komputer untuk menjadikan kualitas citra menjadi lebih baik. Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpertasi oleh manusia atau komputer[16].

2.4.1 Citra Grayscale

Dalam komputasi, suatu citra digital grayscale atau greyscale adalah suatu citra dimana nilai dari setiap pixel merupakan simple tunggal. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan putih pada intensitas yang terkuat. Citra grayscale berbeda dengan citra hitam-putih, dimana pada konteks komputer, citra hitam-putih hanya terdiri dari atas 2 warna yaitu hitam dan putih saja. Citra grayscale disimpan dalam format 8 bit untuk setiap simple pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak[15].

2.4.2 Viola-Jones

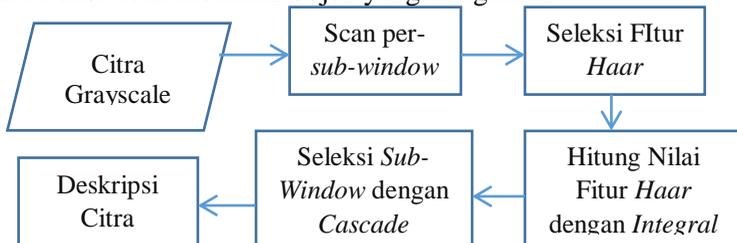
Viola-Jones merupakan salah satu metode yang digunakan untuk mendeteksi wajah manusia atau bagian dari wajah manusia. Metode ini bertujuan untuk membuang bagian-bagian yang tidak termasuk bagian wajah. Metode ini realtive mendapatkan hasil yang cepat, akurat dan efisien. Metode *Viola-Jones* paling banyak digunakan untuk mendeteksi wajah. Objek wajah dicari menggunakan *Viola-Jones* yang mengacu alur seperti Gambar 2.1, dimana citra *grayscale* akan di-*scan* per-sub-window untuk dicari fitur positif dengan *Adaboost* dan *Cascade Clasifier*. Jika wajah terdeteksi, akan dilakukan penggambaran garis persegi pada wajah tersebut. Pendeteksian objek menggolong-kan gambar berdasarkan pada nilai dari fitur sederhana. Operasi dasar dari suatu fitur jauh lebih cepat dibandingkan dengan pengolahan *pixel*. Sejumlah FITur Haar mewakili wilayah persegi pada citra dan menjumlahkan semua piksel pada daerah tersebut.

Viola-Jones mengklasifikasikan citra dari nilai fitur-fitur sederhana dan menggunakan tiga jenis fitur, yaitu fitur persegi, fitur tiga persegi dan fitur empat persegi. Nilai dari fitur-fitur tersebut adalah selisih antara daerah hitam dan putih.

Di dalam tiap sub-*window image*, jumlah total dari fitur Haar sangat besar, jauh lebih besar jika dibandingkan dengan jumlah *pixel*. Untuk memastikan pengklasifikasian dapat dilakukan secara cepat, proses pembelajaran harus menghilangkan fitur-fitur mayoritas yang tersedia, dan memusatkan pada sekumpula kecil fitur yag perlu. *Adaboost* bertujuan untuk membentuk *template* wajah.

Suatu metode klasifikasi yang menggunakan beberapa tingkatan dalam penyeleksian. Pada tiap tingkatan dilakukan penyeleksian menggunakan algoritma *Adaboost* yang telah di-*training* dengan menggunakan Fitur Haar. Penyeleksian berguna untuk memisahkan antara sub-*window* yang mengandung positif objek dengan negatif objek [14]. Positif

objek adalah gambar yang terdeteksi memiliki objek yang diinginkan sedangkan negatif objek, adalah gambar yang terdeteksi tidak memiliki objek yang diinginkan.



Gambar 2.1 Diagram alir metode *Viola-Jones*

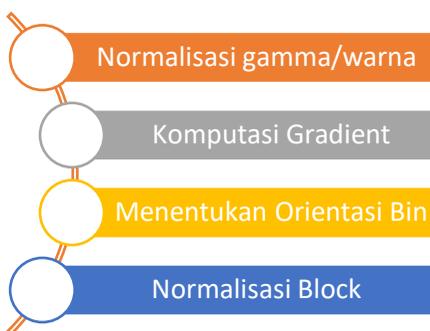
2.5 Ekstraksi Fitur

Fitur adalah atribut individual atau properti dari sebuah objek yang relevan dalam mendeskripsikan dan mengenali objek dan bisa membedakannya dari objek lain. Ekstraksi fitur [10] yaitu proses transformasi dari set input data yang diberikan menjadi himpunan fitur. Feature vector yaitu fitur yang dikelompokkan dari data yang diberikan. Pada proses ini, komponen pada citra diekstrak dan dikarakterisasi.

2.5.1 *Histogram of Oriented Gradient*

HOG adalah ekstraksi fitur yang digunakan pada komputer visi dan pengolahan citra dengan cara menghitung nilai gradien pada suatu citra untuk mendapatkan hasil yang akan digunakan untuk mendeteksi objek [8]. Hal terpenting dari HOG ialah tampilan lokal objek dan bentuk dalam sebuah citra dapat dijelaskan oleh distribusi intensitas gradien atau arah tepi. Implementasi HOG dapat dicapai dengan membagi citra menjadi suatu bagian yang kecil dan saling terhubung yang dapat disebut sebagai sel, dan untuk setiap sel menyusun gradien arah atau orientasi tepi untuk piksel. Untuk peningkatan kinerja, histogram lokal dapat dinormalisasi secara kontras dengan menghitung ukuran intensitas pada

seluruh wilayah besar dari citra yang disebut *block*, kemudian menggunakan nilai ini untuk normalisasi semua sel dalam *block* [7].



Gambar 2.2 Langkah-langkah ekstraksi HOG

Berdasarkan Gambar 2.2, langkah-langkah ekstraksi fitur dengan HOG adalah sebagai berikut :

1) Normalisasi Gamma atau *Square-Root* dan Warna

Tahap pre-processing citra ialah normalisasi warna dan nilai gamma atau *square-root*. Tahap ini bisa diabaikan karena memiliki pengaruh yang kecil pada kinerja [3]. Pada tahap normalisasi gamma dihitung \sqrt{p} (normalisasi *square-root*) atau $\log(p)$ (normalisasi gamma) dari setiap piksel p pada citra input. Definisi normalisasi *square-root* yaitu meng-*compress* (memampatkan) intensitas piksel jauh lebih kecil daripada normalisasi gamma. Berdasarkan penelitian Dalal dan Triggs, normalisasi *square-root* dapat meningkatkan akurasi. Sedangkan dengan menggunakan normalisasi gamma/*power law*, pendekatan ini mungkin merupakan “*over-correction*” dan menurunkan kinerja. Selain dilakukan normalisasi gamma, juga dilakukan normalisasi warna. Citra RGB akan diproses menjadi citra *grayscale*

untuk memudahkan penghitungan gradien citra. Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai *Red* = *Green* = *Blue*. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian intensitas terlemah dan warna putih pada intensitas terkuat.

2) Komputasi Gradien

Menggunakan informasi gradien untuk mengolah citra bisa membantu mendapatkan informasi contour citra dan mengurangi pengaruh perubahan cahaya. Pada tahap ini, akan dihitung gradien horizontal, dan vertical, citra. Setelah mendapatkan nilai informasi gradien, akan dihitung gradien orientasi (θ) dan magnitude (m) untuk setiap pixel pada citra. Kemudian menghitung gradien horizontal $f_x(x, y)$ dengan Persamaan (2.3) dan gradien vertikal $f_y(x, y)$ dengan Persamaan (2.2), sehingga untuk nilai piksel citra tepi tidak berubah [2].

$$f_y(x, y) = I(x, y + 1) - I(x, y - 1) \quad (2.2)$$

$$f_x(x, y) = I(x + 1, y) - I(x - 1, y) \quad (2.3)$$

dengan $I(x, y)$ adalah nilai intensitas citra di posisi x dan y . Setelah mendapatkan nilai informasi gradien, akan dihitung orientasi gradien (θ) dengan Persamaan (2.5) dan magnitude (m) dengan Persamaan (2.6) untuk setiap piksel pada citra

$$m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (2.4)$$

$$\theta = \tan^{-1} \left(\frac{f_y(x, y)}{f_x(x, y)} \right) \quad (2.5)$$

Untuk gambar berwarna, gradien dihitung secara terpisah untuk tiap *channel* warna, dan untuk tiap piksel, dipilih *channel* dengan nilai gradien tertinggi.

3) Menentukan Orientasi Bin

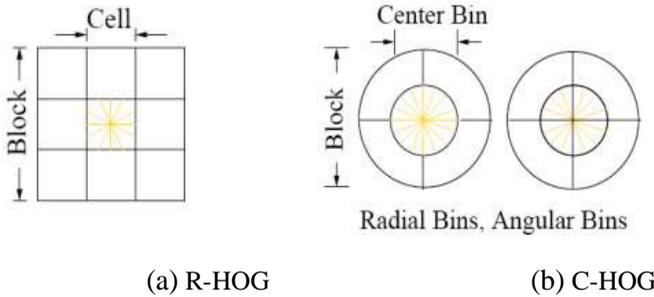
Orientasi bin berjarak merata dari 0° - 180° untuk gradien “*unsigned*” atau 0° - 360° untuk gradien “*signed*”. Window citra kemudian dibagi menjadi daerah spasial kecil yang berbentuk *rectangular* berukuran 8×8 yang disebut *cell*. Setiap piksel di dalam *cell* ditempatkan di dalam bin berdasarkan orientasi mereka. Kemudian sebuah histogram terbentuk untuk setiap *cell* [7].

4) Normalisasi Block

Normalisasi kontras lokal yang efektif adalah hal yang penting untuk menyelesaikan permasalahan pencahayaan dan variasi *foreground-background*. *Cell-cell* digabung untuk membentuk sebuah *block*. *Block* bisa jadi berbentuk *rectangular* (R-HOG) atau *circular* (C-HOG) seperti ditunjukkan pada Gambar 2.3. Kemudian histogram yang besar dibentuk dengan mengkombinasikan semua histogram yang dibuat di dalam sebuah *block*. Setelah dikombinasikan, histogram dinormalisasi dengan Persamaan (2.6) berikut [9]:

$$V_i = \frac{v_i}{\sqrt{\|v\|_i^2 + \epsilon^2}} \quad (2.6)$$

dengan v_i adalah vektor fitur yang berhubungan dengan histogram yang digabungkan untuk satu *block*, i adalah bilangan dari $1 \leq i \leq 27(3 \text{ cell} \times 9 \text{ bin})$, $\|v\|_i^2 = v_1^2 + v_2^2 + \dots + v_{27}^2$, dan ϵ suatu konstanta kecil bernilai 1 untuk menghindari pembagian oleh nol. Setelah itu histogram untuk setiap *block* dikumpulkan untuk membuat vektor fitur.



Gambar 2.3 Bentuk *Cell*

Setelah itu histogram untuk setiap *block* dikumpulkan untuk membuat vektor fitur berukuran 1-D. Selanjutnya vektor fitur tersebut ditempatkan dalam algoritma *learning* yang akan memutuskan apakah citra wajah manusia tersebut dikenali atau tidak berdasarkan data *training*.

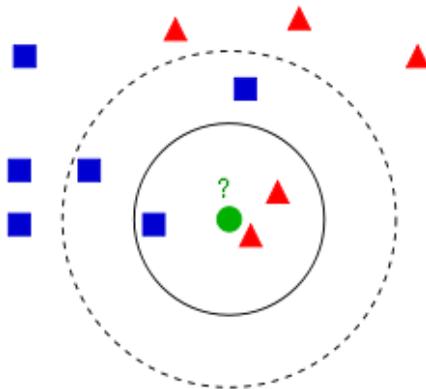
Untuk perhitungan vektor fitur yang dihasilkan, misalkan untuk sebuah citra berukuran 128×128 , ukuran *cell* 8×8 , dan ukuran *block* 16×16 dimana satu *block* terdiri dari dua *cell*, maka menghasilkan 225 *block* dalam sebuah citra karena satu *block* bisa *slid* 15 kali ketika orientasi horizontal dan vertical ($block = 128 : 8 - 1 \times 128 : 8 - 1$). Sementara itu, terdapat 9 *channel bin* dalam satu *cell* dan 36 ($(2 \times 2) \times 9$) fitur dalam satu *block*. Maka, akan didapatkan jumlah fitur adalah 8100 (225×36) untuk citra berukuran 128×128 [7]. Secara ringkas, panjang fitur HOG bisa dihitung sebagai berikut :

$$\begin{aligned}
 \text{Panjang fitur} &= \text{Block} \times \text{Cell Per Block} \times \text{Bin Per Cell} \\
 &= (128 : 8 - 1) \times (128 : 8 - 1) \times (2 \times 2) \times 9 \\
 &= 8100
 \end{aligned}$$

2.6 Klasifikasi Menggunakan Algoritma *K-Nearest Neighbor*

Klasifikasi adalah proses di mana unsur-unsur individu dikelompokkan menurut kesamaan antara elemen dan deskripsi grup. Dengan kata lain, klasifikasi yaitu menghubungkan kategori standar atau label kelas kepada contoh tertentu dari dataset input, kemudian memisahkan data input ke dalam kategori yang berbeda (kelas) dengan fitur-fitur umum.

