

**PENGENALAN RETINA  
DENGAN HIDDEN MARKOV MODEL**

**TUGAS AKHIR**

Oleh :

**MARIA YULIANTIS**

**06 06 04 2720**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

**PENGENALAN RETINA  
DENGAN HIDDEN MARKOV MODEL**

**TUGAS AKHIR**

Oleh :

**MARIA YULIANTIS**

**06 06 04 2720**



**TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN  
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul :

### **Pengenalan Retina Dengan Hidden Markov Model**

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, Juni 2008

**(Maria Yulianti S)**

NPM 0606042720

## **PENGESAHAN**

Tugas Akhir dengan judul :

### **Pengenalan Retina dengan Hidden Markov Model**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang tugas akhir.

Depok, Juni 2008

Dosen Pembimbing

Prof. Dr. Ir. Harry Sudibyo S., DEA

NIP. 130 891 668

## UCAPAN TERIMA KASIH

Puji dan syukur kehadiran Tuhan Yesus Kristus yang memberi kekuatan serta atas segala berkat, hikmat, pimpinan dan rahmat-Nya sehingga tugas akhir ini dapat diselesaikan dengan baik. Tak lupa penulis juga mengucapkan terima kasih kepada:

**Prof. Dr. Ir. Harry Sudibyo S, DEA**

selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan saran, bimbingan, pengarahan, dan kemudahan lain dalam penyelesaian tugas akhir ini. Tak lupa juga penulis mengucapkan terima kasih kepada dr. Ratna Sitompul dan dr. Arvimadao yang telah meluangkan waktu untuk memberikan saran, bimbingan mengenai retina. Dan kepada keluarga, teman-teman POUK Dian Kasih dan Synovate, dalam dukungannya kepada penulis.

Jakarta, Juni 2008

Penulis

Maria Yulianti S

NPM. 06 06 04 2720

Maria Yulianti S NPM 06 06 04 2720 Departemen Teknik Elektro	Dosen Pembimbing Prof. Dr. Ir. Harry Sudibyo S, DEA
<b>Pengenalan Retina dengan <i>HIDDEN MARKOV MODEL</i></b>	
<p><b>ABSTRAK</b></p> <p>Tugas akhir ini dibuat untuk merancang perangkat lunak yang dapat mengidentifikasi retina manusia melalui proses <i>image processing</i> dengan menggunakan <i>Hidden Markov Model</i>.</p> <p>Dalam pembuatan perangkat lunak indentifikasi ini terdiri atas dua bagian, yakni pembentukan database dan proses identifikasi itu sendiri. Pembentukan database akan menghasilkan <i>codebook</i> dan nilai probabilitas HMM. Identifikasi dilakukan dengan mengambil hasil <i>scanning</i> retina dari rumah sakit. Kemudian dilakukan proses normalisasi dan ekstraksi terhadap gambar retina yang didapat untuk memindai pola pembuluh darah kapiler pada retina. Proses pengenalan retina dalam tugas akhir ini menggunakan <i>Hidden Markov Model</i> yang dilakukan melalui dua tahapan yaitu proses pelatihan data (<i>training</i>) yang dilakukan untuk melatih sistem pengenalan yang bekerja agar dapat mengetahui setiap jenis pola pembuluh darah yang ada, serta proses pengenalan retina itu sendiri (<i>recognition</i>) yang digunakan untuk mengenali retina yang ingin diuji. Seluruh proses yang dilakukan dibuat menggunakan sebuah perangkat lunak untuk memecahkan masalah-masalah matematis. Dari hasil uji coba yang diperoleh, sistem ini dapat mengenali retina yang diuji dengan tingkat akurasi mencapai 100%.</p>	
<p><b>Kata Kunci : Pengenalan Retina, <i>Hidden Markov Model</i></b></p>	

Maria Yulianti S NPM 06 06 04 2720 Electrical Department Engineering	Counsellor Prof. Dr. Ir. Harry Sudibyo S, DEA
<b>THE RECOGNITION OF RETINA USING HIDDEN MARKOV MODEL</b>	
<p><b>ABSTRACT</b></p> <p>This final project is created to design a software that can identify human retina through image processing using hidden markov model.</p> <p>This identification software consist of two part, that are database development and identification itself. The database development produce codebook and HMM probability value. Identification using retinal scanning from hospital. Then the images will be normalisated and extracted to separate retinal vessel. The identification process of retinal in this final assignment is using Hidden Markov Model that will be executed in 2 steps. The first step is data training process whose objective is to train the recognition system so it can recognize each kind of retinal vessel, and the second one is the recognition process of all retinal image. All processes are done by using a mathematic problem solution software. From the obtained test results, this system has the ability to recognize the tested retinal image with 100% accuracy.</p>	
<p><b>Key words: Recognition of retina, Hidden Markov Model</b></p>	

# DAFTAR ISI

Pernyataan Keaslian Tugas Akhir .....	ii
Pengesahan .....	iii
Ucapan Terima Kasih.....	iv
Abstrak.....	v
Abstract.....	vi
Daftar Isi.....	vii
Daftar Gambar.....	ix
Daftar Tabel .....	x
Bab 1 Pendahuluan.....	1
1.1 Latar Belakang .....	1
1.2 Tujuan Penulisan .....	2
1.3 Batasan Masalah.....	2
1.4 Sistematika Penulisan.....	3
Bab 2 Tinjauan Pustaka.....	4
2.1 <i>Retina Mata</i> .....	4
2.2 <i>Image processing</i> .....	5
2.2.1 <i>Sampling dan Gray Level Quantization</i> .....	6
2.2.2 Normalisasi data.....	7
2.2.2.1 <i>Thresholding</i> .....	7
2.2.2.2 <i>Resize dan Cropping</i> .....	8
2.2.2.3 <i>Leveling dan Reshaping</i> .....	9
2.3 <i>Image Recognition</i> .....	9
2.3.1 <i>Ekstraksi</i> .....	10
2.3.1.1 <i>Frame Blocking</i> .....	10
2.3.1.2 <i>Windowing</i> .....	11
2.3.1.3 <i>Discrete Fourier Transform</i> .....	11
2.3.1.4 <i>Fast Fourier Transform</i> .....	11
2.3.2 <i>Vector Quantization</i> .....	12
2.3.3 <i>Hidden Markov Model</i> .....	13

Bab 3	Metodologi.....	16
3.1	Pembentukan Basis Data ( <i>database</i> ).....	16
3.1.1	Proses Segmentasi .....	17
3.1.2	Proses Pelabelan.....	18
3.1.3	Pembuatan <i>Codebook</i> .....	20
3.1.4	Pembentukan Parameter HMM.....	22
3.2	Proses Pengenalan Retina .....	24
Bab 4	Hasil Uji Coba dan Analisis.....	26
4.1	Hasil Uji Coba.....	26
4.1.1.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 4, Ukuran <i>Codebook</i> 32.....	27
4.1.2.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 4, Ukuran <i>Codebook</i> 64.....	28
4.1.3.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 4, Ukuran <i>Codebook</i> 128...28	
4.1.4.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 4, Ukuran <i>Codebook</i> 256...29	
4.1.5.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 8, Ukuran <i>Codebook</i> 32.....	29
4.1.6.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 8, Ukuran <i>Codebook</i> 64.....	30
4.1.7.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 8, Ukuran <i>Codebook</i> 128...31	
4.1.8.	Hasil Uji Coba Dengan <i>Training</i> Retina Sebanyak 8, Ukuran <i>Codebook</i> 256...32	
4.2	Pengolahan Hasil Uji Coba .....	33
4.3	Analisis.....	34
4.3.1.	Pengaruh Ukuran <i>Codebook</i> terhadap Tingkat Akurasi Proses Pengenalan Retina.....	34
4.3.2.	Pengaruh Jumlah <i>Training</i> terhadap Tingkat Akurasi Proses	
Bab 5	Kesimpulan.....	38
Daftar Acuan.....		39
Daftar Pustaka.....		41

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Potongan melalui mata dilihat dari atas.....	4
<b>Gambar 2.2</b>	Retina normal.....	5
<b>Gambar 2.3</b>	Matriks Citra Dijital(x,y)[2].....	6
<b>Gambar 2.4</b>	Citra dengan <i>grayscale</i> level 2 (nilai 0 atau 1)	7
<b>Gambar 2.5</b>	Citra Hasil proses konversi citra berwarna menjadi citra hitam putih .....	8
<b>Gambar 2.6</b>	Diagram alir <i>image recognition</i> .....	9
<b>Gambar 2.7</b>	Pemetaan pada proses vektor kuantisasi.....	12
<b>Gambar 2.7</b>	Contoh matriks transisi[7].....	13
<b>Gambar 3.1</b>	Diagram Alir Pembentukan Basis Data.....	17
<b>Gambar 3.2</b>	Hasil pemotongan citra retina.....	17
<b>Gambar 3.3</b>	Citra hitam putih dari retina .....	18
<b>Gambar 3.4</b>	<i>Workspace</i> untuk retina “ajib”.....	19
<b>Gambar 3.5</b>	Hasil Tampilan <i>Codebook</i> .....	21
<b>Gambar 3.6</b>	Probabilitas 1 Label Retina Dengan 10 Iterasi.....	23
<b>Gambar 3.7</b>	Diagram Alir Proses Pengenalan.....	24
<b>Gambar 4.1</b>	Tampilan dari program pengenalan retina.....	26
<b>Gambar 4.2</b>	Hubungan ukuran <i>codebook</i> dengan tingkat akurasi.....	34

## DAFTAR TABEL

<b>Tabel 4.1.1</b>	Hasil uji coba dengan <i>training</i> 4, ukuran <i>codebook</i> 32 .....	27
<b>Tabel 4.1.2</b>	Hasil uji coba dengan <i>training</i> 4, ukuran <i>codebook</i> 64.....	28
<b>Tabel 4.1.3</b>	Hasil uji coba dengan <i>training</i> 4, ukuran <i>codebook</i> 128.....	28
<b>Tabel 4.1.4</b>	Hasil uji coba dengan <i>training</i> 4, ukuran <i>codebook</i> 256.....	29
<b>Tabel 4.1.5</b>	Hasil uji coba dengan <i>training</i> 8, ukuran <i>codebook</i> 32.....	29
<b>Tabel 4.1.6</b>	Hasil uji coba dengan <i>training</i> 8, ukuran <i>codebook</i> 64.....	30
<b>Tabel 4.1.7</b>	Hasil uji coba dengan <i>training</i> 8, ukuran <i>codebook</i> 128.....	31
<b>Tabel 4.1.8</b>	Hasil uji coba dengan <i>training</i> 8, ukuran <i>codebook</i> 256.....	32
<b>Tabel 4.1.9</b>	Hasil Akurasi Keseluruhan Sistem.....	33
<b>Tabel 4.1.10</b>	Nilai LoP Dari 8 model retina milik Ajib untuk Jumlah training sebanyak 8 dengan ukuran <i>codebook</i> 128 dan 256.....	37

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Perkembangan teknologi pengolahan citra saat ini sudah mengalami kemajuan yang cukup signifikan seperti dalam perfilman, fotografi, pendidikan, teknik kedokteran, ilmu komputer, bahkan dalam sistem keamanan. Salah satu sistem pengamanan yang cukup mendapatkan perhatian adalah sistem pengenalan biometrik, yaitu sistem pengenalan yang memanfaatkan ciri khas pada diri manusia, seperti wajah, sidik jari, dan telapak tangan, retina maupun iris. Sistem seperti ini banyak digunakan karena yang digunakan sebagai pembeda adalah ciri yang khas dan unik untuk setiap orang.

Teknologi biometrik digunakan untuk keperluan identifikasi dan otentikasi. Untuk otentikasi, diperlukan akurasi yang tinggi untuk menjamin pengaksesan yang terbatas. Baik *retinal scans* maupun *iris scans* memberikan akurasi yang tinggi. Pengguna pertama *retinal scans* adalah militer dan agen pemerintah Amerika Serikat seperti CIA, FBI, dan NASA. Teknologi ini juga digunakan oleh Cook County Prison di Illinois untuk memastikan identitas para narapidana. Beberapa bank di Jepang menggunakan *retinal scans* di mesin ATM untuk mencegah pihak yang tak punya hak memasuki sistem mereka.

