



UNIVERSITAS INDONESIA

**SISTEM PENGENALAN WAJAH (FACE RECOGNITION)
MENGUNAKAN METODE HIDDEN MARKOV MODEL (HMM)**

SKRIPSI

**SEPRITAHARA
0906603096**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JANUARI 2012**



UNIVERSITAS INDONESIA

**SISTEM PENGENALAN WAJAH (FACE RECOGNITION)
MENGUNAKAN METODE HIDDEN MARKOV MODEL (HMM)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar menjadi
Sarjana Teknik**

**SEPRITAHARA
0906603096**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JANUARI 2012**

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

SISTEM PENGENALAN WAJAH (FACE RECOGNITION) MENGUNAKAN METODE HIDDEN MARKOV MODEL (HMM)

Adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Sepritahara
NPM : 0906603096
Tanda Tangan : 
Tanggal : 20 Januari 2012

PENGESAHAN

Skripsi ini diajukan oleh :

Nama : **Sepritahara**

NPM : **0906603096**

Program Studi : **Teknik Elektro**

Judul Skripsi : **SISTEM PENGENALAN WAJAH
MENGUNAKAN METODE HIDDEN MARKOV
MODEL (HMM)**

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : **Dr. Ir. Dodi Sudiana M.Eng**

Penguji I : **Dr. Ir. Arman D.Diponegoro**

Penguji II : **Ir. Purnomo Sidi Priambodo M.Sc., Ph.D.**

Ditetapkan di : **Depok**

Tanggal : **20 Januari 2012**

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis ucapkan kehadirat Allah SWT karena dengan rahmat, ridho, dan kasih sayang-Nya, penulisan skripsi ini bisa selesai tepat pada waktunya. Salawat dan salam selalu penulis haturkan kepada baginda Rasulullah Muhammad SAW, karena berkat jasa beliau kita dapat hidup di zaman yang terang benderang ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Elektro Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan seminar ini, sangatlah sulit bagi saya untuk menyelesaikan seminar ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Dr. Ir. Dodi Sudiana M.Eng, selaku dosen pembimbing I telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan dalam penyusunan skripsi ini;
2. Dr. Ir. Arman D. Diponegoro, selaku dosen pembimbing II telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan dalam penyusunan skripsi ini;
3. Dona Andika Sukma selaku teman seperjuangan yang telah sangat membantu dalam proses pembuatan skripsi ini.
4. Kedua Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
5. Teman dan sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Depok, 20 Januari 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama : **Sepritahara**
NPM : **0906603096**
Program Studi : **Teknik Elektro**
Departemen : **TEKNIK ELEKTRO**
Fakultas : **TEKNIK**
Jenis Karya : **SKRIPSI**

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Sistem Pengenalan Wajah (*Face Recognition*) menggunakan metode *Hidden Markov Model (HMM)*, beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, Mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 20 Januari 2012

Yang menyatakan



(**Sepritahara**)

ABSTRAK

Nama : Sepritahara
Program Studi : Elektro
Judul : Sistem Pengenalan Wajah (*Face Recognition*) menggunakan Metode *Hidden Markov Model* (HMM)

Sistem pengenalan wajah manusia merupakan salah satu bidang yang cukup berkembang dewasa ini, dimana aplikasi dapat diterapkan dalam bidang keamanan (*security system*) seperti ijin akses masuk ruangan, pengawasan lokasi (*surveillance*), maupun pencarian identitas individu pada *database* kepolisian. Tujuan Penulisan laporan tugas akhir ini adalah untuk membangun sebuah perangkat lunak pengenalan citra wajah manusia menggunakan metode *Hidden Markov Models* (HMM) dengan input *database Pain Ekspresi Subset* dan *database Hasil Foto Sendiri* dengan memanfaatkan aplikasi GUI. Hasil pengujian sistem menunjukkan bahwa sistem pengenalan wajah (*face recognition*) membandingkan percobaan pengenalan sesuai dengan *codebook* (32, 64, 128, 256) dan *iterasi* (5, 10). Sistem pengenalan wajah manusia menggunakan metode *Hidden Markov Models* (HMM) mencapai tingkat akurasi pengenalan sebesar 84,28%, dengan *database* 70 gambar yang terdiri dari 10 individu dengan masing-masing individu memiliki 7 variasi ekspresi yang berbeda.

Kata kunci:

HMM, *centroid*, ukuran *codebook*, jumlah *iterasi*.

ABSTRACT

Name : Sepritahara
Study Program : Electrical Engineering
Title : Face Recognition System Using Hidden Markov Model (HMM)
Method

Human face recognition system is one area that is developing now, where applications can be applied in the field of security (security system) such as permit access into the room, monitoring locations (surveillance), or search for individual identity in the police database. Purpose of this final report is to build a software image of human face recognition using Hidden Markov Models method (HMM) with input Pain Ekspresion Subset database and Image itself database applications of GUI. Test results show that the system of face recognition systems trial comparing the introduction according to the codebook (32, 64, 128, 256) and iteration (5, 10). Human face recognition system using Hidden Markov Models (HMM) reached the level of recognition accuracy of 84,28%, with 70 database that consists of 10 individuals with each individual has 7 variations of expressions.

Key words

HMM, *centroid*, size of codebook, number of iteration.

DAFTAR ISI

JUDUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK.....	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR ISTILAH.....	xiii
1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan masalah.....	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5. Metode Penelitian	3
1.6 Sistematika Penulisan	5
2. TINJAUAN PUSTAKA.....	6
2.1 Citra Digital	6
2.2 Format File Citra.....	7
2.3 Elemen Dasar Citra	8
2.3.1. Pixel (Picture Element).....	8
2.3.2. Bit Depth	9
2.3.3. Resolusi.....	9
2.4. Kecerahan dan Contrass.....	9
2.5. Peningkatan Kualitas Citra.....	10
2.6. Citra Skala Keabuan (<i>Gray Scale</i>)	11
2.7. Fast Fourier Transform (FFT).....	12
2.8. Pengenalan Wajah (FaceRecognition).....	13
2.9. Konsep Pengenalan wajah	14
2.10. Vektor Quantization	15
2.11. Hidden Markov Model (HMM).....	19
3. RANCANG BANGUN PENELITIAN	24
3.1 Prinsip Kerja Sistem	24
3.2 Blok Diagram Sistem.....	26
3.3 Pelabelan, Codebook, dan Training database HMM	29
3.4 Pengenalan Citra wajah (Face Recognition).....	35
4. UJI COBA DAN ANALISA	38
4.1 Daftar Database Variasi Wajah Pada Sistem	38
4.2 Uji Coba Sistem	39

4.2.1 Uji coba sistem menggunakan Pain Ekspression Subset satu gambar	42
4.2.2. Uji coba sistem menggunakan Pain Ekspression Subset empat gambar....	47
4.3 Persentasi Akurasi Satu gambar dan empat Gambar sebagai <i>database</i>	52
4.3.1. Akurasi Pengenalan database Pain Ekspression Subset.....	52
4.3.2. Akurasi Pengenalan database Hasil Foto Sendiri.....	53
4.4. Lama / waktu pengolahan data dengan satu dan empat <i>database</i>	54
4.4.1. Lama Pengolahan database Pain Ekspression Subset	54
4.4.2. . Lama Pengolahan database hasil Foto Sendiri.....	56
4.5 Persentasi akurasi pengenalan wajah Pain ekspression subset dengan database foto sendiri	57
4.6. Persentasi Akurasi pengenalan wajah dengan metode lain.....	59
4.7. Analisa Hasil Uji coba	59
4.7.1. Analisis Pengaruh Codebook terhadap Sistem	60
4.7.2. Analisis Pengaruh Iterasi Terhadap Sistem.....	61
4.7.3. Analisis Pengaruh Jumlah database Terhadap Sistem	61
4.7.4. Analisis Pengaruh Jenis database yang digunakan	62
4.7.5. Analisis Pengaruh codebook, iterasi, jumlah database terhadap waktu	62
5. KESIMPULAN	64
DAFTAR ACUAN	65
DAFTAR PUSTAKA.....	67

DAFTAR GAMBAR

Gambar 2.1	Citra Fungsi Dua Variable.....	6
Gambar 2.2	Pallet skala keabuan	12
Gambar 2.3	Skala Keabuan 4-bit (hitam=0, putih =15).....	12
Gambar 2.4	Kumpulan Codebook.....	16
Gambar 2.5	Codebook Secara Multidimensi	16
Gambar 2.6	Diagram Alir LBG	18
Gambar 2.7	Contoh Probabilitas Transisi A	19
Gambar 3.1	Scan wajah secara <i>Vertical</i>	24
Gambar 3.2	Bentuk matriks gambar	25
Gambar 3.3	BlokDiagram proses training	26
Gambar 3.4	Flowchart Proses training.....	27
Gambar 3.5	Blok Diagram Pengenalan Citra wajah	27
Gambar 3.6	Flowchart Pengenalan Citra Wajah.....	28
Gambar 3.7	Tampilan Menu Utama.....	29
Gambar 3.8	Training Database	30
Gambar 3.9	Proses Pencarian Codebook	30
Gambar 3.10	Tampilan Proses Codebook.....	32
Gambar 3.11	Proses Pengenalan Wajah dengan hasil dikenali.....	36
Gambar 3.12	Proses Pengenalan Wajah dengan hasil tidak dikenali.....	37
Gambar 4.1	Paint Ekspresion Subset database.....	38
Gambar 4.2	Database Hasil Foto Sendiri.....	39

DAFTAR TABEL

Tabel 2. 1	Hubungan Antara <i>Bit Per Pixel dengan</i> Jumlah Warna Maksimum Pada Bitmap.....	8
Tabel 4. 1	Hasil uji coba iterasi 5, codebook 32.....	42
Tabel 4. 2	Hasil uji coba iterasi 5, codebook 64.....	43
Tabel 4. 3	Hasil uji iterasi 5, codebook 128.....	43
Tabel 4. 4	Hasil uji coba iterasi 5, codebook 256.....	44
Tabel 4. 5	Hasil uji coba iterasi 10, codebook 32.....	45
Tabel 4. 6	Hasil uji coba iterasi 10, codebook 64.....	45
Tabel 4. 7	Hasil uji coba iterasi 10, codebook 128.....	46
Tabel 4. 8	Hasil uji coba iterasi 10, codebook 256.....	46
Tabel 4. 9	Hasil uji coba iterasi 5, codebook 32.....	47
Tabel 4. 10	Hasil uji coba iterasi 5, codebook 64.....	48
Tabel 4. 11	Hasil uji coba iterasi 5, codebook 128.....	48
Tabel 4. 12	Hasil uji coba iterasi 5, codebook 256.....	49
Tabel 4. 13	Hasil uji coba iterasi 10, codebook 32.....	50
Tabel 4. 14	Hasil uji coba iterasi 10, codebook 64.....	50
Tabel 4. 15	Hasil uji coba iterasi 10, codebook 128.....	51
Tabel 4. 16	Hasil uji coba iterasi 10, codebook 256.....	52
Tabel 4. 17	Akurasi database satu ekspresi dan empat ekspresi.....	53
Tabel 4. 18	Akurasi database satu ekspresi dan empat ekspresi.....	53
Tabel 4. 19	Akurasi database satu ekspresi dan empat ekspresi.....	54
Tabel 4. 20	Akurasi database satu ekspresi dan empat ekspresi.....	54
Tabel 4. 21	Lama Pengolahan database satu ekspresi dan empat ekspresi.....	55
Tabel 4. 22	Lama Pengolahan database satu ekspresi dan empat ekspresi.....	56
Tabel 4. 23	Lama Pengolahan database satu ekspresi dan empat ekspresi.....	56
Tabel 4. 24	Lama Pengolahan database satu ekspresi dan empat ekspresi.....	57
Tabel 4.25	Akurasi Pengenalan dengan satu ekspresi sebagai database.....	58
Tabel 4.26	Akurasi Pengenalan dengan empat ekspresi sebagai database.....	58
Tabel 4.27	Perbandinagn akurasi dengan metode lain.....	59

DAFTAR ISTILAH

1. **Centroid** : sebuah titik yang mewakili beberapa titik-titik sample (codeword) dalam satu cluster.
2. **Cluster** : suatu ruang dua dimensi dari suatu bidang yang bentuknya tergantung dari reknik yang digunakan.
3. **Codebook** : kumpulan dari sejumlah codeword dari beberapa gelombang.
4. **Codeword** : titik-titik sample dari besaran komponen-komponen spektrum frekuensi dari suatu gelombang yang diperoleh dari hasil transformasi Fourier.
5. **FFT** : Fast Fourier Transform, teknik atau cara untuk mengkoversi suatu gelombang dari domain waktu ke domain frekuensi.
6. **LoP** : *Log of Probability*, besarnya kemungkinan munculnya suatu pengamatan dari suatu sederetan munculnya suatu kejadian-kejadian tertentu (probability of transition) dengan memperhitungkan kemungkinan munculnya kejadian yang mengikuti sebelumnya.
7. **Iterasi** adalah banyaknya pengulangan yang dilakukan dalam pengambilan sample suara gempa
8. **Vector quantization** : proses pemetaan vektor dari ruang vektor yang besar menjadi sebuah wilayah yang terbatas.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Manusia memiliki kemampuan untuk mengenal puluhan bahkan ratusan wajah selama hidupnya. Seseorang dapat mengenali wajah orang lain meskipun tidak beberapa waktu lamanya dan sudah terdapat perubahan pada wajah orang yang dikenal tersebut. Perubahan itu misalnya variasi ekspresi wajah, penggunaan kacamata, perubahan warna dan gaya rambut.

Seiring dengan perkembangan teknologi, dikenallah sebuah istilah *Biometrik*, *biometrik* merupakan metode otomatis untuk mengenali seseorang berdasarkan karakteristik fisik atau perilaku, *biometrik* termasuk di dalamnya *speech recognition*, secara umum digunakan untuk identifikasi dan verifikasi. Identifikasi ialah mengenali identitas seseorang, dilakukan perbandingan kecocokan antara data *biometrik* seseorang dalam database berisi record karakter seseorang. Sedangkan verifikasi adalah menentukan apakah seseorang sesuai dengan apa yang dikatakan terhadap dirinya. sistem *biometrik* merupakan identifikasi secara otomatis terhadap manusia berdasarkan psikological atau karakteristik tingkah laku manusia. Ada beberapa jenis teknologi *biometric* antara lain pengenalan wajah (*face recognition*), pengenalan sidik jari, geometri tangan, pengenalan iris mata (*iris recognition*), suara (*voice recognition*) dan pengenalan tulisan tangan (*handwriting recognition*).