K-Nearest Neighbor (KNN) adalah suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan *training sample*. Klasifikasi menggunakan *K-Nearest Neighbor* (KNN) ditunjukkan pada Gambar 2.4. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah k objek atau (titik *training*) yang paling dekat dengan titik *query*.



Gambar 2.4 Klasifikasi Menggunakan K-Nearest Neighbor

Klasifikasi menggunakan *voting* terbanyak diantara klasifikasi dari k objek. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru.

Algoritma metode KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN-nya. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi *training sample*. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance* yang direpresentasikan sebagai berikut :

$$D(a,b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad (2.7)$$

Dimana matriks $D(a,b)$ adalah jarak skalar dari kedua vektor a dan b dari matriks dengan ukuran d dimensi. Pada fase *training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk *testing data* (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor *training sample* dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi

menjadi semakin kabur. KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar. [6].

BAB III

METODOLOGI PENELITIAN

3.1 Objek dan Aspek Penelitian

Objek yang digunakan dalam penelitian ini adalah citra wajah individu. Sedangkan aspek penelitian yang diteliti adalah mengenali individu melalui identifikasi citra wajah dari individu tersebut dengan menggunakan metode *Histogram of Oriented Gradient*.

3.2 Peralatan

Peralatan utama yang digunakan dalam penelitian ini terdiri perangkat keras dan perangkat lunak, antara lain:

1. Perangkat keras
 - a. Kamera digital dan/atau CCTV
 - b. Personal computer atau laptop dengan spesifikasi:
 - Windows 8, 64-bit Operating System
 - Resolusi 1366x768 Pixel
 - Intel® Pentium® Quad-Core N3700 Processor
 - Memori DDR3L 1600 MHz SDRAM, up to 4 G, OnBoard Memory
2. Perangkat Lunak

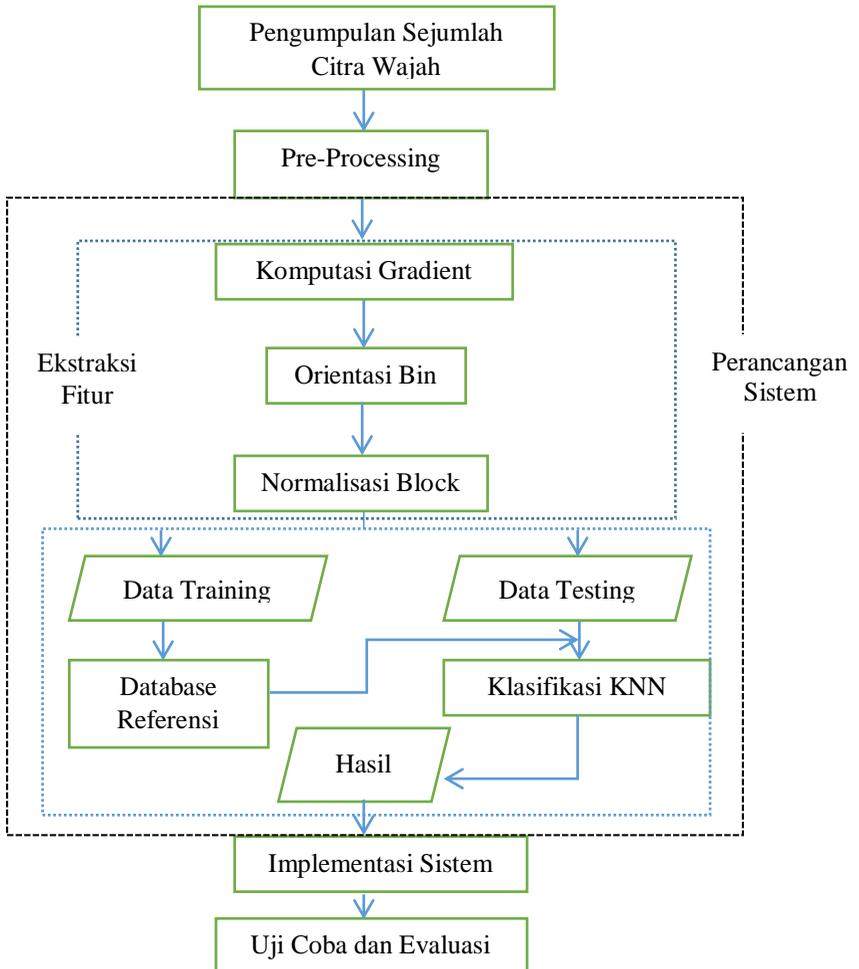
Perangkat lunak yang digunakan untuk membuat simulasi sistem pengenalan individu identifikasi wajah adalah Matlab versi minimum R2010a dan Microsoft Excel.

3.3 Data Penelitian

Data penelitian dalam penelitian ini merupakan citra wajah individu yang ditangkap (*capture*) menggunakan kamera digital atau CCTV.

3.4 Tahap Penelitian

Dalam pengerjaan penelitian ini ada beberapa tahapan, antara lain, dapat dilihat pada diagram alir seperti Gambar 3.1.



Gambar 3.1 Blok diagram tahapan penelitian

1. Pengambilan sejumlah citra wajah

Pengumpulan sejumlah citra wajah dilakukan untuk memperoleh informasi yang dibutuhkan untuk mencapai tujuan dari Tugas Akhir ini. Citra yang digunakan untuk Tugas Akhir ini berupa citra wajah manusia. Sejumlah citra wajah manusia akan dijadikan data set dan data testing dengan citra wajah yang bervariasi.

2. Perancangan sistem pengenalan individu

Aplikasi pengenalan individu melalui identifikasi wajah ini akan dirancang dengan metode *Histogram of Oriented Gradient* sebagai pengekstraksi fitur. Sistem ini dirancang dengan perangkat lunak dan perangkat keras.

3. Preprocessing citra

Proses yang dilakukan untuk meningkatkan kualitas citra wajah sehingga dapat meningkatkan kemungkinan keberhasilan pada tahap pengolahan citra selanjutnya, yaitu proses ekstraksi fitur. Tahap processing citra wajah manusia yang dilakukan adalah konversi sehingga menjadi citra grayscale. Proses grayscale dilakukan untuk membantu manipulasi bit yang tidak terlalu banyak. Kemudian citra dideteksi bagian wajahnya menggunakan library *Viola Jones* bertujuan untuk mendapatkan bagian wajah manusia yang terdapat pada sebuah citra. Hasil deteksi yang didapat kemudian dicrop untuk mengurangi bagian-bagian yang bukan merupakan bagian wajah dan kemudian diubah ukuran dimensinya menjadi 128 x 128. Proses perubahan dimensi ini bertujuan untuk menyeragamkan ukuran dan untuk membantu meringankan kinerja program.

4. Ekstraksi fitur

Pada proses ini akan dilakukan ekstraksi dan karakterisasi komponen pada citra. Ekstraksi fitur dilakukan dengan menggunakan metode *Histogram of Oriented Gradient*.

5. Klasifikasi fitur

Proses klasifikasi dengan *K-Nearest Neighbor* yang menentukan pemilik wajah tersebut.

6. Uji coba dan pembahasan

Uji coba dan pembahasan dilakukan dengan melakukan simulasi sistem deteksi terhadap data citra wajah. Proses ini untuk membantu memahami cara penggunaan sistem dan menganalisis hasil uji coba dan pembahasan sehingga dapat diberikan untuk pengembangan penelitian selanjutnya.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai perancangan sistem dan hasil implementasi semua proses yang telah dirancang sebelumnya. Pembahasan perancangan sistem diawali dengan penjelasan tentang lingkungan perancangan sistem, perancangan data, perancangan proses, serta perancangan antar muka.

4.1 Perancangan Sistem

Tampilan dari sistem pengenalan individu yang melalui identifikasi wajah ini dibangun dengan tampilan yang sederhana. *Software* yang digunakan untuk membangun sistem ini adalah Matlab dan Microsoft Excel.

4.2 Format Data

Data yang digunakan dalam sistem pengenalan individu ini dibagi menjadi 3 macam, yaitu data masukan yaitu data citra wajah manusia, data proses yaitu data ketika tahap-tahap pemrosesan citra wajah manusia sedang dilakukan, dan data keluaran berupa hasil identifikasi wajah berupa nama dari pemilik wajah yang di identifikasi.

4.2.1 Data Masukan

Data masukan dalam sistem pengenalan individu ini adalah citra wajah manusia yang ditangkap (*capture*) menggunakan kamera digital dengan format .jpg dan berupa citra RGB.

Data masukan nantinya akan dibedakan menjadi 2 yaitu data pengujian dan data pelatihan. Pada Gambar 4.1 ditampilkan sebagian citra dari 160 citra yang digunakan untuk database referensi dari sistem identifikasi individu ini. Setiap individu memiliki 4 citra wajah yang akan disimpan di

database referensi dengan setiap individunya memiliki 4 posisi yang berbeda, yaitu posisi depan, kanan, kiri dan atas seperti yang terlihat pada gambar 4.1 yang. Lalu pada Gambar 4.2 ditampilkan beberapa data citra *testing* yang digunakan dalam sistem pengujian. Data masukan dalam sistem ini tidak melihat jenis kelamin dan umur individu. Data latih selengkapnya dapat dilihat pada LAMPIRAN A.



(a)



(b)



(c)



(d)

Gambar 4.1 Sebagian Data Training

(a)Tampak Depan (b)Tampak Kiri (c)Tampak Kanan (d)Tampak Atas



Gambar 4.2 Sebagian Citra *Testing*

4.2.2 Data Proses

Data proses adalah data yang digunakan selama proses berjalannya sistem yang merupakan hasil pengolahan dari data masukan untuk diproses kembali menjadi data keluaran di tahap selanjutnya. Data proses yang digunakan dalam proses ini ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Data Proses

No.	Nama Data	Keterangan
1.	Citra <i>grayscale</i>	Citra keabuan individu.
2.	Citra <i>resize</i>	Citra keabuan yang di <i>resize</i> menjadi berukuran 128×128 .
3.	Fitur citra	Matriks fitur citra yang didapatkan dari proses ekstraksi fitur.
4.	<i>Euclidian Distance</i>	Nilai yang didapat dari perhitungan <i>Euclidian Distance</i> Citra <i>Training</i> dan Citra <i>Testing</i>

4.2.3 Data Luaran

Data luaran yang dihasilkan sistem ini adalah hasil akhir dari proses identifikasi wajah yang menentukan pemilik dari wajah individu tersebut atau apakah wajah teridentifikasi atau tidak. Data berupa informasi ini diperoleh dari proses ekstraksi fitur dengan metode HOG. Setelah mendapatkan fitur dengan menggunakan HOG, dilakukan proses klasifikasi menggunakan K-Nearest Neighbor, dimana dalam proses klasifikasi ini akan dihitung *Euclidian Distance* dari nilai fitur yang didapat saat training data dan testing data. Kemudian nilai dari proses tersebut akan diurutkan dari yang terdekat ke yang terjauh.

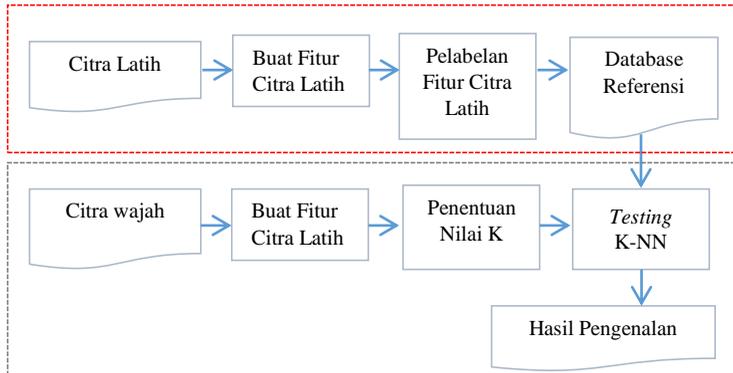
4.3 Gambaran Proses Secara Umum

Pada subbab ini akan diuraikan perancangan proses untuk membangun sebuah sistem pengenalan individu melalui identifikasi wajah. Perancangan proses ini bertujuan untuk memperjelas hubungan antar proses beserta langkah-langkah yang dilakukan pada setiap proses. Secara garis besar proses yang dilakukan adalah:

1. Data input, berupa citra wajah individu yang disimpan dalam media penyimpanan. Selanjutnya data dimasukkan ke dalam sistem.