Retina merupakan bagian dari mata yang bertanggung jawab pada kemampuan melihat manusia. Pola dari pembuluh darah yang membentuk retina mata sama uniknya dengan sidik jari. Prinsip teknologi *retinal scanning* adalah memindai pola pembuluh darah kapiler pada retina dengan sumber cahaya intensitas rendah. *Retinal scanning* berdasar pada jaringan kapiler halus yang memenuhi kebutuhan oksigen dan nutrisi bagi retina. Pembuluh darah ini menyerap cahaya dan dengan mudah dapat divisualisasikan dengan penerangan yang tepat. Untuk itu, diperlukan jarak yang dekat antara mata dengan *scanner*, posisi mata terhadap *scanner* yang pas dan tidak ada pergerakan mata seperti berkedip atau melirik. Agar posisi mata pas, terdapat sebuah titik kecil berwarna hijau yang harus dilihat oleh mata sebagai titik acuan. Cahaya koheren intensitas

rendah ditransmisikan pada mata kemudian refleksi image pola pembuluh darah kapiler retina dicatat oleh komputer.

*Retinal scan* ini tidak bisa 100 persen akurat dan kurang cocok sebagai alat keamanan universal karena meski biasanya seumur hayat manusia pola pembuluh darah kapiler retinanya tidak berubah, namun penyakit diabetes, glaukoma, dan katarak mampu mengubahnya. Pada proses pemasukan data atau pendaftaran orang yang diberi hak akses, *image* kemudian dikonversi menjadi sebuah peta retina dan digunakan untuk mencocokkan.

Terdapat tiga metode deteksi pada pengenalan retina yaitu *Hidden Markov Model* (HMM), *fuzzy logic* atau metode jaringan syaraf tiruan (*Neural network*). Pada tugas akhir ini, digunakan metode *Hidden Markov*, dan metode lainnya dengan judul "PENGEMBANGAN IDENTIFIKASI RETINA DENGAN METODE UNFIS" oleh Nurul Hikmah [1] serta "PENGENALAN RETINA MENGGUNAKAN NEURAL NETWORK" oleh Ajib [2]. Teknik *fuzzy logic* adalah yang paling sederhana, hanya saja variasi kondisi pada tiap *image* retina sangat terbatas. Sedangkan teknik *Neural network* memerlukan proses pembelajaran dan iterasi yang sangat banyak dan panjang. Dengan teknik HMM, waktu proses akan jauh lebih cepat dari *Neural network* dan diharapkan hasil yang didapat akan lebih akurat. Dalam tugas akhir ini akan digunakan teknik HMM dalam proses pengenalan retina.

## **1.2 TUJUAN PENULISAN**

Mengenali retina masing-masing orang dengan *Hidden Markov Model* (HMM) dengan menerapkan prinsip-prinsip pengolahan citra pada proses pra-pengolahan dan ekstraksi fitur citra.

## **1.3 BATASAN MASALAH**

Pembatasan masalah pada penulisan ini adalah citra retina yang digunakan terbatas hanya pada retina itu sendiri. *Image* retina yang digunakan adalah image retina normal yang diperoleh dari hasil *scanning image* dari Rumah Sakit Cipto Mangunkusumo.

#### **1.4 SISTEMATIKA PENULISAN**

Di dalam bab I akan dijelaskan latar belakang dan tujuan penulisan, batasan masalah dan sistematika penulisan.

Di dalam bab II akan dijelaskan dasar teori mengenai *image recognition* dan *Hidden Markov Model* yang diaplikasikan dalam suatu Sistem Identifikasi Citra Biometrik.

Di dalam bab III akan dijelaskan langkah-langkah perancangan program identifikasi retina menggunakan metode *Hidden Markov Model* (HMM).

Di dalam bab IV akan dijelaskan hasil dari simulasi serta analisis dari hasil tersebut.

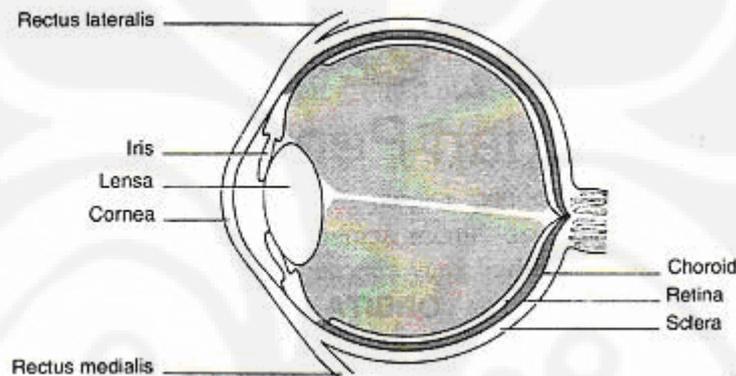
Di dalam bab V akan diberikan kesimpulan dari keseluruhan tugas akhir ini.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 RETINA MATA [3]

Mata merupakan indera penglihatan atau kamera alamiah yang merupakan jendela untuk melihat dunia, fokus pada cahaya dan mengantar gambar ke otak. Lalu otak memproses informasi itu menjadi gambar yang kita lihat. Secara sederhana mata terdiri dari bagian-bagian sebagai berikut:



Gbr 2.1. Potongan melalui mata dilihat dari atas

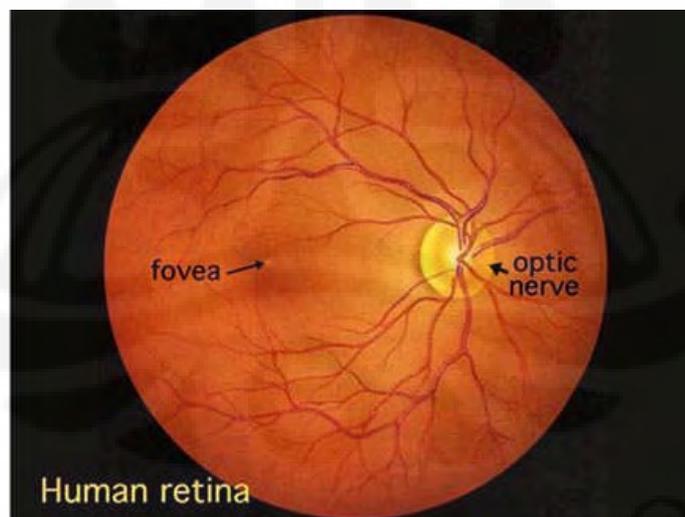
#### Bagian - bagian mata:

1. **Iris** : adalah diafragma yang melingkar dan berpigmen dengan lubang yang agak di tengah yakni pupil. Iris terletak sebagian dibagian depan lensa dan sebagian di depan badan siliaris. Iris terdiri dari serat otot polos. Fungsi iris yakni mengendalikan jumlah cahaya yang masuk.
2. **Kornea** : adalah kubah transparan yang agak pipih yang membentuk seperenam bagian anterior dinding. Kornea memiliki inervasi saraf tetapi *avaskuler* (tidak memiliki suplai darah).
3. **Lapisan koroid** : adalah lapisan berpigmen diantara *sclera* dan retina, lapisan koroid memiliki suplai darah dan membentuk lapisan vascular yang menyuplai mata sesuai dengan kebutuhannya.
4. **Retina** : merupakan bagian saraf pada mata, tersusun oleh sel saraf dan serat-seratnya dan merupakan membran transparan tipis yang melapisi

mata bagian belakang. Retina berperan sebagai reseptor rangsang cahaya, tersusun dari sel kerucut yang bertanggung jawab untuk penglihatan warna dan sel batang yang bertanggung jawab untuk penglihatan di tempat gelap. Hal ini mirip dengan film untuk kamera. Retina mengirimkan lebih dari satu juta serat syaraf melalui syaraf optik ke otak. Serat-serat syaraf itu mengirimkan signal ke bagian gambar pada otak yang memprosesnya menjadi gambar atau *image* visual.

Di tengah retina terdapat makula. Makula merupakan bagian terpenting dari retina. Ini merupakan bagian yang melihat apa saja yang berada di tengah penglihatan anda. Retina yang lainnya berperan untuk daerah-daerah yang tidak bisa dilihat secara jelas yaitu di bagian tepi penglihatan anda atau di sudut-sudut mata anda.

Cairan jernih yang disebut *aqueous* dihasilkan di dalam mata oleh sel berpigmen yang mengelilingi lensa (badan siliar). Cairan ini memberikan tekanan yang dibutuhkan untuk membantu mempertahankan bentuk dari mata. Ini sering disebut tekanan dengan "tekanan *intraokular*". Cairan ini juga memelihara kornea maupun lensa dengan memberikan oksigen dan nutrisi penting. Berikut merupakan gambar retina manusia normal.

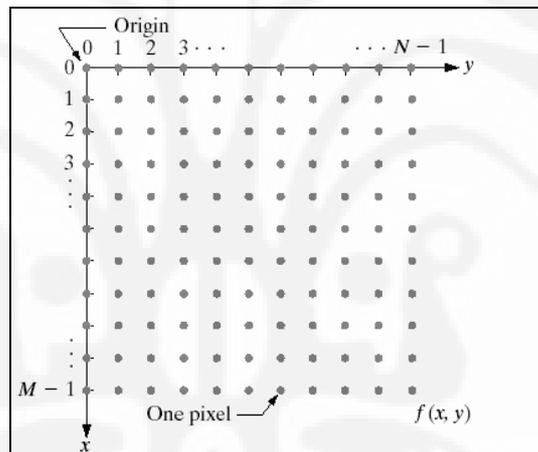


Gambar 2.2 Retina Normal

## 2.2 IMAGE PROCESSING [4]

Sebuah citra monokrom (*monochrome image*) atau citra saja (*image*) mengacu kepada fungsi intensitas dalam bidang dua dimensi yaitu  $f(x,y)$ . Fungsi

ini adalah representasi matematis dari citra, dimana  $f(x,y)$  adalah tingkat kecerahan (*brightness*) pada koordinat spasial  $(x,y)$ . Pemrosesan citra menggunakan komputer membutuhkan citra *digital* sebagai masukannya. Yang dimaksud dengan sebuah citra *digital* adalah sebuah citra dengan fungsi  $f(x,y)$  yang nilai kecerahan maupun posisi koordinatnya telah didiskritkan, sehingga nilainya bukan merupakan nilai yang tak berhingga, melainkan berada didalam jangkauan tertentu. Sebuah citra *digital* dapat direpresentasikan sebagai sebuah matriks, dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/piksel/*pixel*/*picture element*/pels) menyatakan tingkat kecerahan pada titik tersebut. Dibawah ini adalah Gambar 2.3 yang merupakan matriks dari citra digital.



Gambar 2.3. Matriks Citra Digital  $(x,y)$

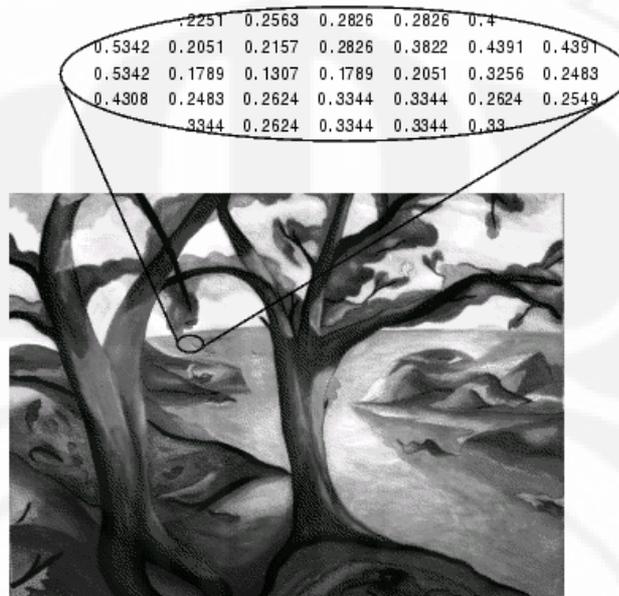
### 2.2.1 Sampling dan Gray Level Quantization [5]

Pada proses MATLAB, fungsi citra  $f(x,y)$  harus didigitasikan secara amplitudo (intensitas) maupun spasial. Proses digitasi dari kordinat spasial  $(x,y)$  disebut *Image sampling*, sedangkan dijitasi amplitudo disebut *gray level quantization*.

