



GABUNGAN METODE FILTER LOG-GABOR DAN PCA UNTUK PENGENALAN INDIVIDU BERBASIS CITRA IRIS MATA

**TESIS
MK-TESIS-SM 142501**

**M. RIZKY FAUNDRA
1214 201 021**

**DOSEN PEMBIMBING
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.**

**PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



COMBINATION OF LOG-GABOR FILTER AND PCA METHOD IN INDIVIDUAL RECOGNITION BASED ON EYE IRIS

**THESIS
MK-THESIS-SM 142501**

**M. RIZKY FAUNDRA
1214 201 021**

**SUPERVISOR
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.**

**MASTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017**

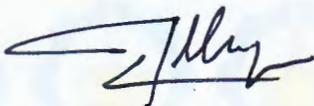
GABUNGAN METODE FILTER LOG-GABOR DAN PCA UNTUK PENGENALAN INDIVIDU BERBASIS CITRA IRIS MATA

Tesis ini disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Magister Sains (M.Si.)
di
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
M. RIZKY FAUNDRA
NRP. 1214201021

Tanggal Ujian : 9 Januari 2017
Periode Wisuda : Maret 2017

Disetujui oleh:



Dr. Dwi Ratna Sulistyoningrum, S.Si., M.T.
NIP: 19690405 199403 2 003

(Pembimbing)



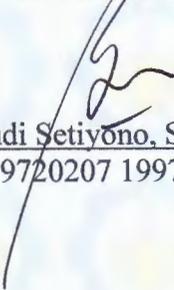
Dr. Chairul Imron, Ml.Komp.
NIP: 19611115 198703 1 003

(Penguji I)



Prof. Dr. Mohammad Isa Irawan, M.T.
NIP: 19631225 198903 1 001

(Penguji II)



Dr. Budi Setiyono, S.Si., M.T.
NIP: 19720207 199702 1 001

(Penguji III)



Direktur Program Pascasarjana
Asisten Direktur

Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP: 19611021 198603 1 001

GABUNGAN METODE FILTER LOG-GABOR DAN PCA UNTUK PENGENALAN INDIVIDU BERBASIS CITRA IRIS MATA

Nama : M. Rizky Faundra
NRP : 1214201021
Dosen Pembimbing : Dr. Dwi Ratna Sulistyaningrum, M.T.

ABSTRAK

Teknologi biometrika berkembang pesat beberapa dekade ini. Beberapa teknologi biometrika seperti pengenalan sidik jari dan suara sudah banyak dikembangkan namun diperlukan teknologi yang handal untuk sistem keamanan yang lebih baik. Salah satu teknologi biometrika yang handal dan perlu dikembangkan yaitu pengenalan individu melalui pola iris mata, hal ini dikarenakan pola iris mata memiliki tekstur unik yang dapat membedakan satu individu dengan individu lain.

Pada penelitian ini dilakukan pengenalan individu melalui pola iris dengan menggunakan gabungan metode filter Log-Gabor dan *Principle Component Analysis* (PCA). Metode ini mengkombinasikan kelebihan antara filter Log Gabor dan PCA, yaitu menghasilkan fitur tekstur iris dan menangkap informasi utama fitur tekstur iris hasil filter Log Gabor dengan cara mengurangi dimensi fitur. Langkah pertama citra iris mata disegmentasi dengan menggunakan transformasi Hough untuk mengisolasi daerah *zigzag collarete*, selanjutnya dilakukan normalisasi citra iris mata. Langkah kedua, citra iris mata hasil normalisasi diekstraksi dengan menggunakan filter Log Gabor, kemudian dilakukan pengurangan dimensi fitur menggunakan PCA. Terakhir dilakukan pencocokan dengan menggunakan normalisasi korelasi. Data yang digunakan pada penelitian ini menggunakan data citra mata keabuan yang diambil dari *CASIA database*.

Hasil penelitian menunjukkan bahwa gabungan metode filter Log-Gabor dan PCA memiliki tingkat akurasi pengenalan tertinggi mencapai 95.83% dengan penggunaan 45 vektor eigen.

Kata Kunci : *pengenalan iris mata, ekstraksi ciri, filter Log-Gabor, PCA*

COMBINATION OF LOG-GABOR FILTER AND PCA IN INDIVIDUAL RECOGNITION BASED ON EYE IRIS

Student's Name : M. Rizky Faundra
Student's Identity Number : 1214201021
Supervisor : Dr. Dwi Ratna Sulistyaningrum, M.T.

ABSTRACT

Biometrics technology developed rapidly the past few decades. Some biometric technologies such as fingerprint recognition and voice has been developed. However, reliable technology required for better security systems, one biometric that is reliable and still needs to be developed is individual recognition through the iris texture because iris pattern has a unique texture that distinguish one individual to another individual.

In this experiment, iris pattern recognition by using Log-Gabor filter and Principle Component Analysis (PCA) is done. First of all, iris images are processed by using Hough Transformation to isolate zigzag collarete area, then the result of the transformation normalized by using Daugman Rubber Sheet Model. Secondly, iris images normalization extracted by Log-Gabor filter then dimensionality reduction by using Principle Component Analysis. Finally, matching is done by using correlation normalization. This method is a kind of combination of advantage between filter Log-Gabor which describes textural iris feature and preserve the discriminant information prior to dimensionality reduction. This experiment will be conducted individual recognition through the iris patterns with a grayscale eye image data taken from a database of iris-Chinese Academy of Sciences Institute of Automation (CASIA). Data iris taken is reliable and widely used to study the iris biometrics.

The results showed that the combination of Log-Gabor filter and PCA has an accuracy rate reached 95.83% in CASIA database with 45 eigenvector.

Keyword : *iris recognition, feature extraction, log-Gabor filter, PCA*

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan segala Rahmat, Hidayah dan Inayah-Nya, serta bimbingan dan penjagaan-NYA kepada penulis sehingga dapat menyelesaikan Tesis yang berjudul “GABUNGAN METODE FILTER LOG-GABOR DAN PCA UNTUK PENGENALAN INDIVIDU BERBASIS CITRA IRIS MATA”. Penulis menyadari bahwa dalam penyusunan Tesis ini masih jauh dari kesempurnaan. Laporan Tesis ini disusun untuk memenuhi persyaratan dalam menyelesaikan studi Strata II Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya. Selama proses penyusunan Laporan Tesis ini penulis telah menerima banyak bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT, atas segala rahmat, hidayah, dan petunjuk-Nya.
2. Orang Tua ku Bapak Suyudi (Alm), Mama Elly Martini, yang telah membesarkan, mendidik, merawat dan memberikan do'a yang tulus dan ikhlas Kakakku Mbak Mia dan Adik-adikku Dik Reza, dan Dik Aldy serta keluarga kecilku Cicik Nurfitriyani dan sang buah hati Arvinza yang memberikan motivasi dan keceriaan selama pengerjaan tesis
3. Ibu Dr. Dwi Ratna Sulistyaningrum, M.T selaku dosen pembimbing yang dengan sabar membimbing penulis selama mengerjakan Tesis.
4. Bapak Dr. Mahmud Yunus, M.Si selaku Ketua Jurusan Pasca Sarjana Matematika ITS yang telah memberikan masukan yang bermanfaat.
5. Bapak Dr. Chairul Imron, M.IKom yang telah memberikan ilmu hadis Arbain selama perkuliahan S2.
6. Bapak Dr. Budi Setiyono, MT yang telah mengajarkan mata kuliah pengolahan citra digital sehingga penulis kembali memilih ke topik ini.
7. Bapak Prof. Isa Irawan yang telah menyelipkan cerita dan motivasi untuk kuliah di luar negeri.
8. Seluruh dosen dan karyawan di Jurusan Matematika ITS yang telah memberikan motivasi untuk bisa lulus

Surabaya, Desember 2016

Penulis

DAFTAR ISI

COVER INDONESIA	i
COVER INGGRIS	iii
HALAMAN PENGESAHAN	v
ABSTRAK	vi
ABSTRACT	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 Biometrika	5
2.2 Pengenalan Iris Mata	6
2.3 Citra Digital	7
2.4 Prapengolahan Citra	8
2.4.1 Deteksi Tepi <i>Canny</i>	8
2.4.2 Transformasi Hough	10
2.4.3 Isolasi Daerah <i>Zigzag Collarete</i> dan Penutupan Bulu Mata	11
2.4.4 Normalisasi Citra	12
2.5 Ekstraksi Ciri	13
2.6 <i>Principle Component Analysis (PCA)</i>	16
2.7 Pencocokan dengan Normalisasi Korelasi	19

2.8 Hitung Akurasi	20
BAB 3 METODE PENELITIAN	21
3.1 Objek Penelitian	21
3.2 Peralatan	21
3.3 Tahapan Penelitian	21
BAB 4 PENERAPAN FILTER LOG-GABOR DAN PCA	25
4.1 Perancangan Sistem Pengenalan Iris Mata	25
4.2 Perancangan Proses	26
4.2.1 Rancangan Prapengolahan Citra	26
4.2.2 Rancangan Ekstraksi	36
4.2.3 Rancangan Pencocokan	41
4.3 Implementasi Proses	42
4.3.1 Implementasi Prapengolahan Citra	42
4.3.2 Implementasi Ekstraksi	49
4.3.3 Implementasi Pencocokan	51
BAB 5 HASIL PENELITIAN DAN PEMBAHASAN	53
5.1 Hasil Pra Pengolahan Citra Iris Mata	53
5.1.1 Hasil Deteksi Tepi Pupil	53
5.1.2 Hasil Pencarian Pusat dan Jari-jari Pupil	55
5.1.3 Hasil Pengisolasian Daerah <i>Zigzag Collarete</i>	56
5.1.4 Hasil Normalisasi Iris	57
5.2 Hasil Ekstraksi Citra Iris Mata	58
5.2.1 Hasil Filter Log-Gabor	58
5.2.2 Hasil PCA	59
BAB 6 PENUTUP	63
6.1 Kesimpulan	63
6.2 Saran	63
DAFTAR PUSTAKA	65
LAMPIRAN	67
BIODATA PENULIS	75