Salah satu teknik *biometric* yang sangat menarik adalah aplikasi yang mampu mendeteksi dan mengidentifikasi wajah. Saat ini, pengenalan wajah melalui aplikasi komputer dibutuhkan untuk mengatasi berbagai masalah, antara

lain dalam identifikasi pelaku kejahatan, pengembangan sistem keamanan, pemrosesan citra maupun film, dan interaksi manusia komputer.

Penelitian dalam bidang pengenalan wajah banyak menarik perhatian pengamat *computer vision* dan menjadi bahan diskusi di antara para praktisi sistem cerdas. Dengan munculnya variasi metode pengenalan wajah, diharapkan aplikasi pengenalan wajah dapat berkembang menjadi lebih akurat agar mampu menjawab kebutuhan dari berbagai permasalahan di atas.

Sistem pengenalan wajah (*Face Recognition*) telah banyak diaplikasikan dengan menggunakan berbagai metode, diantaranya: Metode PCA [5], Metode ICA [6], Metode LDA [7], Metode EP [8], Metode EBGMM [9], Metode Kernel [10], Metode 3-D Morphable model [11], Metode 3-D *Face Recognition* [12], Metode *Bayesian Framework* [13], Metode SVM [14], Metode HMM [15].

Dari beberapa metode diatas, di sini akan dicoba mengembangkan Sistem pengenalan wajah (*Face Recognition*) menggunakan metode *Hidden Markov Model* (HMM), sehingga dalam Tugas Akhir ini akan dikembangkan sebuah aplikasi pengenalan wajah pada citra digital dengan metode *Hidden Markov Models* dengan menggunakan *database Pain Ekspression Subset* sebagai input data untuk diolah oleh program *face recognition*. Dipilih mempergunakan metode HMM sebab HMM mampu memodelkan data 2 dimensi seperti citra dengan baik, serta mendapatkan hasil yang lebih teliti, di samping itu juga penulis ingin membandingkan tingkat akurasi pengenalan wajah menggunakan metode HMM dengan menggunakan metode lain (PCA, LDA, 3D, JST).

1.2. Perumusan Masalah

Berdasarkan uraian diatas, terdapat beberapa permasalahan yang ditemui yaitu :

1. Bagaimana mengenali wajah seseorang dari gambar sesuai dengan *database* yang ada.
2. Bagaimana pengaruh jumlah *database*, *codebook* dan *iterasi* terhadap pengenalan wajah.

1.3 Tujuan

Penulisan laporan tugas akhir ini bertujuan untuk membangun sebuah perangkat lunak pengenalan citra wajah manusia menggunakan metode *Hidden Markov Models* (HMM). Dimana pada tahap pengujian, akan dicari akurasi/tingkat pengenalan yang dicapai oleh perangkat lunak ini untuk mengetahui seberapa besar kemampuan dari penerapan metode *Hidden Markov Models* (HMM) dalam mengenali citra wajah.

1.4. Batasan Masalah

Dalam tugas akhir ini difokuskan pada perancangan dan pembuatan sebuah perangkat lunak pengenalan citra wajah manusia dengan metode *Hidden Markov Models* dengan *database Pain Ekspresi Subset* dan *database Hasil Foto Sendiri* yang berfungsi sebagai input data yang kemudian diolah oleh program *face recognition* dari pengenalan 70 gambar dengan berbagai jumlah *database*, ukuran *codebook* dan *iterasi*.

1.5 Metode Penelitian

Metode penelitian yang digunakan antara lain:

1. Studi Literatur

Penulis menggunakan metode ini untuk memperoleh informasi yang berkaitan dengan penelitian yang penulis buat, mengacu pada buku-buku pegangan, informasi yang didapat dari *internet*, Jurnal-jurnal dan makalah-makalah yang membahas tentang proyek yang penulis buat.

2. Perancangan dan Pembuatan Sistem

Proses perancangan merupakan suatu proses perencanaan bagaimana sistem akan bekerja. Berisi tentang proses perencanaan sistem yang berupa *software*.

3. Pengujian Sistem

Dari sistem yang dibuat, maka dilakukan pengujian terhadap masing-masing bagian dengan tujuan untuk mengetahui kinerja dari sistem tersebut apakah sudah sesuai dengan yang diharapkan atau belum.

4. Pengumpulan Data

Setelah diuji secara keseluruhan sebagai suatu kesatuan sistem, dapat dilihat apakah rancangan *software* sudah dapat bekerja dengan benar atau masih dibutuhkan beberapa perbaikan. Jika sistem sudah dapat bekerja dengan benar, maka dapat dilakukan pengumpulan data yang dianggap penting dan diperlukan.

5. Penulisan Hasil penelitian

Hasil dari pengujian dan pengumpulan data kemudian dianalisa. Dari sini dapat ditarik kesimpulan dari penelitian yang telah dilakukan.

1.6 Sistematika Penulisan

Pada penulisan laporan ini, dapat dibuat urutan bab serta isinya secara garis besar. Diuraikan sebagai berikut :

BAB 1. PENDAHULUAN

Pada bab ini berisi tentang latar belakang, tujuan, pembatasan masalah, dan sistematika penulisan.

BAB 2. TEORI DASAR

Teori dasar berisi landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dalam perancangan dan pembuatan.

BAB 3. PERANCANGAN SISTEM

Proses perancangan merupakan suatu proses perencanaan bagaimana sistem ini akan bekerja. Berisi tentang proses perencanaan *software*. Pada bagian ini akan membahas perancangan dan pembuatan sistem.

BAB 4. IMPLEMENTASI DAN PENGAMBILAN DATA

Bab ini berisi tentang implementasi sistem sebagai hasil dari perancangan sistem. Pengujian akhir nantinya akan dilakukan untuk memastikan bahwa sistem dapat berfungsi sesuai dengan tujuan awal. Setelah sistem berfungsi dengan baik maka akan dilanjutkan dengan pengambilan data untuk memastikan kapabilitas dari sistem yang dibuat.

BAB 5. PENUTUP

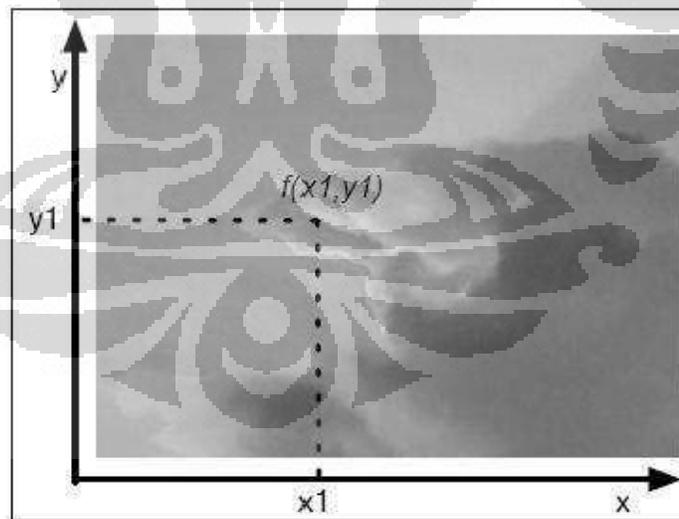
Penutup berisi kesimpulan yang diperoleh dari perancangan, pembuatan, dan implementasi sistem yang telah dilakuka

BAB 2

TINJAUAN PUSTAKA

2.1. Citra Digital

Citra digital adalah suatu citra $f(x,y)$ yang memiliki koordinat spasial, dan tingkat kecerahan yang diskrit. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Fungsi $f(x, y)$ dapat dilihat sebagai fungsi dengan dua unsur. Unsur yang pertama merupakan kekuatan sumber cahaya yang melingkupi pandangan kita terhadap objek (*illumination*). Unsur yang kedua merupakan besarnya cahaya yang direfleksikan oleh objek ke dalam pandangan kita (*reflectance components*). Keduanya dituliskan sebagai fungsi $i(x, y)$ dan $r(x, y)$ yang digabungkan sebagai perkalian fungsi untuk membentuk fungsi $f(x, y)$ seperti gambar 2.1. [2]



Gambar 2.1. Citra fungsi dua variable

Fungsi $f(x, y)$ dapat dituliskan dengan persamaan :

$$f(x, y) = i(x, y) * r(x, y) \text{ di mana } 0 < i(x, y) < \infty \text{ dan } 0 < r(x, y) < 1$$

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra, atau pixel (picture elemen). [2]

Format citra yang digunakan disini adalah Citra Skala Keabuan (*Gray Scale*). Dikatakan format citra skala keabuan karena pada umumnya warna yang dipakai adalah warna hitam sebagai warna minimum dan warna putih sebagai warna maksimalnya, sehingga warna antara ke dua warna tersebut adalah abu-abu.

2.2. Format File Citra

Sebuah format citra harus dapat menyatukan kualitas citra, ukuran file dan kompatibilitas dengan berbagai aplikasi. Saat ini tersedia banyak format grafik dan format baru tersebut dikembangkan, di antaranya yang terkenal adalah BMP, JPEG, dan GIF. Setiap program pengolahan citra biasanya memiliki format citra tersendiri. Format dan metode dari suatu citra yang baik juga sangat bergantung pada jenis citranya. Setiap format file citra memiliki kelebihan dan kekurangan masing-masing dalam hal citra yang disimpan. Citra tertentu dapat disimpan dengan baik (dalam arti ukuran file lebih kecil dan kualitas gambar tidak berubah) pada format file citra tertentu, karena jika disimpan pada format lain, maka terkadang dapat menyebabkan ukuran file menjadi lebih besar dari aslinya dan kualitas citra dapat menurun. Oleh karena itu, untuk menyimpan suatu citra harus diperhatikan citra dan format file citra apa yang sesuai. Misalnya format citra GIF sangat tidak cocok untuk citra fotografi karena biasanya citra fotografi kaya akan warna, sedangkan format GIF hanya mendukung sejumlah warna sebanyak 256

(8 bit) saja. Format JPEG merupakan pilihan yang tepat untuk citra–citra fotografi karena JPEG sangat cocok untuk citra dengan perubahan warna yang halus.[17]

Pada format bitmap, citra disimpan sebagai suatu matriks di mana masing–masing elemennya digunakan untuk menyimpan informasi warna untuk setiap *pixel*. Jumlah warna yang dapat disimpan ditentukan dengan satuan *bit-per-pixel*. Semakin besar ukuran bit-per-pixel dari suatu bitmap, semakin banyak pula jumlah warna yang dapat disimpan. Format bitmap ini cocok digunakan untuk menyimpan citra digital yang memiliki banyak variasi dalam bentuknya maupun warnanya, seperti foto, lukisan, dan frame video. [17]

No	Jumlah bit per <i>pixel</i>	Jumlah warna maksimum
1	1	2
2	4	16
3	8	256
4	16	65536
5	24	16777216

Tabel 2.1. Hubungan Antara *Bit Per Pixel* dengan Jumlah Warna Maksimum Pada Bitmap[17]

2.3. Elemen Dasar Citra

2.3.1. Pixel (*picture element*)

Gambar yang bertipe bitmap tersusun dari pixel –pixel, pixel disebut juga dengan dot. Berbentuk bujur sangkar dengan ukuran relatif kecil yang merupakan penyusun atau pembentuk gambar bitmap. Banyaknya pixel tiap satuan luas tergantung pada resolusi yang digunakan. Keanekaragaman warna pixel tergantung pada *bit depth* yang dipakai. Semakin banyak jumlah pixel tiap satuan

luas, semakin baik kualitas gambar yang dihasilkan dan tentu semakin besar ukuran filenya.[16]

2.3.2. *Bit depth*

Bit depth (kedalaman warna) /*pixel depth/color depth*. *Bit depth* menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan / dicetak dalam setiap pixel, semakin besar nilainya semakin bagus kualitas gambar yang dihasilkan, tentu ukurannya juga semakin besar.[16]

2.3.3. *Resolusi*

Resolusi adalah jumlah pixel persatuan luas yang ada disuatu gambar. Satuan pixel sering dipakai adalah dpi (*dot per inch*)/ppi (*pixel per inch*). Satuan dpi menentukan jumlah pixel yang ada setiap satu satuan luas. Yang dalam hal ini adalah satu inch kuadrat. Resolusi sangat berpengaruh pada detil dan perhitungan gambar. [16]

2.4. **Kecerahan dan Kontras**

Yang dimaksud dengan kecerahan (*brightness*) adalah intensitas yang terjadi pada satu titik citra. Dan lazimnya pada sebuah citra, kecerahan ini merupakan kecerahan rata – rata dari suatu daerah lokal. Sistem visual manusia mampu menyesuaikan dirinya dengan “tingkat kecerahan“ (*brightness level*) dengan jangkauan (dari yang terendah sampai tertinggi). [16]

Untuk menentukan kepekaan kontras (*contrast sensitivity*) pada mata manusia, dilakukan cara pengukuran sebagai berikut. Pada suatu bidang gambar dengan intensitas B , kita perbesar intensitas objek lingkaran sehingga intensitasnya menjadi $B + \Delta B$. Pertambahan intensitas (ΔB) ini dilakukan sampai

mata kita dapat mendeteksi perbedaan ini. Dengan demikian kepekaan kontras dinyatakan dalam rasio weber sebagai $\Delta B/B$. [16]

2.5. Peningkatan Kualitas Citra

Perbaikan citra biasanya dilakukan pada awal proses pengolahan citra. Proses ini dilakukan guna menghasilkan kualitas citra yang lebih baik untuk diolah pada tahap selanjutnya. Kualitas suatu citra dipengaruhi oleh faktor utama yaitu Pencahayaan, pemfokusan objek, kuantitas pendeteksi obyek, frekuensi sampling.