2. *Preprocessing* sebelum dilakukan proses ekstraksi fitur. Pada proses ini, citra input yang masuk ke dalam sistem akan dirubah dari citra RGB menjadi citra *grayscale*. Setelah citra wajah dirubah menjadi citra keabuan, akan dilakukan proses deteksi bagian wajah menggunakan *viola-jones*. Bagian wajah yang terdeteksi kemudian dicrop dan terapkan proses *resizing*. Pada proses *resizing* akan dilakukan perubahan ukuran dimensi citra menjadi 128 x 128.
3. Proses ekstraksi fitur dengan metode HOG yaitu proses yang dilakukan terhadap citra input yang sebelumnya telah dilakukan *preprocessing* untuk memperoleh ciri citra wajah manusia yang akan digunakan untuk proses berikutnya, yaitu proses klasifikasi.
4. Proses klasifikasi dengan *K-Nearest Neighbor* merupakan proses pengklasifikasian citra wajah terhadap pemilik dari wajah tersebut. Ada 2 proses yang dilakukan pada proses ini yaitu:
 - a. Proses *training* (pelatihan), adalah proses untuk melatih sistem dengan data masukan yang telah diolah sebelumnya sehingga mampu mengenali apabila diberi masukan yang baru. Pada proses ini didapat fitur yang akan disimpan di database referensi. Fitur data training di labeli secara manual menggunakan *software* Microsoft excel. Nilai yang ada di database referensi ini akan digunakan untuk perhitungan *Euclidian Distance* yang akan menentukan hasil dari kepemilikan wajah tersebut.
 - b. Proses *testing* (pengujian), yaitu proses pengambilan keputusan terhadap citra uji yang dimasukkan. Dengan mendapatkan jarak antara data training dan data testing menggunakan perhitungan *Euclidian Distance* yang kemudian diurutkan dari yang terdekat ke yang terjauh. Setelah diurutkan, ditentukan nilai **K** dimana **K** merupakan jumlah

tetangga terdekat. Dari proses tersebut menghasilkan nama dari pemilik wajah tersebut.



Gambar 4.3 Diagram Alir Proses Pengenalan Individu Secara Umum

Gambar 4.3 menunjukkan digram alir secara keseluruhan proses pengenalan wajah melalui identifikasi wajah secara umum, dimana terdapat proses *training* dengan urutan proses citra latih masuk ke dalam sistem lalu dilakukan proses ekstraksi fitur dari citra latih tersebut untuk mendapatkan fitur dari citra latih, kemudian dilakukan pelabelan untuk setiap fitur citra latih. Fitur citra latih yang didapat dihitung bersamaan dengan fitur citra testing untuk dihitung nilai *Euclidian Distance*-nya.

4.3.1 Preprocessing

Data input yang masuk ke dalam sistem pengenalan individu melalui identifikasi wajah sebelum melangkah ke tahap ekstraksi fitur akan dilakukan proses *preprocessing*. Pada tahap *preprocessing* ini terdapat beberapa proses, yaitu yang pertama adalah perubahan citra wajah RGB menjadi citra wajah *grayscale*. Kemudian dari citra *grayscale* ini

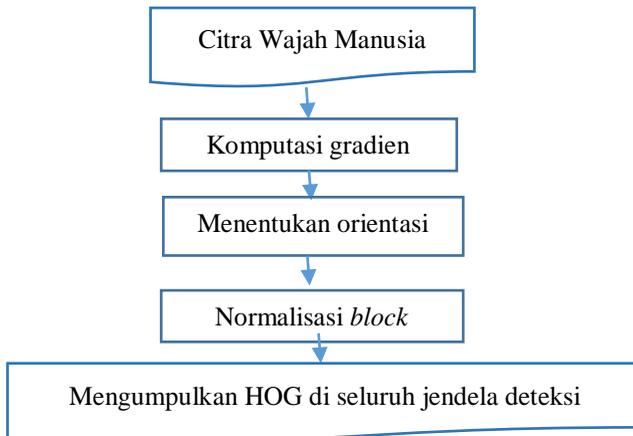
dideteksi bagian mana yang termasuk bagian wajah individu menggunakan metode viola-jones. Setelah didapat bagian wajah individu, bagian wajah individu tersebut dicrop yang kemudian hasil *cropping* di *resize* menjadi citra berukuran dimensi 128 x 128.

4.3.2 Proses Ekstraksi Fitur dengan *Histogram of Oriented Gradient* (HOG)

Ekstraksi fitur merupakan proses yang bertujuan untuk menentukan ciri dari suatu citra yang mampu membedakan wajah individu satu dengan yang lainnya. Metode *Histogram of Oriented Gradient* (HOG) merupakan salah satu metode yang digunakan untuk ekstraksi fitur ke dalam kelas-kelas yang berbeda.

Pada proses ekstraksi fitur dengan HOG ini akan dihitung *magnitude* gradien dan orientasi gradien pada bidang lokal yang disebut *cell* dan dikonversi menjadi histogram. Gambar 4.4 menunjukkan diagram alir proses ekstraksi fitur dengan HOG. Untuk menghitung *descriptor* HOG, pertama dihitung gradien horizontal dan vertikal, selanjutnya didapatkan *magnitude* dan orientasi gradiennya. Citra dibagi ke dalam *cell-cell* berukuran 8×8 . Kemudian nilai orientasi tiap pikselnya dikuantisasi ke dalam 9 *bin* (kanal) dalam sudut 180° menggunakan histogram. Dalam Tugas Akhir ini digunakan *unsigned gradient* dengan 9 bin setiap *cell* (1 bin = 20°). Kontribusi piksel terhadap tiap *bin* bergantung pada nilai *magnitude* gradiennya. Nilai-nilai dari seluruh *bin* dari tiap *cell* akan membentuk garis vektor dengan jumlah garis sama dengan jumlah *bin* yang dibentuk, yaitu 9 garis dengan sudut 20° antara garis satu dengan garis yang selanjutnya. Untuk memaksimalkan fitur HOG maka dibentuk sebuah *block* yang merupakan gabungan dari beberapa *cell* dengan ukuran 16×16 . Teknik *overlapping block* dilakukan dengan membuat *block* yang bersebelahan saling *overlap* (tumpang

tindih) dan menghitung kembali bagian yang *overlap*. Kemudian histogram yang terbentuk pada tiap *block* dinormalisasi, setelah itu digabungkan semua vektor pada semua *cell*.



Gambar 4.4 Diagram Alir Proses Ekstraksi Fitur dengan HOG

Contoh perhitungan manual HOG adalah sebagai berikut, dengan diberikan matriks A berdimensi 16×16 dimana matriks A merupakan representasi dari citra wajah.

$$A = \begin{bmatrix} 155 & 89 & 102 & 149 & 158 & 154 & 150 & 145 & 139 & 130 & 121 & 116 & 112 & 83 & 70 & 85 \\ 100 & 89 & 140 & 170 & 169 & 173 & 173 & 160 & 151 & 143 & 130 & 119 & 117 & 103 & 71 & 70 \\ 86 & 98 & 160 & 173 & 177 & 180 & 177 & 167 & 156 & 145 & 135 & 125 & 119 & 113 & 76 & 65 \\ 81 & 116 & 175 & 172 & 170 & 181 & 176 & 167 & 160 & 152 & 134 & 116 & 112 & 120 & 84 & 60 \\ 77 & 146 & 181 & 163 & 159 & 157 & 159 & 168 & 157 & 144 & 129 & 119 & 111 & 120 & 94 & 54 \\ 95 & 149 & 183 & 172 & 152 & 132 & 141 & 163 & 142 & 133 & 136 & 110 & 112 & 124 & 97 & 50 \\ 114 & 160 & 182 & 157 & 136 & 104 & 122 & 161 & 138 & 126 & 127 & 96 & 101 & 120 & 98 & 51 \\ 97 & 166 & 179 & 175 & 167 & 153 & 151 & 156 & 129 & 121 & 143 & 132 & 126 & 126 & 92 & 51 \\ 83 & 162 & 181 & 181 & 179 & 167 & 151 & 149 & 128 & 103 & 146 & 155 & 138 & 128 & 89 & 49 \\ 90 & 170 & 179 & 185 & 193 & 154 & 152 & 170 & 147 & 113 & 123 & 169 & 149 & 132 & 89 & 52 \\ 76 & 101 & 166 & 185 & 178 & 155 & 169 & 160 & 143 & 116 & 123 & 159 & 148 & 130 & 67 & 48 \\ 62 & 61 & 149 & 183 & 170 & 162 & 162 & 154 & 140 & 119 & 115 & 144 & 143 & 118 & 50 & 45 \\ 58 & 58 & 105 & 182 & 175 & 155 & 147 & 149 & 129 & 1108 & 107 & 147 & 142 & 86 & 39 & 45 \\ 58 & 60 & 60 & 136 & 176 & 168 & 173 & 165 & 145 & 134 & 122 & 138 & 108 & 48 & 43 & 49 \\ 59 & 59 & 55 & 68 & 156 & 173 & 175 & 169 & 155 & 136 & 121 & 112 & 55 & 40 & 49 & 51 \\ 57 & 56 & 57 & 61 & 148 & 164 & 172 & 174 & 160 & 136 & 113 & 76 & 39 & 45 & 49 & 51 \end{bmatrix}$$

Langkah pertama pada HOG adalah normalisasi warna, proses ini dianggap sudah dilakukan. Proses normalisasi warna disini adalah proses konversi citra RGB menjadi citra keabuan. Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, sedangkan pada citra RGB terdapat kombinasi dari 3 kanal yang memungkinkan nilai intensitas pikselnya mencapai nilai jutaan atau lebih. Sehingga dilakukan proses ini untuk memudahkan penghitungan gradien citra. Kemudian setelah melalui proses normalisasi gamma, dilakukan proses komputasi gradient yaitu menghitung gradient horizontal dan vertikal dari matriks citra menggunakan Persamaan 2.3 untuk gradient horizontal, sehingga didapat hasil dari perhitungan gradient horizontal adalah:

$$f_x = \begin{bmatrix} 69 & -9 & -58 & -24 & -19 & -26 & -27 & -22 & -17 & -15 & -14 & -9 & -7 & -30 & -6 & 20 \\ 19 & -27 & -35 & -2 & -1 & -8 & -3 & -7 & -9 & -9 & -4 & 3 & 5 & -17 & -13 & 10 \\ 9 & -48 & -21 & 10 & 18 & 23 & 18 & -1 & -1 & 1 & 6 & 6 & -7 & -7 & -18 & 11 \\ -14 & -33 & -8 & 0 & 18 & 49 & 35 & 4 & 18 & 19 & -2 & 6 & 0 & -4 & -13 & 10 \\ -37 & -14 & -1 & 6 & 23 & 53 & 37 & 7 & 19 & 18 & 2 & 23 & 10 & 0 & -4 & 3 \\ -2 & -17 & 4 & -3 & -15 & -21 & -10 & 7 & 13 & 12 & -7 & -22 & -14 & -2 & 5 & -1 \\ 31 & -2 & 1 & -24 & -43 & -63 & -29 & 12 & 10 & 23 & -19 & -59 & -37 & -8 & 9 & 2 \\ 7 & -4 & 0 & -10 & -26 & -1 & -1 & 14 & -18 & 8 & 20 & -37 & -23 & -6 & 3 & -1 \\ 7 & 61 & 15 & -4 & 1 & 12 & -18 & -11 & -15 & -13 & 23 & -4 & -10 & -2 & 22 & 1 \\ 28 & 109 & 30 & 2 & 23 & -8 & -10 & 16 & 7 & -6 & 8 & 25 & 6 & 14 & 39 & 7 \\ 18 & 43 & 61 & 3 & 3 & 0 & 22 & 11 & 14 & 8 & 16 & 12 & 6 & 44 & 28 & 3 \\ 4 & 1 & 89 & 47 & -6 & -6 & -11 & -11 & -5 & -15 & -7 & 6 & 35 & 70 & 7 & -4 \\ -1 & -1 & 50 & 114 & 19 & -18 & -28 & -20 & -26 & -28 & -14 & 35 & 87 & 46 & -10 & -6 \\ 1 & 4 & 3 & 75 & 28 & 4 & 1 & -9 & -15 & -2 & 9 & 62 & 69 & 3 & -6 & -3 \\ 59 & 59 & 55 & 68 & 156 & 173 & 175 & 169 & 155 & 136 & 121 & 112 & 55 & 40 & 49 & 51 \\ 57 & 56 & 57 & 61 & 148 & 164 & 172 & 174 & 160 & 136 & 113 & 76 & 39 & 45 & 49 & 52 \end{bmatrix}$$

untuk gradient vertikal menggunakan Persamaan 2.2 dan didapat hasil dari perhitungan gradient vertikal adalah:

$$f_y = \begin{bmatrix} 53 & -60 & -56 & -5 & 8 & 9 & 11 & 15 & 18 & 14 & 9 & 33 & 42 & -2 & 70 & 85 \\ -40 & -81 & -29 & -3 & -4 & 13 & 22 & 17 & 21 & 24 & 13 & 16 & 46 & 33 & 71 & 70 \\ -74 & -75 & -17 & -7 & 0 & 13 & 21 & 22 & 21 & 20 & 16 & 12 & 43 & 48 & 76 & 65 \\ -94 & -56 & 5 & -9 & -6 & 14 & 16 & 15 & 26 & 36 & 22 & -4 & 28 & 60 & 84 & 60 \\ -104 & -17 & 22 & 6 & 0 & -11 & 2 & 24 & 28 & 25 & 18 & -1 & 17 & 66 & 94 & 54 \\ -88 & -23 & 31 & 40 & 11 & -31 & -1 & 30 & 6 & 23 & 24 & -14 & 15 & 74 & 97 & 50 \\ -68 & 3 & 46 & 53 & 14 & -57 & -16 & 35 & 11 & 30 & 26 & -24 & 3 & 69 & 98 & 51 \\ -82 & -9 & 12 & 22 & 16 & -3 & 22 & 35 & -14 & -11 & 17 & 6 & 34 & 75 & 92 & 51 \\ -98 & -19 & 2 & 14 & 28 & 18 & 23 & 46 & -18 & -52 & 8 & 27 & 49 & 79 & 89 & 49 \\ -89 & -15 & -14 & 31 & 41 & -16 & 5 & 57 & 24 & -56 & -26 & 37 & 60 & 80 & 89 & 52 \\ -90 & -84 & -12 & 30 & 9 & -5 & 26 & 44 & 20 & -43 & -25 & 29 & 81 & 82 & 67 & 48 \\ -87 & -122 & -21 & 21 & 8 & 8 & 22 & 35 & 25 & -25 & -28 & 26 & 93 & 73 & 50 & 45 \\ -47 & -124 & -70 & 27 & 28 & 6 & 18 & 41 & 22 & -39 & -35 & 61 & 103 & 41 & 39 & 45 \\ -2 & -76 & -116 & -32 & 3 & 3 & 28 & 31 & 23 & -4 & 14 & 90 & 65 & -1 & 43 & 49 \\ 4 & -9 & -101 & -105 & -19 & 4 & 20 & 33 & 34 & 24 & 66 & 72 & 6 & -11 & 49 & 51 \\ 0 & -5 & -91 & -103 & -24 & -10 & 12 & 38 & 47 & 60 & 74 & 31 & -10 & -7 & 49 & 52 \end{bmatrix}$$