DAFTAR TABEL

No. Tabel

2.1	Perbandingan Karakteristik Biometrika	5
2.2	Pengelompokkan Arah tepian	9
4.1	Hasil Percobaan Menentukan Batas Ambang	29
5.1	Proses Deteksi Tepi Pupil dari Citra Mata CASIA	53
5.2	Pusat dan Jari-jari Pupil dari Citra Mata CASIA	55
5.3	Proses Isolasi Daerah Zigzag Collarette dari Citra Mata CASIA	57
5.4	Hasil Normalisasi Iris	58
5.5	Hasil Pemfilteran Iris	58
5.6	Hasil Vektor Bobot Pelatihan dengan 10 Vektor Eigen.....	59
5.7	Hasil Perhitungan Vektor Bobot Uji dengan Vektor Bobot Latih	60
5.8	Hasil Akurasi Terhadap Jumlah Vektor Eigen.....	60
5.9	Hasil Akurasi Terhadap Berbagai Metode	62
5.10	Waktu Kinerja Pengenalan Terhadap Berbagai Metode	62

DAFTAR GAMBAR

No. Gambar

2.1	Bagian-bagian Mata	6
2.2	ATM pada Amman Bank	7
2.3	Bandara Manchester, Inggris	7
2.4	Perubahan Citra Kontinyu ke Diskrit	7
2.5	Hasil Deteksi Tepi Canny	10
2.6	Ilustrasi Transformasi Hough Lingkaran	10
2.7	Daerah <i>Zigzag Collarette</i>	11
2.8	Ilustrasi Proses Normalisasi	12
2.9	Perbandingan Filter Gabor dan Filter Log-Gabor	13
3.1	Contoh Citra Iris pada CASIA V-1	22
3.2	Diagram Alir Metode Penelitian	24
4.1	Perancangan Sistem Pengenalan Iris Mata	25
4.2	Diagram Alir Deteksi Tepi Pupil CASIA	27
4.3	Diagram Alir Pencarian Pusat dan Jari-jari Pupil	30
4.4	Grafik Penentuan Batas Maksimal Jari-jari	32
4.5	Diagram Alir Isolasi <i>Zigzag Collarette</i>	33
4.6	Penggambaran Lingkaran dalam Koordinat Citra	34
4.7	Diagram Alir Normalisasi	35
4.8	Diagram Alir Filter Log-Gabor	37
4.9	Diagram Alir Tahap Pelatihan PCA	39
4.10	Diagram Alir Tahap Pengujian PCA	40
4.11	Diagram Alir Pencocokan	41
5.1	Hasil Deteksi Tepi Canny dari Citra Mata CASIA	54
5.2	Penggambaran Lingkaran dalam Citra Asli	56
5.3	Penggambaran Lingkaran yang Tidak Berhasil dalam Citra Asli	56
5.4	Perbandingan Metode Filter Log-Gabor PCA dengan PCA	61

DAFTAR LAMPIRAN

A	HASIL IDENTIFIKASI PENGENALAN IRIS MATA	67
B	WAKTU RATA-RATA PENGENALAN TIAP IRIS MATA	71

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi biometrika telah berkembang dengan pesat beberapa dekade ini, teknologi yang memanfaatkan karakteristik manusia dapat digunakan sebagai *password* diri menggantikan kartu pada ATM atau sebagai akses masuk pada tempat penting yang membutuhkan tingkat keamanan yang cukup tinggi (Jain, 2003). Beberapa teknologi biometrika seperti pengenalan sidik jari dan suara sudah banyak dikembangkan dan mendapat perhatian menarik namun diperlukan teknologi yang cukup handal sehingga sistem keamanan menjadi lebih baik. Salah satu teknologi biometrika yang cukup handal dan masih perlu dikembangkan adalah pengenalan individu melalui tekstur iris (Ma, dkk, 2002a).

Tekstur iris mata manusia sangat banyak dan kompleks terdiri atas 2 layer yaitu *epithelium* layer dan *stromal* layer. Kombinasi kedua layer tersebut membentuk sebuah pola warna yang sangat beragam dan terjadi secara alami sejak tiga bulan kelahiran (Adler, 1965). Selain itu pola iris unik bahkan iris mata kiri dan kanan pada setiap individu memiliki pola yang berbeda. Bila dibandingkan dengan tekstur yang lain seperti wajah, sidik jari atau suara, tekstur iris lebih stabil karena letak iris berada di dalam tubuh sehingga dapat digunakan sebagai pengenalan individu yang handal (Daugman, 2004).

Proses pengenalan iris terdiri dari proses akuisisi citra, pra-pengolahan citra, ekstraksi ciri, dan pencocokan. Metode yang digunakan untuk ekstraksi ciri merupakan bagian yang penting guna memperoleh fitur yang bisa membedakan pola iris satu dengan pola iris yang lain. Penelitian tentang ekstraksi iris mata diperkenalkan oleh Daugman dengan menggunakan *Complex 2D Filter Gabor* untuk menangkap informasi tekstur secara global dan mengkodekan struktur iris dengan kuantisasi quadratur kemudian jarak Hamming digunakan untuk memutuskan cocok atau tidak cocok (Daugman, 2004).

Penelitian selanjutnya dikembangkan oleh Ma dengan menggunakan *Multichannel Gabor Filter* dan *Circular Symmetric Filter* untuk menangkap

informasi tekstur lokal dan global dari iris sedangkan *Weighted Euclidean Distance* dan *Nearest Feature Line* digunakan untuk mencocokkan iris (Ma, dkk, 2002a; 2002b). Kemudian dilakukan analisis penggunaan filter Gabor pada pengkodean iris yang telah dilakukan oleh Yao, dalam penelitiannya menyebutkan bahwa filter Gabor memiliki kelemahan yaitu pengkodean filter pada bagian real yang dihasilkan tidak secara total dapat terbentuk yang disebabkan adanya komponen DC (frekuensi titik asal tidak sama dengan nol), kemudian Yao mengusulkan penggunaan filter Log-Gabor sebagai alternatif filter Gabor untuk mengatasi adanya komponen DC (Yao, dkk, 2006a; 2006b). Permasalahan selanjutnya adalah penambahan dimensi data yang digunakan untuk pengenalan iris menjadi sangat besar, hal ini dikarenakan variasi pencahayaan alami pada saat akuisisi mempengaruhi pengenalan iris pada tiap individu akibatnya proses pengenalan tiap iris menjadi tidak efektif. Chowhan mengusulkan sebuah metode yang disebut dengan *eigen-iris* dengan menggunakan *Principle Component Analysis*. Metode ini digunakan untuk mengurangi dimensi data yang besar menjadi dimensi data yang kecil dengan masih mempertahankan informasi dari data yang besar (Chowhan, dkk 2009). Patil dalam penelitiannya melakukan analisa terhadap metode filter Log-Gabor saja dan PCA saja berdasarkan kinerja, dalam penelitiannya menyebutkan bahwa metode filter Log-Gabor memiliki akurasi yang cukup tinggi namun waktu yang dibutuhkan cukup tinggi bila dibandingkan dengan metode PCA saja (Patil, 2012). Yu dalam penelitiannya mencoba menggabungkan metode filter Log-Gabor dan PCA dalam pengenalan wajah, hasil yang diperoleh menunjukkan bahwa metode yang diusulkan oleh Yu dapat mengenali wajah dengan akurasi 86% (Yu, 2015).

Pada penelitian tesis ini dikombinasikan metode filter Log-Gabor dan PCA untuk pengenalan individu melalui pola iris. Metode ini mengkombinasikan kelebihan antara filter Log Gabor dan PCA, yaitu menghasilkan fitur tekstur iris dan menangkap informasi utama fitur tekstur iris hasil filter Log Gabor dengan cara mengurangi dimensi fitur, sedangkan untuk pencocokan menggunakan normalisasi korelasi. Data iris mata yang digunakan menggunakan data *CASIA* yang memiliki karakteristik pencahayaan yang seragam dengan citra warna

grayscale, data ini terbebas dari cahaya kamera pada saat akuisisi dan banyak digunakan pada penelitian biometrika iris mata.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah yang dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan metode filter Log-Gabor dalam pengenalan pola iris mata?
2. Bagaimana menerapkan metode PCA dalam pengenalan pola iris mata?
3. Bagaimana kinerja gabungan metode filter Log-Gabor dan PCA dalam pengenalan pola iris mata?

1.3 Batasan Masalah

Pada penelitian ini dibatasi ruang lingkup sebagai berikut. Data citra yang digunakan berupa gambar iris mata dengan format bitmap (*.bmp) yang diambil dari *CASIA iris database* sebanyak 210 gambar mata, kemudian dilakukan pengenalan per citra iris mata dengan menggunakan bahasa pemrograman MATLAB.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan gabungan metode filter Log-Gabor dan PCA kemudian menganalisa kinerja pengenalan pola iris mata.

1.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah

1. Bidang Biometrika, hasil dari penelitian dapat dijadikan sebagai alternatif pemilihan metode ekstraksi dalam pengenalan pola iris mata.
2. Bidang Keamanan, pemanfaatan dalam kontrol akses bandara atau gedung pemerintahan

1.6 Sistematika Penulisan

Penulisan tesis ini dibagi menjadi beberapa bab sebagai berikut:

BAB I PENDAHULUAN, bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penulisan

BAB II TINJAUAN PUSTAKA, bab ini berisi tentang landasan teori yang mendukung dan mendasari penelitian tesis ini

BAB III METODE PENELITIAN, bab ini berisi tentang langkah-langkah yang dilakukan dalam penelitian

BAB IV PENERAPAN FILTER LOG-GABOR DAN *PRINCIPLE COMPONENT ANALYSIS (PCA)*, bab ini berisi tentang rancangan proses-proses dalam mengenali pola iris mata dengan menggunakan Filter Log-Gabor dan PCA, serta pengimplementasian ke dalam bahasa pemrograman MATLAB berdasarkan rancangan yang telah dibuat.