2.5.1. Pencahayaan

Pencahayaan berhubungan dengan cahaya atau sumber cahaya yang terdapat disekitar lingkungan sistem. Secara langsung masalah pencahayaan mempengaruhi kualitas citra yang diperoleh. Teknik pencahayaan yang baik akan menghasilkan citra yang baik atau tidak mengandung derau yang berlebihan. Hal ini penting untuk pemrosesan selanjutnya di bagian pengolahan citra, sehingga dihasilkan informasi yang akurat mengenai obyek dan lingkungan sistem pada umumnya. [16]

Terkait dengan perangkat lunak yang digunakan, kualitas citra dapat ditingkatkan dengan dua metoda yang sederhana yaitu mengatur nilai *contras* dan *brighness*. Kombinasi yang tepat antara kedua variabel tersebut dapat meningkatkan kualitas citra yang terbentuk, sehingga proses identifikasi unsur – unsur obyek dapat dilakukan dengan baik. [16]

2.5.2. Pemfokusan obyek

Pemfokusan obyek terkait erat dengan lensa kamera yang digunakan sebagaimana diketahui, lensa merupakan peralatan optik yang berfungsi

Universitas Indonesia

mengumpulkan cahaya yang dipantulkan oleh obyek dan juga daerah sekitarnya. Pemfokusan obyek akan menentukan ketajaman atau kejelasan citra yang diperoleh.[16]

2.5.3. Kuantitas pendeteksi obyek

Pada sebuah kamera, pendeteksi obyek berupa dioda – dioda yang peka terhadap cahaya. Dioda – dioda ini memiliki kemampuan mendeteksi muatan cahaya dan mengkonversikannya menjadi muatan listrik. Disamping itu dioda – dioda akan membentuk array citra. Semakin banyak dioda yang menyusun array citra, resolusi citra yang mampu dihasilkan akan semakin besar, berarti kualitas citra semakin baik pula.[16]

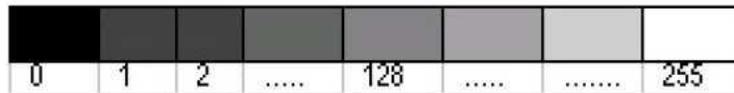
2.5.4. Frekuensi sampling

Faktor ini berhubungan dengan periode dan frekuensi sampling itu sendiri. Periode sampling merupakan waktu yang diperlukan untuk proses pengambilan sample citra analog. Sedangkan frekuensi sampling adalah banyaknya sample citra yang diambil dalam satu periode sampling. Frekuensi sampling untuk satu periode harus tepat agar citra yang dihasilkan dapat diidentifikasi dengan baik . frekuensi sampling sangat menentukan kualitas citra yang terbentuk pada proses digitalisasi ini. Frekuensi sampling yang berlebihan atau sebaliknya sangat kurang akan berdampak pada kejelasan citra. [16]

2.6. Citra Skala Keabuan (*Gray Scale*)

Citra Skala Keabuan (*Gray Scale*) Dikatakan format citra skala keabuan karena pada umumnya warna yang dipakai adalah warna hitam sebagai warna minimum dan warna putih sebagai warna maksimalnya, sehingga warna antara ke

dua warna tersebut adalah abu-abu, dimana Jumlah maksimum warna sesuai dengan bit penyimpanan yang digunakan .[17]

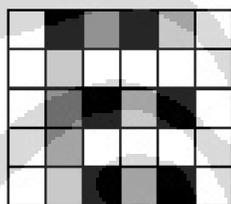


Gambar 2.2. Palet skala keabuan [18]

Contoh :skala keabuan 4 bit

jumlah kemungkinan $2^4 = 16$ warna

kemungkinan warna 0 (min) sampai 15 (max)



= 15 0 6 0 13 15
 = 15 12 15 15 15 15
 = 15 5 0 12 0 15
 = 15 8 15 15 15 15
 = 15 10 0 13 0 15

Gambar 2.3. Skala keabuan 4 bit (hitam=0, putih=15) [18]

2.7. Fast Fourier Transform (FFT)

FFT mengubah masing-masing *frame* dari domain waktu ke domain frekuensi. FFT adalah *fast algorithm* untuk mengimplementasikan *Discrete Fourier Transform (DFT)* yang mana didefinisikan sebagai N sampel $\{x_n\}$, yaitu:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn / N} \dots\dots\dots (2.5)$$

- b. *Pre-processing module*, merupakan proses penyesuaian citra input yang meliputi, normalisasi ukuran citra, histogram equalization untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya, media filtering untuk menghilangkan *noise* akibat kamera atau pergeseran frame, *high pass filtering* untuk menunjukkan bagian tepi dari citra, *background removal* untuk menghilangkan *background* sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input. Bagian pre-processing ini untuk menghilangkan masalah yang akan timbul pada proses pengenalan wajah seperti yang dijelaskan sebelumnya.
- c. *Feature Extraction module*, digunakan untuk mengutip bagian terpenting sebagai suatu vektor yang merepresentasikan wajah dan bersifat unik.
- d. *Classification module*, pada modul ini, dengan bantuan pemisahan pola, fitur wajah yang dibandingkan dengan fitur yang telah tersimpan di *database* sehingga dapat diketahui apakah citra wajah tersebut dikenali.
- e. *Training set*, modul ini digunakan selama proses pembelajaran, proses pengenalan, semakin kompleks dan sering proses pengenalan wajah akan semakin baik.
- f. *Database*, berisi kumpulan citra wajah

2.9. Konsep Pengenalan Wajah

Pengenalan wajah adalah suatu metoda pengenalan yang berorientasi pada wajah. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : Dikenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola yang sebelumnya disimpan di dalam *database*. Metoda ini juga harus mampu mengenali objek bukan wajah. Perhitungan model pengenalan wajah memiliki beberapa masalah. Kesulitan muncul ketika wajah direpresentasikan dalam suatu pola yang berisi informasi unik yang membedakan dengan wajah yang lain.

Metoda pengenalan wajah memakai dua prosedur, yaitu :

- a. Pengenalan kontur wajah dengan mengenali bentuk hidung, mata dan mulut dan bentuk korelasi diantara keduanya. Karakteristik organ tersebut kemudian dinyatakan dalam bentuk vektor.
- b. Analisis komponen yang prinsipil, berdasarkan informasi dari konsep ini, mencari perhitungan model terbaik yang menjelaskan bentuk wajah dengan mengutip informasi yang paling relevan yang terkandung di dalam wajah tersebut. Dibalik kemudahan mengenali wajah, ada beberapa masalah yang mungkin timbul dalam proses pengenalan wajah, yaitu: adanya perubahan skala, perubahan posisi, perubahan pencahayaan, atau adanya perubahan detail dan ekspresi wajah. [1].

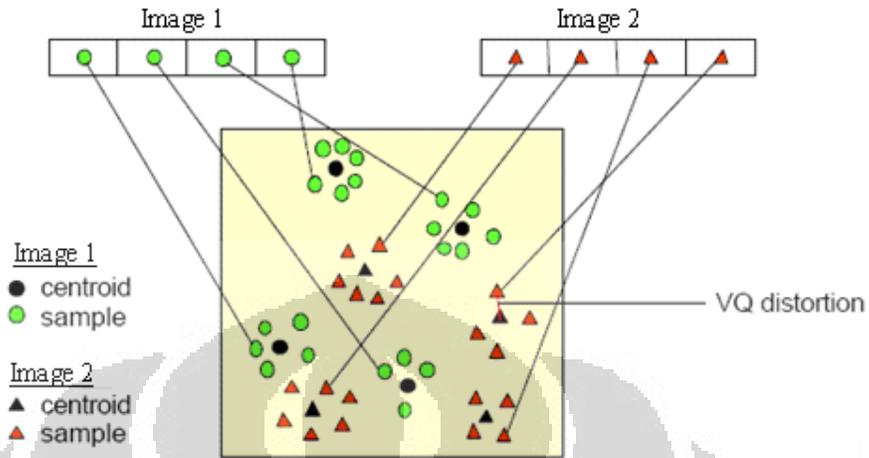
Pada sistem ini, metode pengenalan wajah yang dipilih yaitu: Pengenalan kontur wajah dengan sistem *scan* secara vertical.

2.10. *Vector Quantization*

Vector quantization (VQ) adalah proses pemetaan vektor dari ruang vektor yang besar menjadi sebuah wilayah yang terbatas. VQ sangat baik digunakan dalam *face recognition* karena mengurangi kesalahan dan memiliki akurasi yang tinggi. VQ akan mengkompresi sinyal dalam domain frekuensi. Sinyal akan dikompresi sesuai dengan ukuran *codebook*. [3]

Tiap wilayah disebut sebagai *cluster* dan dapat direpresentasikan oleh *centroid* yang disebut *codeword*. Kumpulan dari *codeword* disebut *codebook*. Gambar 2.4 Menunjukkan diagram yang mengilustrasikan proses *recognition*. Pada gambar terdapat dua *user* dan dua dimensi. Lingkaran menunjukkan vektor dari *user* satu. Segitiga adalah *user* dua. Pada saat *training*, tiap *user* akan dikelompokkan dengan meng-*cluster* tiap-tiap vektornya. Jarak antara vektor yang dekat dengan *codeword* disebut sebagai *distortion*. [3]

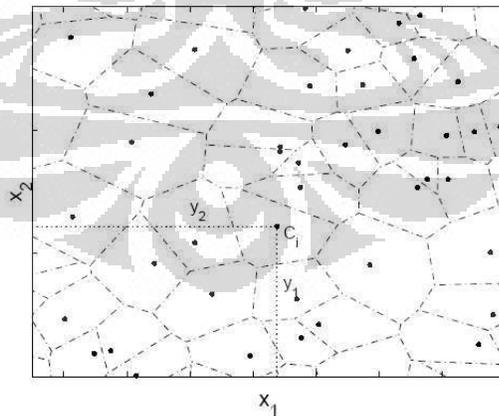
Universitas Indonesia



Gambar 2.4 Kumpulan codebook

VQ diinterpretasikan dengan skalar kuantisasi. Sinyal masukan akan dikuantisasi menjadi *codebook* $C = \{y_k \mid k = 1, \dots, N\}$. Sinyal masukan yang digunakan merupakan sebuah vektor yang harus dikodekan ke dalam ruang multidimensi. Gambar 2.5 adalah contoh ruang dua dimensi dari *codebook*. Pada gambar menunjukkan partisi dari ruang multidimensi sebuah masukan vektor yang dibagi menjadi L wilayah yang dapat dinotasikan sebagai $P = \{C_1, C_2, \dots, C_L\}$ dimana [3] :

$$C_i = \{x \mid d(x, y_i) \leq d(x, y_j), j \neq i\} \dots\dots\dots(2.2)$$



Gambar 2.5 Codebook secara multidimensi.

Pada tahap *recognition*, sinyal masukan akan di vektor-kuantisasi menggunakan semua *trained codebook* dan selanjutnya dihitung total VQ, *distortion*-nya. Total *distortion* yang paling kecil antara *codeword* dari salah satu sinyal dalam *database* dan VQ *codebook* dari sinyal masukan diambil sebagai hasil identifikasi.

Dalam pembentukan *codebook* untuk iterasi guna memperbaiki VQ digunakan *General Lloyd Algorithm* (GLA) atau yang sering disebut dengan LBG *algorithm*. LBG VQ *algorithm* tersebut dapat diimplementasikan dengan prosedur rekursif sebagai berikut:

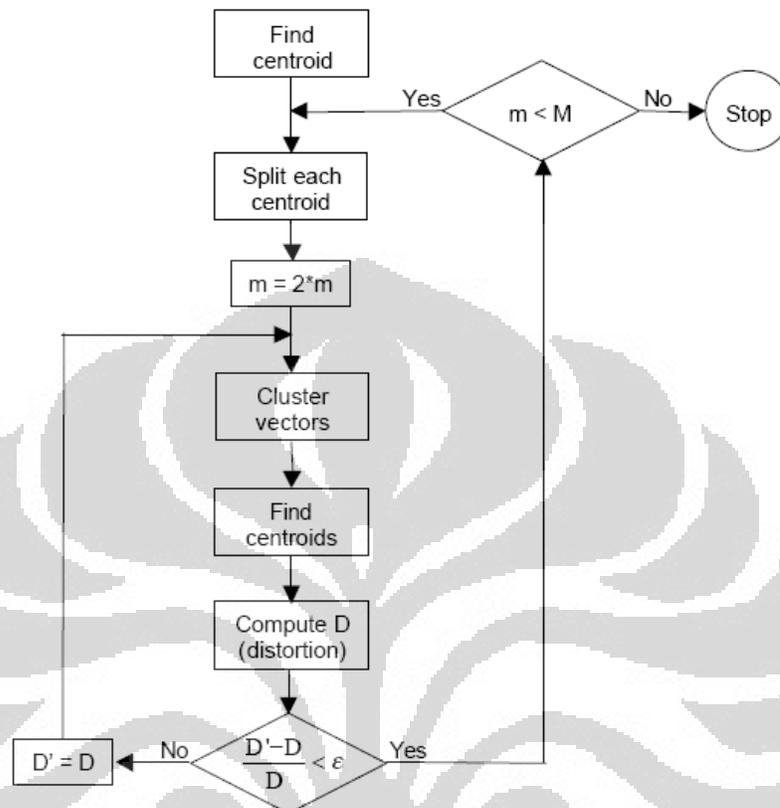
1. Merencanakan vektor *codebook* yang merupakan *centroid* dari keseluruhan vektor *training*
2. Melipatgandakan ukuran dari *codebook* dengan membagi masing-masing *codebook* C_n menurut aturan

$$C_n^+ = C_n(1 + \varepsilon) \dots\dots\dots(2.3)$$

$$C_n^- = C_n(1 - \varepsilon) \dots\dots\dots(2.4)$$

Dimana n bervariasi dari satu sampai dengan *current size codebook* dan ε adalah parameter *splitting* ($\varepsilon = 0.01$)

3. *Nearest Neighbour Search*
Mengelompokkan *training vector* yang mengumpul pada blok tertentu. Selanjutnya menentukan *codeword* dalam *current codebook* yang terdekat dan memberikan tanda vektor yaitu *cell* yang diasosiasikan dengan *codeword-codeword* yang terdekat.
4. *Centroid Update*
Menentukan *centroid* baru yang merupakan *codeword* yang baru pada masing-masing *cell* dengan menggunakan *training vector* pada *cell* tersebut.
5. Iterasi 1
Mengulang step 3 dan 4 sampai jarak rata-rata dibawah *present threshold*
6. Iterasi 2
Mengulang step 2, 3, dan 4 sampai *codebook* berukuran M .