Jika suatu citra  $f(x,y)$  disampling dan menjadi  $N \times M$  array maka setiap elemen dari *array* merupakan kuantitas diskrit seperti dapat dilihat pada gambar 2.4. Setiap elemen dari array dapat disebut *image element* atau *picture element* atau *pixel*. Untuk sebuah *image digital*, setiap *pixel* memiliki nilai integer yakni *gray level* yang menunjukkan amplitudo atau intensitas dari *pixel* tersebut. Citra

merupakan fungsi dua dimensi yang kedua variabelnya yaitu nilai amplitudo dan koordinatnya merupakan nilai integer.

Gambar 2.4 memperlihatkan sebuah citra dengan *grayscale level 2* dimana pixel yang bernilai 0 merepresentasikan warna hitam sempurna dan 1 merepresentasikan warna putih sempurna.



Gambar 2.4 Citra dengan *grayscale level 2* ( nilai 0 sampai 1 )

## 2.2.2 Normalisasi Data

Data berupa image retina yang berasal dari hasil *scanner* harus dinormalisasi terlebih dahulu. Hal ini berguna untuk menyesuaikan data agar dapat diproses melalui berbagai proses selanjutnya dan untuk mendapatkan data yang baik. Berikut ini merupakan beberapa proses yang terjadi dalam normalisasi data:

### 2.2.2.1 Thresholding

Citra pada umumnya terdiri dari pixel-pixel dengan variasi nilai RGB (komponen warna merah/red, hijau/green, dan biru/blue). Citra jenis ini disebut sebagai citra berwarna. Citra abu-abu (*grayscale*) adalah citra dengan satu macam warna yakni abu-abu, dengan variasi derajat keabuan. Proses mengubah citra warna menjadi citra *grayscale* digunakan dalam *image processing* untuk menyederhanakan elemen matriks dari suatu citra. Citra berwarna terdiri dari 3 layer matriks yaitu R-layer, G-layer, B-layer. Untuk mengubah citra berwarna

menjadi cita abu-abu maka 3 layer tersebut diubah menjadi 1 layer matrik *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

*Thresholding* merupakan proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimilikinya. Piksel yang memiliki derajat keabuan lebih kecil dari nilai batas yang ditentukan akan diberikan nilai 0, sementara piksel yang memiliki derajat keabuan yang lebih besar dari batas akan diubah menjadi bernilai 1. Metode ini digunakan untuk mengkonversi data *image* menjadi data biner dengan tujuan agar proses selanjutnya menjadi lebih mudah. Sebagai contoh hasil konversi dari citra berwarna pada retina menjadi citra biner dapat dilihat pada Gambar 2.5.



(i) Citra Awal



(ii) Citra Akhir

Gambar 2.5. Citra Hasil proses konversi citra berwarna menjadi citra hitam putih

#### 2.2.2.2 *Resize dan Cropping*

Ukuran data yang berupa resolusi gambar  $N \times M$  terdiri dari kumpulan pixels dari hasil scanner memiliki ukuran yang berbeda-beda tergantung dari ukuran asli dari masing-masing retina dan spesifikasi alat scanner (dpi). Untuk itu diperlukan suatu perubahan terhadap ukuran tersebut sehingga ukuran seragam di setiap data. Inilah yang dinamakan *resize*. Metode yang digunakan dalam *resize* image adalah *bilinear interpolation* yang terdapat pada MATLAB.

Istilah *cropping* dalam hal ini dapat diartikan sebagai memotong suatu image pada suatu image pada daerah tertentu untuk diambil dan diolah. Tidak semua daerah dari image yang diolah sebagai data akan tetapi hanya daerah tertentu yang termasuk dalam lingkaran retina. Hal ini dilakukan untuk

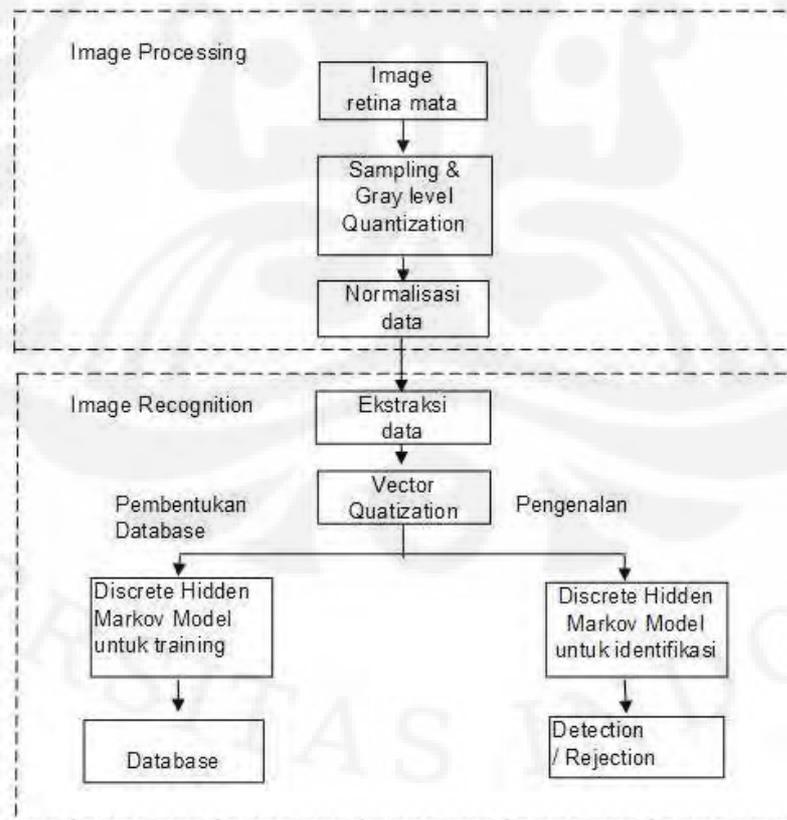
mendapatkan data yang tepat dan berukuran kecil sehingga memudahkan dalam proses komputasi data.

### 2.2.2.3 Leveling dan Reshaping

Leveling merupakan proses pemberian level nilai gray-level image dari suatu array atau matriks ke suatu nilai diskrit tertentu. Sedangkan *reshaping* adalah membentuk matriks dengan ukuran tertentu menjadi matriks dengan ukuran lain yang memiliki jumlah array yang sama, misalnya matriks  $N \times M$  menjadi matriks  $1 \times NM$ .

## 2.3 IMAGE RECOGNITION[4]

*Recognition* adalah suatu proses pemberian label kepada suatu objek berdasarkan informasi yang diberikan oleh seorang deskriptor. Secara umum *image recognition* adalah suatu bentuk pengenalan dari gambar/objek dalam hal ini objek 2 dimensi. Gambar berikut merupakan bagan proses pengenalan *image*:



Gambar 2.6 Diagram alir *Image Recognition*

### 2.3.1 Ekstraksi

Data yang telah dinormalisasi selanjutnya akan diekstraksi untuk mengubah data dalam domain frekuensi. Dalam proses ekstraksi terdapat proses yaitu:

#### 2.3.1.1 Frame Blocking

Sinyal kontinu diblok dalam frame dari N sampel, dengan frame berdekatan yang terpisah oleh M ( $M < N$ ). Frame pertama terdiri dari N sampel pertama. Frame kedua mulai M sampel setelah frame pertama dan saling menyusul dengan N-M sampel. Begitu pula dengan frame ketiga mulai sampel 2M setelah frame pertama (atau M sampel setelah frame kedua) dan menyusul N-2M sampel. Proses ini berlanjut sampai semua sinyal dihitung dalam satu frame atau lebih.

#### 2.3.1.2 Windowing

Langkah selanjutnya adalah windowing masing-masing individual frame untuk meminimalisasikan diskontinuitas sinyal pada permulaan dan akhir dari masing-masing frame. Windowing ini untuk meminimalisasikan *spectral distortion* dengan menggunakan *window* untuk mendapatkan batas sinyal ke nol pada permulaan dan akhir masing-masing *frame*. Jenis windowing yang digunakan adalah metode *Hamming*. Hasil windowing adalah sinyal yang dinyatakan dalam persamaan :

$$y_1(n) = x_1(n) w(n), 0 \leq n \leq N-1 \dots \dots \dots (2.1)$$

Persamaan *Hamming windowing* adalah :

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \dots \dots \dots (2.2)$$

#### 2.3.1.3 Discrete Fourier Transform

*Discrete Fourier Transform* (DFT) digunakan untuk mengubah *frame-frame* dari domain spasial ke domain frekuensi. Transformasi diskrit merupakan transformasi dimana input dan output bernilai diskrit yang digunakan untuk

manipulasi di komputer. Rumus DFT untuk mengubah N data dari domain spasial ke domain frekuensi adalah sebagai berikut :

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N} \dots\dots\dots(2.3)$$

dimana :  $F(u)$  : transformasi fourier

$f(x)$  : fungsi *image* dalam domain spasial

N : jumlah data

#### 2.3.1.4 Fast Fourier Transform

*Fast Fourier Transform* (FFT) merupakan algoritma yang lebih cepat dari *Discrete Fourier transform* (DFT). FFT dapat mereduksi jumlah perhitungan untuk setiap N data yang sama pada perhitungan DFT sehingga perhitungan yang ada menjadi lebih cepat khususnya ketika nilai N yang digunakan cukup besar menggunakan persamaan:

$$F(u) = \frac{1}{2} \left[ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{ux} W_{2M}^u \right] \dots\dots\dots(2.4)$$

$W_M^{ux}$  dapat dituliskan sebagai

$$W_M^{ux} = e^{-j2ux\pi/M} \dots\dots\dots(2.5)$$

$W_{2M}^u$  dapat dituliskan sebagai

$$W_{2M}^u = e^{-j\pi u/M} \dots\dots\dots(2.6)$$

Dimana :  $M = \frac{1}{2} N$

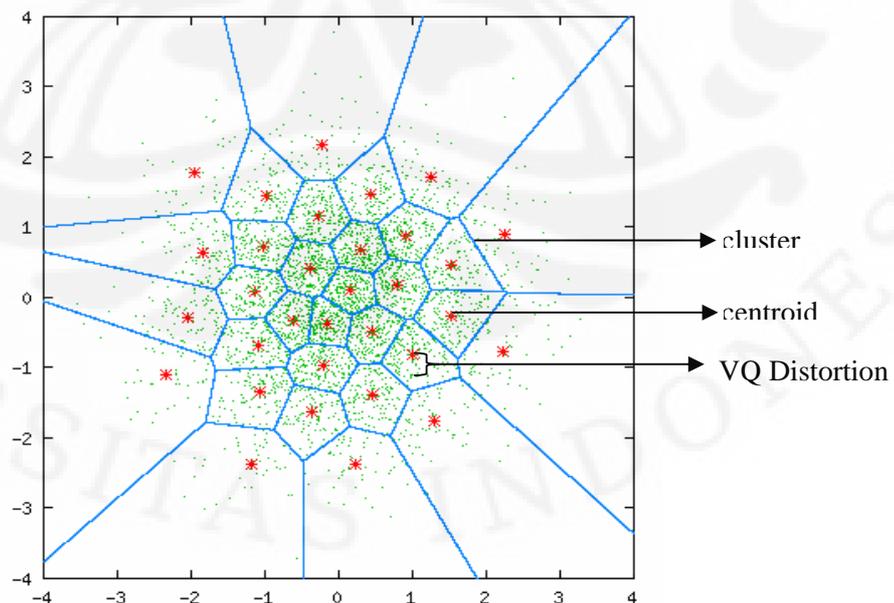
Baik DFT maupun FFT akan menghasilkan spektrum frekuensi berupa kumpulan titik-titik dimana masing-masing titik terdiri dari komponen real (fungsi *Cosinus*) dan komponen imajiner (fungsi *sinusoidal*) disebabkan adanya bilangan eksponensial. Kumpulan titik-titik ini kemudian akan digunakan dalam *Vector quantization* yang akan dijelaskan dalam sub bab berikut ini.

### 2.3.2 Vector Quantization

*Vector quantization* (VQ) adalah proses pemetaan vektor data yang merupakan titik-titik hasil dari proses FFT ke dalam sebuah wilayah yang terbatas dalam grafik dua dimensi (X-Y) dimana sumbu X merupakan komponen real dari masing-masing titik dan sumbu Y merupakan komponen imajiner dari masing-masing titik. Pemetaan titik-titik tersebut ditunjukkan oleh Gambar 2.6.