BAB V HASIL PENELITIAN DAN PEMBAHASAN, memaparkan hasil penelitian pengenalan pola iris dan analisa kinerja filter Log-Gabor dan PCA dalam pengenalan pola iris mata

BAB VI PENUTUP, bab ini berisi kesimpulan dan saran untuk penelitian berikutnya

BAB II

TINJAUAN PUSTAKA

2.1 Biometrika

Biometrika (berasal dari bahasa Yunani *bios* yang artinya hidup dan *metron* yang artinya mengukur) adalah studi tentang metode otomatis untuk mengenali manusia berdasarkan satu atau lebih bagian tubuh manusia atau kelakuan dari manusia itu sendiri yang memiliki keunikan. Secara karakteristik biometrika dibagi menjadi 2 yaitu:

1. Berdasarkan Karakteristik Perilaku

Contoh: Suara, Tandatangan, Gaya Berjalan, Cara Mengetik

2. Berdasarkan Karakteristik Fisiologis

Contoh: Telapak Tangan, Wajah, Sidik Jari, Iris Mata, Retina Mata (Putra, 2009)

Tidak semua bagian tubuh atau perilaku seseorang dapat digunakan sebagai biometrika (Putra, 2009). Syarat utama biometrika adalah sebagai berikut:

1. Universalitas (*universality*) yaitu karakteristik yang dipilih harus dimiliki oleh setiap orang
2. Membedakan (*distinctiveness*) yaitu karakteristik yang dipilih memiliki kemampuan membedakan antara satu orang dengan orang lain.
3. Permanen (*permanence*) yaitu karakteristik yang dipilih tidak cepat berubah dalam periode waktu yang lama
4. Kolektabilitas (*collectability*) yaitu karakteristik yang diperoleh mudah diperoleh dan dapat diukur secara kuantitatif

Tabel 2.1 Perbandingan Karakteristik Biometrika

Biometrika	Sidik Jari	Wajah	Telapak Tangan	Iris Mata	Suara
					
Universal	Tinggi	Rendah	Sedang	Tinggi	Rendah
Membedakan	Tinggi	Sedang	Sedang	Tinggi	Rendah

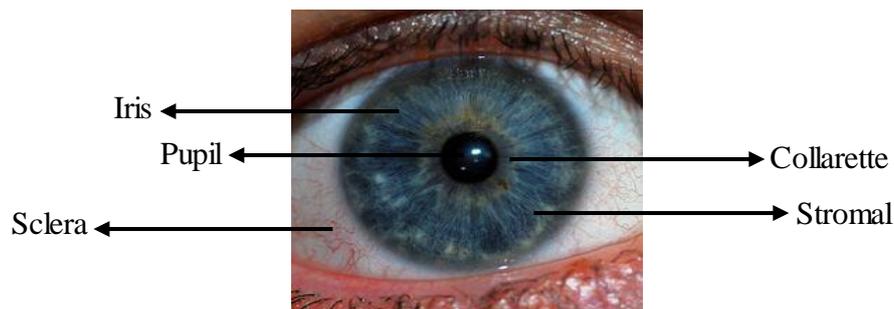
Biometrika	Sidik Jari	Wajah	Telapak Tangan	Iris Mata	Suara
Permanen	Sedang	Tinggi	Tinggi	Sedang	Sedang
Kolektabilitas	Tinggi	Rendah	Sedang	Tinggi	Rendah

Sumber: Jain, 2003

Pada Tabel 2.1 menunjukkan karakteristik biometrika iris mata termasuk baik karena memiliki rata-rata yang tinggi bila dibandingkan dengan karakteristik biometrika yang lain, sehingga penelitian tentang pengenalan iris mata telah mendapat banyak perhatian beberapa tahun terakhir ini.

2.2 Pengenalan Iris Mata

Iris atau selaput pelangi pada mata merupakan bagian dari mata yang berfungsi untuk mengontrol pupil dan mengatur banyaknya cahaya yang mencapai pupil. Iris memiliki 2 layer yaitu *epithelium* layer dan *stromal* layer, kedua kombinasi layer tersebut membentuk sebuah pola warna yang sangat beragam. Pola warna layer tersebut terbentuk sejak 3 bulan dari kelahiran dan tidak ada kaitannya dengan genetika, pola iris mata anak kembar berbeda bahkan pola iris mata kanan maupun iris mata kiri pun berbeda (Adler, 1965).



Gambar 2.1 Bagian-bagian Mata (Adler, 1965)

Pengenalan iris mata adalah pengenalan berdasarkan pola unik iris mata. Karena karakteristik iris mata yang membedakan antara pola satu dengan pola yang lain serta stabil sehingga pengenalan iris mata dapat digunakan sebagai pengenalan identitas atau individu (Adler, 1965). Beberapa penerapan pengenalan

iris mata telah diaplikasikan seperti pada ATM di Yordania dan bandara udara di Inggris.



Gambar 2.2 ATM pada Amman Bank
(Sumber: Bank ATM Recognition by Iris)

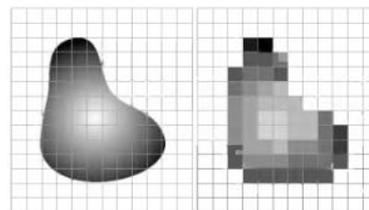


Gambar 2.3 Bandara Manchester, Inggris
(Sumber: Iris Recognition Immigration System)

Pengenalan iris mata dilakukan oleh instansi perbankan dan pemerintahan di luar negeri untuk memberikan keamanan pada setiap orang dalam beraktivitas sehingga kejahatan seperti pembobolan ATM atau terorisme dapat diminimalkan.

2.3 Citra Digital

Istilah “citra” yang digunakan dalam bidang pengolahan citra dapat diartikan sebagai suatu fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Pemrosesan citra dengan komputer digital membutuhkan citra digital sebagai masukannya. Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit seperti pada Gambar 2.4, baik koordinat ruang maupun intensitas cahayanya (Gonzalez, 2002).



Gambar 2.4 Perubahan Citra Kontinyu ke Diskrit (Gonzalez, 2002)

Gonzales mengatakan bahwa citra digital dapat dinyatakan dalam matriks dua dimensi $f(x,y)$, dimana (x,y) merupakan koordinat piksel dan f merupakan derajat intensitas cahaya yang dinyatakan dalam bentuk matriks berukuran $M \times N$

$$f(x, y) = \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \dots & f(2,N-1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{pmatrix} \quad (2.1)$$

Suatu citra $f(x, y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M - 1$$

$$0 \leq y \leq N - 1$$

$$0 \leq f(x, y) \leq G - 1$$

Dimana : M = banyaknya baris pada array citra

: N = banyaknya kolom pada array citra

: G = banyaknya skala keabuan (*graylevel*)

2.4 Prapengolahan Citra

Bagian prapengolahan citra (*preprocessing*) yaitu melakukan pengolahan awal citra agar dapat diambil ROI (*region of interest*) dalam hal ini adalah ROI iris, bagian ini terdiri atas beberapa metode pengolahan citra digital dalam mengambil ROI iris yaitu deteksi tepi Canny, transformasi Hough dan normalisasi.

2.4.1 Deteksi Tepi Canny

Deteksi tepi Canny memiliki kelebihan yaitu tepi yang dihasilkan sama seperti tepi pada gambar sebenarnya. Proses deteksi tepi Canny (Putra, 2009) sebagai berikut, sebelum dilakukan pendeteksian tepi, terlebih dahulu dilakukan penghalusan citra dengan filter Gaussian dengan tujuan mengurangi *noise* sehingga terlihat perbedaan tepi dan yang bukan tepi, selanjutnya dilakukan pendeteksian tepi dengan operator Sobel. Pada persamaan (2.2) dan (2.3) berikut merupakan operator Sobel dalam arah vertikal dan horizontal.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2.2)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.3)$$

Hasil dari kedua operator diatas kemudian dicari nilai tepi dan arah tepi dengan rumus:

$$G = \sqrt{(G_x)^2 + (G_y)^2} \quad (2.4)$$

$$\theta = \text{arc tan} \left(\frac{G_y}{G_x} \right) \quad (2.5)$$

Setelah arah tepi diketahui, langkah selanjutnya menghubungkan arah tepi pada suatu arah tertentu dalam citra seperti pada Tabel 2.2. Tujuan dilakukan pengelompokkan arah agar arah tepi lebih terfokus pada suatu sudut tertentu.

Tabel 2.2 Pengelompokan Arah Tepian

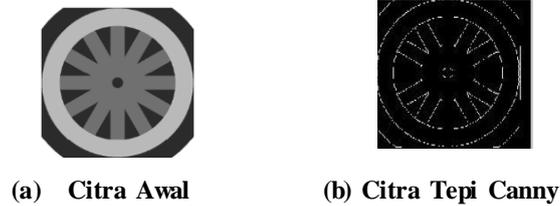
Arah Gradien	Daerah
$0^0 < \theta < 22.5^0$ $157.5^0 \leq \theta \leq 180^0$	Horizontal (0^0)
$22.5^0 \leq \theta < 67.5^0$	Diagonal Kanan (45^0)
$67.5^0 \leq \theta < 112.5^0$	Vertikal (90^0)
$112.5^0 \leq \theta < 157.5^0$	Diagonal Kiri (135^0)

Arah tepi terlihat pada Tabel 2.2 di atas memiliki empat arah kemungkinan yaitu piksel 0^0 (dalam arah horizontal), 45^0 (dalam posisi sepanjang positif diagonal), 90^0 (dalam arah vertikal) atau 135^0 (dalam posisi sepanjang negatif diagonal). Sehingga arah tepi dapat dikelompokkan ke dalam 4 arah bergantung pada arah mana yang mendekati (misalnya: jika sudut orientasi diperoleh 3^0 maka sudut orientasi tersebut dibuat menjadi 0^0).

Sudut tepi yang telah dikelompokkan kemudian dilakukan non maxima surpression untuk menemukan nilai piksel terbesar berdasarkan arah atau sudut tepi. Semua nilai piksel sepanjang arah sudut tepi yang bukan merupakan nilai terbesar menjadi nol.

Hasil pendeteksian tepi yang masih terlihat seperti garis-garis putus kemudian dilakukan penggunaan dua nilai ambang yaitu T_1 sebagai ambang bawah dan T_2 sebagai ambang atas, piksel yang memiliki nilai lebih besar dari T_2 diasumsikan sebagai tepi sedangkan piksel yang lebih besar dari T_1 dan terhubung

dengan T_1 juga ditandai sebagai tepi. Pada Gambar 2.5 menunjukkan contoh hasil deteksi tepi Canny.