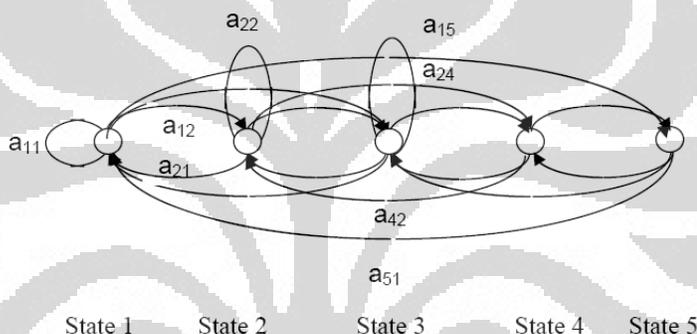


Gambar 2.6 Diagram Alir LBG

Gambar 2.6 menunjukkan diagram alir dari algoritma LBG. *Cluster vector* menerapkan prosedur *nearest neighbour search* yang mana menandai masing-masing *training vector* ke sebuah *cluster* yang diasosiasikan dengan *codeword* terdekat. '*find centroid*' merupakan prosedur meng-*update centroid* untuk menentukan *codeword* yang baru. '*Compute D (distortion)*' berarti menjumlah jarak semua *training vector* dalam *nearest neighbour search* terhadap *centroid* untuk menentukan besarnya distorsi. [3]

2.11. Hidden Markov Models (HMM)

Hidden Markov model merupakan pemodelan probabilitas suatu sistem dengan mencari parameter-parameter markov yang tidak diketahui untuk memperoleh analisis sistem tersebut. Metode Hidden Markov Model (HMM) mampu menangani perubahan statistik dari gambar, dengan memodelkan elemen-elemen menggunakan probabilitas. Salah satu aplikasinya adalah pada *image processing*, HMM memiliki tiga parameter utama yang harus dicari nilainya terlebih dahulu, ketiga parameter tersebut sebagai berikut:



Gambar 2.7. Contoh Probabilitas Transisi a [3]

Parameter a dalam HMM dinyatakan dalam sebuah matriks dengan ukuran MxM dengn M adalah jumlah state yang ada. Pada gambar ada 5 (lima) satate sehingga setiap state memiliki 5 hubungan transisi, maka parameter A dapat dituliskan dalam bentuk matriks seperti pada gambar berikut:

$$A = a_{ij} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \dots\dots\dots (2.5)$$

Parameter B disebut sebagai probabilitas state, merupakan proses kemunculan suatu state dalam deretan seluruh state yang ada.

Parameter B dalam HMM dituliskan dalam bentuk matriks kolom dengan ukuran $M \times 1$, dimana M merupakan jumlah seluruh state yang ada. Misalnya terdapat 5 buah state dalam suatu kondisi, maka matriks B yang terbentuk ditunjukkan oleh persamaan berikut:

$$B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \dots\dots\dots(2.6)$$

Parameter Π , disebut sebagai parameter awal, merupakan probabilitas kemunculan suatu state di awal.

Sama halnya dengan parameter B, parameter Π juga dituliskan dalam bentuk matriks kolom dengan ukuran $M \times 1$, dimana M adalah jumlahnya, jadi jika terdapat 5 (lima) buah state maka parameter Π yang dihasilkan akan ditunjukkan seperti pada gambar berikut.

$$\Pi = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \dots\dots\dots (2.7)$$

Hidden Markov Model dapat dituliskan sebagai $\lambda = (A, B, \pi)$. Dengan diketahuinya N, M, A, B , dan π , *Hidden Markov Model* dapat menghasilkan urutan observasi $O = O_1 O_2 \dots O_T$ dimana masing-masing observasi O_t adalah simbol dari V , dan T adalah jumlah urutan observasi. *Hidden Markov Model* dapat dituliskan sebagai $\lambda = (A, B, \pi)$.

Perhitungan yang efisien dari $P(O|\lambda)$, yaitu probabilitas urutan observasi apabila diberikan urutan observasi $O = O_1 O_2 \dots O_T$ dan sebuah model $\lambda = (A, B, \pi)$.

Misalkan diberikan urutan *state*

$$Q = q_1 q_2 \dots q_T \quad (2.8)$$

dimana q_1 adalah inisial *state*. Dengan demikian probabilitas urutan observasi O untuk urutan *state* pada persamaan (2.8) adalah

$$P(O | Q, \lambda) = \prod_{t=1}^T P(O_t | q_t, \lambda) \quad (2.9)$$

sehingga didapatkan

$$P(O | Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T) \quad (2.10)$$

Probabilitas dari urutan *state* Q dapat dituliskan

$$P(Q | \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \quad (2.11)$$

Probabilitas gabungan dari O dan Q yaitu probabilitas dari O dan Q yang terjadi secara bersamaan. Probabilitas gabungan ini dapat dituliskan

$$P(O, Q | \lambda) = P(O | Q, \lambda)P(Q | \lambda) \quad (2.12)$$

Probabilitas observasi O yang diberikan, diperoleh dengan menjumlahkan seluruh probabilitas gabungan terhadap semua kemungkinan urutan *state* q , yaitu

$$P(O | \lambda) = \sum_{all Q} P(O | Q, \lambda)P(Q | \lambda) \quad (2.13)$$

$$= \sum_{q_1 q_2 \dots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \quad (2.14)$$

Untuk menghitung persamaan (2.14) dengan menggunakan prosedur *forward*. Variabel *forward* $\alpha_1(i)$ didefinisikan sebagai probabilitas sebagian urutan observasi $O_1 O_2 \dots O_t$ (hingga waktu t) dan *state* S_i pada waktu t , dari model λ yang diberikan.

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \quad (2.15)$$

untuk menyelesaikan $\alpha_1(i)$ adalah sebagai berikut :

a. Inisialisasi

$$\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N \quad (2.16)$$

b. Induksi

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad 1 \leq t \leq T-1$$

$$1 \leq j \leq N \quad (2.17)$$

c. Terminasi

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.18)$$

Dari nilai probabilitas yang telah didapatkan, dapat dihitung nilai *Log of Probability*-nya (LoP) :

$$LoP = \text{Log } P(O | \lambda) \quad (2.19)$$



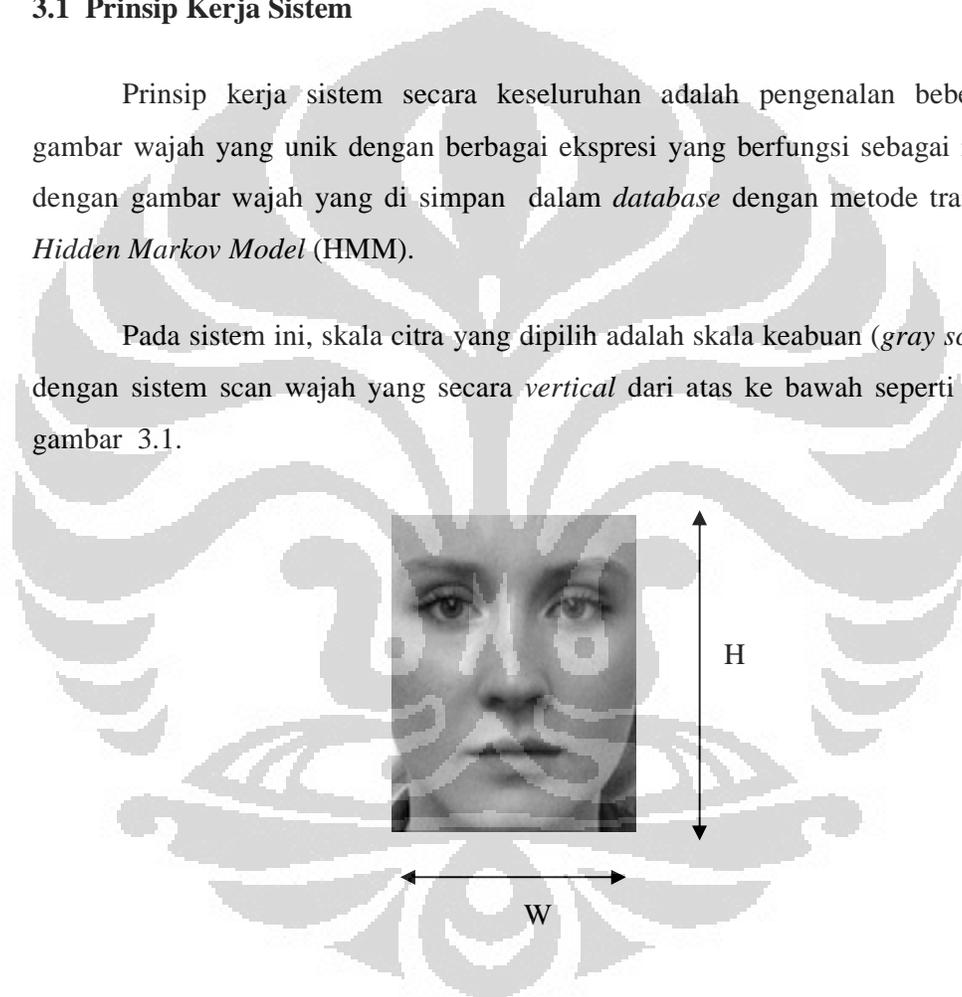
BAB 3

RANCANG BANGUN PENELITIAN

3.1 Prinsip Kerja Sistem

Prinsip kerja sistem secara keseluruhan adalah pengenalan beberapa gambar wajah yang unik dengan berbagai ekspresi yang berfungsi sebagai input dengan gambar wajah yang di simpan dalam *database* dengan metode training *Hidden Markov Model* (HMM).

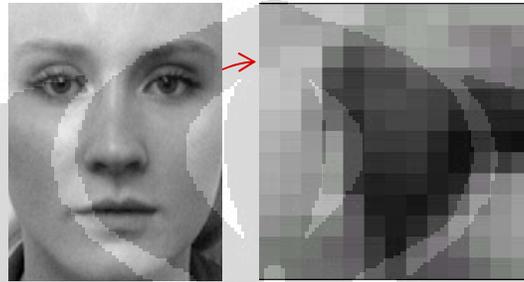
Pada sistem ini, skala citra yang dipilih adalah skala keabuan (*gray scale*), dengan sistem scan wajah yang secara *vertical* dari atas ke bawah seperti pada gambar 3.1.



Gambar 3.1. *Scan wajah secara vertical*

Pada wajah tersebut dengan lebar W dan tinggi H , dimana wajah diproses dengan cara scan secara *vertical* dari atas ke bawah dengan proses sebagai berikut:

- Gambar di scan secara vertical dari atas ke bawah
- *Contrass* gambar diubah (penambahan *Contrass*)
- Gambar tersebut di ubah ke dalam bentuk matriks



201	188	181	185	180	147	140	149	155	138	144	144	145
199	200	201	188	139	132	147	150	143	123	112	102	117
207	221	222	136	90	111	125	145	140	138	122	104	97
231	219	200	90	65	84	84	107	95	92	92	99	89
227	223	181	74	72	89	92	86	77	63	50	55	65
217	211	166	85	47	75	82	83	75	42	42	39	40
208	195	179	131	54	68	66	72	46	21	15	24	19
198	187	181	141	53	54	55	59	37	21	37	66	90
195	184	170	134	52	38	42	45	35	43	98	152	172
186	175	171	169	100	34	34	27	44	85	139	170	184
167	156	142	144	112	48	32	46	84	133	166	172	186
142	139	131	120	108	67	30	76	102	123	153	171	178
145	134	128	125	117	70	38	91	101	105	125	146	157

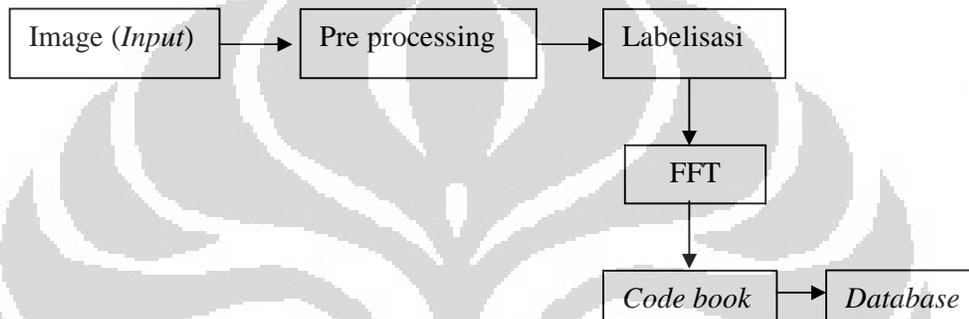
Gambar 3.2. Bentuk matriks dari gambar

- Selanjutnya matriks tersebut diubah ke dalam bentuk sinyal diskrit
- Sinyal diskrit tersebut dipotong-potong dalam bentuk *frame*
- *Frame* tersebut di FFT untuk mendapatkan *sampel point*
- Sample point tersebut di *Vector kuantisasi* untuk mendapatkan *clusterisasi* yang dapat direpresentasikan oleh *centroid* yang disebut *codeword*. Kumpulan *codeword* inilah yang disebut *codebook*.
- Dari proses clusterisasi ini diperoleh *state* dan LOP (*Log Of Probability*).
- Nilai LOP yang tertinggi merupakan hasil identifikasi.