Tujuan dari proses vektor kuantisasi adalah untuk menyederhanakan panjang data masukan agar proses selanjutnya menjadi lebih mudah. Tiap komponen dari spektrum frekuensi yang merupakan hasil FFT memiliki beberapa titik yang masing-masing memiliki komponen real dan imajiner. Kumpulan dari titik-titik yang memiliki jarak berdekatan membentuk suatu *cluster* dan setiap *cluster* yang terbentuk dapat direpresentasikan dengan *centroid* yang disebut *codeword*. Koleksi dari semua *codeword* disebut *codebook*. Jarak antara satu titik dengan titik lain dalam sebuah *cluster* disebut *VQ Distortion*. Semakin kecil *VQ Distortion*-nya, maka *cluster* yang terbentuk menjadi lebih akurat.

Luas daerah *cluster* ditentukan oleh ukuran *codebook* dimana semakin besar ukuran *codebook*-nya, maka luas daerah masing-masing *cluster* menjadi lebih kecil dan jumlah *cluster* yang terbentuk menjadi lebih banyak disertai nilai *VQ distortion* yang semakin kecil sehingga *codeword* yang terbentuk akan semakin mewakili informasi dari masukannya.

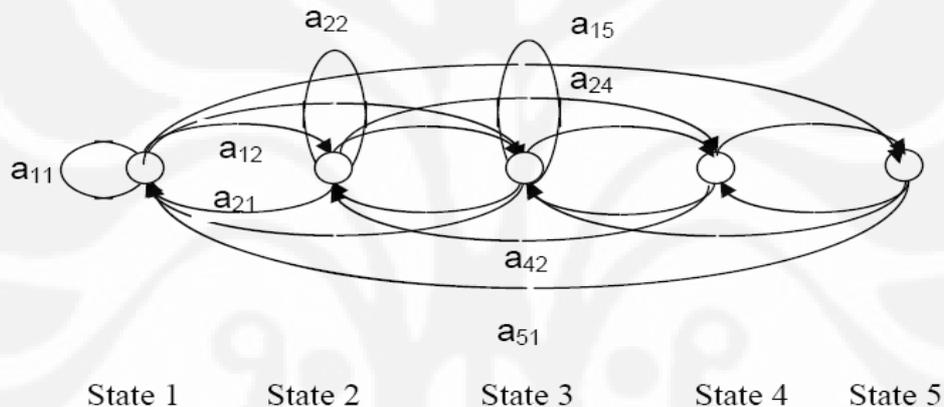


Gambar 2.7. Pemetaan pada proses vektor kuantisasi

### 2.3.3 Hidden Markov Model

*Hidden Markov Model* (HMM) adalah pemodelan probabilitas suatu sistem dengan mencari parameter-parameter yang tidak diketahui untuk mempermudah proses analisis sistem tersebut. HMM memiliki 3 parameter utama yang harus dicari nilainya terlebih dahulu. Ketiga parameter itu adalah sebagai berikut :

1. Parameter A : disebut sebagai probabilitas transisi, merupakan probabilitas kedudukan suatu *state* terhadap semua *state* yang ada, termasuk kedudukan terhadap *state* itu sendiri. Contoh dari matriks transisi dapat dilihat pada Gambar 2.7.



Gambar 2.8 : Contoh matriks transisi

Parameter A pada HMM dinyatakan dalam sebuah matriks dengan ukuran  $M \times M$  dengan  $M$  adalah jumlah *state* yang ada. Matriks transisi pada Gambar 2.6 terdiri dari 5 *state* sehingga setiap *state* memiliki 5 hubungan transisi, maka parameter A dapat dituliskan dalam bentuk matriks seperti pada persamaan 2.14.

$$A = a_{ij} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \dots\dots\dots(2.7)$$

2. Parameter B : disebut sebagai probabilitas *state*, merupakan probabilitas 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. Sebagai contoh, jika terdapat 5 buah *state* dalam suatu kondisi, maka matriks B yang terbentuk ditunjukkan oleh persamaan 2.15.

$$B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \dots\dots\dots(2.8)$$

3. Parameter  $\pi$  : disebut sebagai probabilitas awal, merupakan probabilitas kemunculan suatu *state* di awal.

Sama halnya dengan parameter B, parameter  $\pi$  juga dituliskan dalam bentuk matriks kolom dengan ukuran  $M \times 1$  dimana  $M$  adalah jumlah *statenya*. Jadi jika terdapat 5 *state*, maka parameter  $\pi$  yang dihasilkan akan ditunjukkan seperti pada persamaan 2.16.

$$\Pi = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \dots\dots\dots(2.9)$$

Dari ketiga parameter utama maka HMM dapat dituliskan dalam bentuk  $\lambda = (A, B, \pi)$ . Dari kesemua parameter yang ada maka bisa diperoleh suatu probabilitas observasi(O). Fungsi untuk probabilitas O ditunjukkan oleh persamaan 2.17

$$P(O) = \sum_{i=1}^N P(A_{ij}) * P(B_i) \dots\dots\dots(2.10)$$

Berikut adalah contoh perhitungan untuk mencari probabilitas observasi:

Citra 1  $\rightarrow (w_1, w_1, w_2, w_1, w_2) \rightarrow P(O) \text{ citra1} = c_1 * a_{11} * b_1 * a_{12} * b_2 * a_{21} * b_2 * a_{12} * b_1$

Citra 2  $\rightarrow (w_2, w_1, w_1, w_3, w_2) \rightarrow P(O) \text{ citra2} = c_2 * a_{21} * b_2 * a_{11} * b_1 * a_{13} * b_1$

$a_{32} * b_3$

Proses terjadinya nilai probabilitas HMM adalah sebagai berikut:

1. Data dibagi menjadi data-data kecil melalui proses *frame blocking* kemudian dicocokkan berdasarkan *codebook* yang dimiliki. Pada proses pencocokan dengan *codebook* akan dihitung jarak dari tiap data dengan *centroid-centroidnya*. Jarak yang paling dekat akan menentukan urutan kode observasi.
2. Data yang telah dikenali berdasarkan *codebook* akan dicocokkan dengan nilai pada parameter HMM. Parameter HMM sesuai dengan urutan kode observasi.

## **BAB III**

### **METODOLOGI**

Untuk menampilkan retina yang akan diidentifikasi atau dikenal (*recognized*) dilakukan dengan menggunakan teknik *Hidden Markov Model* (HMM). Sebelum proses pengenalan dilakukan, terlebih dahulu dibentuk basis data (*database*) untuk menyimpan *codeword* dan parameter HMM dari seluruh retina yang ada. Parameter - parameter di dalam basis data selanjutnya digunakan sebagai pembanding dalam proses pengenalan.

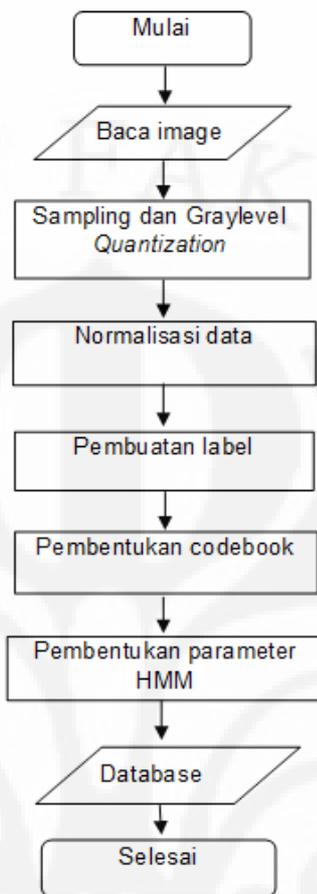
#### **3.1. PEMBENTUKAN BASIS DATA (DATABASE)**

Pada proses pembentukan basis data ini terdapat tiga tahapan, yakni tahap pelabelan, tahap pembuatan *codebook*, dan tahap pembentukan *Hidden Markov Model* (HMM).

Seluruh proses yang dilakukan dibuat menggunakan perangkat lunak untuk memecahkan masalah matematis dimana komputer yang digunakan memiliki spesifikasi sebagai berikut :

- a. Performa : Intel® Core™ 2 Duo T5250@ 1,5 GHz
- b. VGA Card : Realtek High Definition Audio

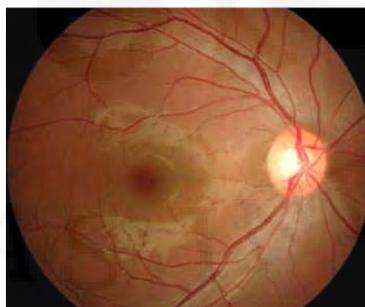
Gambar 3.1 memperlihatkan diagram alir pembentukan basis data.



Gambar 3.1 Diagram alir pembentukan basis data

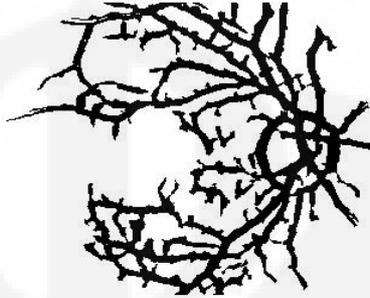
### 3.1.1. Proses Segmentasi

Tahap pertama dari pembentukan basis data retina adalah penentuan batas kanan dan kiri dari gambar retina tersebut yang dilakukan dengan cara memotong bagian hitam latar belakang retina. Dimana hasil pemotongan citra retina tersebut ditunjukkan oleh Gambar 3.2 di bawah ini:



Gambar 3.2 Hasil pemotongan citra retina

Setelah didapatkan hasil pemotongan citra retina, langkah berikutnya adalah ekstraksi citra retina yang terdiri dari proses segmentasi terhadap citra retina. Pada citra retina yang didapatkan dari hasil pemotretan pertama-tama diubah ke dalam citra hitam putih yang terdiri dari piksel 0 dan piksel 1 sebagai penyusun dari citra retina tersebut seperti ditunjukkan oleh Gambar 3.2 berikut:



Gambar 3.3 Citra hitam putih dari retina

Berikut ini adalah algoritma dari proses segmentasi yang dilakukan:

Mulai

Input *image*;

Hapus bagian kanan dan kiri retina sampai hanya bagian retina saja yang tersisa;

Image RGB diubah menjadi hitam putih;

Segmentasi retina;

Hapus bagian *noise* yang ikut tersegmentasi;

Lakukan proses normalisasi terhadap retina hasil segmentasi;

Selesai

### 3.1.2 Proses Pelabelan

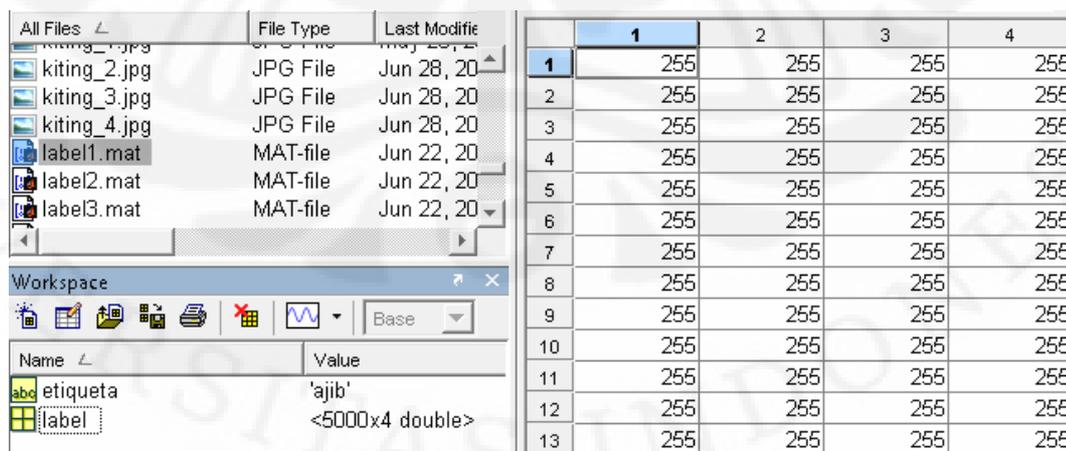
Pada proses pelabelan ini, masing-masing gambar retina yang akan didaftarkan pada *database* diberi label sesuai dengan pemilik retina tersebut. Sebagai contoh, retina “ajib” diberi label “ajib”, retina “denny” diberi label “denny” dan seterusnya, Nama Label inilah yang nantinya akan menjadi keluaran pada proses pengenalan retina.