Gambar 2.5 Hasil Deteksi Tepi Canny (a) Citra Awal, (b) Citra Tepi Canny (Sumber: Aggrawal, 2004)

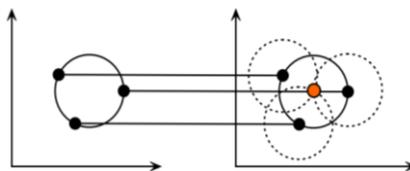
Gambar 2.5(a) merupakan citra awal sebelum dilakukan deteksi tepi Canny, sedangkan Gambar 2.5(b) merupakan citra tepi *Canny*, tepi terlihat lebih akurat karena dapat menggambarkan dengan tepat perbedaan intensitas pada citra awal.

2.4.2 Transformasi Hough

Transformasi ini pertama kali diperkenalkan oleh Paul Hough, kemudian digunakan untuk mendeteksi bentuk-bentuk garis maupun lingkaran oleh (Duda, 1972). Tujuan dilakukan transformasi ini adalah mendeteksi bentuk-bentuk lingkaran pada bagian pupil. Proses transformasi Hough lingkaran sebagai berikut, nilai jari-jari diatur dengan batas maksimum dan minimum, kemudian dilakukan penggambaran koordinat lingkaran yang dibentuk dengan tepi sebagai pusat lingkaran dengan persamaan

$$(x - a)^2 + (y - b)^2 = r^2$$

dimana (a, b) merupakan koordinat piksel tepi, r adalah jari-jari dengan batas tertentu sedangkan (x, y) adalah titik-titik pada lingkaran, setelah digambarkan pada tiap tepi sebagai titik pusat dan jari-jari tertentu maka dilakukan voting pada titik yang merupakan perpotongan lingkaran buatan, titik terbanyak yang dilewati oleh lingkaran buatan menjadi titik pusat lingkaran asli sedangkan jari-jari lingkaran buatanya menjadi jari-jari lingkaran asli. Gambar 2.6 berikut menunjukkan ilustrasi dari transformasi Hough Lingkaran:



Gambar 2.6 Ilustrasi Transformasi Hough Lingkaran (Sumber: Murmu, 2009)

Sebagai contoh perhitungan diberikan tiga titik pada koordinat citra yaitu $A(3,3)$, $B(2,4)$, $C(3,5)$ sebagai koordinat piksel tepi. Jika tiap titik merupakan pusat lingkaran dengan jari-jari sama dengan satu maka untuk menentukan koordinat yang merupakan perpotongan terbanyak dari ketiga lingkaran yang dibuat yaitu dengan menentukan persamaan lingkaran terlebih dahulu pada masing-masing titik sehingga diperoleh:

Persamaan lingkaran pada titik A adalah $L_1 \equiv (x - 3)^2 + (y - 3)^2 = 1$

Persamaan lingkaran pada titik B adalah $L_2 \equiv (x - 2)^2 + (y - 4)^2 = 1$

Persamaan lingkaran pada titik C adalah $L_3 \equiv (x - 3)^2 + (y - 5)^2 = 1$

Titik potong persamaan lingkaran A dan B adalah $(2,3)$ dan $(3,4)$

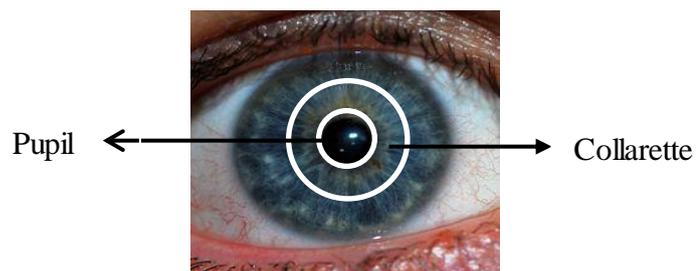
Titik potong persamaan lingkaran B dan C adalah $(3,4)$ dan $(2,5)$

Titik potong persamaan lingkaran A dan C adalah $(3,4)$

Sehingga titik potong terbanyak adalah $(3,4)$ merupakan pusat lingkaran dengan jari-jari samadengan satu.

2.4.3 Isolasi Daerah Zigzag Collarete dan Penutupan Bulu Mata

Daerah *zigzag collarete* merupakan daerah yang terpilih untuk ekstraksi fitur iris dikarenakan daerah tersebut memiliki pola iris yang sangat kompleks dan tidak banyak terpengaruh oleh kelopak mata maupun bulu mata (Rai, 2014). Pada Gambar 2.7 merupakan gambar daerah zigzag collarete.



Gambar 2.7 Daerah Zigzag Collarete

Daerah tersebut konsentris dengan pupil sedangkan jari-jari pada bagian ini memiliki jarak tertentu. Pencarian daerah *zigzag collarete* dihitung dengan menggunakan rumus berikut:

$$l = r_z - r_p$$

dimana r_z adalah jari-jari *zigzag collarete* dan r_p adalah jari-jari pupil. Rai mengatakan bahwa untuk lebar pencarian daerah *zigzag collarete* memiliki nilai sebesar 24 piksel untuk database *CASIA* (Rai, 2014).

Pada saat mengisolasi daerah *zigzag collarete* terdapat beberapa gangguan (*noise*) yang disebabkan oleh penutupan bulu mata, untuk mengatasi hal ini maka digunakan piksel rata-rata dari daerah *zigzag collarete*. Proses yang dilakukan adalah menelusuri intensitas piksel yang kurang dari nilai *threshold* yang berarti piksel tersebut tertutup bulu mata. Setiap piksel dengan intensitas yang kurang dari *threshold* dilakukan penutupan bulu mata dengan nilai rata-rata citra *zigzag collarete* (Masek, 2003).

2.4.4 Normalisasi Citra

Proses normalisasi yaitu mengubah bentuk lingkaran menjadi bentuk persegi panjang dengan tujuan agar lebih mudah dianalisa tiap piksel dari atas ke bawah atau dari bawah ke atas sehingga dapat memudahkan dalam pencocokan. Proses normalisasi iris dilakukan dengan memetakan tiap piksel sesuai dengan jari-jari (r) dan sudut (θ) menggunakan transformasi berikut:

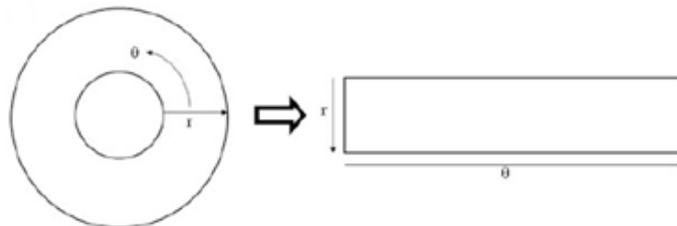
$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \quad (2.6)$$

dengan

$$x(r, \theta) = x_0 + r \cos \theta$$

$$y(r, \theta) = y_0 + r \sin \theta$$

(x_0, y_0) menyatakan pusat koordinat, r menyatakan selisih jari-jari pupil dan jari-jari *zigzag* sedangkan θ menyatakan besar sudut. Pada Gambar 2.8 berikut menunjukkan ilustrasi normalisasi citra.



Gambar 2.8 Ilustrasi Proses Normalisasi
(Sumber: Masek, 2003)

2.5 Ekstraksi Ciri

Bagian ekstraksi merupakan bagian yang penting guna menghasilkan informasi citra iris sehingga dapat membedakan dengan yang lain. Metode filter gabor banyak digunakan dalam ekstraksi karena dapat menghasilkan informasi tekstur lokal baik secara spasial maupun frekuensi, namun disisi lain selalu menghasilkan komponen DC (frekuensi titik asal tidak nol) pada bagian real sehingga cenderung untuk menghasilkan frekuensi rendah (Yao, 2006).

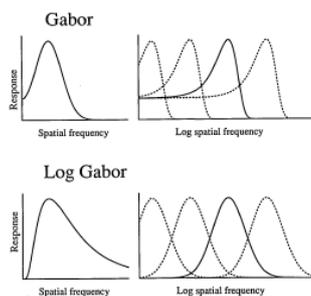
Field mengatakan bahwa untuk mengatasi adanya komponen DC maka dibuat sebuah fungsi filter Gabor dengan menggunakan skala logaritma pada domain frekuensi yang disebut dengan filter Log-Gabor. Secara matematis bentuk filter Log-Gabor sebagai berikut (Field, 1987)

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right) \quad (2.7)$$

dengan f_0 = pusat frekuensi filter

σ = lebar filter

Field juga melakukan percobaan dengan membandingkan antara filter Gabor dan filter Log-Gabor. Hasil percobaan yang dilakukan oleh Field menunjukkan bahwa komponen DC (frekuensi titik asal menjadi tidak nol) seperti pada Gambar 2.9 berikut:



**Gambar 2.9 Perbandingan Filter Gabor dan Filter Log-Gabor
(Sumber: Field, 1987)**

Bentuk filter log-gabor kemudian dikembangkan menjadi 2D filter Log-Gabor dengan memperhatikan komponen sudut filter, filter ini telah digunakan

untuk mengekstraksi citra wajah oleh (Yu, 2015). Berikut secara matematis filter Log-Gabor 2D:

$$G(f, \theta) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right) \times \exp\left(\frac{-(\theta - \theta_0)}{2\sigma_\theta^2}\right) \quad (2.8)$$

dengan f_0 = pusat frekuensi filter

f = frekuensi

θ = sudut orientasi

$\frac{\sigma}{f_0}$ = lebar filter terhadap pusat frekuensi

θ_0 = sudut awal

σ_θ = lebar sudut filter

Untuk mengekstraksi ciri dengan menggunakan filter Log-Gabor 2D, citra dengan ukuran $M \times N$ difilter dengan filter Log-Gabor pada domain frekuensi. Langkah-langkah pemfilteran pada domain frekuensi adalah sebagai berikut:

- Menghitung transformasi Fourier pada citra dengan ukuran $M \times N$ dengan rumus berikut:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \left(\cos\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) - j \sin\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) \right)$$