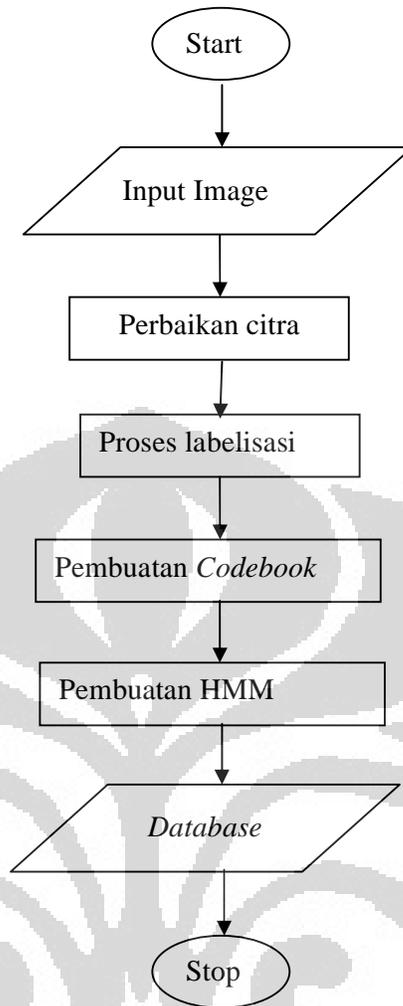
3.2. Blok Diagram Sistem

Secara garis besar sistem kerja dari proses pengenalan citra dibagi menjadi dua yaitu: Proses Training (Pelatihan *face database*) dan Proses Pengenalan Citra wajah (*Face Recognition*).

3.2.1. Proses Training (Pelatihan *face database*)

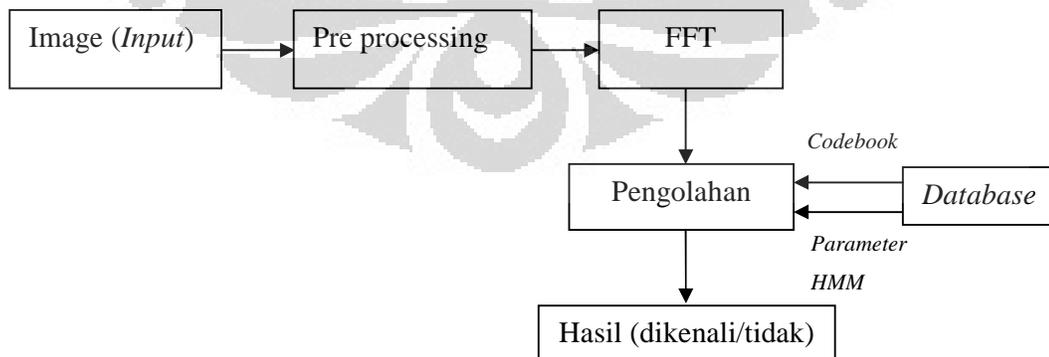


Gambar 3.3. Blok diagram proses training (Pelatihan *face database*)

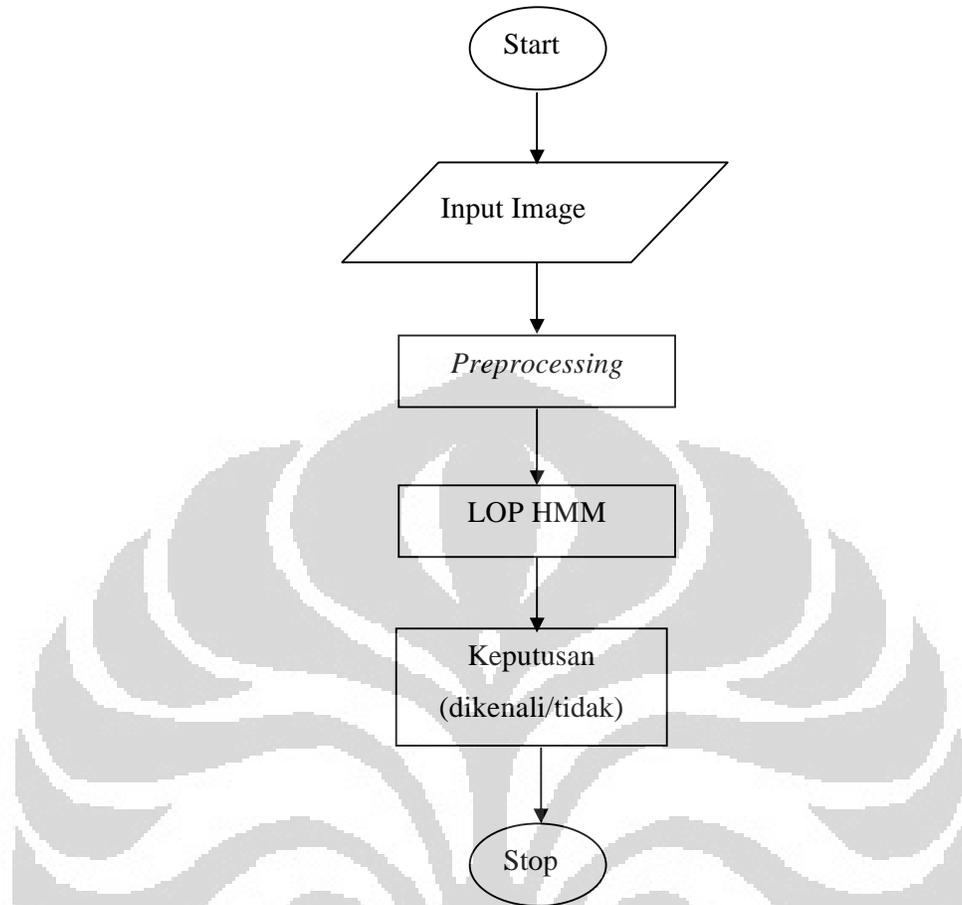


Gambar 3.4. *Flowchart* proses training (Pelatihan *face database*)

3.2.2. Proses Pengenalan Citra wajah (*Face Recognition*)



Gambar 3.5. Blok diagram Pengenalan Citra wajah (*Face Recognition*)



Gambar 3.6. Flowchart Pengenalan Citra wajah (*Face Recognition*)

Perangkat lunak pengenal wajah ini dilakukan dengan menggunakan komputer dengan spesifikasi sebagai berikut :

Sistem Operasi : *Windows 7*

Prosesor : Intel Atom N450 1,66GHz

Memory : 1 GB

Perangkat lunak ini dibuat dengan menggunakan GUI untuk mempermudah penggunaan dimana didalamnya terdapat *function-function* untuk

Universitas Indonesia

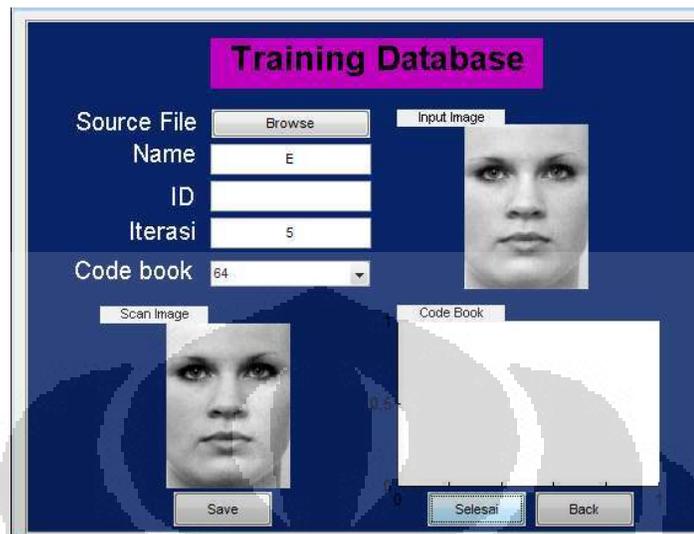
melakukan perhitungan-perhitungan. Untuk memulai memanggil GUI akan terlihat tampilan seperti Gambar 3.7



Gambar 3.7. Tampilan Menu Utama

3.3. Pelabelan, Codebook dan Training database HMM

Selanjutnya dengan menekan tombol *Training database*, maka terlihat seperti gambar 3.8.



Gambar 3.8. Training Database

Selanjutnya apabila ditekan tombol selesai, maka sistem tersebut akan melakukan proses pencarian *codebook*, seperti terlihat pada gambar 3.9.



Gambar 3.9. Proses pencarian Codebook

3.3.1. Pelabelan

Pada proses ini dilakukan pembuatan label untuk masing-masing jenis gambar wajah. Tiap-tiap gambar wajah akan dikelompokkan dalam satu label dan label tersebut diberi nama sesuai dengan nama yang dimaksud, sehingga jumlah label sama dengan jumlah gambar wajah. Nama label inilah yang nantinya akan menjadi keluaran akhir pada simulasi.

Adapun algoritma pelabelan seperti berikut:

mulai

Input gambar

Input nama

Ubah citra menjadi bentuk matrik kolom

Label = nama

Simpan

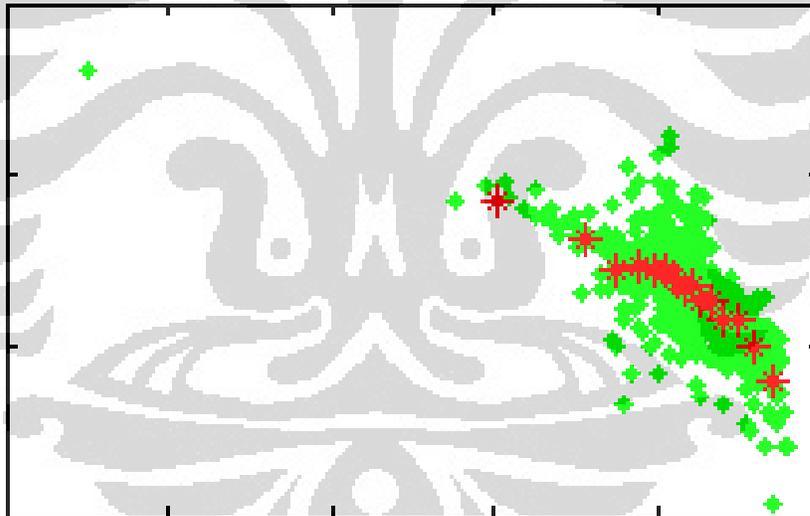
Selesai

3.3.2. Codebook

Setelah proses pembuatan *label* dilakukan, langkah selanjutnya adalah membuat *codebook* dari *label-label* yang telah dibentuk. Pembuatan *codebook* ini terlaksana ketika tombol *codebook* ditekan dan memilih *codebook* berapa yang diinginkan (32, 64, 128, 256, 512), selanjutnya menginput *iterasi* yang diinginkan, dan menekan tombol save untuk memulai proses pencarian *codebook*.

Data dari hasil proses pembuatan *Label* akan *diload* untuk di bentuk *frame-frame* (*frame blocking*) untuk memecah-mecah signal menjadi bagian-bagian kecil. Bagian-bagian kecil ini kemudian dirubah menjadi domain waktu dengan menggunakan fungsi FFT. Sinyal-sinyal dalam domain frekuensi ini di hitung untuk mendapatkan *centroid-centroid*. Kumpulan-kumpulan dari *centroid* inilah yang disebut *codebook*.

Pada pembuatan program *codebook* ini dilengkapi dengan fasilitas status proses berupa persentase dari jalannya program guna memudahkan *user* untuk mengetahui sejauh mana proses pembuatan *codebook* sudah berjalan. Berikut adalah Gambar 3.10 yang merupakan tampilan *codebook* dari salah satu *training*:



Gambar 3.10. Tampilan proses Codebook

Algoritma pembuatan *codebook* sebagai berikut:

```
Mulai

Tentukan besar nilai N

Untuk I = 1 sampai M
    Hitung FFT untuk setiap sample[i];
    Sample point[i] = nilai FFT;
kembali
Tentukan cluster;
Untuk j =1 sampai cluster
    Hitung centroid;
    Simpan centroid[j] berdasarkan urutan
    labelnya;
Kembali
Selesai
```

3.3.3. Training database HMM

Setelah proses pembuatan *codebook* dilakukan, langkah selanjutnya adalah membuat training *database* model HMM dari *label-label* dan *codebook-codebook* yang telah terbentuk dari proses-proses sebelumnya. Untuk membuat model HMM, *user* harus terlebih dahulu menekan tombol *Save* yang terdapat dalam tampilan menu training *database*, seperti gambar 3.8.

Algoritma pembuatan HMM sebagai berikut:

Mulai

Input gambar, codebook dan iterasi

Tingkatkan kontrass pada gambar

Ubah gambar dalam bentuk gray scale

Filter gambar

Bagi matriks gambar kedalam bentuk frame

Hitung FFT masing-masing frame

Panggil file codebook pada database sesuai dengan jumlah codebooknya yang diletakkan pada folder iterasi.

Untuk $i=1$ sampai dengan jumlah data base

Panggil parameter HMM database

Kembali

Selesai

3.4. Pengenalan Citra wajah (*Face Recognition*)

Setelah proses pembuatan training model HMM dilakukan, langkah selanjutnya adalah proses pengenalan wajah (*Face recognition*) dengan menggunakan *codebook-codebook* dan model HMM yang telah terbentuk dari proses-proses sebelumnya. Untuk melakukan proses *recognition* ini *user* harus terlebih dahulu menekan tombol proses pengenalan pada main menu seperti pada Gambar 3.7.

Pada proses *recognition* ini terdapat 2 macam variabel yang sama dengan variabel yang terdapat didalam proses pembuatan *codebook* dan pembuatan HMM pada proses training, yaitu ukuran *codebook*, dan besar iterasi. Variabel ini digunakan untuk menentukan variasi yang diinginkan oleh *user*, proses ini juga mengharuskan *user* untuk memasukkan nama file gambar yang akan dikenali (*direcognition*).

Algoritma pengenalan wajah sebagai berikut:

Mulai

Input gambar yang akan dikenali

Ubah gambar ke dalam bentuk grayscale

Lakukan perbaikan gambar dengan meningkatkan

Contrast.

Ubah gambar dalam bentuk matrik kolom

Matrik kolom dikelompokkan ke dalam beberapa frame

Hitung FFT masing-masing frame

Universitas Indonesia

Untuk $h=1$ sampai jumlah data base

hitung LOP dengan menggunakan parameter

HMM

$LOP(h) = LOP$

Kembali

Tentukan nilai $LOP(h)$ tertinggi

Tentukan nama label untuk $LOP(h)$ tertinggi

Hasil = label untuk $LOP(h)$ tertinggi

Selesai

Gambar 3.11. berikut merupakan hasil pengenalan wajah (*face recognition*) dengan hasil dikenali.