Pada program pelabelan terdapat tiga masukan, yakni index label, jumlah training, dan nama label. Index label menentukan urutan retina pada keseluruhan

database yang ada. Jumlah training diisi dengan angka yang diinginkan dan nama label diisi sesuai dengan nama retina yang dimasukkan dalam database. Proses pelabelan ini dilakukan dengan menekan tombol *execute* pada tampilan program proses pelabelan. Tombol *execute* ini kemudian akan memanggil fungsi `make_labels(rep)` dalam *file* `labelisasi.m` dimana `rep` merupakan nilai dari jumlah *training* yang dimasukkan.

Untuk melakukan proses pelabelan dibutuhkan *file-file* gambar dari masing-masing retina yang telah *dipraprocessing* dalam format “\*.jpg” dimana setiap retina harus memiliki file sebanyak jumlah training. Misalnya, akan dibentuk label untuk retina “ajib” dengan jumlah training 4, maka harus tersedia file “ajib1.jpg” sampai file “ajib4.jpg”. Pada proses labelisasi, *image-image* dari tiap retina diubah ke dalam bentuk matriks  $N \times M$  (ukuran matriks  $N \times M$  akan sama dengan ukuran *image* yang dimasukkan) dengan fungsi matlab `imread` dan kemudian digunakan fungsi `reshape` untuk mengubah matriks  $N \times M$  tersebut menjadi matriks kolom.

Keluaran dari pelabelan ini adalah kumpulan matriks-matriks kolom dari tiap retina dengan jumlah kolom sebanyak jumlah *training*. Matriks-matriks ini akan disimpan dalam sebuah file dengan format “\*.mat” dan diberi nama “Label+index label” di dalam folder yang sama dengan `labelisasi.m`. Sebagai contoh, hasil proses pelabelan untuk Retina “ajib” diperlihatkan oleh Gambar 3.4.



The screenshot shows the MATLAB workspace with a file browser on the left and a data table on the right. The file browser lists files: kiting\_2.jpg, kiting\_3.jpg, kiting\_4.jpg, label1.mat, label2.mat, and label3.mat. The workspace table shows the following data:

	1	2	3	4
1	255	255	255	255
2	255	255	255	255
3	255	255	255	255
4	255	255	255	255
5	255	255	255	255
6	255	255	255	255
7	255	255	255	255
8	255	255	255	255
9	255	255	255	255
10	255	255	255	255
11	255	255	255	255
12	255	255	255	255
13	255	255	255	255

The workspace also shows variables: `etiqueta` with value 'ajib' and `label` with value <5000x4 double>.

Gambar 3.4 *Workspace* untuk Retina “ajib”

Pada Gambar 3.4 tampak empat deret kolom. Hal ini sesuai dengan jumlah *training* yang dilakukan pada proses pelabelan, dimana keempat *training* yang mengandung data dengan lima gambar Retina “ajib” yang sedikit berbeda antara satu dengan yang lain. Nilai 5000 pada baris menyatakan ukuran gambar retina yang dimasukkan dalam database yaitu 100 x 50.

Algoritma dari proses pelabelan retina adalah:

```
Mulai
Untuk I=1 sampai 5
Input nama retina;
Tulis nama retina;
Ubah retina menjadi bentuk matriks kolom;
Nama label retina[i] = nama retina;
Kembali
Selesai
```

### 3.1.3 Pembuatan *Codebook*

Setelah dilakukan proses pelabelan, maka langkah selanjutnya adalah melakukan proses penggabungan dari semua label yang telah dibuat ke dalam sebuah file *codebook* dengan format “\*.mat”. Pada pembuatan *codebook* ini juga dilakukan proses *vector quantization* yang telah dijelaskan dalam bab sebelumnya. Proses pembuatan *codebook* ini dilakukan pada file *codebook.m*.

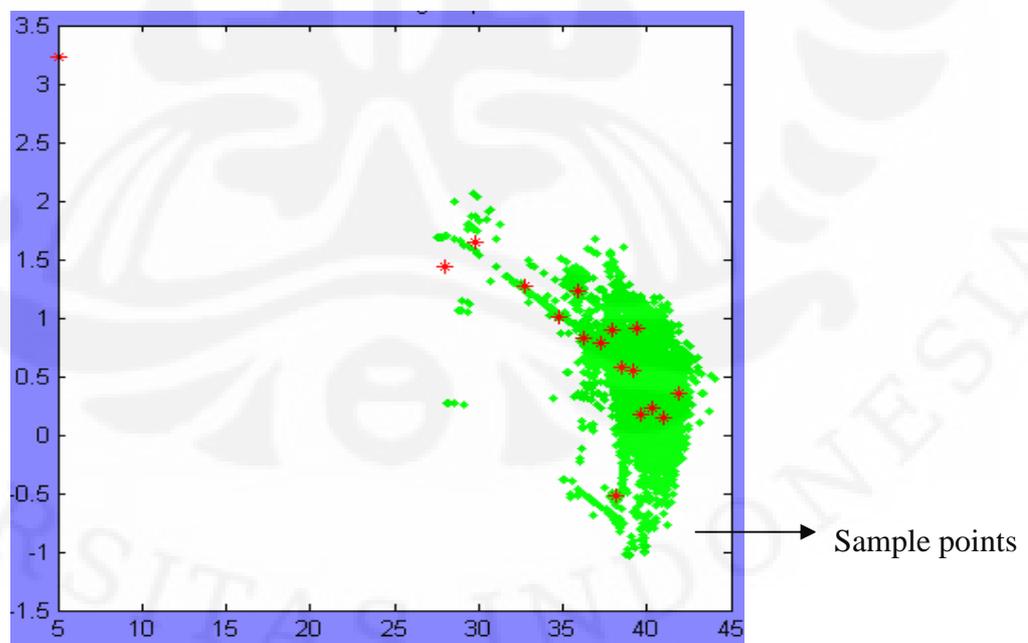
Dalam proses pembuatan *codebook* ini akan digunakan satu buah *codebook* untuk meningkatkan akurasi dari hasil pengenalan retina akan dicocokkan dalam basis data retina sehingga tidak akan terjadi kesalahan. Pada program ini terdapat empat buah masukan, yakni nama *file*, ukuran *codebook*, jumlah iterasi, dan jumlah label. Nama *file* dimasukkan sesuai dengan nama *file* yang diinginkan yang nantinya akan tersimpan dalam format “\*.mat” di dalam folder yang sama dengan *codebook.m*. Misalnya *codebook* ini diberi nama *file* “*codebook\_retina*”, maka hasil dari proses pembuatan *codebook* ini akan tersimpan dengan nama *file* “\**codebook\_retina.mat*”.

Ukuran *codebook* yang tersedia dalam program ini adalah 32, 64, dan 128, 256. Dimana keempat ukuran *codebook* ini akan dijadikan bahan perbandingan

pada tugas akhir ini untuk dilihat berapa nilai *codebook* yang paling sesuai pada proses pengenalan retina. Jumlah iterasi merupakan banyaknya proses pengulangan yang dilakukan dalam menentukan *centroid* guna mendapatkan *centroid* yang cukup presisi. Semakin besar jumlah iterasinya, maka akan semakin presisi letak *centroid* yang didapat, namun dengan mengambil iterasi yang sangat tinggi proses pembuatan *codebook* akan berjalan sangat lambat, oleh karena itu iterasi yang dilakukan juga tidak perlu teralu besar. Dalam ini ditentukan *default* untuk besarnya iterasi adalah 10 dengan harapan letak *centroid* yang diperoleh cukup presisi dan waktu proses relatif cepat.

Jumlah label dimasukkan sesuai dengan jumlah label yang telah dibuat sebelumnya pada proses pelabelan. Pada pembuatan *codebook* retina, jumlah label dimasukkan sebanyak 5 label karena terdapat 5 jenis retina yang dimasukkan dalam *database* retina.

Program *codebook.m* 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 3.6 yang merupakan tampilan *codebook* dari salah satu *training*:



Gambar 3. 5. Hasil tampilan *codebook*

Algoritma dari proses pembuatan *codebook* adalah 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

#### 3.1.4 Pembentukan Parameter HMM

Proses pembentukan HMM ini dilakukan dengan menggunakan program *hmm.m*. Pada program ini terdapat tiga masukan, yakni *file* *hmm*, *file* *codebook*, dan jumlah iterasi. *File* *hmm* dimasukan sesuai dengan keinginan dimana hasil dari proses parameter *hmm* akan disimpan dengan nama *file* tersebut dalam format “\*.mat” di dalam folder yang dengan sama *hmm.m*. Misalnya, akan dimasukkan *file* *hmm* dengan nama “hmm\_retina”, maka hasil dari proses pembentukan *hmm* ini akan disimpan dalam file “hmm\_retina.mat” di dalam folder untuk Retina. *File* *codebook* dimasukan sesuai dengan nama *file* *codebook* yang telah diisi sebelumnya pada proses pembuatan *codebook*. Disini akan dibuat *file* *hmm.m* yaitu pembentukan parameter HMM untuk retina karena alasan yang sudah dijelaskan dalam proses pembentukan *codebook*.

Jumlah iterasi merupakan banyaknya proses pengulangan yang dilakukan dalam mengolah parameter HMM guna mendapatkan probabilitas yang paling baik. Semakin besar jumlah iterasinya, maka akan semakin baik probabilitas yang didapat, namun terdapat suatu nilai iterasi dimana nilai probabilitas yang didapat akan konstan atau relatif sangat kecil perubahannya. Dengan demikian tidak perlu diambil nilai iterasi yang sangat tinggi karena dengan mengambil iterasi yang sangat tinggi proses akan berjalan sangat lambat. Sama halnya pada proses

pembuatan *codebook*, dalam pembentukan hmm juga ditentukan *default* untuk besarnya iterasi sebesar 10 dengan harapan probabilitas yang didapat bernilai tinggi dan waktu proses relatif cepat.

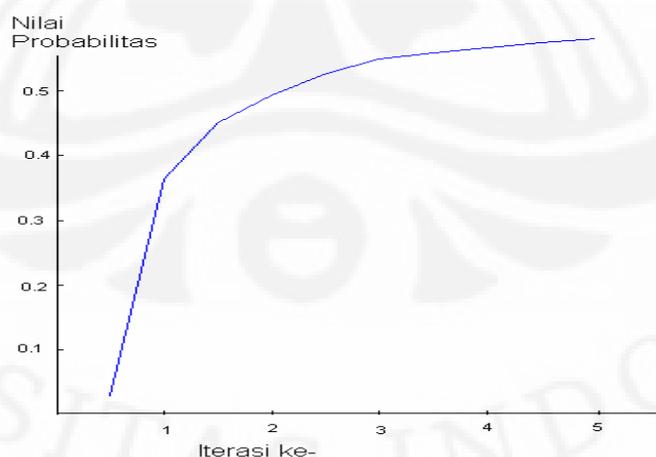
Algoritma dari proses pembentukan HMM yang bertujuan untuk memperoleh nilai parameter dari setiap label berdasarkan *file codebook* yang telah dibuat adalah sebagai berikut :

```

Mulai
Ambil file *.mat dari codebook yang telah
dibuat;
HMM Retina = file codebook retina;
Tentukan letak centroid dari tiap retina;
Inisialisasi matriks A, B dan  $\pi$  dengan nilai
acak;
Pelatihan dengan memasukkan semua data image
sampai nilai matriks tidak berubah;
Hitung probabilitas observasi HMM untuk setiap
image;
Simpan hasil dalam file format *.mat;
Selesai

```

Gambar 3.6 dibawah ini adalah tampilan grafik probabilitas dalam pembuatan model HMM :

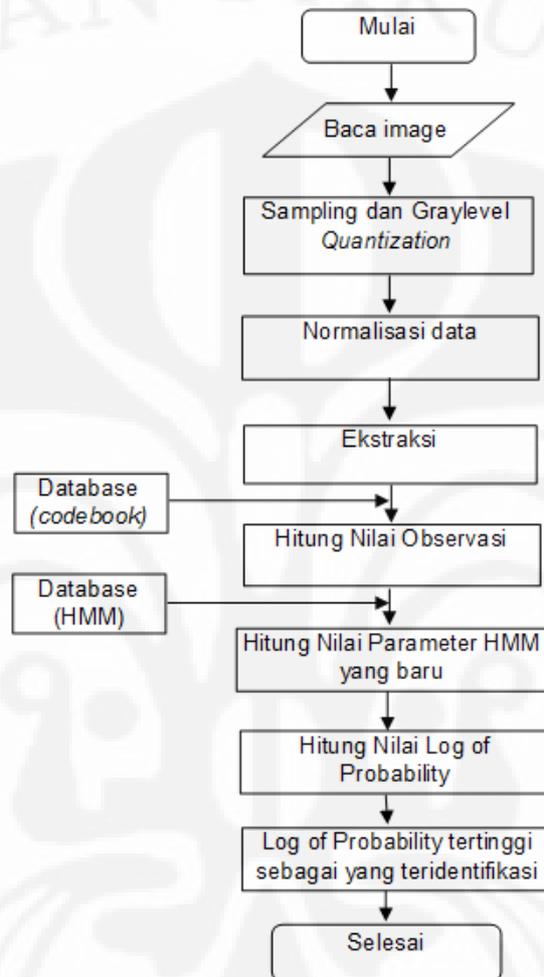


Gambar 3.6. Probabilitas 1 label Retina dengan 10 iterasi

### 3.2 PROSES PENGENALAN RETINA

Setelah basis data dibuat, maka proses pengenalan retina bisa dilakukan.