Kemudian setelah di dapat gradientnya, langkah berikutnya adalah mengitung magnitudo (m) dengan

Persamaan 2.4 dimana akan diakarkan jumlah dari nilai kuadrat gradient horizontal dan nilai kuadrat gradient vertikal seperti berikut:

$$\begin{aligned}
 m(x, y) &= \sqrt{69^2 + 53^2} \\
 &= \sqrt{7570} \\
 &= 87.0057469
 \end{aligned}$$

dari hasil perhitungan gradient kemudian dicari nilai angle (θ) dari tiap elemennya dengan Persamaan 2.5 seperti berikut:

$$\begin{aligned}
 \theta(x, y) &= \tan^{-1}\left(\frac{69}{53}\right) \\
 &= 127.5284
 \end{aligned}$$

Langkah berikutnya adalah menentukan orientasi bin dari nilai $m(x, y)$ dan $\theta(x, y)$. Proses menentukan orientasi seperti berikut:

$m(x, y)$	60.6712	80.6226	24.5153	20.6155	27.5136	29.1548	26.6271
87.0057	60.6712	80.6226	24.5153	20.6155	27.5136	29.1548	26.6271
44.2832	85.3815	45.4533	3.6056	4.1231	15.2643	22.2036	18.3848
74.5453	89.0449	27.0185	12.2066	18.000	26.4197	27.6586	22.0227
95.0368	55.000	9.4240	9.0000	18.9737	50.9600	38.4838	15.5242
110.3857	22.0227	22.0227	8.4853	23.000	54.1295	37.0540	25.000
88.0227	31.2570	31.2570	40.1123	18.6011	37.4433	10.0499	30.8058
74.7329	3.6056	46.0109	58.1808	45.2217	84.9588	33.1210	37.0000
82.2982	63.6946	12.000	24.1661	30.5287	3.1623	22.0227	37.6962

$\theta(x, y)$	171.4692	133.9949	101.7683	67.1163	70.9065	67.8337	55.7131
127.528	171.4692	133.9949	101.7683	67.1163	70.9065	67.8337	55.7131
25.4077	161.5651	129.6442	146.3099	70.9065	31.6075	7.7652	22.3801
6.9343	147.3808	128.9910	55.0080	31.6075	119.4759	139.3987	2.6026
171.5289	149.4898	57.8046	0	.90	105.9459	114.5672	165.0686
160.4160	140.5275	2.6026	135	71.5651	78.2749	93.0941	163.7398
178.6980	143.5308	17.26476	4.2892	53.7462	145.8855	95.7106	166.8660
24.5074	33.6901	178.7546	24.3625	71.9657	132.1378	118.8866	161.0754
4.8793	156.0375	180.000	24.4440	58.3925	161.5651	2.6026	21.8014

$m(x, y)$	22.0227	22.0227					87.0057	87.0057	
$\theta(x, y)$	0	20	40	60	80	100	120	140	160

proses di atas dilakukan pada *cell* berukuran 8x8 yang kemudian membentuk 9-bin histogram sehingga didapat hasil dari orientasi bin adalah:

Tabel 4.2 Hasil proses penentuan Orientasi Bin

$m(x, y)$	473.9409	74.5514	41.8151	83.8932	245.4427	710.4377	826.7447	827.7447	776.4837
$\theta(x, y)$	0	20	40	60	80	100	120	140	160

Proses terakhir HOG adalah normalisasi block. dimana tiap elemen yang tumpang tindih pada satu block akan di normalisasi menggunakan persamaan 2.6. Hasil normalisasi akan menghasilkan sebuah vektor berukuran $1 \times D$ seperti berikut:

$$V = [0.1260 \ 0.0665 \ 0.0250 \ 0.0640 \ 0.043 \ 0.0465 \ 0.129 \ 0.1261 \ 0.174 \ 0.200 \ 0.0602 \ 0.0286 \ 0.0124 \ 0.0313 \ 0.0287 \ 0.0401 \ 0.0729 \ 0.200 \ 0.2000 \ 0.0407 \ 0.1735 \ 0.200 \ 0.1523 \ 0.0144 \ 0.0507 \ 0.153 \ 0.1591 \ 0.0250 \ 0.0140 \ 0.0281 \ 0.082 \ 0.200 \ 0.200 \ 0.200 \ 0.200]$$

Dari proses HOG diatas menghasilkan matriks V yang merupakan vektor fitur yang mewakili dari sebuah citra.

4.3.3 Proses Klasifikasi dengan *K-Nearest Neighbor*

Tahapan setelah proses ekstraksi fitur dengan *Histogram of Oriented Gradient* (HOG) adalah proses klasifikasi dengan metode *K-Nearest Neighbor*. Proses klasifikasi ini adalah proses uji data dimana akan diklasifikasi sejumlah citra *testing* yang telah ditentukan dan citra tersebut akan dikenali pemiliknya melalui identifikasi wajah. Pada proses training, klasifikasi fitur dari wajah individu dilakukan secara manual dengan memberi nomer pada fitur secara satu persatu menggunakan excel.

Pada proses testing ini yang akan menentukan hasil uji dari data yang diinput. Misal, diberikan tiga fitur hasil ekstraksi berukuran 1 x D yang ditentukan secara acak.

$F_1 = [0.512 \ 0.634 \ 0.765 \ 0.878 \ 0.912]$ dengan label 1,

$F_2 = [0.234 \ 0.124 \ 0.788 \ 0.112 \ 0.347]$ dengan label 2,

$F_3 = [0.666 \ 0.542 \ 0.966 \ 0.439 \ 0.455]$ dengan label 3.

Akan dicari nilai *Euclidian distance* atau jarak Euclid dari masing masing vektor fitur yang dihitung bersamaan dengan nilai vector fitur data testing. Misal, diberikan fitur hasil ekstraksi data testing berukuran 1xD.

$$F_4 = [0,513 \ 0,654 \ 0,785 \ 0,877 \ 0,921]$$

Pertama yaitu menentukan nilai k atau jumlah tetangga terdekat sesuai keinginan user atau pengguna. Misal dipilih jumlah tetangga terdekatnya atau nilai **K** adalah 1. Kemudian menghitung *Euclidian distance* dari 3 fitur training ke fitur testing, dengan rumus *Euclidian distance* seperti Persamaan 2.7.

$$\begin{aligned} D(a, b) &= \sqrt{(0.512 - 0.513)^2 + (0.634 - 0.654)^2 \\ &\quad + (0.765 - 0.785)^2 + (0.878 - 0.877)^2 \\ &\quad + (0.912 - 0.921)^2} \\ &= \sqrt{(-0.001)^2 + (-0.02)^2 + (-0.02)^2 \\ &\quad + (0.1)^2 + (-0.009)^2} \\ &= \sqrt{0.000001 + 0.0004 + 0.0004 \\ &\quad + 0.01 + 0.000081} \\ &= \sqrt{0.000883} \end{aligned}$$

$$= 0.2970000$$

Perhitungan diatas adalah salah satu contoh perhitungan Euclidian Distance dari vektor fitur yang diberikan. Perhitungan ini dilakukan terhadap 2 matriks lainnya yang kemudian didapatkan hasil Euclidian distance dengan jarak tiap matriksnya yang dapat dilihat pada Tabel 4.2

Tabel 4.3 Euclidian Distance

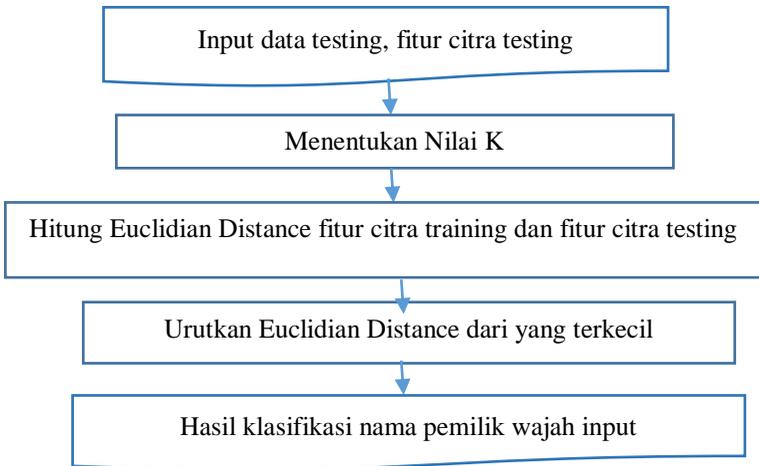
Euclidian Distance	Label
0.0297	1
1.1285	2
0.6912	3

Selanjutnya, hasil dari perhitungan Euclidian distance diurutkan dari yang terkecil hingga yang terbesar yang dapat dilihat pada Tabel 4.3,

Tabel 4.4 Tabel urutan dari yang terkecil ke yang terbesar

Euclidian Distance	Label
0.0297	1
0.6912	3
1.1285	2

Berdasarkan Tabel 4.3 dapat dilihat bahwa yang memiliki jarak terkecil adalah yang berlabel 1. Nilai K menentukan jumlah tetangga terdekat, juga menentukan nilai yang paling sering muncul atau modus. Maka F_4 merupakan vektor fitur dari individu yang berlabel 1.



Gambar 4.5 Diagram alir proses testing K-Nearest Neighbor

BAB V

IMPLEMENTASI

Pada bab ini, perancangan program yang telah dibangun selanjutnya diimplementasikan pada bahasa pemrograman dengan menggunakan Matlab. Pembahasan dalam implementasi sistem meliputi implementasi antarmuka (*interface*) sistem, implementasi tahap ekstraksi fitur dengan *Histogram of Oriented Gradient* (HOG), dan terakhir tahap klasifikasi dengan metode *K-Nearest Neighbor*.

5.1 Implementasi Antarmuka

Pada penelitian ini, antarmuka sistem dibangun dengan menggunakan form dan kontrol yang terdapat pada Matlab. Adapun antarmuka-antarmuka yang diimplementasikan untuk menunjang penelitian Tugas Akhir ini adalah sebagai berikut.

5.1.1 Halaman Utama

Halaman utama merupakan antarmuka yang berisi menu-menu untuk menampilkan antarmuka-antarmuka lainnya dalam sistem. Hasil implementasi halaman utama dapat dilihat pada Gambar 5.1



Gambar 5.1 Halaman Utama

Halaman utama dibuat dalam bentuk yang sederhana, terdiri dari 3 bagian utama, diantaranya :

1. Bagian *title bar*
2. Bagian *menu bar*
3. Bagian *main window*

Title bar merupakan bagian yang menunjukkan judul dari antarmuka yang sedang ditampilkan. Di bawah *title bar* terdapat *menu bar* yang terdiri 2 pilihan yaitu start dan exit. Adapun kegunaan menu-menu yang ditampilkan pada antarmuka utama sistem disajikan dalam Tabel 5.1.

Tabel 5.1 Kegunaan Menu Sistem

Menu	Kegunaan
Start	Masuk kedalam halaman menu utama program yang terdiri dari beberapa menu
Exit	Keluar dari program.

Main window merupakan bagian antarmuka yang digunakan untuk menampilkan berbagai antarmuka lain di dalam sistem.

5.1.2 Antarmuka Menu Utama

Antarmuka menu utama ini terdiri dari sederetan menu-menu yang digunakan oleh sistem. Menu-menu ini dirancang dengan kegunaan yang berbeda-beda. Adapun kegunaan menu-menu yang ditampilkan pada antarmuka menu utama disajikan dalam Tabel 5.2.