Gambar 3.11. Proses pengenalan wajah (*face recognition*) dengan hasil dikenali

Pada gambar 3.11. dapat dilihat bahwa dengan iterasi 5, serta menggunakan *codebook* 64, maka gambar wajah tersebut dapat dikenali dengan nama E (*database* diberi label E), untuk mencoba masing-masing *iterasi* dan *codebook* sesuai dengan yang diinginkan, dapat dilakukan dengan cara yang sama, serta hasilnya dapat dilihat pada bab IV.

Pada gambar 3.12. dibawah ini merupakan proses pengenalan wajah (*face recognition*) dengan hasil tidak dikenali.



Gambar 3.12. Proses pengenalan wajah (*face recognition*) dengan hasil tidak dikenali

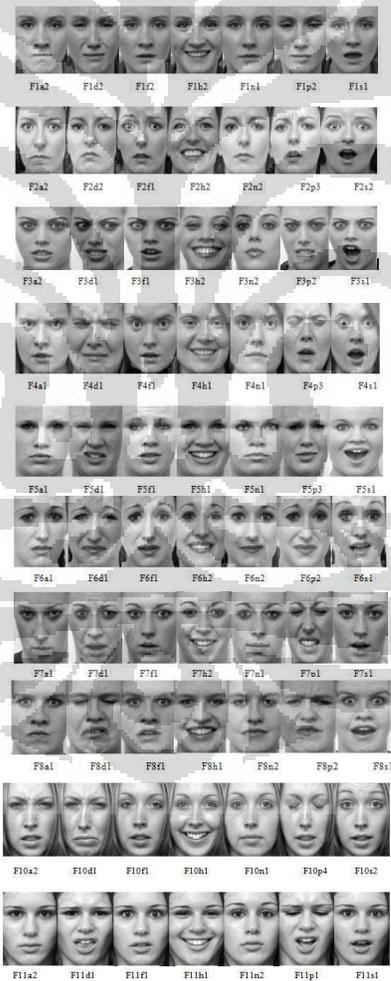
Pada gambar 3.12. dapat dilihat bahwa dengan iterasi 5, serta menggunakan *codebook* 64, maka gambar wajah tersebut dapat dikenali dengan nama C (*database* diberi label E), hal ini menunjukkan bahwa gambar tersebut tidak dapat dikenali sesuai dengan label yang telah diberikan pada *data*

BAB 4

UJI COBA DAN ANALISA

4.1. Daftar *Database* Variasi Wajah pada Sistem

Dalam sistem ini, data base yang digunakan adalah *Pain ekspresion subset database*, seperti gambar 4.1. dan gambar *database* hasil foto sediri, seperti gambar 4.2.



Gambar 4.1. *Pain ekspresion subset database*



Gambar 4.2. Database Gambar hasil foto sendiri

4.2. Uji Coba Sistem

Uji coba yang dilakukan dalam skripsi ini menggunakan 70 gambar wajah dengan berbagai ekspresi. Uji coba ini menggunakan seluruh variasi dari *codebook* dan *iterasi* yang terdapat dalam perangkat lunak sistem pengenalan wajah (*face recognition*). Proses ini dilakukan dengan memasukkan nama file gambar ketika dibrowse dari tampilan proses *recognition*, selanjutnya menekan

Universitas Indonesia

tombol pengenalan. Perangkat lunak ini akan mengolah file yang masuk dengan melakukan perhitungan-perhitungan sampai mendapatkan nilai *log of probability* dan menentukan *label* nama yang dikenali. Secara lengkap hasil-hasil uji coba terdiri dari :

4.2.1. Uji Coba Sistem Menggunakan *Pain Ekspresiin Subset Database*

1. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 32.
2. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 64.
3. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 128.
4. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 256.
5. Hasil uji coba 70 gambar wajah, dengan iterasi 10 dan ukuran *codebook* 32.
6. Hasil uji coba 70 gambar wajah, dengan iterasi 10 dan ukuran *codebook* 64.
7. Hasil uji coba 70 gambar wajah dengan iterasi 10 dan ukuran *codebook* 128.
8. Hasil uji coba 70 gambar wajah dengan iterasi 10 dan ukuran *codebook* 256.

4.2.2. Uji Coba Sistem Menggunakan *Database Hasil Foto Sendiri*

1. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 32.
2. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 64.
3. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 128.
4. Hasil uji coba 70 gambar wajah, dengan iterasi 5 dan ukuran *codebook* 256.

5. Hasil uji coba 70 gambar wajah, dengan iterasi 10 dan ukuran *codebook* 32.
6. Hasil uji coba 70 gambar wajah, dengan iterasi 10 dan ukuran *codebook* 64.
7. Hasil uji coba 70 gambar wajah dengan iterasi 10 dan ukuran *codebook* 128.
8. Hasil uji coba 70 gambar wajah dengan iterasi 10 dan ukuran *codebook* 256.

Langkah uji coba dalam proses *training* Pengenalan wajah (*face recognition*) ini dapat dilihat pada GUI dengan langkah-langkah sebagai berikut:

1. Memasukkan gambar sebagai *database*, dengan cara menekan tombol *browse*, memilih gambar yang akan dijadikan *database* dan memberikan nama *database*, menentukan jumlah iterasi yang diinginkan serta memilih ukuran *codebook* selanjutnya *save*, setelah semua *database* dimasukkan, dengan menekan tombol selesai maka proses *codebook* dan *training* HMM akan tersimpan sebagai *database*, seperti terlihat pada gambar 3.8
2. Setelah proses *database* selesai, selanjutnya adalah proses pengenalan wajah (*face recognition*) dengan cara memasukkan gambar input yang akan dikenali, mengisi jumlah iterasi yang diinginkan serta memilih *codebook* yang dibutuhkan, selanjutnya tekan tombol kenali, maka akan muncul hasil yang diinginkan apakah dikenali atau tidak, seperti gambar 3.11. dan gambar 3.12.

4.2.1. Uji Coba Sistem Menggunakan *Pain Ekspresion Subset Database*

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database Pain Ekspresion Subset*, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, hasil percobaan tersebut dapat dilihat pada tabel 4.1 sampai 4.8.

1. Hasil uji coba sistem dengan iterasi 5 dan ukuran *codebook* 32

Pada Tabel 4.1. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresion Subset* dan iterasi 5 serta ukuran *codebook* 32, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.1. Hasil uji coba sistem dengan iterasi 5 dan ukuran *codebook* 32

No	Nama	Jumlah Uji / Ekspresi	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	3	4	7	0
2	B	10/7	4	3	5	2
3	C	10/7	5	2	5	2
4	D	10/7	6	1	6	1
5	E	10/7	4	3	6	1
6	F	10/7	5	3	7	0
7	G	10/7	3	4	6	1
8	H	10/7	5	2	7	0
9	I	10/7	5	2	7	0
10	J	10/7	3	4	5	2
Jumlah hasil identifikasi			43	28	61	9

2. Hasil uji coba sistem dengan iterasi 5 dan ukuran *codebook* 64

Pada Tabel 4.2. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresion Subset* dan iterasi 5 serta ukuran *codebook* 64, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.2. Hasil uji coba sistem dengan iterasi 5 dan ukuran *codebook* 64

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	7	0	7	0
2	B	10/7	4	3	6	1
3	C	10/7	5	2	6	1
4	D	10/7	6	1	6	1
5	E	10/7	3	4	6	1
6	F	10/7	6	1	7	0
7	G	10/7	3	4	6	1
8	H	10/7	3	4	7	0
9	I	10/7	6	1	7	0
10	J	10/7	5	2	5	2
Jumlah hasil Identifikasi			48	22	63	7

3. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 128

Pada Tabel 4.3. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspression Subset* dan iterasi 5 serta ukuran *codebook* 128, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.3. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 128

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	5	2	7	0
2	B	10/7	5	2	6	1
3	C	10/7	5	2	6	1
4	D	10/7	6	1	6	1
5	E	10/7	5	2	6	1
6	F	10/7	7	0	7	0
7	G	10/7	3	4	6	1
8	H	10/7	4	3	7	0
9	I	10/7	7	0	7	0
10	J	10/7	4	3	5	2
Jumlah hasil Identifikasi			51	19	63	7

4. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 256

Pada Tabel 4.4. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 5 serta ukuran *codebook* 256, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.4. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 256

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	7	0	7	0
2	B	10/7	5	2	7	0
3	C	10/7	4	3	6	1
4	D	10/7	6	1	6	1
5	E	10/7	4	3	6	1
6	F	10/7	6	1	6	1
7	G	10/7	6	1	6	1
8	H	10/7	4	3	7	0
9	I	10/7	6	1	7	0
10	J	10/7	4	3	6	1
Jumlah hasil Identifikasi			52	18	64	6

5. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 32

Pada Tabel 4.5. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 10 serta ukuran *codebook* 32, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.5. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 32

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	6	1	7	0
2	B	10/7	6	1	6	1
3	C	10/7	4	3	7	0

Universitas Indonesia

4	D	10/7	6	1	5	2
5	E	10/7	4	3	6	1
6	F	10/7	6	1	7	0
7	G	10/7	6	1	5	2
8	H	10/7	7	0	7	0
9	I	10/7	6	1	7	0
10	J	10/7	4	3	5	2
Jumlah hasil Identifikasi			55	15	62	8

6. Hasil uji coba iterasi 10 dan ukuran *codebook* 64

Pada Tabel 4.6. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 10 serta ukuran *codebook* 64, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.6. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 64

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 4 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	6	1	7	0
2	B	10/7	6	1	6	1
3	C	10/7	4	3	6	1
4	D	10/7	5	2	6	1
5	E	10/7	2	5	6	1
6	F	10/7	6	1	7	0
7	G	10/7	3	4	6	1
8	H	10/7	5	2	7	0
9	I	10/7	6	1	7	0
10	J	10/7	4	3	5	2
Jumlah hasil Identifikasi			47	23	63	7

7. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 128

Pada Tabel 4.7. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 10 serta

ukuran *codebook* 128, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.7. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 128

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	3	4	7	0
2	B	10/7	6	1	7	0
3	C	10/7	6	1	6	1
4	D	10/7	5	2	6	1
5	E	10/7	3	4	6	1
6	F	10/7	6	1	6	1
7	G	10/7	4	3	7	0
8	H	10/7	5	2	7	0
9	I	10/7	6	1	7	0
10	J	10/7	6	1	6	1
Jumlah hasil Identifikasi			50	20	65	5

8. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 256

Pada Tabel 4.8. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 10 serta ukuran *codebook* 256, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.8. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 256

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	A	10/7	5	2	7	0
2	B	10/7	7	0	7	0
3	C	10/7	4	3	6	1
4	D	10/7	6	1	6	1
5	E	10/7	6	1	6	1
6	F	10/7	7	0	6	1
7	G	10/7	4	3	7	0
8	H	10/7	6	1	7	0
9	I	10/7	6	1	7	0

10	J	10/7	6	1	7	0
Jumlah hasil Identifikasi			57	13	66	4

4.2.2. Uji Coba Sistem Menggunakan *Database* Hasil Foto Sendiri

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database* Hasil Foto Sendiri, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, hasil percobaan tersebut dapat dilihat pada tabel 4.9. sampai 4.16.

1. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 32

Pada Tabel 4.9. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 32, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.9. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 32

No	Nama	Jumlah Uji / Ekspresi	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	4	3	7	0
2	Bagus	10/7	7	0	7	0
3	Dena	10/7	3	4	6	1
4	Dodi	10/7	2	5	4	3
5	Ginde	10/7	6	1	7	0
6	Robi	10/7	4	3	6	1
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			47	23	61	9

2. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 64

Pada Tabel 4.10. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 64, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.10. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 64

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	4	3	7	0
2	Bagus	10/7	6	1	7	0
3	Dena	10/7	4	3	6	1
4	Dodi	10/7	3	4	6	1
5	Ginde	10/7	6	1	7	0
6	Robi	10/7	5	2	6	1
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			49	21	63	7

3. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 128

Pada Tabel 4.11. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 128, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.11. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 128

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	4	3	7	0
2	Bagus	10/7	6	1	7	0
3	Dena	10/7	4	3	6	1

Universitas Indonesia

4	Dodi	10/7	3	4	6	1
5	Ginde	10/7	6	1	7	0
6	Robi	10/7	5	2	7	0
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			49	21	64	6

4. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 256

Pada Tabel 4.12. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 256, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.12. Hasil uji coba dengan iterasi 5 dan ukuran *codebook* 256

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	4	3	7	0
2	Bagus	10/7	6	1	7	0
3	Dena	10/7	5	2	6	1
4	Dodi	10/7	3	4	6	1
5	Ginde	10/7	7	0	7	0
6	Robi	10/7	5	2	7	0
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			51	19	64	6

5. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 32

Pada Tabel 4.13. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran

codebook 32, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.13. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 32

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	5	2	7	0
2	Bagus	10/7	6	1	7	0
3	Dena	10/7	4	3	6	1
4	Dodi	10/7	3	4	6	1
5	Ginde	10/7	6	1	7	0
6	Robi	10/7	5	2	6	1
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			50	20	63	7

6. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 64

Pada Tabel 4.14. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran *codebook* 64, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.14. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 64

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	6	1	7	0
2	Bagus	10/7	7	0	7	0
3	Dena	10/7	4	3	6	1
4	Dodi	10/7	3	4	6	1
5	Ginde	10/7	7	0	7	0
6	Robi	10/7	5	2	7	1
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0

Universitas Indonesia

10	Yoga	10/7	4	3	6	1
Jumlah hasil Identifikasi			53	17	64	7

7. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 128

Pada Tabel 4.15. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran *codebook* 128, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.15. Hasil uji wajah dengan iterasi 10 dan ukuran *codebook* 128

No	Nama	Jumlah Uji / Ekspresi	Database = 1 Ekspresi		Database = 4 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	6	1	7	0
2	Bagus	10/7	7	0	7	0
3	Dena	10/7	4	3	6	1
4	Dodi	10/7	4	3	6	1
5	Ginde	10/7	7	0	7	0
6	Robi	10/7	6	1	7	1
7	Sugeng	10/7	3	4	4	3
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	6	1	6	1
Jumlah hasil Identifikasi			57	13	64	7

8. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 256

Pada Tabel 4.16. merupakan hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran *codebook* 256, dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.16. Hasil uji coba dengan iterasi 10 dan ukuran *codebook* 256

No	Nama	Jumlah Uji / Ekspresi	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 1 Ekspresi	
			Dikenali	Tidak dikenali	Dikenali	Tidak dikenali
1	Alif	10/7	6	1	7	0
2	Bagus	10/7	7	0	7	0
3	Dena	10/7	4	3	7	0
4	Dodi	10/7	4	3	6	1
5	Ginde	10/7	7	0	7	0
6	Robi	10/7	6	1	7	1
7	Sugeng	10/7	4	3	5	2
8	Widya	10/7	7	0	7	0
9	Wily	10/7	7	0	7	0
10	Yoga	10/7	6	1	7	0
Jumlah hasil Identifikasi			58	12	67	4

4.3. Persentasi Akurasi Pengenalan Wajah Menggunakan Satu Ekspresi dan Empat Ekspresi sebagai *Database*

4.3.1. Akurasi Pengenalan dengan Beberapa Ekspresi dari *Database Pain Ekspresion Subset*

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database Pain Ekspresion Subset*, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, persentasi tingkat akurasi dari hasil percobaan tersebut dapat dilihat pada tabel 4.17. sampai 4.18.