Gambar 3.7 adalah diagram alir dari proses pengenalan.



Gambar 3.7. Diagram alir proses pengenalan

Pada proses pengenalan, masukan berupa citra retina akan diubah kedalam citra biner terlebih dahulu. Setelah itu dilakukan proses segmentasi pada citra biner untuk mendapatkan bentuk kerangka dari setiap retina yang ada. Hasil dari proses segmentasi dan normalisasi tersebut kemudian diubah ke dalam domain frekuensi dengan menggunakan FFT. Spektrum frekuensi yang merupakan hasil dari FFT tersebut akan membentuk nilai vektor *real* dan *imaginer* yang akan dipetakan dalam *codebook* dalam bentuk *sample points*. Selanjutnya beberapa *sample point* yang terdekat dikuantisasikan ke satu titik vektor yang dinamakan

*centroid*. Letak dari *centroid* ini kemudian dicocokkan dengan letak *centroid* yang ada pada *codeword* dalam basis data untuk dicari nilai observasinya. Kemudian matriks dari nilai observasi yang didapat akan dicocokkan dengan matriks-matriks dari parameter-parameter HMM dalam basis data. Berdasarkan nilai parameter-parameter HMM tersebut maka dihitung besar *Log of Probability* (LoP) untuk semua retina yang akan dikenali.

Untuk retina akan didapat 5 buah nilai LoP per retina yang akan dikenali karena adanya 5 jenis retina pada basis data retina. Retina yang memiliki kemiripan paling banyak dengan retina dalam basis data akan memiliki nilai LoP yang paling besar sehingga retina yang memiliki nilai LoP terbesar dibandingkan 4 nilai LoP dari retina lainnya akan dianggap sebagai hasil pengenalan. Algoritma dari proses pengenalan dengan HMM dapat ditulis sebagai berikut:

Mulai

```
imread image retina.jpg;
```

```
Hapus bagian kanan dan kiri retina sampai hanya bagian retina saja yang tersisa;
```

```
Image RGB diubah menjadi hitam putih;
```

```
Hapus bagian noise yang ikut tersegmentasi;
```

```
Hitung FFT;
```

```
Retina = matriks codeword retina;
```

```
Cari centroid codeword di basis data;
```

```
Tentukan nilai observasi;
```

```
Tentukan parameter HMM;
```

```
untuk h = 1 sampai jml_label
```

```
    Baca data parameter HMM untuk retina  
    lainnya dari basis data;
```

```
    Hitung log of probability (LoP) untuk semua label  
    retina;
```

```
    LoP[jml_label] = LoP;
```

```
    Cari LoP[jumlah_label] = tertinggi;
```

```
    Tentukan nama label untuk LoP tertinggi;
```

Kembali

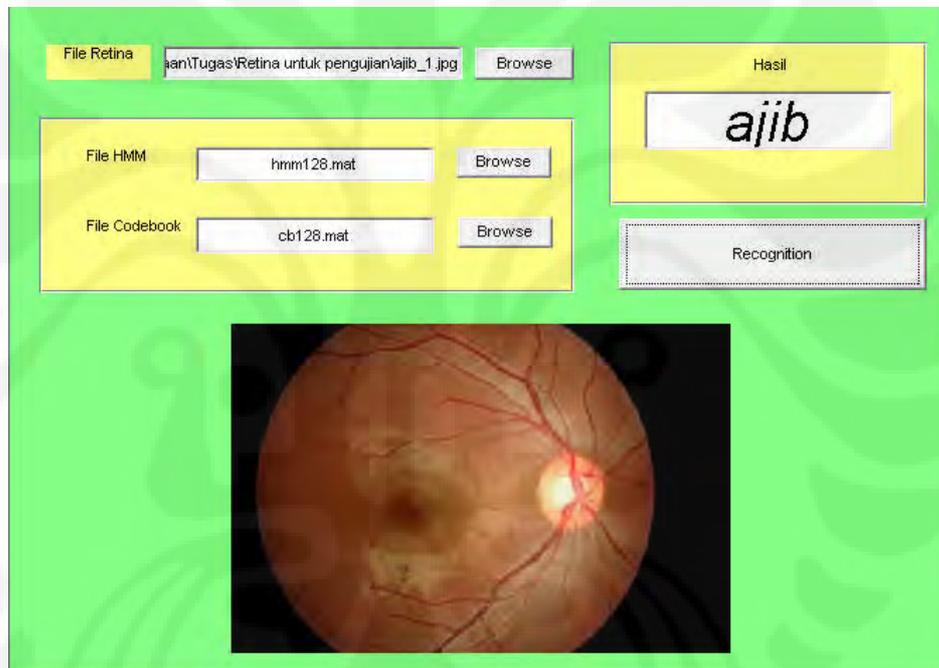
Selesai

## BAB IV

### HASIL UJI COBA DAN ANALISIS

#### 4.1 HASIL UJI COBA

Dalam uji coba pengenalan akan diuji coba 5 buah retina. *File - file* dari 5 buah citra retina tersebut kemudian akan dicoba pada program pengenalan yang telah dibuat dalam program Pengenalan Retina Menggunakan *Hidden Markov Model* untuk kemudian dibandingkan *log of probability* nya untuk dikenali. Gambar 4.1 berikut ini adalah tampilan dari program pengenalan retina.



Gambar 4.1. Tampilan dari program pengenalan retina

Citra retina dimasukkan dengan menekan tombol *browse* yang terletak di sebelah kanan "*File retina*". Setelah itu dimasukkan *file-file* HMM dan *codebook* melalui tombol *browse* yang telah disediakan. Kemudian ditekan tombol "*Recognition*" untuk memulai proses pengenalan retina dimana hasil dari proses pengenalan ini akan muncul pada textbox yang tersedia. Hasil dari pengenalan retina tersebut merupakan tampilan dari kumpulan retina dengan probabilitas terbesar dari keseluruhan retina yang ada dalam database.

Uji coba dilakukan dengan memvariasikan ukuran *codebook* dari retina yang kemudian dihitung tingkat keberhasilan dari tiap variasi yang dilakukan dan

model gambar dari tiap-tiap retina. Untuk training 4, maka model gambar dari retina milik Ajib misalnya, juga ada 4, sedangkan untuk training 8, maka model gambarnya juga ada 8. Model gambar menyatakan model yang berbeda dari retina tersebut dilihat dari perubahan sudut. Ujicoba yang dilakukan pada program ini adalah

1. Uji coba dengan *training* sebanyak 4 ukuran *codebook* 32.
2. Uji coba dengan *training* sebanyak 4 ukuran *codebook* 64.
3. Uji coba dengan *training* sebanyak 4 ukuran *codebook* 128.
4. Uji coba dengan *training* sebanyak 4 ukuran *codebook* 256.
5. Uji coba dengan *training* sebanyak 8 ukuran *codebook* 32.
6. Uji coba dengan *training* sebanyak 8 ukuran *codebook* 64.
7. Uji coba dengan *training* sebanyak 8 ukuran *codebook* 128.
8. Uji coba dengan *training* sebanyak 8 ukuran *codebook* 256.

#### 4.1.1 Hasil Uji Coba Dengan *Training* Sebanyak 4, Ukuran *Codebook* 32

Pada Tabel 4.1 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 5, ukuran *codebook* 32:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	32	1
2	Denny	Kiting	Tidak	32	1
3	Gerald	Kiting	Tidak	32	1
4	Kiting	Kiting	Terkenali	32	1
5	Nurul	Kiting	Tidak	32	1
6	Ajib	Kiting	Tidak	32	2
7	Denny	Kiting	Tidak	32	2
8	Gerald	Kiting	Tidak	32	2
9	Kiting	Kiting	Terkenali	32	2
10	Nurul	Kiting	Tidak	32	2
11	Ajib	Kiting	Tidak	32	3
12	Denny	Kiting	Tidak	32	3
13	Gerald	Kiting	Tidak	32	3
14	Kiting	Kiting	Tidak	32	3
15	Nurul	Kiting	Tidak	32	3
16	Ajib	Kiting	Tidak	32	4
17	Denny	Kiting	Tidak	32	4
18	Gerald	Kiting	Tidak	32	4
19	Kiting	Kiting	Terkenali	32	4
20	Nurul	Kiting	Tidak	32	4

Tabel 4.1 Hasil Uji Coba Dengan *Training* 4, Ukuran *Codebook* 32

#### 4.1.2 Hasil Uji Coba Dengan *Training* Sebanyak 4, Ukuran *Codebook* 64

Pada Tabel 4.2 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 5, ukuran *codebook* 64:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	64	1
2	Denny	Denny	Terkenali	64	1
3	Gerald	Gerald	Terkenali	64	1
4	Kiting	Ajib	Tidak	64	1
5	Nurul	Nurul	Terkenali	64	1
6	Ajib	Ajib	Terkenali	64	2
7	Denny	Denny	Terkenali	64	2
8	Gerald	Gerald	Terkenali	64	2
9	Kiting	Denny	Tidak	64	2
10	Nurul	Nurul	Terkenali	64	2
11	Ajib	Ajib	Terkenali	64	3
12	Denny	Denny	Terkenali	64	3
13	Gerald	Gerald	Terkenali	64	3
14	Kiting	Ajib	Tidak	64	3
15	Nurul	Nurul	Terkenali	64	3
16	Ajib	Ajib	Terkenali	64	4
17	Denny	Denny	Terkenali	64	4
18	Gerald	Gerald	Terkenali	64	4
19	Kiting	Denny	Tidak	64	4
20	Nurul	Nurul	Terkenali	64	4

Tabel 4.2 Hasil Uji Coba Dengan *Training* 4, Ukuran *Codebook* 64

#### 4.1.3 Hasil Uji Coba Dengan *Training* Sebanyak 4, Ukuran *Codebook* 128

Pada Tabel 4.3 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 5, ukuran *codebook* 128:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	128	1
2	Denny	Ajib	Tidak	128	1
3	Gerald	Gerald	Terkenali	128	1
4	Kiting	Kiting	Terkenali	128	1
5	Nurul	Kiting	Tidak	128	1
6	Ajib	Ajib	Terkenali	128	2
7	Denny	Nurul	Tidak	128	2
8	Gerald	Gerald	Terkenali	128	2
9	Kiting	Kiting	Terkenali	128	2
10	Nurul	Nurul	Terkenali	128	2
11	Ajib	Ajib	Terkenali	128	3
12	Denny	Ajib	Tidak	128	3

13	Gerald	Gerald	Terkenali	128	3
14	Kiting	Kiting	Terkenali	128	3
15	Nurul	Kiting	Tidak	128	3
16	Ajib	Ajib	Terkenali	128	4
17	Denny	Kiting	Tidak	128	4
18	Gerald	Gerald	Tidak	128	4
19	Kiting	Kiting	Terkenali	128	4
20	Nurul	Nurul	Terkenali	128	4