- Melakukan perkalian dot antara hasil transformasi Fourier dengan filter Log-Gabor sebagai berikut:

$$H(f, \theta) = G(f, \theta) * F(u, v)$$

- Mengubah ke domain spasial dengan invers transformasi Fourier

$$f(x, y) = \frac{1}{MN} \sum_{x=1}^{M-1} \sum_{y=1}^{N-1} H(f, \theta) \left(\cos\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) + j \sin\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) \right)$$

Hasil invers transformasi Fourier diatas merupakan bentuk bilangan kompleks yang terdiri atas bagian real dan imajiner dimana

$$R(x, y) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N H(f, \theta) \left(\cos\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) \right)$$

$$I(x, y) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N H(f, \theta) \left(\sin\left(2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right) \right)$$

dengan $R(x,y)$ adalah bagian real dan $I(x,y)$ adalah bagian imajiner

- d. Menghitung spektrum Fourier yang merepresentasikan hasil fitur

$$|f(x,y)| = \sqrt{R^2(x,y) + I^2(x,y)}$$

Sebagai contoh perhitungan diberikan matriks A dengan ukuran 2×2 yaitu, $A = \begin{bmatrix} 2 & 4 \\ 1 & 5 \end{bmatrix}$, matriks A dianggap sebagai nilai piksel dari suatu citra, akan dilakukan pemfilteran dengan matriks $G = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$ dimana matriks G merupakan contoh matriks filter Log-Gabor. Berikut ini merupakan contoh perhitungan pada domain frekuensi:

- a. Mengubah citra A pada domain frekuensi dengan transformasi Fourier diperoleh

$$F(0,0) = 2 \left(\cos \left(2\pi \left(\frac{0.0}{2} + \frac{0.0}{2} \right) \right) - j \sin \left(2\pi \left(\frac{0.0}{2} + \frac{0.0}{2} \right) \right) \right) + \dots = 12$$

$$F(0,1) = 2 \left(\cos \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) - j \sin \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) \right) + \dots = -6$$

$$F(1,0) = 2 \left(\cos \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) - j \sin \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) \right) + \dots = 0$$

$$F(1,1) = 2 \left(\cos \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) - j \sin \left(2\pi \left(\frac{0.0}{2} + \frac{1.0}{2} \right) \right) \right) + \dots = 2$$

- b. Melakukan perkalian dot dengan filter Log-Gabor diperoleh

$$H = \begin{bmatrix} 12.1 & -6.2 \\ 0.0 & 3.2 \end{bmatrix} = \begin{bmatrix} 12 & -12 \\ 0 & 6 \end{bmatrix}$$

- c. Mengubah hasil pemfilteran pada domain spasial dengan invers transformasi Fourier diperoleh

$$f(0,0) = 1.5 \quad f(0,1) = 4.5$$

$$f(1,0) = -1.5 \quad f(1,1) = 7.5$$

- d. Menghitung spektrum Fourier dari matriks f diperoleh

$$f = \begin{bmatrix} 1.5 & 4.5 \\ -1.5 & 7.5 \end{bmatrix}$$

Matriks hasil perhitungan spektrum Fourier merupakan hasil pemfilteran terhadap matriks A

.

2.6 Principle Component Analysis (PCA)

Principle Component Analysis (PCA) merupakan metode yang terkenal untuk menangkap informasi utama dari tiap fitur dengan cara mengurangi dimensi fitur citra. Metode ini mampu mendeskripsikan data masukan yang besar tanpa mengurangi informasi dari data itu (Chowhan, 2009). Ide dasar dari PCA adalah melakukan transformasi terhadap data masukan yang berupa n vektor dari n citra yang ditumpuk. Jika tiap n citra tersebut memiliki ukuran $M \times N$ maka terdapat sebanyak $M \times N \times n$ vektor kolom. Hal ini berakibat pada ukuran matriks fitur menjadi sangat besar sehingga diperlukan transformasi fitur dengan PCA, secara matematis langkah-langkah PCA sebagai berikut:

Merepresentasikan tiap citra ke dalam bentuk matriks kolom diberi dengan ukuran $M \times N \times 1$. Bila dinotasikan dalam bentuk matriks maka diperoleh

$$X = \begin{bmatrix} J_{1,1} & J_{1,2} & \cdots & J_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{MN,1} & J_{MN,2} & \cdots & J_{MN,n} \end{bmatrix} \quad (2.9)$$

Dimana $X_{k,1}$ adalah vektor kolom dari citra pertama dan seterusnya. Kemudian menghitung rata-rata citra dengan persamaan

$$m_k = \frac{1}{n} \sum_{i=1}^n J_{k,i} \quad (2.10)$$

dengan n adalah jumlah citra, dan m_k adalah rata-rata dari baris ke- k pada matriks X . Fitur ekstraksi PCA merupakan hasil pengurangan antara data tiap citra X dengan rata-rata citra m yang dinyatakan dalam persamaan

$$S_{k,i} = X_{k,i} - m_k \quad (2.11)$$

Hasil ekstraksi PCA kemudian dicari matriks kovarian dengan rumus

$$C = \frac{1}{n-1} S S^T \quad (2.12)$$

setelah didapat matriks kovarian kemudian dicari nilai eigen dan vektor eigen dengan persamaan

$$C v = \lambda v \quad (2.13)$$

dimana λ adalah nilai eigen dan v adalah vektor eigen. Proses mencari nilai eigen adalah sebagai berikut:

$$\begin{aligned} C v &= \lambda v \\ C v - \lambda v &= 0 \\ C v - \lambda I v &= 0 \\ (C - \lambda I) v &= 0 \end{aligned} \tag{2.14}$$

Agar mempunyai solusi maka

$$|C - \lambda I| = 0 \tag{2.15}$$

Persamaan diatas menghasilkan nilai eigen kemudian nilai eigen diurutkan secara menurun dari nilai paling besar menuju ke nilai paling kecil sedangkan vektor eigen disusun dengan urutan nilai eigen yang telah diurutkan. Langkah berikutnya adalah melakukan transformasi fitur atau juga disebut proyeksi fitur dengan rumus

$$w = v^T S \tag{2.16}$$

Proyeksi fitur merupakan hasil kali antara vektor eigen terpilih dengan setiap citra

Sebagai contoh perhitungan diberikan tiga buah matriks dengan ukuran 2×2 yaitu $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ dan $C = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$, matriks ini dianggap sebagai nilai piksel dari citra.

Dari masing-masing citra diubah menjadi matriks kolom kemudian digabungkan sehingga diperoleh bentuk matriks baru yaitu

$$X = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Setiap baris pada citra baru kemudian dicari rata-rata, hasilnya adalah matriks rata-rata tiap baris

$$m_x = \begin{bmatrix} 0.67 \\ 0.33 \\ 0.33 \\ 0.67 \end{bmatrix}$$

Dari matriks rata-rata kemudian dihitung selisih antara tiap-tiap matriks kolom dengan matriks rata-rata satu demi satu, matriks ini disebut matriks fitur. Proses perhitungannya adalah sebagai berikut:

$$S = \begin{bmatrix} 1 - 0.67 & 1 - 0.67 & 0 - 0.67 \\ 0 - 0.33 & 0 - 0.33 & 1 - 0.33 \\ 0 - 0.33 & 1 - 0.33 & 0 - 0.33 \\ 1 - 0.67 & 0 - 0.67 & 1 - 0.67 \end{bmatrix} = \begin{bmatrix} 0.33 & 0.33 & -0.67 \\ -0.33 & -0.33 & 0.67 \\ -0.33 & 0.67 & -0.33 \\ 0.33 & -0.67 & 0.33 \end{bmatrix}$$

Selanjutnya adalah menghitung matriks kovarian dengan rumus matriks kovarian, proses perhitungannya adalah sebagai berikut:

$$C = \frac{1}{2} \begin{bmatrix} 0.33 & 0.33 & -0.67 \\ -0.33 & -0.33 & 0.67 \\ -0.33 & 0.67 & -0.33 \\ 0.33 & -0.67 & 0.33 \end{bmatrix} \begin{bmatrix} 0.33 & -0.33 & -0.33 & 0.33 \\ 0.33 & -0.33 & 0.67 & -0.67 \\ -0.67 & 0.67 & -0.33 & 0.33 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.33 & -0.33 & 0.16 & -0.16 \\ -0.33 & 0.33 & -0.16 & 0.16 \\ 0.16 & -0.16 & 0.33 & -0.33 \\ -0.16 & 0.16 & -0.33 & 0.33 \end{bmatrix}$$

Dari matriks kovarian kemudian dicari nilai eigen dan vektor eigen diperoleh

$$V = \begin{bmatrix} -0.23 & -0.66 & 0.5 & -0.5 \\ -0.23 & -0.66 & -0.5 & 0.5 \\ -0.66 & 0.23 & -0.5 & -0.5 \\ -0.66 & 0.23 & 0.5 & 0.5 \end{bmatrix} \text{ dan } D = [0 \quad 0 \quad 0.33 \quad 1]$$

Vektor eigen kemudian diurutkan berdasarkan urutan nilai eigen dari terbesar ke terkecil diperoleh

$$V = \begin{bmatrix} -0.5 & 0.5 & -0.66 & -0.23 \\ 0.5 & -0.5 & -0.66 & -0.23 \\ -0.5 & -0.5 & 0.23 & -0.66 \\ 0.5 & 0.5 & 0.23 & -0.66 \end{bmatrix}$$

Langkah terakhir yaitu membuat vektor bobot latih dengan memproyeksikan matriks fitur terhadap vektor eigen diperoleh

$$w = \begin{bmatrix} 0 & -1 & 1 \\ 0.67 & -0.33 & -0.3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Matriks w merupakan matriks bobot yang terdiri atas vektor bobot (vektor kolom). Tiap-tiap vektor bobot mewakili bobot setiap citra latih, misal vektor kolom pertama pada matriks w merupakan bobot dari setiap citra latih pertama dan seterusnya.