1. Tingkat Akurasi dengan Iterasi 5

Pada Tabel 4.17. merupakan tingkat akurasi dari hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresion Subset* dan iterasi 5 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.17. Akurasi dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Pengujian	Akurasi (%)	
		<i>Database</i> = 1 Ekspresi	<i>Database</i> = 4 Ekspresi
1	Iterasi 5, <i>Codebook</i> 32	61,42%	87,14%
2	Iterasi 5, <i>Codebook</i> 64	68,57%	90,00%
3	Iterasi 5, <i>Codebook</i> 128	72,85%	90,00%
4	Iterasi 5, <i>Codebook</i> 256	74,68%	91,42%

2. Tingkat Akurasi dengan Iterasi 10

Pada Tabel 4.18. merupakan tingkat akurasi dari hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspression Subset* dan iterasi 10 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.18. Akurasi dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Pengujian	Akurasi (%)	
		<i>Database</i> = 1 Ekspresi	<i>Database</i> = 4 Ekspresi
1	Iterasi 5, <i>Codebook</i> 32	78,57%	88,57%
2	Iterasi 5, <i>Codebook</i> 64	67,17%	90,00%
3	Iterasi 5, <i>Codebook</i> 128	72,42%	92,85%
4	Iterasi 5, <i>Codebook</i> 256	82,85%	94,28%

4.3.2. Akurasi Pengenalan dengan Beberapa Ekspresi dari *Database* Hasil Foto Sendiri

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database* Hasil Foto Sendiri, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, persentasi tingkat akurasi dari hasil percobaan tersebut dapat dilihat pada tabel 4.19. sampai 4.20.

1. Tingkat Akurasi dengan Iterasi 5

Pada Tabel 4.19. merupakan tingkat akurasi dari hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.19. Akurasi dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Pengujian	Akurasi (%)	
		<i>Database</i> = 1 Ekspresi	<i>Database</i> = 4 Ekspresi
1	Iterasi 5, <i>Codebook</i> 32	67,14%	87,14%
2	Iterasi 5, <i>Codebook</i> 64	70,00%	90,00%
3	Iterasi 5, <i>Codebook</i> 128	70,00%	90,00%
4	Iterasi 5, <i>Codebook</i> 256	72,85%	91,42%

2. Tingkat Akurasi dengan Iterasi 10

Pada Tabel 4.20. merupakan tingkat akurasi dari hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.20. Akurasi dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Pengujian	Akurasi (%)	
		<i>Database</i> = 1 Ekspresi	<i>Database</i> = 4 Ekspresi
1	Iterasi 5, <i>Codebook</i> 32	72,85%	90,00%
2	Iterasi 5, <i>Codebook</i> 64	77,14%	91,42%
3	Iterasi 5, <i>Codebook</i> 128	82,85%	91,42%
4	Iterasi 5, <i>Codebook</i> 256	84,28%	95,71%

4.4. Lama Pengolahan Data Pengenalan Wajah Menggunakan Satu Ekspresi dan Empat Ekspresi sebagai *Database*

4.4.1. Lama Pengenalan dengan *database* *Pain Expression Subset*

Universitas Indonesia

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database Pain Ekspresi Subset*, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, maka lama pemrosesan data dari hasil percobaan tersebut dapat dilihat pada tabel 4.21. sampai 4.22.

1. Tingkat Akurasi dengan Iterasi 5

Pada Tabel 4.21. merupakan lama waktu pemrosesan data hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 5 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.21. Lama pengolahan data dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Ukuran <i>codebook</i>	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
		Pencarian <i>Codebook</i>	Pengenalan	Pencarian <i>Codebook</i>	Pengenalan
1	32	5 menit 2 detik	4 detik	7 menit 33 detik	7 detik
2	64	6 menit 14 detik	5 detik	8 menit 24 detik	8 detik
3	128	7 menit 17 detik	6 detik	10 menit 49 detik	10 detik
4	256	9 menit 3 detik	8 detik	12 menit 45 detik	11 detik

2. Tingkat Akurasi dengan Iterasi 10

Pada Tabel 4.22. merupakan lama waktu pemrosesan data hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database Pain Ekspresi Subset* dan iterasi 10 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.22. Lama pengolahan data dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Ukuran <i>codebook</i>	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
		Pencarian <i>Codebook</i>	Pengenalan	Pencarian <i>Codebook</i>	Pengenalan
1	32	5 menit 55 detik	5 detik	8 menit 29 detik	8 detik
2	64	6 menit 49 detik	6 detik	9 menit 34 detik	9 detik
3	128	7 menit 57 detik	7 detik	12 menit 57 detik	11 detik
4	256	9 menit 58 detik	9 detik	13 menit 45 detik	13 detik

4.4.2. Lama Pengenalan dengan *database* Hasil Foto Sendiri

Setelah melakukan uji coba sistem pengenalan wajah menggunakan *database* Hasil Foto Sendiri, dimana *database* yang dipakai yaitu satu ekspresi dan empat ekspresi dengan berbagai ukuran *codebook* dan iterasi, maka lama pemrosesan data dari hasil percobaan tersebut dapat dilihat pada tabel 4.23. sampai 4.24.

1. Tingkat Akurasi dengan Iterasi 5

Pada Tabel 4.23, merupakan lama waktu pemrosesan data hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 5 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.23. Lama pengolahan data dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Ukuran <i>codebook</i>	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
		Pencarian <i>Codebook</i>	Pengenalan	Pencarian <i>Codebook</i>	Pengenalan
1	32	5 menit	3 detik	7 menit 12 detik	7 detik
2	64	6 menit 3 detik	4 detik	8 menit 4 detik	8 detik
3	128	7 menit 5 detik	5 detik	10 menit 24 detik	9 detik
4	256	8 menit 57 detik	7 detik	12 menit 13 detik	11 detik

2. Tingkat Akurasi dengan Iterasi 10

Pada Tabel 4.24. merupakan lama waktu pemrosesan data hasil uji coba pengenalan 70 gambar wajah dengan menggunakan *database* Hasil Foto Sendiri dan iterasi 10 serta ukuran *codebook* 32,64,128 dan 256. Dimana gambar yang digunakan sebagai *database* yaitu satu ekspresi dan empat ekspresi.

Tabel 4.24. Lama pengolahan data dengan *database* Satu Ekspresi dan Empat Ekspresi

No	Ukuran <i>codebook</i>	<i>Database</i> = 1 Ekspresi		<i>Database</i> = 4 Ekspresi	
		Pencarian <i>Codebook</i>	Pengenalan	Pencarian <i>Codebook</i>	Pengenalan
1	32	6 Menit	4 detik	8 menit 9 detik	8 detik
2	64	7 menit 30 detik	5 detik	9 menit 25 detik	9 detik
3	128	8 menit 46 detik	6 detik	12 menit 54 detik	12 detik
4	256	9 menit 35 detik	8 detik	13 menit 41 detik	13 detik

4.5. Persentasi Akurasi Pengenalan Wajah Menggunakan *database Pain Ekspression Subset* dengan *Database Hasil Foto Sendiri*

Setelah melakukan uji coba pengenalan wajah menggunakan *database Pain Ekspression Subset* dan *database Hasil Foto Sendiri*, maka pada tabel 4.25 dan 4.26 dapat dilihat perbandingan tingkat akurasi pengenalan wajah menggunakan *database Pain Ekspression Subset* dengan *database Hasil Foto Sendiri* dengan lama pengolahan data pada masing-masing *database*.

1. Akurasi Pengenalan Wajah dengan Satu Ekspresi sebagai *Database*

Pada tabel 4.25. merupakan perbandingan akurasi pengenalan dan lama pengolahan data antara *database Pain Ekspression Subset* dan *database Hasil Foto Sendiri* dengan menggunakan satu ekspresi sebagai *database*.

Tabel 4.25. Akurasi Pengenalan dengan satu Ekspresi sebagai Database

Iterasi	Ukuran Codebook	Tingkat/Persentase Akurasi Pengenalan		Lama Pengolahan Data			
				Database Pain Ekspression Subset		Database Hasil Foto Sendiri	
		Database Pain Ekspression Subset	Database hasil foto sendiri	Pembuatan Codebook	Pengenalan	Pembuatan Codebook	Pengenalan
Iterasi 5	32	61,42%	67,14%	5 Menit 2 Detik	4 Detik	5 Menit	3 Detik
	64	68,57%	70,00%	6 Menit 14 Detik	5 Detik	6 Menit 3 Detik	4 Detik
	128	72,85%	70,00%	7 Menit 17 Detik	6 Detik	7 Menit 5 Detik	5 Detik
	256	74,68%	72,85%	9 Menit 3 Detik	8 Detik	8 Menit 57 Detik	7 Detik
iterasi 10	32	78,57%	72,85%	5 Menit 55 Detik	5 Detik	6 Menit 3 Detik	4 Detik
	64	67,17%	77,14%	6 Menit 49 Detik	6 Detik	7 Menit 30 Detik	5 Detik
	128	72,42%	82,85%	7 Menit 57 Detik	7 Detik	8 Menit 46 Detik	6 Detik
	256	82,85%	84,28%	9 Menit 58 Detik	9 Detik	9 Menit 35 Detik	8 Detik

2. Akurasi Pengenalan Wajah dengan Empat Ekspresi sebagai Database

Pada tabel 4.26. merupakan perbandingan akurasi pengenalan dan lama pengolahan data antara *database Pain Ekspression Subset* dan *database Hasil Foto Sendiri* dengan menggunakan empat ekspresi sebagai *database*.

Tabel 4.26. Akurasi Pengenalan dengan empat Ekspresi sebagai Database

Iterasi	Ukuran Codebook	Tingkat/Persentase Akurasi Pengenalan		Lama Pengolahan Data			
				Database Pain Ekspression Subset		Database Hasil Foto Sendiri	
		Database Pain Ekspression Subset	Database hasil foto sendiri	Pembuatan Codebook	Pengenalan	Pembuatan Codebook	Pengenalan
Iterasi 5	32	87,14%	87,14%	7 menit 33 Detik	7 Detik	7 Menit 12 Detik	7 Detik
	64	90,00%	90,00%	8 Menit 24 Detik	8 Detik	8 Menit 4 Detik	8 Detik
	128	90,00%	90,00%	10 Menit 49 Detik	10 Detik	10 Menit 24 Detik	9 Detik
	256	91,42%	91,42%	12 Menit 45 Detik	11 Detik	12 Menit 13 Detik	11 Detik
iterasi 10	32	88,57%	90,00%	8 Menit 29 Detik	8 Detik	8 Menit 9 Detik	8 Detik
	64	90,00%	91,42%	9 Menit 34 Detik	9 Detik	9 Menit 25 Detik	9 Detik
	128	92,85%	91,42%	12 Menit 57 Detik	11 Detik	12 Menit 54 Detik	12 Detik
	256	94,28%	95,71%	13 Menit 45 Detik	13 Detik	13 Menit 41 Detik	13 Detik

4.6. Perbandingan Persentasi Akurasi dengan metode lain

Pada tabel 4.27. Perbandingan persentasi HMM yang diambil adalah dengan menggunakan satu gambar sebagai *database*, karena pada beberapa metode yang lain tersebut juga menggunakan satu gambar sebagai *database*.

Tabel 4.27. Perbandingan tingkat akurasi pengenalan wajah antara metode HMM dengan metode yang lain (PCA, 3D, JST, LDA)

No	Metode Pengenalan	Tingkat akurasi pengenalan Wajah
1	PCA	91%
2	3D	89%
3	JST	84.50 %
4	HMM	84,28%
5	LDA	79,82 %

4.7. Analisis Hasil Uji Coba

Pada penelitian ini, sampel gambar yang ingin diidentifikasi dibandingkan dengan sampel gambar yang ada pada *database*. Hanya ada dua kondisi yang mungkin keluar sebagai hasil identifikasi, yaitu :

1. Sistem dapat mengenali gambar

Kondisi ini terjadi jika hasil pengenalan sesuai dengan masukan. *Label* atau nama dengan probabilitas tertinggi merupakan gambar yang sesuai dengan masukan.

2. Sistem salah mengenali gambar

Kondisi ini terjadi jika hasil pengenalan berbeda dengan masukan. *Label* atau nama dengan probabilitas tertinggi bukan merupakan gambar yang sesuai

dengan masukan. Dalam hal ini, sistem akan menjadikan *label* dengan nilai *log of probability* yang tertinggi sebagai hasil pengenalan.

Keberhasilan sistem dalam mengidentifikasi sampel uji tergantung dari beberapa faktor, diantaranya adalah ukuran *codebook* dan *iterasi*.

4.7.1 Analisis Pengaruh Variasi Ukuran *Codebook* Terhadap Sistem

Berdasarkan Tabel 4.25 dan 4.26 dapat dilihat bahwa untuk jumlah iterasi yang sama, namun ukuran *codebook* meningkat mengakibatkan peningkatan besarnya persen akurasi pada keseluruhan sistem.