Tabel 4.3 Hasil Uji Coba Dengan *Training* 4, Ukuran *Codebook* 128

#### 4.1.4 Hasil Uji Coba Dengan *Training* Sebanyak 4, Ukuran *Codebook* 256

Pada Tabel 4.4 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 5, ukuran *codebook* 256:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	256	1
2	Denny	Denny	Terkenali	256	1
3	Gerald	Gerald	Terkenali	256	1
4	Kiting	Kiting	Terkenali	256	1
5	Nurul	Nurul	Terkenali	256	1
6	Ajib	Ajib	Terkenali	256	2
7	Denny	Denny	Terkenali	256	2
8	Gerald	Gerald	Terkenali	256	2
9	Kiting	Kiting	Terkenali	256	2
10	Nurul	Nurul	Terkenali	256	2
11	Ajib	Ajib	Terkenali	256	3
12	Denny	Denny	Terkenali	256	3
13	Gerald	Gerald	Terkenali	256	3
14	Kiting	Kiting	Terkenali	256	3
15	Nurul	Nurul	Terkenali	256	3
16	Ajib	Ajib	Terkenali	256	4
17	Denny	Denny	Terkenali	256	4
18	Gerald	Gerald	Terkenali	256	4
19	Kiting	Kiting	Terkenali	256	4
20	Nurul	Nurul	Terkenali	256	4

Tabel 4.4 Hasil Uji Coba Dengan *Training* 4, Ukuran *Codebook* 256

#### 4.1.5 Hasil Uji Coba Dengan *Training* Sebanyak 8, Ukuran *Codebook* 32

Pada Tabel 4.5 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 8, ukuran *codebook* 32:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Kiting	Tidak	32	1
2	Denny	Denny	Terkenali	32	1

3	Gerald	Kiting	Tidak	32	1
4	Kiting	Kiting	Terkenali	32	1
5	Nurul	Kiting	Tidak	32	1
6	Ajib	Kiting	Tidak	32	2
7	Denny	Denny	Terkenali	32	2
8	Gerald	Nurul	Tidak	32	2
9	Kiting	Kiting	Terkenali	32	2
10	Nurul	Nurul	Terkenali	32	2
11	Ajib	Kiting	Tidak	32	3
12	Denny	Denny	Terkenali	32	3
13	Gerald	Nurul	Tidak	32	3
14	Kiting	Kiting	Terkenali	32	3
15	Nurul	Kiting	Tidak	32	3
16	Ajib	Kiting	Tidak	32	4
17	Denny	Denny	Terkenali	32	4
18	Gerald	Kiting	Tidak	32	4
19	Kiting	Kiting	Terkenali	32	4
20	Nurul	Nurul	Terkenali	32	4
21	Ajib	Kiting	Tidak	32	5
22	Denny	Kiting	Terkenali	32	5
23	Gerald	Kiting	Tidak	32	5
24	Kiting	Kiting	Terkenali	32	5
25	Nurul	Kiting	Tidak	32	5
26	Ajib	Kiting	Tidak	32	6
27	Denny	Denny	Terkenali	32	6
28	Gerald	Nurul	Tidak	32	6
29	Kiting	Kiting	Terkenali	32	6
30	Nurul	Nurul	Terkenali	32	6
31	Ajib	Nurul	Tidak	32	7
32	Denny	Kiting	Tidak	32	7
33	Gerald	Kiting	Tidak	32	7
34	Kiting	Kiting	Terkenali	32	7
35	Nurul	Kiting	Tidak	32	7
36	Ajib	Nurul	Tidak	32	8
37	Denny	Denny	Terkenali	32	8
38	Gerald	Kiting	Tidak	32	8
39	Kiting	Kiting	Terkenali	32	8
40	Nurul	Nurul	Terkenali	32	8

Tabel 4.5 Hasil Uji Coba Dengan *Training* 8, Ukuran *Codebook* 32

#### 4.1.6 Hasil Uji Coba Dengan *Training* Sebanyak 8, Ukuran *Codebook* 64

Pada Tabel 4.6 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 8, ukuran *codebook* 64:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	64	1
2	Denny	Denny	Terkenali	64	1
3	Gerald	Gerald	Terkenali	64	1

4	Kiting	Kiting	Terkenali	64	1
5	Nurul	Nurul	Terkenali	64	1
6	Ajib	Ajib	Terkenali	64	2
7	Denny	Denny	Terkenali	64	2
8	Gerald	Gerald	Terkenali	64	2
9	Kiting	Kiting	Terkenali	64	2
10	Nurul	Nurul	Terkenali	64	2
11	Ajib	Ajib	Terkenali	64	3
12	Denny	Denny	Terkenali	64	3
13	Gerald	Gerald	Terkenali	64	3
14	Kiting	Kiting	Terkenali	64	3
15	Nurul	Nurul	Terkenali	64	3
16	Ajib	Ajib	Terkenali	64	4
17	Denny	Denny	Terkenali	64	4
18	Gerald	Gerald	Terkenali	64	4
19	Kiting	Kiting	Terkenali	64	4
20	Nurul	Nurul	Terkenali	64	4
21	Ajib	Ajib	Terkenali	64	5
22	Denny	Denny	Terkenali	64	5
23	Gerald	Gerald	Terkenali	64	5
24	Kiting	Kiting	Terkenali	64	5
25	Nurul	Nurul	Terkenali	64	5
26	Ajib	Ajib	Terkenali	64	6
27	Denny	Denny	Terkenali	64	6
28	Gerald	Gerald	Terkenali	64	6
29	Kiting	Kiting	Terkenali	64	6
30	Nurul	Nurul	Terkenali	64	6
31	Ajib	Ajib	Terkenali	64	7
32	Denny	Denny	Terkenali	64	7
33	Gerald	Gerald	Terkenali	64	7
34	Kiting	Kiting	Terkenali	64	7
35	Nurul	Nurul	Terkenali	64	7
36	Ajib	Ajib	Terkenali	64	8
37	Denny	Denny	Terkenali	64	8
38	Gerald	Gerald	Terkenali	64	8
39	Kiting	Kiting	Terkenali	64	8
40	Nurul	Nurul	Terkenali	64	8

Tabel 4.6 Hasil Uji Coba Dengan *Training* 8, Ukuran *Codebook* 64

#### 4.1.7 Hasil Uji Coba Dengan *Training* Sebanyak 8, Ukuran *Codebook* 128

Pada Tabel 4.7 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 8, ukuran *codebook* 128:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	128	1
2	Denny	Denny	Terkenali	128	1
3	Gerald	Gerald	Terkenali	128	1
4	Kiting	Ajib	Tidak	128	1

5	Nurul	Denny	Tidak	128	1
6	Ajib	Ajib	Terkenali	128	2
7	Denny	Kiting	Tidak	128	2
8	Gerald	Gerald	Terkenali	128	2
9	Kiting	Kiting	Terkenali	128	2
10	Nurul	Kiting	Tidak	128	2
11	Ajib	Ajib	Terkenali	128	3
12	Denny	Denny	Terkenali	128	3
13	Gerald	Gerald	Terkenali	128	3
14	Kiting	Kiting	Terkenali	128	3
15	Nurul	Kiting	Tidak	128	3
16	Ajib	Ajib	Terkenali	128	4
17	Denny	Denny	Terkenali	128	4
18	Gerald	Gerald	Terkenali	128	4
19	Kiting	Kiting	Terkenali	128	4
20	Nurul	Kiting	Tidak	128	4
21	Ajib	Ajib	Terkenali	128	5
22	Denny	Denny	Terkenali	128	5
23	Gerald	Gerald	Terkenali	128	5
24	Kiting	Ajib	Tidak	128	5
25	Nurul	Ajib	Tidak	128	5
26	Ajib	Ajib	Terkenali	128	6
27	Denny	Kiting	Tidak	128	6
28	Gerald	Gerald	Terkenali	128	6
29	Kiting	Kiting	Terkenali	128	6
30	Nurul	Kiting	Tidak	128	6
31	Ajib	Ajib	Terkenali	128	7
32	Denny	Denny	Terkenali	128	7
33	Gerald	Gerald	Terkenali	128	7
34	Kiting	Kiting	Terkenali	128	7
35	Nurul	Kiting	Tidak	128	7
36	Ajib	Ajib	Terkenali	128	8
37	Denny	Denny	Terkenali	128	8
38	Gerald	Gerald	Terkenali	128	8
39	Kiting	Kiting	Terkenali	128	8
40	Nurul	Kiting	Tidak	128	8

Tabel 4.7 Hasil Uji Coba Dengan *Training* 8, Ukuran *Codebook* 128

#### 4.1.8 Hasil Uji Coba Dengan *Training* Sebanyak 8, Ukuran *Codebook* 256

Pada Tabel 4.8 dapat dilihat hasil uji coba program dengan *training* retina sebanyak 8, ukuran *codebook* 256:

No	Retina	Hasil pengenalan	Terkenali/Tidak	Codebook	Model gambar
1	Ajib	Ajib	Terkenali	256	1
2	Denny	Denny	Terkenali	256	1
3	Gerald	Gerald	Terkenali	256	1
4	Kiting	Kiting	Terkenali	256	1
5	Nurul	Nurul	Terkenali	256	1

6	Ajib	Kiting	Tidak	256	2
7	Denny	Denny	Terkenali	256	2
8	Gerald	Gerald	Terkenali	256	2
9	Kiting	Kiting	Terkenali	256	2
10	Nurul	Nurul	Terkenali	256	2
11	Ajib	Ajib	Terkenali	256	3
12	Denny	Denny	Terkenali	256	3
13	Gerald	Gerald	Terkenali	256	3
14	Kiting	Kiting	Terkenali	256	3
15	Nurul	Nurul	Terkenali	256	3
16	Ajib	Ajib	Terkenali	256	4
17	Denny	Denny	Terkenali	256	4
18	Gerald	Gerald	Terkenali	256	4
19	Kiting	Kiting	Terkenali	256	4
20	Nurul	Nurul	Terkenali	256	4
21	Ajib	Ajib	Terkenali	256	5
22	Denny	Denny	Terkenali	256	5
23	Gerald	Kiting	Terkenali	256	5
24	Kiting	Kiting	Terkenali	256	5
25	Nurul	Nurul	Terkenali	256	5
26	Ajib	Ajib	Terkenali	256	6
27	Denny	Denny	Terkenali	256	6
28	Gerald	Gerald	Terkenali	256	6
29	Kiting	Kiting	Terkenali	256	6
30	Nurul	Nurul	Terkenali	256	6
31	Ajib	Kiting	Tidak	256	7
32	Denny	Denny	Terkenali	256	7
33	Gerald	Gerald	Terkenali	256	7
34	Kiting	Kiting	Terkenali	256	7
35	Nurul	Nurul	Terkenali	256	7
36	Ajib	Ajib	Terkenali	256	8
37	Denny	Denny	Terkenali	256	8
38	Gerald	Gerald	Terkenali	256	8
39	Kiting	Kiting	Terkenali	256	8
40	Nurul	Nurul	Terkenali	256	8

Tabel 4.8 Hasil Uji Coba Dengan *Training* 8, Ukuran *Codebook* 256

#### 4.2 PENGOLAHAN HASIL UJI COBA

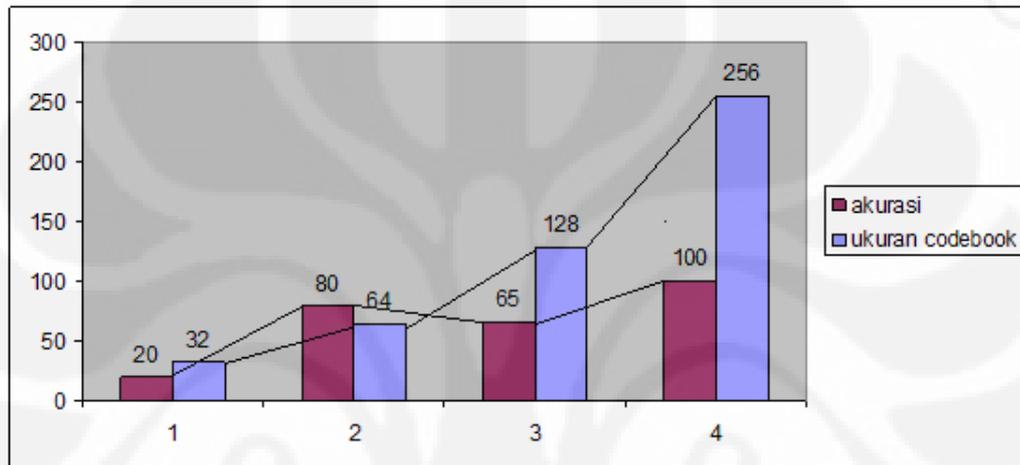
Dari hasil yang didapatkan dari uji coba dapat dihitung persen akurasi keseluruhan dari sistem. Persen akurasi didapat dengan cara membagi jumlah retina yang berhasil dikenali secara sempurna dengan jumlah seluruh *sample* retina yang digunakan.