2.7 Pencocokan dengan Normalisasi Korelasi

Pencocokan merupakan tahap pengujian dengan mencari nilai jarak suatu vektor uji dan vektor referensi satu demi satu. Metode pencocokan yang dipakai adalah normalisasi korelasi. Normalisasi korelasi antara vektor uji dengan vektor referensi pertama kali diperkenalkan oleh Lewis pada tahun 1995 kemudian digunakan oleh (Wildes, 1997) sebagai pencocokan citra iris. Bentuk normalisasi korelasi yang diberikan memiliki bentuk yang sama seperti koefisien korelasi oleh (Putra, 2009) yaitu menghitung selisih antara tiap koordinat dengan rata-ratanya kemudian dibagi dengan masing-masing varians dan hasil kali ukuran citra. Secara matematis bentuk normalisasi korelasi sebagai berikut:

$$d = \sum_{i=1}^n \sum_{j=1}^m \frac{(p_1(i,j) - \mu_1)(p_2(i,j) - \mu_2)}{nm \sigma_1 \sigma_2} \quad (2.17)$$

dimana $p_1(i,j)$, $p_2(i,j)$ merupakan vektor uji dan vektor referensi, μ_1, μ_2 adalah rata-rata vektor uji dan vektor referensi, $\sigma_1 \sigma_2$ adalah hasil kali varians dari vektor uji dan vektor referensi dan nm hasil kali ukuran vektor baris dan kolom. Nilai normalisasi korelasi berada di antara -1 dan 1 yang mana semakin tinggi nilainya maka dua citra semakin mirip sebaliknya semakin rendah nilainya maka dua citra tidak mirip. Sebagai contoh perhitungan normalisasi yaitu diberikan dua buah vektor $A = [0, 3, 4, 5]$ dan $B = [7, 6, 3, -1]$, normalisasi korelasi dicari dengan menggunakan langkah sebagai berikut:

1. Hitung rata-rata dari setiap vektor

$$\mu_1 = \frac{0 + 3 + 4 + 5}{4} = 3$$

$$\mu_2 = \frac{7 + 6 + 3 + (-1)}{4} = 3.75$$

2. Hitung normalisasi korelasi dari vektor A dan B

$$d = \frac{(0-3)(7-3.75) + (3-3)(6-3.75) + (4-3)(3-3.75) + (5-3)(-1-3.75)}{([(0-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2][(7-3.75)^2 + (6-3.75)^2 + (3-3.75)^2 + (-1-3.75)^2])^{\frac{1}{2}}}$$

$$d = \frac{-9.75 + 0 - 0.75 - 9.5}{\sqrt{14 \times 38.75}} = \frac{-20}{\sqrt{542.5}} = -0.859$$

Kesimpulan dari kedua vektor diatas adalah memiliki ketidakmiripan karena nilai normalisasi korelasinya sangat rendah.

2.8 Hitung Akurasi

Besar akurasi dihitung dengan persentase banyak citra uji yang dikenali dengan benar, semakin banyak citra uji yang dikenal maka akurasi pencocokan semakin besar (Ali, 2003).

$$akurasi = \frac{U}{T} \times 100\% \quad (2.18)$$

dimana U adalah banyak citra uji yang dikenali dengan benar dan T adalah total citra yang diuji.

BAB III

METODE PENELITIAN

Bab ini menjelaskan tahapan penelitian dan peralatan untuk menyelesaikan pengenalan individu melalui pola iris dengan filter Log-Gabor dan PCA. Adapun tahapannya sebagai berikut:

3.1 Objek Penelitian

Pada penelitian ini objek penelitian adalah gambar iris mata yang didapat melalui database iris CASIA sebanyak 210 gambar dari 30 orang yang masing-masing memiliki 7 citra mata.

3.2 Peralatan

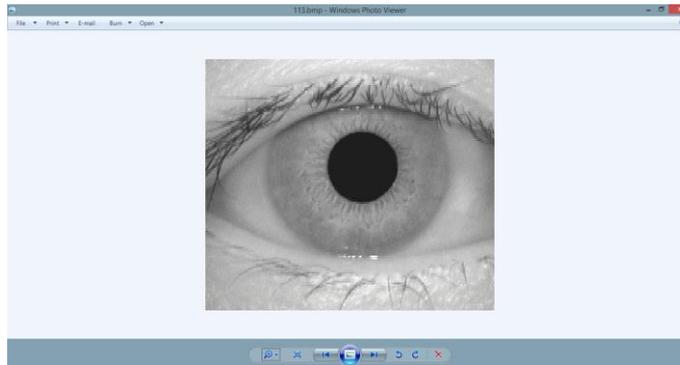
Peralatan yang digunakan yaitu software MATLAB, Microsoft Excel dan hardware Processor AMD A6, Memory 2GB

3.3 Tahapan Penelitian

Tahapan penelitian yang dilakukan untuk menyelesaikan pengenalan individu melalui pola iris meliputi:

a. Akuisisi Data

Pada tahap ini dilakukan akuisisi data mata, akuisisi data mata dilakukan dengan mendownload di website <<http://www.cbsr.ia.ac.cn/IrisDatabase>> untuk database CASIA. Database CASIA yang dipilih adalah CASIA V-1 dikarenakan citra iris mata memiliki karakteristik seragam serta iluminasi atau berkas cahaya putih pada daerah pupil sudah dihilangkan dengan kamera yang sudah dikembangkan. Pada Gambar 3.1 merupakan contoh citra iris CASIA V-1.



**Gambar 3.1 Contoh Citra Iris pada CASIA V-1
(Sumber: CASIA iris database)**

Data mata CASIA memiliki dua sesi pengambilan yaitu sesi pertama sebanyak 3 mata dan sesi kedua sebanyak 4 mata. Kedua sesi tersebut diambil dari mata yang sama. Berikut merupakan contoh pengkodean database CASIA pada sesi pertama dan sesi kedua:

'001_1_1'

Keterangan: 001 = mata orang pertama

1 = mata diambil pada sesi pertama

1 = mata diambil pada urutan pertama

'001_2_1'

Keterangan: 001 = mata orang pertama

2 = mata diambil pada sesi kedua

1 = mata diambil pada urutan pertama

b. Prapengolahan Citra

Pada tahap ini dilakukan proses prapengolahan citra, tahapan ini terdiri atas tiga bagian utama yaitu mendeteksi tepi pupil dengan menggunakan metode deteksi tepi *Canny*, kemudian dilakukan penelusuran tepi berbentuk lingkaran dengan Transformasi Hough serta mengisolasi daerah *zigzag collarette*, kemudian mengubah bentuk lingkaran hasil deteksi menjadi bentuk persegi panjang. Pada tahap mendeteksi tepi pupil pada database *CASIA* dilakukan prapengolahan citra. Prapengolahan pada database *CASIA* dimulai dengan menentukan *threshold* agar bisa memisahkan antara pupil dan iris setelah itu dilakukan deteksi tepi *Canny* dan Transformasi Hough untuk menelusuri bentuk lingkaran pupil

c. Ekstraksi Ciri

Pada tahap ini dilakukan ekstraksi ciri iris yang sudah diubah ke bentuk persegi panjang, tahapan ini meliputi mensubstitusi parameter filter Log-Gabor, kemudian melakukan pemfilteran pada citra dengan filter Log Gabor dan pembuatan vektor latih dan vektor uji dengan *Principle Component Analysis (PCA)*.

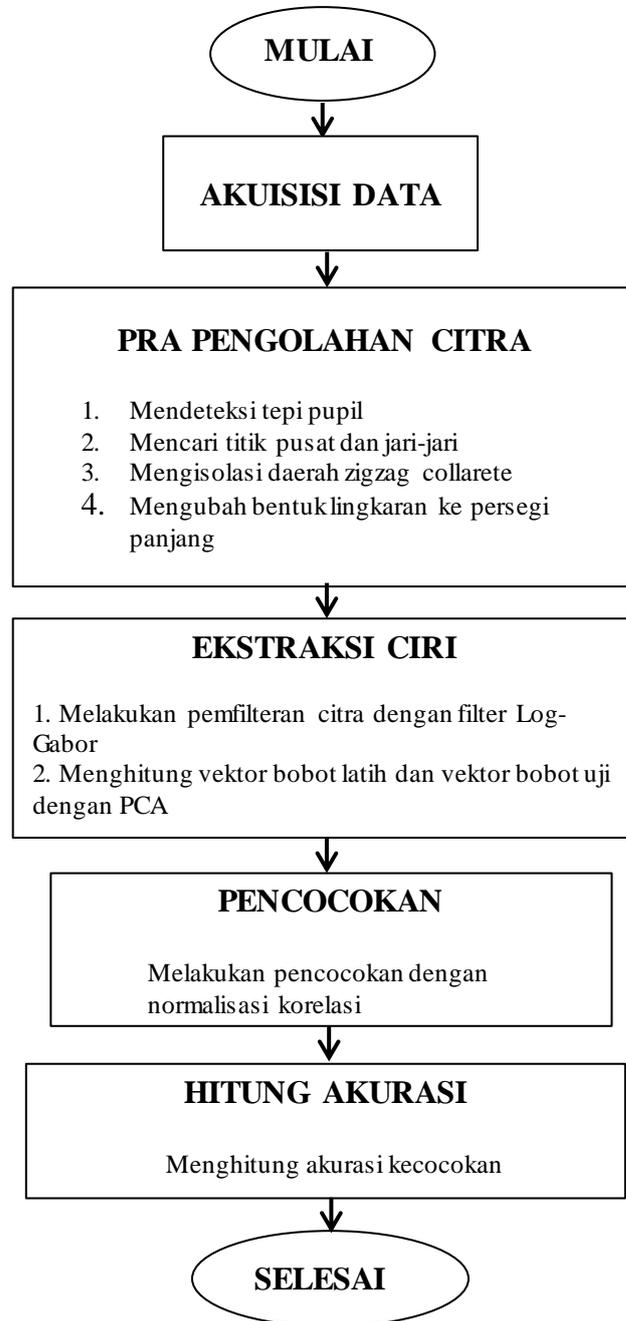
d. Pencocokan

Pada tahap ini dilakukan pencocokan terhadap dua pola iris dengan menggunakan normalisasi korelasi. Proses pencocokan dilakukan per citra iris dengan memilih jarak minimal dari setiap pencocokan antara vektor uji dengan vektor latih

e. Hitung Akurasi

Pada tahap ini dilakukan perhitungan akurasi serta dilakukan perbandingan akurasi dari beberapa metode pengenalan iris mata, dan menganalisa kinerja dari tiap-tiap metode