Tabel 5.2 Kegunaan Menu Utama

Menu	Kegunaan
Recognize	Melakukan pengujian, pengamatan, & penilaian pada sistem, seberapa baik hasil yang didapatkan pada data masukan tertentu.
About	Tentang pengenalan program dan metode <i>Histogram of Oriented Gradient</i> serta metode <i>K-Nearest Neighbor</i>
Help	Petunjuk dan tahap tahap penggunaan program

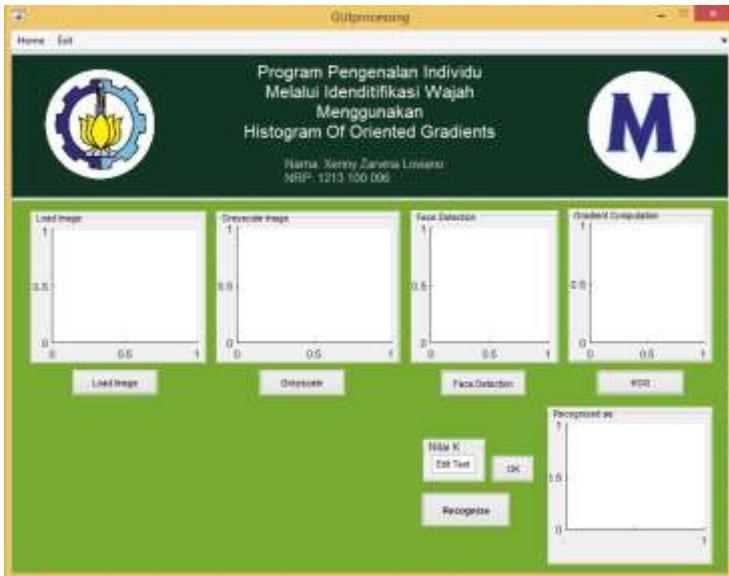
Hasil implementasi antar muka menu utama dapat dilihat pada Gambar 5.2.



Gambar 5.2 Antarmuka Menu Utama Program

5.1.3 Antarmuka Pengujian

Antarmuka Pengujian digunakan untuk melakukan pengujian terhadap sistem berdasarkan algoritma *K-Nearest Neighbor*. Hasil implementasi antarmuka Pengujian dapat dilihat pada Gambar 5.3.



Gambar 5.3 Antarmuka Pengujian Sistem Pengenalan Individu

Pada antarmuka Pengujian terdapat enam tombol. Fungsi dari tombol-tombol tersebut adalah sebagai berikut :

1. Tombol *load image* digunakan untuk mengambil citra wajah individu yang tersimpan pada computer dan akan ditampilkan pada *axes* yang telah dibuat pada halaman pengujian.
2. Tombol *grayscale* digunakan untuk mengubah citra RGB wajah individu menjadi citra keabuan dan akan ditampilkan pada *axes* yang telah dibuat pada halaman pengujian.
3. Tombol *face detection* digunakan untuk mendeteksi bagian wajah dari citra yang diinputkan dan akan ditampilkan pada *axes* yang telah dibuat pada halaman pengujian.

4. Tombol ekstraksi HOG digunakan untuk melakukan proses ekstraksi fitur dengan HOG terhadap citra yang akan diuji. Citra hasil pemrosesan tersebut akan ditampilkan pada *axes* yang telah dibuat pada halaman Pengujian.
5. Tombol OK digunakan untuk menentukan nilai k yang sudah diinputkan oleh user.
6. Tombol *Recognize* digunakan untuk melakukan proses pengujian citra, sehingga didapatkan keputusan pemilih dari citra yang diuji. Hasil pengujian tersebut akan ditampilkan pada *textfield* Hasil Pengujian. Gambar 5.4 di bawah ini menunjukkan hasil pengujian sistem pengenalan individu melalui identifikasi wajah.



Gambar 5.4 Hasil Running Program Pengujian Sistem Pengenalan Individu

5.2 Implementasi Tahap Input Data

Proses input data membutuhkan interaksi pengguna untuk mencari dan mengambil citra yang berukuran dibutuhkan di dalam media penyimpanan. Proses ini diimplementasikan menjadi sebuah program ke dalam fungsi yang telah tersedia di Matlab. Fungsi dijalankan adalah ketika tombol “*Load Image*” dipilih..

5.3 Implementasi Tahap *Preprocessing*

Pada tahap *preprocessing*, citra wajah dirubah dari citra RGB menjadi citra *grayscale* kemudian citra dideteksi bagian wajahnya menggunakan library *viola-jones*. Setelah dideteksi untuk mendapatkan bagian, bagian wajah tersebut dicorp dan kemudian *resize* menjadi ukuran 128×128 . Kode program implementasi tahap *preprocessing* ini diimplementasikan menjadi sebuah program berikut:

```
im2 = rgb2gray(im3);
%face detection
box = step(faceDetector, im2);
%red box
figure,imshow(im2);
    for i = 1:size(box,1)
        rectangle('Position',box(i,:), 'LineWidth'
            ',2, 'LineStyle', '-', 'EdgeColor', 'r');
    end
title('Face Detection');
%crop image
figure,
crop_result=imcrop(im2,box);
imshow (crop_result);
title('Face Detection');
im=imresize(crop_result,[128 128]);
im=double(im);
```

Kode Sumber 5.1 Kode program untuk pre-processing

5.4 Implementasi Tahap Ekstraksi Fitur

Proses selanjutnya adalah tahap ekstraksi fitur wajah individu dengan menggunakan metode *Histogram of Oriented Gradient*. Proses ini merupakan proses pembuatan fitur-fitur dari wajah individu sebagai database untuk sistem pengenalan individu melalui identifikasi wajah.

Pada proses ekstraksi fitur HOG ini, gradien horizontal dan vertikal citra dihitung untuk mencari *magnitude* dan orientasi gradiennya. Lalu citra dibagi menjadi *block* 16×16 piksel, dan pada saat yang sama, tiap *block* dibagi menjadi *cell* berukuran 8×8 piksel. Histogram dibentuk pada tiap piksel berdasarkan pada nilai *magnitude* yang berhubungan dengan orientasi bin. Bin berjarak merata dalam sudut $0^\circ - 180^\circ$, dan tiap *bin* berisi 9 garis sudut, dimana setiap garis sudut bernilai 20° . Histogram yang terbentuk pada tiap *block* kemudian digabungkan dan dinormalisasi.

```
function [feature] = ekstraksi(im)
```

Kode Sumber 5.2 Kode program untuk ekstraksi fitur HOG

Output dari fungsi ini adalah vektor fitur citra. Kode program selengkapnya dapat dilihat pada lampiran A1.

Fitur citra berbentuk matriks $1 \times D$ dengan D adalah dimensi ruang fitur. Vektor fitur kemudian di-*transpose* menjadi matriks $D \times 1$ untuk data input di proses klasifikasi.

5.5 Implementasi Tahap Klasifikasi

Setelah dilakukan proses pembuatan fitur, maka proses selanjutnya adalah proses klasifikasi. Klasifikasi dengan metode *K-Nearest Neighbor* ini dilakukan saat proses *testing*. Pada tahap klasifikasi ini, penulis menggunakan Matlab R2015a.

5.5.1 Proses *Training*

Proses *training* merupakan proses untuk mendapatkan database referensi dari citra wajah *training*. Pada tahap ini, 200 citra *training* yang ditaruh didalam satu folder dipanggil kemudian dilakukan proses ekstraksi fitur tiap citra *training*. Setelah didapat fitur dari 200 citra, kemudian vektor fitur dijadikan satu didalam satu file excel untuk dijadikan database referensi dalam sistem pengenalan individu.

Kode program implementasi tahap pemanggilan 200 citra didalam satu folder ini dapat dilihat pada Kode Sumber 5.2.

```
data = dir('D:\116\MATLAB TA\Data  
Wajah\*.jpg');
```

Kode Sumber 5.3 Kode program untuk memanggil seluruh data *training*.

Setelah melalui proses ekstraksi fitur, vektor fitur citra *training* disimpan didalam satu file excel dengan implementasi kode program sebagai berikut:

```
save('Database Wajah.xlsx', 'features');
```

Kode Sumber 5.4 Kode program untuk menyimpan vektor fitur citra *training*

Fitur-fitur yang disimpan didalam file tersebut dilabeli secara manual. Individu yang pertama di labeli 1 dan individu berikutnya dilabel $n + 1$.

5.5.2 Proses *Testing*

Proses *testing* bertujuan untuk menentukan pemilik dari citra wajah yang di inputkan. Pemilik citra ditentukan dari

label yang paling banyak muncul berdasarkan nilai k atau jumlah tetangga terdekat sesuai dengan yang ditentukan user.

Proses pengujian diimplementasikan menjadi sebuah program berikut :

```
%--memanggil database wajah training
filename = 'Database Wajah.xlsx';
label = xlsread(filename,1,'KY01:KY0270');
a = xlsread(filename,1,'A1:KYN270');
[baris,kolom]=size(a);
nilai_K = str2double(get(handles.edit5,
'String'))
    if isempty(nilai_K)
        msgbox('Nilai tidak valid');
    else
        msgbox('Nilai berhasil diinputkan');
    end
%--menghitung jarak matriks uji dengan matriks
latih
y=[];
for i=1:baris;
    for j=1:kolom;
        y(i,j)=(feature(j)-a(i,j))^2;
    end
end
...
```

Kode program selengkapny dapat dilihat pada Lampiran B.

BAB VI UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan mengenai proses pengujian yang dilakukan terhadap sistem pengenalan individu melalui identifikasi wajah dengan menggunakan metode *Histogram of Oriented Gradient* dan *K-Nearest Neighbor*. Hasil pengujian kemudian dibahas untuk mengetahui unjuk kerja sistem secara keseluruhan dalam menjalankan fungsi yang diinginkan. Selanjutnya dijelaskan mengenai hasil pengujian terhadap sistem yang telah diimplementasikan pada bab sebelumnya, yaitu pengujian tahap *preprocessing*, pengujian tahap ekstraksi fitur dengan *Histogram of Oriented Gradient (HOG)* dan pengujian tahap klasifikasi dengan metode *K-Nearest Neighbor* dengan beberapa variasi pengujian untuk mengetahui keakuratan sistem. Bab ini diakhiri dengan pembahasan hasil pengujian yang telah dilakukan.

6.1 Lingkungan Pengujian Sistem

Lingkungan pengujian dari sistem pengenalan individu meliputi perangkat keras dan lunak komputer. Detail dari perangkat keras dan lunak yang digunakan dapat dilihat pada Tabel 6.1.

Tabel 6.1 Lingkungan Pengujian Sistem

Perangkat Keras	Prosesor : Intel® Pentium® Quad Core @ 2.10GHz 2.10GHz
	Memory : 2.00 GB
Perangkat Lunak	Sistem Operasi : Windows 8 Ultimate 64-bit
	Tools : MATLAB R2015a

6.2 Pengujian Tahap Preprocessing

Pengujian ini dilakukan terhadap proses-proses pada tahap pengolahan citra. Pengujian bertujuan untuk mengetahui bahwa proses-proses pada tahap pengolahan citra sudah benar. Pada tahap ini terdapat beberapa proses, yaitu:

1. Pengujian proses *grayscale*
Pengujian pada proses ini bertujuan untuk memudahkan penghitungan gradien citra. Gambar 6.1 adalah citra hasil *grayscale*.



(a)

(b)

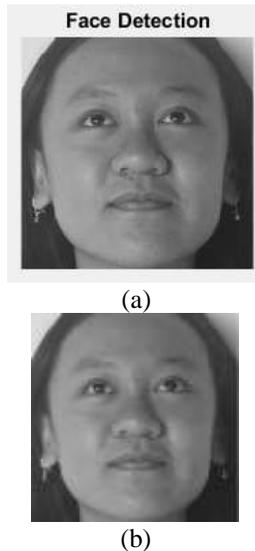
Gambar 6.1 Pengujian proses *grayscale* untuk citra wajah
(a) Citra RGB (b) Citra *Grayscale*

2. Pengujian proses deteksi wajah menggunakan *Viola-Jones*
Pengujian pada proses ini mengklasifikasikan citra dari nilai fitur-fitur sederhana. Gambar 6.2 adalah hasil deteksi wajah menggunakan *Viola-Jones*.



Gambar 6.2 Pengujian proses deteksi wajah menggunakan *Viola-Jones*

3. Pengujian proses *cropping* dan *resizing*.
Pengujian proses ini bertujuan untuk memperkecil dimensi citra menjadi 128×128 . Gambar 6.3 adalah citra hasil yang sudah melalui proses *cropping* dan *resizing*.



Gambar 6.3 Pengujian proses terhadap citra wajah manusia
(a) *cropping* (b) *resizing*

6.3 Pengujian Tahap Ekstraksi Fitur

Pengujian proses ini menggunakan data citra yang telah diproses pada pra-pemrosesan. Tahap ekstraksi fitur ini menggunakan metode *Histogram of Oriented Gradient*. Pada tahap ini terdapat 3 proses yaitu:

1. Pengujian proses komputasi gradien
Pengujian proses ini bertujuan untuk mendapatkan informasi *contour* citra dan mengurangi perubahan

cahaya. Pengujian proses komputasi gradien ditunjukkan pada Gambar 6.4.