Jumlah Training	Codebook	Akurasi (%)
4	32	20
4	64	80

4	128	65
4	256	100
8	32	47,5
8	64	100
8	128	70
8	256	95

Tabel 4.9 Hasil Akurasi Keseluruhan Sistem

Hubungan ukuran *codebook* dengan tingkat akurasi pada identifikasi nilai nominal dengan jumlah training 4 ditunjukkan gambar 4.2 berikut:



Gambar 4.2 Hubungan ukuran *codebook* dengan tingkat akurasi

### 4.3. ANALISIS

#### 4.3.1. Pengaruh Ukuran *Codebook* Terhadap Tingkat Akurasi Proses Pengenalan Retina

Pengaruh ukuran *codebook* terhadap tingkat keberhasilan proses pengenalan retina dapat dilihat dari variasi jumlah *codebook*. Apabila jumlah training dibuat tetap dan ukuran *codebook* diubah-ubah dari 32 sampai 256 maka akan didapat peningkatan besarnya tingkat akurasi seperti yang dapat dilihat pada tabel 4.9.

Peningkatan besarnya persen akurasi ini disebabkan karena ukuran *codebook* yang lebih besar membuat jumlah *codeword* (*centroid*) semakin banyak. Banyaknya *centroid* ini membuat proses kuantisasi pemilihan nilai vektor data semakin teliti, sehingga pemetaan terhadap vektor data dapat dilakukan dengan

jarak yang lebih kecil. Dengan kata lain, distorsi VQ (jarak antara sebuah vektor data dengan *codeword* terdekat) pada akhir iterasi akan semakin kecil.

Meskipun peningkatan ukuran *codebook* dapat meningkatkan persen akurasi secara keseluruhan namun bukan berarti retina yang telah dikenali pada ukuran *codebook* yang lebih kecil akan dikenali juga pada ukuran *codebook* yang lebih besar. Hal ini dapat dilihat pada beberapa retina seperti retina Denny pada model 1 yang telah dikenali pada *codebook* 64, namun menjadi tidak dikenali secara sempurna pada *codebook* 128. Contoh lainnya dapat dilihat pada retina Gerald model 4 yang dikenali pada *codebook* 64 dan menjadi tidak dikenali pada *codebook* 128. Menurunnya tingkat akurasi untuk beberapa retina ini ketika ukuran *codebook* meningkat tidak mempengaruhi tingkat akurasi untuk keseluruhan sistem sebab tingkat akurasi untuk keseluruhan sistem tetap meningkat.

Penurunan akurasi hasil pengenalan dari beberapa retina ini disebabkan kemiripan karakteristik elemen matriks sehingga *centroid -centroid* dari keseluruhan sistem menjadi sangat berdekatan yang menyebabkan kesulitan dalam proses pengenalan. Faktor penyebab lainnya mungkin juga disebabkan oleh proses kuantisasi yang dilakukan pada pembentukan *codebook* sehingga ada beberapa data dari citra asli yang terbuang yang menyebabkan elemen matriks baik dalam proses pembuatan basis data maupun pada proses pengenalan tidak sepenuhnya mewakili elemen matriks dari citra aslinya.

#### **4.3.2. Pengaruh jumlah *Training* Terhadap Tingkat Akurasi Proses Pengenalan retina**

Berdasarkan Tabel 4.9 dapat dilihat bahwa untuk ukuran *codebook* yang sama, peningkatan jumlah training mengakibatkan peningkatan besarnya persen akurasi pada keseluruhan sistem.

Pada proses pembuatan basis data dimasukkan citra atau gambar dari masing retina sebagai pembanding. Banyaknya data untuk masing-masing retina ini dinyatakan dengan jumlah training. Sebagai contoh, jika jumlah training retina adalah 8, maka masing-masing retina memiliki 8 buah data pembanding yang berbeda-beda. Semakin banyak gambar retina yang dimasukkan dalam basis data,

maka proses pengenalan akan lebih akurat karena dengan semakin banyaknya basis data untuk masing-masing retina, masalah kemiripan antar retina akan semakin berkurang.

Dari Tabel 4.9 dapat dilihat untuk ujicoba dengan *training* sebanyak 4, ukuran *codebook* 32, tingkat akurasi yang didapat adalah sebesar 20%. Sedangkan untuk ujicoba dengan *training* 8, ukuran *codebook* 32, tingkat akurasi yang didapat adalah sebesar 47,5%, yang berarti terjadi peningkatan akurasi sebesar 2,375 kali lipat dengan penambahan jumlah training sebanyak 4 buah. Untuk ukuran *codebook* 64 untuk ujicoba training 8 terhadap ujicoba training 4 peningkatan akurasi sebesar 1,25 kali lipat, sedangkan untuk ukuran *codebook* 128 akurasinya sebesar 1,07 kali, dan ukuran *codebook* 256 akurasinya sebesar 0,95 kali.

Dengan penambahan ukuran *codebook* peningkatan maksimum yang terjadi hanyalah berkisar antara 0,7 kali (yang didapat dari perbandingan uji coba *training* sebanyak 8, ukuran *codebook* 128 dengan uji coba *training* sebanyak 8, ukuran *codebook* 64) sampai 5 kali (yang didapat dari perbandingan uji coba *training* sebanyak 4, ukuran *codebook* 256 dengan uji coba *training* sebanyak 4, ukuran *codebook* 32). Dari perbandingan hasil peningkatan tingkat akurasi antara penambahan jumlah training dengan penambahan ukuran *codebook* maka dapat dikatakan bahwa peningkatan tingkat akurasi dengan menambah jumlah training lebih baik dibandingkan dengan peningkatan tingkat akurasi ketika ukuran *codebook* ditambah.

Namun peningkatan jumlah training jika dilihat dari tiap retina belum tentu meningkatkan besar tingkat akurasi. Sama seperti pada peningkatan ukuran *codebook*, ada beberapa retina yang tingkat akurasinya menurun seperti pada hasil pengenalan retina milik Ajib pada model 2 dan model 7 yang dapat dikenali sempurna pada hasil uji coba dengan *training* 8, ukuran *codebook* 128 namun menjadi tidak dikenali secara sempurna pada uji coba dengan *training* 8 ukuran *codebook* 256, dimana hasil pengenalannya menjadi retina milik Kiting.

Kesalahan hasil pengenalan ini bisa terjadi karena penambahan jumlah training berarti menambah data untuk membuat label dalam basis data. Ketika jumlah *training* bertambah maka masing-masing label akan mempunyai data baru

terhadap keseluruhan elemen matriks dari citra retina yang direpetisi yang berbeda dengan kondisi sebelumnya. Data baru ini akan mempengaruhi proses pembentukan *codebook* dan parameter HMM sehingga perhitungan untuk mendapatkan nilai LoP juga berubah yang menyebabkan hasil pengenalan yang didapat dengan mencari nilai LoP yang tertinggi juga menjadi berubah. Tetapi menurunnya tingkat akurasi untuk beberapa retina ini tidak mempengaruhi kemampuan keseluruhan sistem sebab tingkat akurasi untuk keseluruhan sistem tetap meningkat. Tabel 4.10 dibawah ini adalah nilai-nilai LoP dari seluruh jenis model untuk hasil pengenalan retina milik Ajib dengan ukuran *codebook* yang 256.

Model	Log Of Probability	
	Codebook=128	Codebook=256
1	-551,6	-831,7
2	-441,6301	-850,5727
3	-550,5304	-809,5
4	-429,2142	-843,1541
5	-542,6453	-776,6
6	-542,0667	-758
7	-425,1578	-837,3194
8	-507,6941	-834,5670

Tabel 4.10 Nilai LoP Dari 8 Model Retina Milik Ajib Untuk Jumlah *Training* Sebanyak 8 Dengan Ukuran *Codebook* 128 dan 256

Dari Tabel 4.10 diatas dapat dilihat bahwa nilai LoP untuk model 2 berubah dari -441,6301 pada *codebook* 128 menjadi -850,5727 pada *codebook* 256, nilai LoP untuk model 6 berubah dari -425,1578 pada *codebook* 128 menjadi -837,3194 pada *codebook* 256 sehingga sistem salah mengenali dan yang dikenali menjadi kiting untuk model 2 dan 7.

## **BAB V**

### **KESIMPULAN**

1. Pengenalan retina dengan metode HMM memiliki tingkat akurasi hingga 100% pada uji coba dengan jumlah *training* sebanyak 4 dan ukuran *codebook* 256, dan dengan jumlah *training* sebanyak 8 dan ukuran *codebook* 64.
2. Penambahan ukuran *codebook* meningkatkan persen akurasi sebesar 0,7 - 5 kali lipat. Sedangkan penambahan jumlah *training* meningkatkan persen akurasi sebesar 0,95 - 2.375 kali lipat.
3. Faktor yang paling signifikan dalam meningkatkan akurasi adalah besarnya *training* dan *codebook*.

## DAFTAR ACUAN

- [1] Nurul Hikmah. " Pengembangan Identifikasi Retina dengan Metode Unfis". Skripsi. Program Sarjana Fakultas Teknik Universitas Indonesia. 2008.
- [2] Ajib. " Pengenalan Retina dengan Neural Network". Seminar. Program Sarjana Fakultas Teknik Universitas Indonesia. 2008.
- [3] [www.indofarma.com/Newsroom/Health Info/Mata](http://www.indofarma.com/Newsroom/Health Info/Mata)
- [4] Ferry Sugiarto. " Sistem Pengenalan Plat Mobil Dengan Skletonisasi dan Hidden Markov Model ". Skripsi. Program Sarjana Fakultas Teknik Universitas Indonesia. 2007.
- [5] Robby Nelson. " Sistem Pengenalan Plat Mobil Menggunakan Hidden Markov Model ". Skripsi. Program Sarjana Fakultas Teknik Universitas Indonesia. 2007.
- [6] Ridzky Arya Pradana. "Perangkat Lunak Identifikasi Nilai Nominal Dan Keaslian Mata Uang Kertas Rupiah Dengan Proses Image Recognition Menggunakan Hidden Markov Model". Skripsi. Program Sarjana Fakultas Teknik Universitas Indonesia. 2007.
- [7] Arman Djohan Diponegoro, "Penentuan jenis ikan dengan menggunakan hidden markov model dari pendeteksian fase penerimaan sinyal akustik", disertasi, Program Studi Teknologi Kelautan Program Pasca Sarjana Institut Pertanian Bogor, 2006.
- [8] Gonzalez, Rafael, Woods, Richard, " *Digital Image Processing*", Addison Wesley Publishing co., USA, 1992.

[9] Thitiporn Chanwimaluang dan Guoliang Fan. Corrections to Hybrid Retinal Image Registration. Journal. 2006.

[10] Adam Hoover, Threshold Probing to Segment Blood Vessels, 2000

[12] Thitiporn Chanwimaluang dan Guoliang Fan, Blood Vessel Extraction



## DAFTAR PUSTAKA

[www.pikiran-rakyat.com/Teknologi Identifikasi Biometrik/](http://www.pikiran-rakyat.com/Teknologi%20Identifikasi%20Biometrik/)

Gonzalez, Rafael, Woods, Richard, “ *Digital Image Processing Using MATLAB*”,  
Prentice Hall , New Jersey, 2004

Hasudungan, Michael, ”Identifikasi species ikan dengan analisis echogram *fish finder*  
menggunakan *Hidden Markov Model*.” Skripsi, Departemen Teknik Elektro,  
Fakultas Teknik, Universitas Indonesia, 2006.