Metode penelitian digambarkan melalui diagram alir yang ditunjukkan pada Gambar 3.2 berikut:



Gambar 3.2 Diagram Alir Metode Penelitian

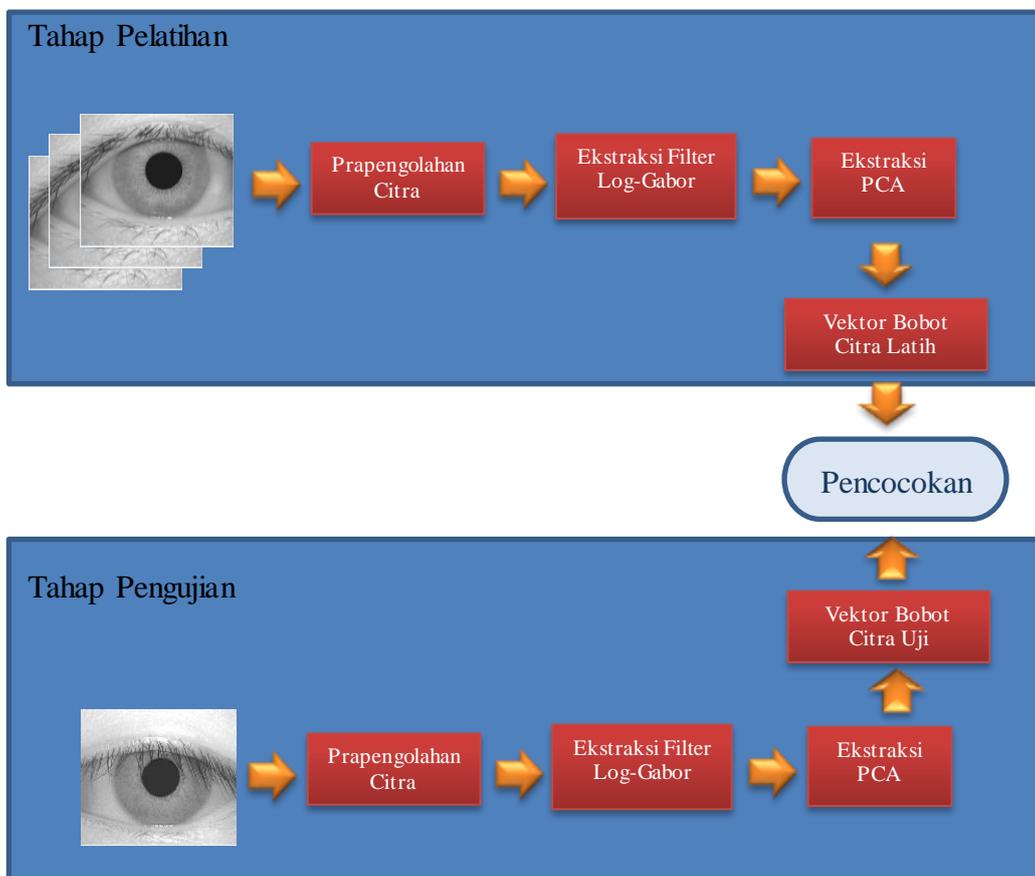
BAB IV

PENERAPAN FILTER LOG-GABOR DAN *PRINCIPLE COMPONENT ANALYSIS (PCA)*

Pada bab ini dibahas mengenai rancangan proses-proses yang terjadi pada pengenalan iris mata yang terdiri atas prapengolahan citra, ekstraksi menggunakan filter Log-Gabor dan PCA, serta pencocokan terhadap vektor bobot uji dan latih. Proses yang dirancang kemudian dilakukan pengimplementasian ke dalam bahasa pemrograman untuk dilakukan pengujian dan analisa hasil.

4.1 Perancangan Sistem Pengenalan Iris Mata

Sebagai langkah awal pembuatan sistem pengenalan iris maka dirancang sistem pengenalan iris yang berupa proses-proses yang terjadi dalam sistem pengenalan iris. Berikut merupakan perancangan sistem pengenalan iris mata



Gambar 4.1 Perancangan Sistem Pengenalan Iris Mata

Pada Gambar 4.1, merupakan rancangan sistem pengenalan iris mata yang terdiri atas tahap pelatihan, pengujian, dan pencocokan.

a. Tahap Pelatihan

Pada tahap ini citra masukan berupa himpunan citra mata, setiap citra mata dilakukan proses prapengolahan citra yang terdiri atas segmentasi dan normalisasi untuk mengambil bagian iris yaitu daerah *zigzag collarette*, citra *zigzag collarette* dilakukan proses ekstraksi dengan filter Log-Gabor untuk menghasilkan citra filter, hasil dari citra filter kemudian dilakukan pencarian vektor eigen yang signifikan dengan PCA, pada proses ini diakhiri dengan menghitung vektor bobot dengan menggunakan vektor eigen yang signifikan.

b. Tahap Pengujian

Pada tahap ini dilakukan pengujian terhadap citra uji dengan proses yang sama seperti pada tahap pelatihan yaitu melakukan prapengolahan citra dan ekstraksi dengan filter Log-Gabor, pada saat mengekstraksi dengan PCA diambil vektor eigen dari tahap pengujian, kemudian dilanjutkan dengan menghitung vektor bobot dengan menggunakan vektor eigen dari citra latih.

c. Tahap Pencocokan

Pada tahap ini dilakukan pencocokan terhadap vektor bobot citra latih dengan vektor bobot citra uji, kemudian dicari jarak minimal untuk memutuskan kecocokan.

4.2 Perancangan Proses

Perancangan pada sistem pengenalan iris terdiri atas rancangan prapengolahan citra, ekstraksi dan pencocokan. Berikut ini dijelaskan rancangan algoritma dan beberapa hal yang berkaitan dengan sistem pengenalan iris mata.

4.2.1 Rancangan Prapengolahan Citra

Pada proses prapengolahan citra iris mata terdiri atas 4 bagian yaitu

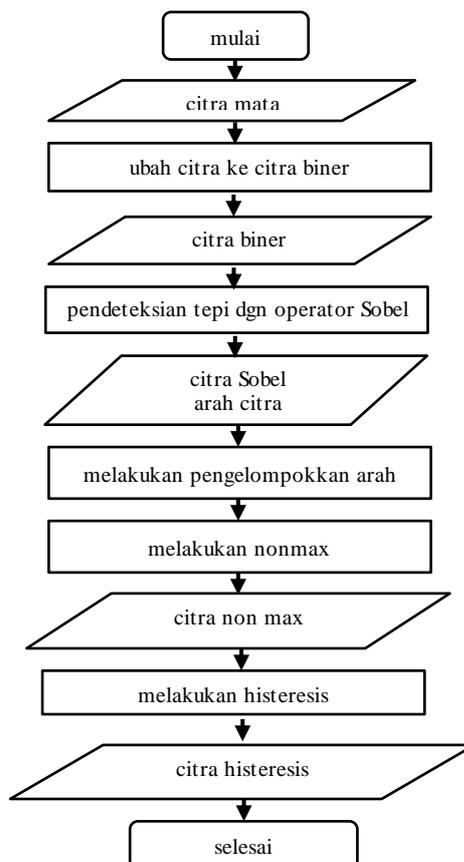
- a. Mendeteksi tepi pupil pada tiap citra iris mata
- b. Mencari pusat dan jari-jari pupil dengan Transformasi Hough Lingkaran

c. Mengisolasi daerah *zigzag collarette*

d. Mengubah bentuk lingkaran ke persegi panjang atau normalisasi iris

4.2.1.1 Rancangan Deteksi Tepi Pupil

Rancangan deteksi tepi pupil adalah merancang proses pendeteksian tepi pada daerah lingkaran pupil. Pupil merupakan daerah berwarna kecoklatan yang berada di tengah-tengah mata, warna pupil sangat dominan dan seragam sehingga bisa terlihat perbedaan antara pupil dengan yang bukan pupil. Pada penelitian ini pendeteksian tepi pupil diperlukan untuk mencari batas perbedaan antara pupil dengan iris, pada database CASIA karakteristik warna pupil sudah seragam untuk semua citra sehingga bisa dilakukan konversi ke citra biner untuk mendeteksi awal pupil tanpa dilakukan filter Gaussian. Kemudian untuk mempercepat pendeteksian tepi pupil maka dilakukan deteksi tepi sehingga terlihat tepi-tepi di sekeliling pupil. Pada Gambar 4.2 merupakan diagram alir deteksi tepi pupil



Gambar 4.2 Diagram Alir Deteksi Tepi Pupil CASIA

Berikut merupakan penjelasan langkah-langkah mendeteksi tepi pupil pada citra mata CASIA:

1. Menginputkan citra iris mata kemudian dilakukan perubahan ke citra biner untuk mendapatkan bentuk pupil dengan batas ambang sebesar 0.3, yang berarti citra dibawah skala 0.3 menjadi warna hitam sedangkan citra diatas skala 0.3 menjadi warna putih
2. Melakukan operator sobel pada citra hitam putih untuk mendapatkan intensitas gradien dan arah tepi
3. Mengelompokkan arah tepi pada citra tepi hasil operator Sobel pada sudut $0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$ sehingga tepi menjadi terfokus pada sudut tertentu.
4. Melakukan proses *nonmaxsup* yaitu piksel tepi hasil operator Sobel yang tidak bernilai maksimal dihilangkan dengan membandingkan 2 piksel tetangga sesuai dengan arah tepi hasil pengelompokkan
5. Meredam piksel yang menyerupai garis terputus dengan menggunakan dua nilai ambang, proses ini disebut histeresis. Pada penelitian ini nilai ambang bawah memiliki skala T_1 dan ambang atas T_2 yang berarti piksel diatas T_2 menjadi piksel tepi bernilai 255 (putih) sedangkan piksel di bawah T_1 bukan merupakan piksel tepi bernilai 0 (hitam), untuk piksel diantara T_1 dan T_2 diberi nilai 255 dengan syarat delapan piksel tetangga bernilai 255. Hasil yang diperoleh merupakan citra biner karena hanya memiliki dua nilai piksel yaitu 0 dan 255. Dalam menentukan batas ambang maka dilakukan percobaan terhadap citra latih pada database CASIA sebanyak 90 buah, percobaan dilakukan dengan mensubstitusi batas ambang kemudian dilakukan pengamatan terhadap hasil deteksi tepi, jika bentuk lingkaran berhasil digambarkan tepat di sekeliling tepi pupil maka batas ambang tersebut dipilih. Berikut pada Tabel 4.1 merupakan hasil percobaan terhadap berbagai nilai ambang