(a)



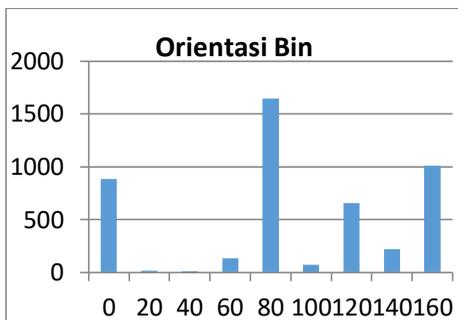
(b)



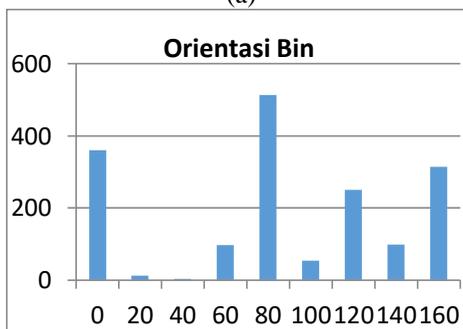
(c)

Gambar 6.4 Pengujian proses komputasi gradien untuk citra wajah
(a) Adit (b) Amel (c) Anas

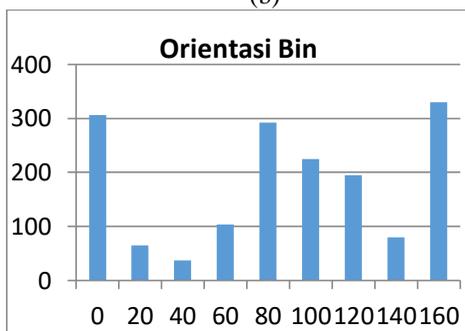
2. Pengujian proses orientasi *bin*.
Pengujian proses ini bertujuan untuk mendapatkan orientasi *bin* berdasarkan komputasi gradien tiap piksel citra. Pengujian proses orientasi *bin* dapat ditunjukkan pada Gambar 6.5.



(a)



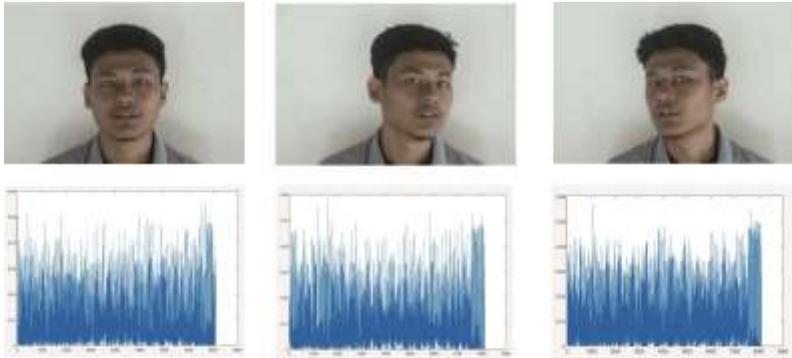
(b)



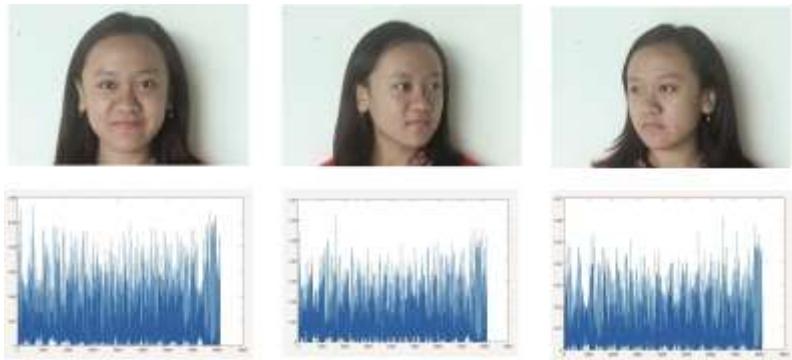
(c)

Gambar 6.5 Pengujian proses orientasi bin untuk citra wajah
 (a) Adit (b) Anas (c) Amel

3. Pengujian proses visualisasi fitur HOG.
Pengujian proses ini bertujuan untuk menampilkan fitur HOG secara keseluruhan.



(a)



(b)

Gambar 6.6 Pengujian proses ekstraksi 2 individu berbeda
(a) Adit (b) Anas

Gambar 6.6 (a) dan (b) menunjukkan hasil ekstraksi dari 2 individu yang berbeda. Hasil dari ekstraksi kedua

individu itu memiliki hasil yang berbeda. Tetapi untuk individu yang sama, memiliki kemiripan hasil ekstraksi, meskipun dari hasil tetap terlihat perbedaan.

6.4 Pengujian Tahap Klasifikasi dengan K-Nearest

6.4.1 Proses Training

Data pada sistem pengenalan individu melalui identifikasi wajah ini terdiri dari 220 citra. Data dibagi menjadi data pelatihan dan data uji. Data pelatihan terdiri dari 180 citra wajah dari 45 individu yang berbeda. Sedangkan data uji merupakan sisa dari jumlah keseluruhan data yang tidak digunakan untuk data *training*. Terdapat 45 data uji yang merupakan citra wajah individu dari 45 individu yang berbeda.

Pengujian proses ini menggunakan fitur yang telah diperoleh dari proses ekstraksi fitur dengan *Histogram of Oriented Gradient* dan bertujuan untuk mengetahui data yang diperlukan dalam proses berikutnya.

Data keluaran yang dihasilkan akan disimpan dalam file matlab dengan format .xlsx. Data-data yang disimpan akan berbentuk vektor fitur dari data training yang kemudian dijadikan satu dalam satu file excel sebagai database referensi. Setelah dijadikan satu, vektor fitur setiap individu di labelkan secara manual untuk menentukan pemilik dari vektor fitur tersebut. Data label akan berbentuk informasi seperti yang terlihat pada Tabel.6.2.

Tabel 6.2 Label pada vektor fitur hasil ekstraksi HOG

Nama	Label
Adit	1
Afif	2
Agus	3
Alfin	4
Amel	5
Anas	6

Anin	7
Anshar	8
Ardi F	9
Ardi	10
Athya D.	11
Aufa	12
Ayu R	13
Bhara	14
Brian F	15
Corry	16
Dasilva	17
Dian	18
Dita	19
Fadhlan	20
Faizin	21
Fella	22
Feri	23
Gery	24
Hayu	25
Indra	26
Intan	27
Isabela	27
Ivan	29
Khoirul	30
Lingga	31
Nadya	32
Nirwana	33
Nuke	34
Prima	35
Rohim	36
Romli	37
Rozi	38
Satria	39
Syifaul	40

Uyun	41
Wawan	42
Wikjung	43
Zaki	44
Zani	45

6.4.2 Proses Testing

Pengujian sistem menggunakan 45 citra wajah manusia yang terdiri dari 45 individu yang berbeda. Data yang digunakan adalah 45 data training dan 40 data testing. Pengujian proses ini untuk menentukan siapa pemilik dari wajah ter. Hasil dari pengujian sistem dapat dilihat pada Tabel 6.3 sampai dengan Tabel 6.10.

Tabel 6.3 Hasil Proses *Testing* dengan nilai $K = 1$

Data Input Hasil Deteksi	Training	Testing
Benar	41	35
Salah	4	5
Total	45	40
Akurasi	92.5%	87.5%

Tabel 6.4 Hasil Proses *Testing* dengan nilai $K = 2$

Data Input Hasil Deteksi	Training	Testing
Benar	41	34
Salah	4	6
Total	45	40
Akurasi	92.5%	85%

Tabel 6.5 Hasil Proses *Testing* dengan nilai $K = 3$

Data Input Hasil Deteksi	Training	Testing
Benar	43	32
Salah	2	8
Total	45	40
Akurasi	95%	80%

Tabel 6.6 Hasil Proses *Testing* dengan nilai $K = 4$

Data Input Hasil Deteksi	Training	Testing
Benar	43	31
Salah	2	9
Total	45	40
Akurasi	95%	77.5%

Tabel 6.7 Hasil Proses *Testing* dengan nilai $K = 5$

Data Input Hasil Deteksi	Training	Testing
Benar	43	31
Salah	2	9
Total	45	40
Akurasi	95%	77.5%

Tabel 6.8 Hasil Proses *Testing* dengan nilai $K = 6$

Data Input Hasil Deteksi	Training	Testing
Benar	41	31
Salah	4	9
Total	45	40
Akurasi	92.5%	77.5%

Tabel 6.9 Hasil Proses *Testing* dengan nilai $K = 7$

Data Input Hasil Deteksi	Training	Testing
Benar	41	31
Salah	4	9
Total	45	40
Akurasi	92.5%	77.5%

Tabel 6.10 Hasil Proses *Testing* dengan nilai $K = 8$

Data Input Hasil Deteksi	Training	Testing
Benar	38	31
Salah	7	9
Total	45	40
Akurasi	82.5%	77.5%

Pada penelitian ini dilakukan pengujian menggunakan nilai $K = [1, 2, 3, 4, 5, 6, 7, 8]$. Pemilihan K adalah pemilihan secara bebas oleh user. Pada penelitian ini, karena setiap

individu memiliki 4 posisi pada database referensi oleh karena itu penulis menggunakan K seperti terlihat pada Tabel 6.3 dan selanjutnya sampai Tabel 6.10. Berdasarkan 8 Tabel pengujian di atas, dapat dilihat bahwa pada $K = [3,4,5]$ memiliki tingkat akurasi yang paling tinggi yaitu 95% pada pengujian sistem menggunakan data training. Sedangkan pada pengujian sistem menggunakan data testing, berdasarkan tabel pengujian di atas dapat dilihat bahwa pada $K = 1$ memiliki tingkat akurasi sebesar 87,5 %.

6.5 Pembahasan Hasil Pengujian

Pembahasan hasil pengujian difokuskan pada hasil pengujian dan proses pengenalan individu melalui identifikasi wajah. Pada penelitian ini pembahasan hasil pengujian ini digunakan untuk mengetahui kinerja system pengenalan individu melalui identifikasi wajah menggunakan *Histogram of Oriented Gradient*. Dari hasil yang didapatkan pada subbab sebelumnya, sistem mampu mengenali dengan cukup baik. Pada subbab ini akan diberikan cara menghitung tingkat akurasi sistem dengan menggunakan Persamaan 6.1.

$$Akurasi = \frac{Hasil\ Benar}{Jumlah\ citra\ uji} \times 100\% \quad (6.1)$$

dengan Hasil Benar adalah jumlah citra wajah yang dikenali oleh sistem.

Perhitungan akurasi dilakukan pada data *training* dan data *testing*. Perhitungan ini dilakukan untuk setiap $K = [1,2,3,4,5,6,7,8]$ sehingga terdapat 8 nilai akurasi pada data training dan 8 nilai akurasi pada data testing. Data *training* yang diuji terdiri dari 45 citra wajah manusia, ditunjukkan pada Tabel 6.11. Sedangkan data *testing* yang diuji terdiri dari 40 citra wajah manusia, ditunjukkan pada Tabel 6.12. Berdasarkan hasil pengujian pada Tabel 6.3 sampai dengan Tabel 6.10 di dapatkan akurasi sistem sebagai berikut,

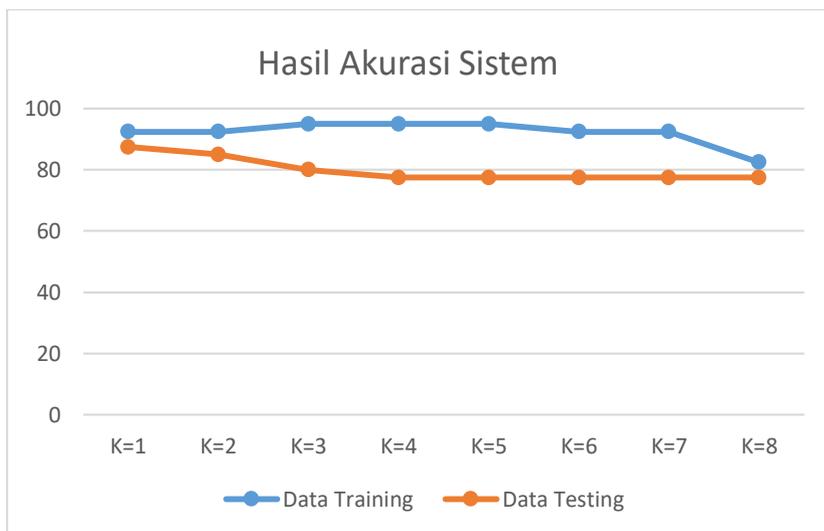
Tabel 6.11 Hasil perhitungan akurasi sistem menggunakan data *training*

Nilai K	1	2	3	4	5	6	7	8
Akurasi (%)	92,5	92,5	95	95	95	92,5	92,5	82,5

Tabel 6.12 Hasil perhitungan akurasi sistem menggunakan data *testing*

Nilai K	1	2	3	4	5	6	7	8
Akurasi (%)	87,5	85	80	77,5	77,5	77,5	77,5	77,5

Dilihat dari hasil akurasi sistem bahwa untuk nilai akurasi pada sistem nilai K sangat berpengaruh terhadap hasil pada sistem. Nilai K berpengaruh karena nilai K menentukan jumlah tetangga terdekat atau modus dari nilai label yang muncul saat pengujian. Hasil akurasi sitem secara keseluruhan dapat dilihat pada Gambar 6.7.