Peningkatan besarnya persen akurasi ini disebabkan karena ukuran *codebook* yang semakin besar sehingga mengakibatkan jumlah *centroid* semakin banyak, dengan banyaknya jumlah *centroid* ini membuat proses *kuantisasi* pengenalan nilai vektor data semakin teliti sehingga pemetaan vektor data dapat dilakukan dengan jarak yang semakin kecil.

Sebelumnya telah diketahui bahwa peningkatan ukuran *codebook* dapat meningkatkan persen akurasi secara keseluruhan, akan tetapi bila dilihat dari tiap *label* maka peningkatan ukuran *codebook* belum tentu meningkatkan persen akurasi. Ada *label* dari pengenalan wajah yang persen akurasinya menurun ketika ukuran *codebook*nya meningkat. Kondisi ini dapat dilihat dari Tabel 4.25 dan 4.26. Berkurangnya persentase akurasi untuk tiap *label* ketika ukuran *codebook* meningkat tidak mempengaruhi kemampuan keseluruhan sistem sebab persen akurasi untuk keseluruhan sistem tetap meningkat.

Berkurangnya persen akurasi untuk tiap *label* ini dikarenakan oleh beberapa faktor, diantaranya yaitu: karena adanya karakteristik dari input uji untuk jenis label tertentu memiliki kemiripan dengan karakteristik citra untuk

label lain pada *database* seperti intensitas cahaya saat pengambilan gambar (posisi kemiringan gambar input tidak sama dengan posisi kemiringan gambar pada *database*).

4.7.2. Analisis Pengaruh Iterasi Terhadap Sistem

Perubahan besar *iterasi* yang semakin meningkat akan membuat persen akurasi dari sistem pengenalan gambar semakin meningkat, peningkatan ini dapat dilihat pada Tabel 4.25 dan 4.26. Hal ini dikarenakan oleh iterasi tersebut melakukan perulangan yang semakin sering untuk mencari tata letak / posisi *centroid*, dimana pada awalnya posisi *centroid* tersebut tidak tetap, dengan semakin seringnya dilakukan iterasi maka pada suatu tingkat itersi tertentu posisi *centroid* tersebut bisa tetap, dan hasil yang diidentifikasi tersebut sudah berada pada posisi *centroid* yang tetap, sehingga tingkat akurasi pengenalan wajah tersebut semakin meningkat. Tabel 4.25 dan 4.26 juga dapat dilihat peningkatan persen akurasi ini lebih baik dibandingkan dengan peningkatan persen akurasi ketika ukuran *codebook* meningkat.

4.7.3. Analisis Pengaruh Banyaknya Jumlah *Database* pada Sistem

Banyaknya Jumlah *database* yang digunakan dalam percobaan akan meningkatkan persentase akurasi pengenalan yang diperoleh sistem, hal ini dapat dilihat pada tabel 4.25 dan 4.26. Semakin banyak jumlah *database* yang masukkan, maka kemungkinan kemiripan karakteristik gambar input dengan *database* pada label yang sama semakin tinggi sehingga tingkat pengenalan sistem semakin besar. Dan sebaliknya semakin sedikit jumlah *database* yang dimasukkan kedalam sistem, maka kemungkinan kemiripan karakteristik gambar input dengan *database*

pada label yang sama semakin rendah sehingga tingkat pengenalan sistem semakin kecil.

4.7.4. Analisis Pengaruh Jenis *Database* yang digunakan pada Sistem

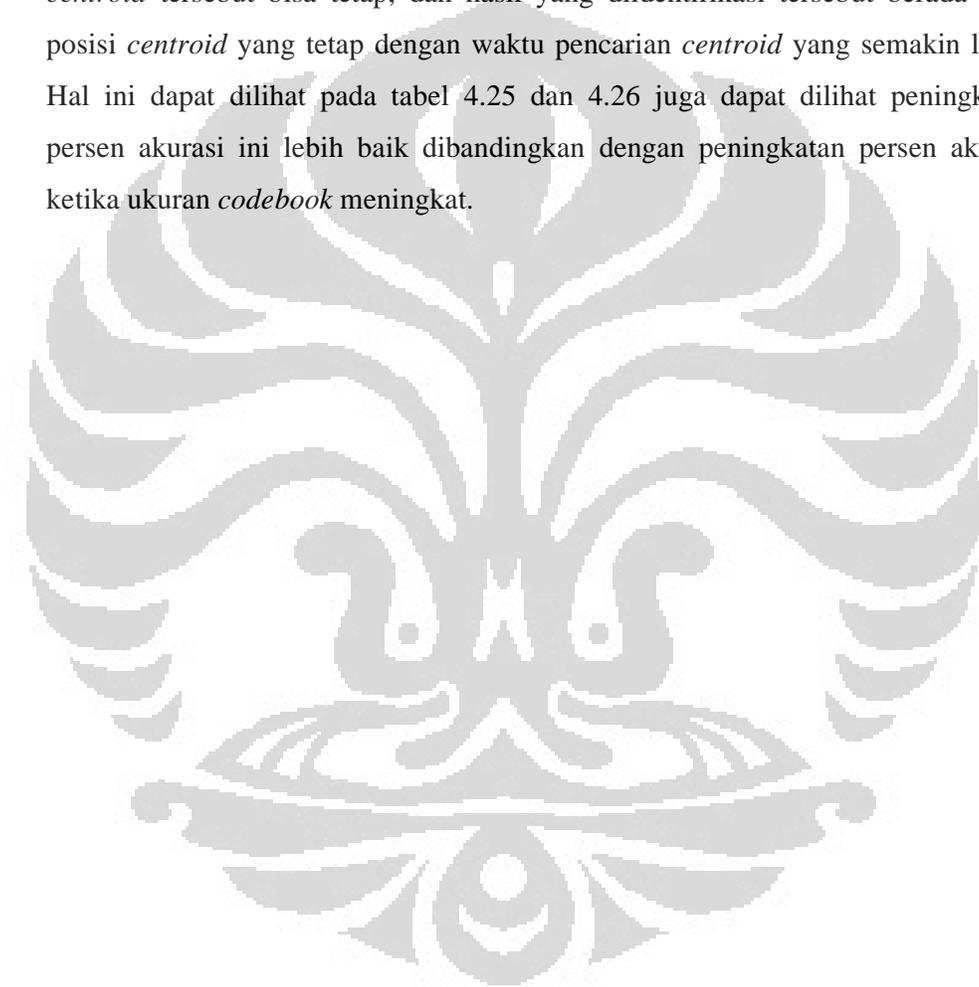
Tingginya tingkat akurasi pengenalan sistem juga dipengaruhi oleh jenis *database* yang digunakan, seperti pada tabel 4.25 dan 4.26. dapat dilihat bahwa persentasi tingkat pengenalan *database* Hasil Foto Sendiri lebih tinggi dibandingkan *database Pain Ekspresion Subset*, hal ini dikarenakan pada *database* hasil foto sendiri dimana wajah antara individu lebih bervariasi dan lebih unik dibandingkan dengan wajah pada gambar *database Pain Ekspresion Subset*, serta pada *database* Hasil Foto Sendiri masing-masing individu memiliki karakteristik yang lebih khas sehingga pada saat pengenalan, sistem bisa mengenali dengan tingkat akurasi pengenalan yang lebih tinggi.

4.7.5. Analisis Pengaruh Jumlah *database*, Ukuran *Codebook* dan Iterasi terhadap Waktu/Lama Pengolahan Data pada Sistem

Semakin banyak jumlah *database* yang dimasukkan pada suatu percobaan, maka lama pencarian *codebook* dan proses pengenalan gambar semakin lama, hal ini dikarenakan semakin banyaknya data yang akan diolah dan dikenali oleh sistem, hal ini dapat dilihat pada tabel 4.25 dan 4.26

Semakin besar ukuran *Codebook* yang diuji pada suatu percobaan, maka lama pencarian *codebook* dan proses pengenalan gambar semakin lama, hal ini ukuran *codebook* yang semakin besar sehingga mengakibatkan jumlah centroid semakin banyak, dengan banyaknya jumlah *centroid* ini membuat proses kuantisasi pengenalan nilai vektor data semakin teliti sehingga pemetaan vektor data dapat dilakukan dengan jarak yang semakin kecil sehingga waktu pencarian *codebook* semakin lama. hal ini dapat dilihat padatabel 4.25 dan 4.26

Pada pengujian sistem, semakin sering Iterasi diberikan, maka waktu pemrosesan data akan semakin lama, hal ini dikarenakan iterasi tersebut melakukan perulangan yang semakin sering untuk mencari tata letak / posisi *centroid*, dimana pada awalnya posisi *centroid* tersebut tidak tetap, dengan semakin seringnya dilakukan iterasi maka pada suatu tingkat itersi tertentu posisi *centroid* tersebut bisa tetap, dan hasil yang diidentifikasi tersebut berada pada posisi *centroid* yang tetap dengan waktu pencarian *centroid* yang semakin lama. Hal ini dapat dilihat pada tabel 4.25 dan 4.26 juga dapat dilihat peningkatan persen akurasi ini lebih baik dibandingkan dengan peningkatan persen akurasi ketika ukuran *codebook* meningkat.



BAB 5

KESIMPULAN

Setelah melakukan perencanaan pemrograman perangkat lunak sistem pengenalan wajah (*face recognition*) serta melalui pengujian yang telah dilakukan maka dapat diambil kesimpulan antara lain:

1. Sistem pengenalan wajah (*face recognition*) dengan menggunakan metode *Hidden Markov Model* (HMM) dapat mengenali gambar sesuai dengan label (nama) yang diberikan pada *database* dan tidak dapat mengenali gambar (namanya tidak sesuai dengan nama yang diberikan pada *database*).
2. Penambahan ukuran *codebook* besar iterasi dan jumlah *database* pada dasarnya akan membuat persen akurasi dari sistem semakin tinggi, namun pada tabel 4.25 dan 4.26 ada beberapa data yang menunjukkan bahwa penambahan ukuran *codebook* tidak mempengaruhi tingginya akurasi, hal ini disebabkan karena adanya karakteristik dari input uji untuk jenis label tertentu memiliki kemiripan dengan karakteristik citra untuk label lain pada *database*, seperti intensitas cahaya saat pengambilan gambar (posisi kemiringan gambar input tidak sama dengan posisi kemiringan gambar pada *database*).
3. Pada penelitian ini, dengan menggunakan *database* Hasil Foto Senidiri, ukuran *codebook* yang optimal adalah 256 dengan jumlah iterasi 10 dan diperoleh persentase akurasi sebesar 84.28%
4. Semakin banyak jumlah *database*, *codebook* dan iterasinya maka waktu training *database* dan pengenalan semakin lama.

DAFTAR ACUAN

- [1] Ni Wayan Marti, Pemanfaatan GUI Dalam *Pengembangan Perangkat Lunak Pengenalan Citra Wajah Manusia menggunakan Metode Eigenface*, 2010
- [2] Esty Vidyaningrum, Prihandoko, *Human Face Detection by using Eigenface Methode for Various Pose of Human Face*, 2009
- [3] Chandra Sasmita, *Pengenalan Golongan darah Jenis ABO Dengan Mempgunakan Pemodelan Hidden Markov*, 2008
- [4] Arie Wirawan Margono, Ibnu Gunawan, Resmana Lim, *Pelacakan dan Pengenalan Wajah Menggunakan Metode Embedded Hidden Markov Models*, Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra: 2004
- [5] M. Turk, A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991
- [6] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, *Face Recognition by Independent Component Analysis*, IEEE Trans. on Neural Networks, Vol. 13, No. 6, November 2002
- [7] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, *Face Recognition Using LDA-Based Algorithms*, IEEE Trans. on Neural Networks, Vol. 14, No. 1, January 2003
- [8] C. Liu, H. Wechsler, *Evolutionary Pursuit and Its Application to Face Recognition*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 6, June 2000
- [9] L. Wiskott, J.-M. Fellous, N. Krueger, C. von der Malsburg, *Face Recognition by Elastic Bunch Graph Matching*, Chapter 11 in Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., CRC Press, 1999
- [10] M.-H. Yang, *Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods*, Proc. of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 20-21 May 2002, Washington D.C. USA
- [11] B. Moghaddam, J.H. Lee, H. Pfister, R. Machiraju, *Model-Based 3D Face Capture with Shape-from-Silhouettes*, Proc. of the IEEE International

Workshop on Analysis and Modeling of Faces and Gestures, AMFG, 17 October 2003

- [12] A. Bronstein, M. Bronstein, R. Kimmel, and A. Spira, *3D face recognition without facial surface reconstruction*, in Proceedings of ECCV 2004, Prague, Czech Republic, 2004
- [13] B. Moghaddam, T. Jebara, A. Pentland, *Bayesian Face Recognition*, Pattern Recognition, Vol. 33, Issue 11, November 2000
- [14] B. Heisele, P. Ho, T. Poggio, *Face Recognition with Support Vector Machines: Global versus Component-based Approach*, Proc. of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vol. 2, 09-12 July 2001
- [15] A.V. Nefian, M.H. Hayes III, *Hidden Markov Models for Face Recognition*, Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98, Vol. 5, 12-15 May 1998
- [16] Hartaman, *Pengolahan Citra Digital*, Jurnal Telekomunikasi, Universitas Indonesia.
- [17] Makalah Pengolahan Citra, Universitas Sumatera Utara
- [18] Idhawati Hestningsih, *Pengolahan Citra*.

DAFTAR PUSTAKA

Saaddatuddaroin. “Perancangan Perangkat Lunak Sendor Tsunami dengan Teknik Hidden Markov Model”. Skripsi, Program Sarjana Fakultas Teknik Universitas Indonesia, Depok, 2009.

Andriani Evi, “Analisa dan Identifikasi berbagai Penyakit Paru-Paru dengan Metode Hidden Markov Model”, Skripsi, Program Sarjana Fakultas Teknik Universitas Indonesia, Depok, 2009.

Sugiarto Ferry, “Pengenalan Plat Nomor Mobil dengan Skeletonisasi dan Hidden Markov Model”, Skripsi, Program Sarjana Fakultas Teknik Universitas Indonesia, Depok, 2007/2008.