Gambar 6.7 Grafik hasil nilai akurasi sistem

Setelah didapat semua nilai akurasi pada sistem pengenalan individu ini, dilakukan juga perhitungan rata-rata nilai akurasi sistem pengenalan individu, seperti yang ditunjukkan pada Tabel 6.13. Perhitungan ini menentukan nilai akurasi secara keseluruhan. Perhitungan rata-rata akurasi menggunakan Persamaan 6.2.

$$\text{Rata - rata} = \frac{\text{Jumlah seluruh nilai akurasi}}{\text{Banyak nilai akurasi}} \quad (6.2)$$

Berdasarkan Tabel 6.10 maka didapat perhitungan rata-rata nilai akurasi sistem menggunakan data *training* sebagai berikut:

$$\begin{aligned} \text{Rata - rata} &= \frac{92.5 + 92.5 + 95 + 95 + 95 + 92.5 + 92.5 + 82.5}{8} \\ &= 92.1875\% \end{aligned}$$

sedangkan berdasarkan Tabel 6.11 maka didapat perhitungan rata-rata nilai akurasi sistem menggunakan data *testing* sebagai berikut:

$$\begin{aligned} \text{Rata - rata} &= \frac{87.5 + 85 + 80 + 77.5 + 77.5 + 77.5 + 77.5 + 77.5}{8} \\ &= 80\% \end{aligned}$$

Tabel 6.13 Hasil Perhitungan Rata-rata Akurasi

Citra Input	Rata-rata Akurasi (%)
Data <i>Training</i>	92,1875
Data <i>Testing</i>	80

6.6 Pembahasan Penyebab Besar Kecil Akurasi

Penyebab utama rendahnya akurasi sistem pengenalan individu melalui identifikasi wajah menggunakan

Histogram of Oriented Gradient adalah kurang banyaknya data *training* citra wajah yang dilatih, sehingga menyebabkan berkurangnya tingkat akurasi dari sistem ini.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan eksperimen dan pembahasan terhadap hasil pengujian yang telah dilakukan terhadap sistem pengenalan individu melalui identifikasi wajah menggunakan *Histogram of Oriented Gradient* maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil membuat sebuah sistem pengenalan individu melalui identifikasi wajah dengan tahapam prosesnya adalah *pre-processing*, ekstraksi fitur menggunakan metode HOG dan proses klasifikasi menggunakan metode *K-Nearest Neighbor*.
2. Sistem pengenalan individu ini dapat digunakan dengan tingkat akurasi rata-rata sebesar 80%. Pengujian dilakukan dengan 40 citra wajah manusia. Hasil pengujian sistem terbaik menggunakan $K=1$ dengan tingkat akurasi 87.5 %

7.2 Saran

Dengan melihat hasil yang dicapai pada penelitian ini, penulis menyarankan untuk menambah keberagaman data *training* untuk citra wajah individu. Pada pengembangan selanjutnya dapat mengaplikasikan metode HOG pada obyek biometrik yang lain.

DAFTAR PUSTAKA

- [1] N. Dalal, B. Triggs. (2005). “Histograms of oriented gradient for human detections”. Proc. IEEE Conf.on Computer Vision & Pattern Recognition. 886-893.
- [2] Frans Irawan, dkk. “Deteksi Mobil pada Citra Digital Menggunakan C-HOG dan *Support Vector Machine*”
- [3] Yeunghak Lee, dkk. “Spatial Regions Periodicity based Detection of Two-wheelers using Histogram of Oriented Gradient”. International Journal of Multimedia and Ubiquitous Engineering. 10(4): 325-336, 2015.
- [4] Cahyo Permata, I Ketut Eddy Purnama, Muhtadin. “Deteksi Mobil Menggunakan Histogram of Oriented Gradient”.
- [5] Dian Esti Pratiwi, Agus Harjoko (2013). “Implementasi Pengenalan Wajah Menggunakan PCA (*Principal Component Analysis*)”
- [6] Muhammad Ilyas Sikki (2009). “Pengenalan Wajah Menggunakan *K-Nearest Neighbor* dengan Praproses Transformasi Wavelet
- [7] Tuffahatul Ummah (2016)` “Deteksi Kendaraan Menggunakan *Histogram of Oriented Gradient* dan *Real Adaboost*”.
- [8] Chen, P.Y., dkk. (2014). “An Efficient Hardware Implementation of HOG Feature Extraction for Human

- Detection”. IEEE Transactions On Intelligent Transportation Systems.15(2), 656-662.
- [9] Gonzalez, R.C dan Rafael E.W. *Digital Image Processing*. Prentice-Hall. Inc. United State, America. 2008.
- [10] M. Nixon dan A. Aguado, “Feature Extraction and Image Processing”. Elsevier Ltd., 2008.
- [11] Ni Wayan Marti, Pemanfaatan GUI Dalam Pengembangan Perangkat Lunak Pengenalan Citra Wajah Manusia menggunakan Metode Eigenface, 2010.
- [12] SHU Chang, DING Xiaoqing dan FANG Chi, Histogram of Oriented Gradient for Face Recognition, 2011
- [13] Rosmawati Dwi, Koerdianto, Achmad Rizal, ”Desain dan Simulasi Sistem Identifikasi Manusia dengan Analisis Ciri Fisis Citra Palmprint Berbasis Image Processing dan K-Nearest Neighbor”, Bandung, 2010.
- [14] Dedi Ary Prasetya, Imam Nurviyanto, “Deteksi Wajah Metode *Viola Jones* Pada Opencv Menggunakan Pemrograman Python’, Surakarta, 2012
- [15] Hanif Al Fatta, “Konversi Format Citra RGB ke Format Grayscale Menggunakan Visual Basic’, Yogyakarta 2007
- [16] Shinta Nur Desmia Sari, Abdul Fadlil, “Sistem Identifikasi Citra Jahe (*Zingiber Officinale*) Menggunakan Metode Jarak”, Yogyakarta 2014

LAMPIRAN A
Citra Wajah

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Aditya Barul-hadi				
Afif				
Agus				
Alfin Asmara				
Amel				
Anas				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Anin				
Anshar				
Ardi				
Ardi F				
Athya Danni				
Aufa				
Ayu Riska				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Brian Fadi				
Corry				
Dasilva				
Dian Eka				
Rizka Yoni				
Fadhlan				
Faizin				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Feri Winata				
Gery D.C				
Hayu				
Indra				
Intan				
Isabela				
Ivan				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Khoiril				
Lingga				
Nadya				
Nirwana				
Nuke				
Prima				
Rohim				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Romli				
Rozi				
Satria				
Syifaul				
Uyun				
Wawan				
Wikjuning				

Nama	Posisi			
	Depan	Kanan	Kiri	Atas
Zaki				
Zani				

LAMPIRAN B

Source Code

```
[namafilename, formatfile] = uigetfile({'*.jpg'},  
'load image'); %memilih gambar  
image = imread([formatfile, namafilename]); %membaca  
gambar  
  
axes(handles.axes1); %memilih axes1 sebagai  
letak gambar yang dimunculkan  
imshow(image); %memunculkan gambar  
handles.image=image;  
guidata(hObject, handles);  
  
image1 = handles.image;  
gray = rgb2gray(image1);  
axes(handles.axes2);  
imshow(gray);  
  
handles.image2 = gray;  
guidata(hObject, handles);  
  
% --- PushButton (Face Detection)  
function pushbutton3_Callback(hObject,  
 eventdata, handles)  
faceDetector = vision.CascadeObjectDetector;  
image2 = handles.image2;  
box = step(faceDetector, image2);  
%hold on  
for i = 1:size(box,1)  
  
rectangle('Position', box(i,:), 'LineWidth', 1, 'Lin  
eStyle', '-', 'EdgeColor', 'w');  
end  
  
hasil_crop=imcrop(image2, box);
```

```

axes(handles.axes3);
imshow(hasil_crop);

handles.image3 = hasil_crop;
guidata(hObject,handles);

%--Histogram of Oriented Gradient
function pushbutton6_Callback(hObject,
 eventdata, handles)
global feature
im11 = handles.image3;
im=imresize(im11,[128 128]);
im=double(im);

rows=size(im,1);
cols=size(im,2);
Ix=im;
Iy=im;

% Gradient in X and Y direction. Iy is the
gradient in X direction and Iy
% is the gradient in Y direction
for i=1:rows-2
    Iy(i,:)=(im(i,:)-im(i+2,:));
end
for i=1:cols-2
    Ix(:,i)=(im(:,i)-im(:,i+2));
end

gauss=fspecial('gaussian',8); %% Initialized a
gaussian filter with sigma=0.5 * block width.

angle=atand(Ix./Iy); % Matrix containing the
angles of each edge gradient
angle=imadd(angle,90); %Angles in range (0,180)

```

```

magnitude=sqrt(Ix.^2 + Iy.^2);
axes(handles.axes4);
imshow(uint8(magnitude));
% figure,imshow(uint8(angle));
% figure,imshow(uint8(magnitude));

% Remove redundant pixels in an image.
angle(isnan(angle))=0;
magnitude(isnan(magnitude))=0;

feature=[]; %initialized the feature vector

% Iterations for Blocks
for i = 0: rows/8 - 2
    for j= 0: cols/8 -2
        %disp([i,j])

        mag_patch = magnitude(8*i+1 :
8*i+16 , 8*j+1 : 8*j+16);
        %mag_patch =
imfilter(mag_patch,gauss);
        ang_patch = angle(8*i+1 : 8*i+16 ,
8*j+1 : 8*j+16);

        block_feature=[];

        %Iterations for cells in a block
        for x= 0:1
            for y= 0:1
                angleA
=ang_patch(8*x+1:8*x+8, 8*y+1:8*y+8);
                magA
=mag_patch(8*x+1:8*x+8, 8*y+1:8*y+8);
                histr =zeros(1,9);

```

```

%Iterations for pixels in one cell
    for p=1:8
        for q=1:8

            alpha= angleA(p,q);

            % Binning Process
            (Bi-Linear Interpolation)
            if alpha>10 &&
alpha<=30
histr(1)=histr(1)+ magA(p,q) * (30-alpha)/20;
histr(2)=histr(2)+ magA(p,q) * (alpha-10)/20;
            elseif alpha>30 &&
alpha<=50
histr(2)=histr(2)+ magA(p,q) * (50-alpha)/20;
histr(3)=histr(3)+ magA(p,q) * (alpha-30)/20;
            elseif alpha>50 &&
alpha<=70
histr(3)=histr(3)+ magA(p,q) * (70-alpha)/20;
histr(4)=histr(4)+ magA(p,q) * (alpha-50)/20;
            elseif alpha>70 &&
alpha<=90
histr(4)=histr(4)+ magA(p,q) * (90-alpha)/20;
histr(5)=histr(5)+ magA(p,q) * (alpha-70)/20;
            elseif alpha>90 &&
alpha<=110
histr(5)=histr(5)+ magA(p,q) * (110-
alpha)/20;

```

```

histr(6)=histr(6)+ magA(p,q) * (alpha-90)/20;
                                elseif alpha>110 &&
alpha<=130

histr(6)=histr(6)+ magA(p,q) * (130-
alpha)/20;

histr(7)=histr(7)+ magA(p,q) * (alpha-
110)/20;
                                elseif alpha>130 &&
alpha<=150

histr(7)=histr(7)+ magA(p,q) * (150-
alpha)/20;

histr(8)=histr(8)+ magA(p,q) * (alpha-
130)/20;
                                elseif alpha>150 &&
alpha<=170

histr(8)=histr(8)+ magA(p,q) * (170-
alpha)/20;

histr(9)=histr(9)+ magA(p,q) * (alpha-
150)/20;
                                elseif alpha>=0 &&
alpha<=10

histr(1)=histr(1)+ magA(p,q) * (alpha+10)/20;

histr(9)=histr(9)+ magA(p,q) * (10-alpha)/20;
                                elseif alpha>170 &&
alpha<=180

histr(9)=histr(9)+ magA(p,q) * (190-
alpha)/20;

```

```

histr(1)=histr(1)+ magA(p,q)*(alpha-
170)/20;
                                end

                                end
                                end
                                % Concatenation of Four
                                histograms to form one block feature
block_feature=[block_feature histr];

                                end
                                end
                                % Normalize the values in the block
                                using L1-Norm

block_feature=block_feature/sqrt(norm(block
_feature)^2+.01);
                                %Features concatenation
                                feature=[feature block_feature];
                                end
                                end
                                %Removing Infinitiy values
                                feature(isnan(feature))=0;

                                % Normalization of the feature vector using
                                L2-Norm
                                feature=feature/sqrt(norm(feature)^2+.001);

                                for z=1:length(feature)
                                    if feature(z)>0.2
                                        feature(z)=0.2;
                                    end
                                end
                                end
                                msgbox('Ekstraksi Sukses');

```

```

% --- KNN button.
function pushbutton9_Callback(hObject,
eventdata, handles)
% hObject    handle to pushbutton9 (see
GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)
global nilai_K
nilai_K = str2double(get(handles.edit5,
'String'))

if isempty(nilai_K)
    msgbox('Nilai tidak valid');
else
    msgbox('Nilai berhasil diinputkan');
end

% --- Recognize face
function pushbutton7_Callback(hObject,
eventdata, handles)
% hObject    handle to pushbutton7 (see
GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)
global feature;
global nilai_K;

%--memanggil database wajah training
filename = 'Database Wajah.xlsx';
label = xlsread(filename,1,'KY01:KY0270');
a = xlsread(filename,1,'A1:KYN270');
[baris,kolom]=size(a);

```

```

%--menghitung jarak matriks uji dengan
matriks latih
y=[];
for i=1:baris;
    for j=1:kolom;
        y(i,j)=(feature(j)-a(i,j))^2;
    end
end

%--menjumlahkan semua elemen satu baris
matrik
z=[];
for i=1:baris;
    z(i)=sum(y(i,:));
end

%--mengakarkan setiap elemen matriks hasil
z=sqrt(z);
z=reshape(z,[],1);
tampung = [z label];

%mengurutkan elemen pada matriks hasil
m = sortrows(tampung);
mat = m(:,2);
nn = mat(1:nilai_K);
result = mode(nn);

%--validasi hasil
if (result == 1)
    b='Aditya Barulhadi'
    set(handles.text12,'String',b);
elseif (result == 2)
    c='Afif'
    set(handles.text12,'String',c);
elseif (result == 3)
    d='Agus Setiawan'

```

```

set(handles.text12, 'String',d);
elseif (result == 4)
    e='Alfin Asmara'
    set(handles.text12, 'String',e);
elseif (result == 5)
    f='Amalia Margaretha'
    set(handles.text12, 'String',f);
elseif (result == 6)
    g='Anas'
    set(handles.text12, 'String',g);
elseif (result == 7)
    h='Anin'
    set(handles.text12, 'String',h);
elseif (result == 8)
    k='Anshar'
    set(handles.text12, 'String',k);
elseif (result == 9)
    l='Ardi'
    set(handles.text12, 'String',l);
elseif (result == 10)
    bm='Ardi Firmansyah'
    set(handles.text12, 'String',bm);
elseif (result == 11)
    s='Athya Danni'
    set(handles.text12, 'String',s);
elseif (result == 12)
    n='Aufa Biadillah'
    set(handles.text12, 'String',n);
elseif (result == 13)
    o='Ayu Riska'
    set(handles.text12, 'String',o);
elseif (result == 14)
    p='Maulana Bhara'
    set(handles.text12, 'String',p);
elseif (result == 15)
    q='Bryan Fadi'
    set(handles.text12, 'String',q);

```

```
elseif (result == 16)
    r='Corry Ayu'
    set(handles.text12, 'String', r);
elseif (result == 17)
    t='Dasilva Ayu'
    set(handles.text12, 'String', t);
elseif (result == 18)
    u='Dian Eka Wardani'
    set(handles.text12, 'String', u);
elseif (result == 19)
    v='Rizka Yoni'
    set(handles.text12, 'String', v);
elseif (result == 20)
    w='Fadhlan Septiawan'
    set(handles.text12, 'String', w);
elseif (result == 21)
    x='Faizin Anshori'
    set(handles.text12, 'String', x);
elseif (result == 22)
    by='Fella Diandra'
    set(handles.text12, 'String', by);
elseif (result == 23)
    bz='Feri Winata'
    set(handles.text12, 'String', bz);
elseif (result == 24)
    aa='Gery Dias Claudio'
    set(handles.text12, 'String', aa);
elseif (result == 25)
    aaa='Hayu'
    set(handles.text12, 'String', aaa);
elseif (result == 26)
    ab='Indra Darussalam'
    set(handles.text12, 'String', ab);
elseif (result == 27)
    ac='Roselina Dwi Intan'
    set(handles.text12, 'String', ac);
elseif (result == 28)
    ad='Noni Isabella'
```

```
set(handles.text12, 'String', ad);
elseif (result == 29)
    ae='Ivan Octaviano'
    set(handles.text12, 'String', ae);
elseif (result == 30)
    af='Khoirul'
    set(handles.text12, 'String', af);
elseif (result == 31)
    ag='Lingga'
    set(handles.text12, 'String', ag);
elseif (result == 32)
    ah='Nadya'
    set(handles.text12, 'String', ah);
elseif (result == 33)
    ai='Nirwana'
    set(handles.text12, 'String', ai);
elseif (result == 34)
    aj='Nuke'
    set(handles.text12, 'String', aj);
elseif (result == 35)
    ak='Prima Aditya'
    set(handles.text12, 'String', ak);
elseif (result == 36)
    akk='Abdur Rohim'
    set(handles.text12, 'String', akk);
elseif (result == 37)
    al='Romli'
    set(handles.text12, 'String', al);
elseif (result == 38)
    am='Fakhrur Rozi'
    set(handles.text12, 'String', am);
elseif (result == 39)
    an='M. Satria D. U.'
    set(handles.text12, 'String', an);
elseif (result == 40)
    ao='M. Syifaul Rizky'
    set(handles.text12, 'String', ao);
```

```

elseif (result == 41)
    ap='Uyun Pane'
    set(handles.text12, 'String', ap);
elseif (result == 42)
    aq='Hartanto Setiawan'
    set(handles.text12, 'String', aq);
elseif (result == 43)
    bak='Wikjung'
    set(handles.text12, 'String', bak);
elseif (result == 44)
    cak='Zaki Mandala'
    set(handles.text12, 'String', cak);
elseif (result == 45)
    dak='M. Sabila Zamani'
    set(handles.text12, 'String', dak);
else
    zz='Wajah tidak dikenali'
    set(handles.text12, 'String', zz);
end

hasil = handles.image;
axes(handles.axes8);
imshow(hasil);

```

LAMPIRAN C

Matriks Hasil

$\rho(\lambda, j) =$

127.5284	171.6692	133.9949	101.7003	67.2663	70.9965	63.6337	53.7131	43.1614	46.9749	57.2648	33.2531	9.4623	93.8141	4.8991	106.7595
25.4077	161.5651	129.6442	146.2099	165.9638	71.6075	7.7852	22.2801	21.1986	20.5560	17.1027	166.3803	173.2966	27.2753	10.3738	171.8699
6.9143	147.3808	128.9910	55.0080	90.0000	119.4739	139.2687	2.6026	2.7263	177.1376	159.4440	153.4349	169.4608	8.2971	13.3345	170.3648
171.5280	149.4808	57.9946	0	71.5651	105.0454	114.5672	165.0686	145.3048	152.1739	5.1944	56.3099	1.882.000	3.8141	8.2974	170.5377
160.4160	140.5275	2.6026	153.000	90.000	78.2749	93.6941	165.7908	145.8403	144.2461	173.6198	97.5104	146.5345	180.000	2.4366	178.6302
178.6680	145.5308	172.0476	4.2892	53.7462	145.8835	95.7106	166.8260	114.7751	152.4472	10.2002	112.4712	4.92051	1.5482	177.0402	11.4438
24.2074	31.6901	178.7546	24.2625	71.9657	132.1376	118.8866	161.0754	137.7263	142.5738	367.582	112.1355	85.3665	6.6135	174.7529	177.7543
4.8193	156.0375	180.600	24.4440	58.2925	161.5651	2.6026	21.8014	127.8750	36.0278	130.2645	80.7890	34.0772	4.3739	178.1123	11.235
4.0834	72.6695	97.2946	15.9434	177.9346	146.2099	38.0470	13.4480	140.1944	165.9638	109.1790	8.4270	11.5340	1.4502	166.1153	178.8308
17.4639	82.1645	64.8931	176.2086	150.7086	153.4349	63.4340	164.3205	163.7598	173.8845	17.1027	142.9343	174.2804	179.0738	156.3369	172.3332
11.3099	27.1081	78.8708	174.2804	161.5651	0	139.7636	165.9638	145.0080	10.5392	32.6192	157.5306	175.7636	151.7826	157.1194	176.4217
2.0234	0.46696	76.7236	114.0735	36.8699	36.8699	20.2661	17.6472	11.2999	149.0362	165.9638	167.0054	158.2765	156.2018	172.0364	5.0796
178.7811	179.5379	35.5377	163.5345	145.8403	71.5651	57.2648	26.6033	49.7636	144.5336	158.1986	150.1541	131.7108	14.3814	7.5946	
26.5651	3.0128	1.4815	66.8937	96.1155	126.8699	177.9546	161.892	33.1113	153.4349	147.2648	145.4375	153.2002	71.5651	7.4635	3.5015
93.6785	81.3288	28.5797	32.9279	83.0559	91.3245	96.5198	101.0489	101.3722	100.0080	118.6105	122.7952	96.2238	74.6237	135.000	135.000
90.000	84.8978	32.0619	30.6354	80.7890	86.3107	93.9969	102.3194	106.3702	113.8059	123.2195	112.1903	75.6196	81.1582	135.000	135.000

LAMPIRAN C (Lanjutan)

$w(x, y) =$

87.0057	60.6712	80.0226	24.5153	20.6155	27.5156	29.1548	26.6271	24.7588	20.5183	16.6433	34.2053	42.4793	30.0066	70.2567	87.3312
44.2832	85.3815	45.4533	1.6056	4.1231	15.2643	22.2016	18.9848	22.8473	25.6120	13.6015	16.2788	46.2709	57.1214	72.1803	70.7107
74.5453	89.0440	27.0185	12.2066	18.0000	26.4197	17.6588	12.0227	21.0238	20.0250	17.0880	13.4164	43.7379	48.5077	78.1025	65.9242
95.0368	65.0000	9.4360	9.0000	18.9737	50.9608	38.4838	15.5342	31.6228	40.7963	22.0907	7.2111	28.0000	60.1332	85.0000	60.8276
110.3817	22.0227	22.0227	8.4853	23.0000	54.1295	37.0540	15.0000	33.8378	30.8958	18.1108	23.0217	10.7231	66.0000	94.0856	54.0853
88.0227	28.6007	31.2570	40.1123	18.6011	37.4433	10.0499	30.8038	14.3178	25.9422	25.000	26.0768	20.5183	74.0270	97.1288	50.0100
74.7329	3.4056	46.0109	58.1808	45.2217	84.8588	33.1210	37.0000	14.8661	37.8021	32.2025	63.6946	37.1214	68.4627	98.4124	51.0392
82.2982	9.8489	12.0000	24.1661	30.5287	3.1623	22.0227	37.6992	22.8035	13.6015	26.2488	37.4833	41.0488	75.2396	92.0489	51.0098
98.2497	63.8905	15.1572	14.5602	28.0179	21.8385	11.8003	59.2010	25.000	56.3205	27.2020	48.6542	60.2993	81.1158	97.1700	52.4690
93.3006	110.0273	33.1059	31.0644	47.0106	17.8885	11.8003	59.2010	25.000	56.3205	27.2020	48.6542	60.2993	81.1158	97.1700	52.4690
91.7824	94.3663	62.1691	50.1496	9.4888	5.0000	34.0588	45.3542	24.4131	45.7379	29.6816	31.9847	81.2219	93.0591	72.6154	48.0937
87.0919	122.0041	91.4440	51.4782	10.0000	24.5967	56.6879	25.4951	20.1548	28.8617	26.6833	99.2480	101.1385	56.4876	43.1774	
47.0106	124.0040	86.0233	117.1537	33.8378	18.9737	33.2866	45.6180	34.0588	48.0194	37.6862	70.3278	134.8258	61.6108	40.2616	45.5982
2.2681	70.1052	116.0388	81.5414	28.1663	5.0000	28.0179	31.2880	27.4391	4.4721	16.6433	09.2880	94.2945	3.1623	45.4166	49.0938
59.1154	59.0815	115.0043	125.0950	157.1528	173.0462	176.1391	172.1818	158.6852	138.1014	137.8206	133.1465	50.3161	41.6649	69.2965	72.1249
57.000	56.2228	107.3778	119.7080	149.9333	164.3040	172.4181	178.1011	166.7603	148.6472	135.0741	82.0792	40.4016	45.5412	69.2965	73.5991

BIODATA PENULIS



Penulis bernama lengkap **Xenny Zarvina Loviano**, yang biasa dipanggil Xenny. Penulis adalah anak ketiga dari pasangan Marzalio dan Indriany Machmud serta memiliki dua kakak Laki-Laki yaitu Chappy Marzalio Junior dan Febrizar Aviliano. Penulis dilahirkan di Surabaya pada tanggal 25 Agustus 1995. Penulis

menempuh pendidikan di SD Al-Falah Tropodo, SMP Al-Falah Tropodo, SMA Negeri 17 Surabaya dan melanjutkan pendidikan di Matematika ITS pada tahun 2013. Saat belajar di Departemen Matematika, penulis mengambil bidang minat yaitu Ilmu Komputer. Selama kuliah penulis mempelajari bahasa pemrograman. Bahasa Pemrograman yang pernah penulis pelajari adalah Java dan C++. Semasa menempuh jenjang pendidikan S-1, penulis juga aktif dalam kegiatan non-akademis diantaranya aktif di Himpunan Mahasiswa Matematika ITS sebagai staff Dalam Negeri HIMATIKA ITS periode 2014/2017, staff Internal Affair HIMATIKA ITS periode 2015/2016, dan Steering Committee PADAMU HIMATIKA 2015. Untuk mendapatkan informasi yang berhubungan dengan Tugas Akhir ini dapat ditunjukkan ke alamat email: xennyzarvina@gmail.com.