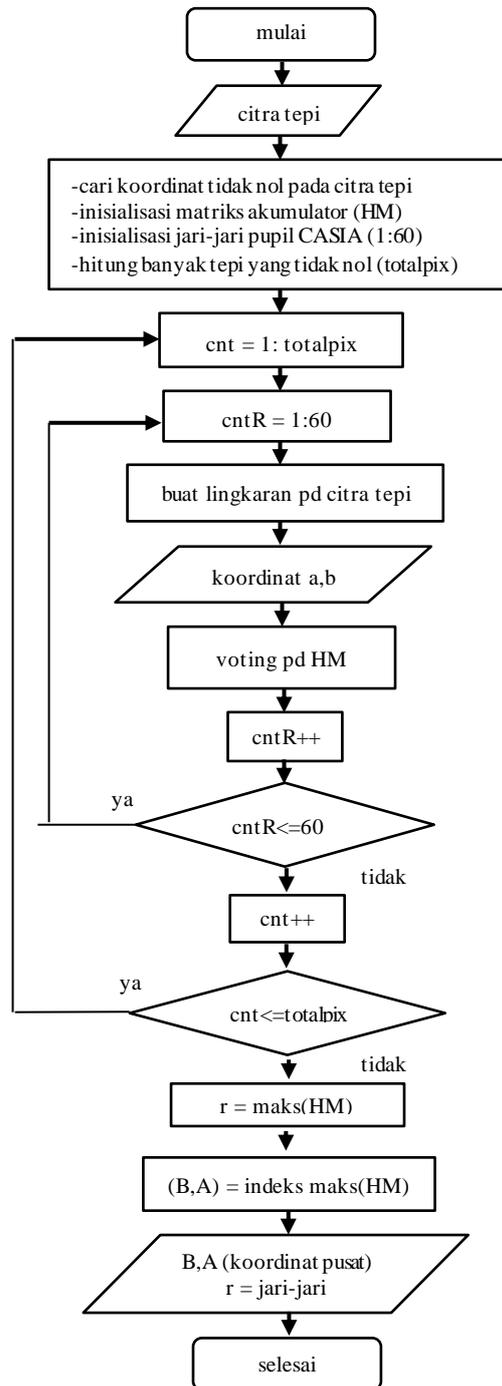
Tabel 4.1 Hasil Percobaan Menentukan Batas Ambang

T_1	T_2	Hasil Percobaan		Waktu
		Tepat	Tidak Tepat	
0.2	0.3	71	19	338.30
0.3	0.4	63	27	341.03
0.4	0.5	72	18	251.16
0.5	0.6	71	19	253.92
0.6	0.7	70	20	239.48
0.7	0.8	81	9	178.09
0.8	0.9	78	12	127.736

Pada Tabel 4.1 terlihat bahwa pada saat batas ambang $T_1 = 0.2$ dan $T_2 = 0.3$ diperoleh banyak tepi pupil yang terdeteksi dengan tepat sebanyak 71 namun membutuhkan waktu kinerja yang cukup lama pada citra latih, sedangkan pada saat $T_1 = 0.8$ dan $T_2 = 0.9$ diperoleh banyak tepi pupil yang berhasil terdeteksi sebanyak 78 citra dengan waktu kinerja lebih cepat daripada batas ambang yang lain. Pada perancangan deteksi tepi pupil dipilih $T_1 = 0.7$ dan $T_2 = 0.8$ karena diperoleh banyak lingkaran pupil pada citra latih yang terdeteksi dengan waktu yang cukup efisien.

4.2.1.2 Rancangan Pencarian Pusat dan Jari-jari Pupil

Rancangan pencarian pusat dan jari-jari pupil adalah merancang proses pencarian terhadap pusat dan jari-jari pupil. Bentuk pupil pada mata menyerupai bentuk geometris lingkaran sehingga batas keseluruhan antara pupil dan iris dapat digambarkan dalam bentuk lingkaran sehingga diperlukan parameter lingkaran yaitu koordinat pusat dan jari-jari lingkaran pupil. Salah satu metode pengolahan citra untuk mencari koordinat pusat dan jari-jari lingkaran pupil adalah Transformasi Hough Lingkaran.



Gambar 4.3 Diagram Alir Pencarian Pusat dan Jari-jari Pupil

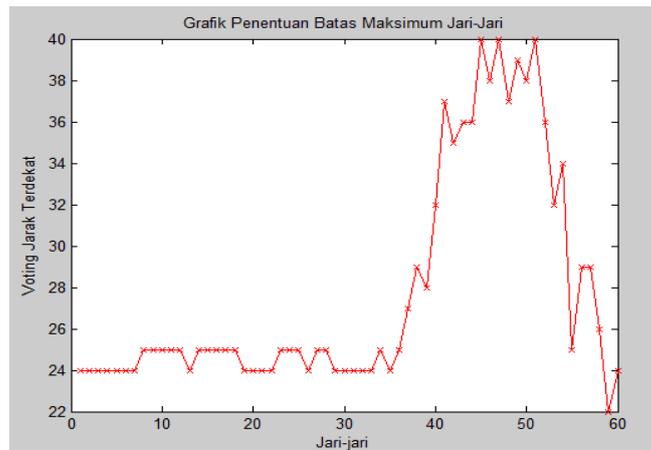
Metode ini memerlukan inputan citra tepi biner untuk menggambarkan lingkaran-lingkaran buatan, kemudian dari lingkaran-lingkaran buatan dicari titik potong maksimal untuk mencari titik pusat. Pada penelitian ini dilakukan

percobaan untuk mencari batas jari-jari maksimal dan minimal dengan Transformasi Hough Lingkaran. Berikut merupakan penjelasan algoritma Transformasi Hough Lingkaran:

1. Mencari koordinat tepi biner yang tidak nol pada citra tepi
2. Mengalokasikan batas jari-jari minimal dan maksimal piksel pada database *CASIA*, nilai inisialisasi pada percobaan ini adalah 1-60 piksel
3. Melakukan voting pada koordinat lingkaran dari tiap-tiap tepi biner yang dilewati dalam matriks akumulator, proses ini dilakukan sesuai batas jari-jari minimal dan maksimal
4. Mencari nilai voting maksimal sebagai jari-jari lingkaran pupil pada matriks akumulator
5. Mencari indeks yang menunjukkan nilai voting maksimal sebagai koordinat pusat lingkaran

Setelah didapatkan koordinat pusat lingkaran, dilakukan perhitungan jarak terdekat dengan koordinat lingkaran pada matriks akumulator dengan jarak *Euclidean*. semakin banyak koordinat lingkaran yang melewati koordinat pusat lingkaran maka jari-jarinya semakin akurat.

Pencarian jari-jari maksimal dilakukan dengan melakukan voting pada database *CASIA* sebanyak 90 citra, pada penelitian yang dilakukan oleh Masek diperoleh jari-jari minimal sebesar 28 piksel. Pada percobaan ini dilakukan pencarian jari-jari maksimal dengan hasil percobaan seperti pada Gambar 4.4 berikut:

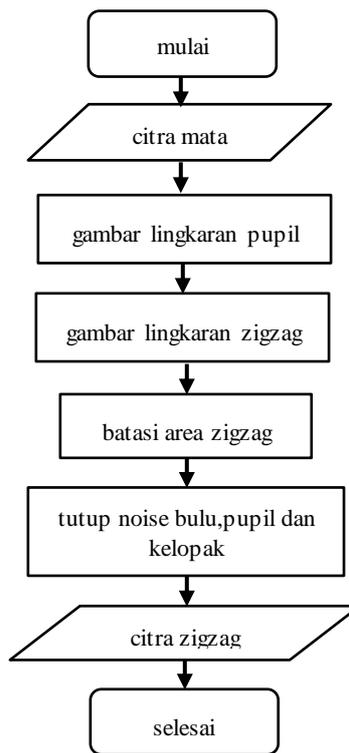


Gambar 4.4 Grafik Penentuan Batas Maksimal Jari-jari

Grafik pada Gambar 4.4 menunjukkan bahwa pada saat jari-jari 45 sampai 51 piksel menunjukkan voting tertinggi yang berarti banyak koordinat lingkaran yang melewati pusat lingkaran pupil. Sehingga pada penelitian ini digunakan pencarian minimal jari-jari sebesar 28 piksel (Masek, 2003) dan maksimal pencarian jari-jari sebesar 51 piksel yang didapat dari percobaan.

4.2.1.3 Rancangan Isolasi Daerah Zigzag Collarette

Rancangan isolasi daerah *zigzag collarette* adalah merancang proses pengisolasian daerah *zigzag collarette* yang merupakan bagian dari iris. Daerah *zigzag collarette* dipilih karena menangkap informasi penting dari pola kompleks iris, selain itu tidak banyak berdampak pada *noise* seperti kelopak mata dan bulu mata. Daerah ini secara umum *concentric* dan dekat dengan pupil sehingga untuk mendeteksi daerah ini digunakan informasi pusat pupil sedangkan jari-jarinya terbatas pada jarak tertentu. Dalam penelitian yang dilakukan oleh Rai (2011) disimpulkan bahwa jarak *zigzag collarette* adalah 24 piksel dari tepi pupil. Dalam penelitian tesis ini, proses isolasi daerah *zigzag collarette* pada citra CASIA masih ada *noise* disekitar pupil yang disebabkan bentuk geometris pupil yang tidak sepenuhnya lingkaran, kemudian masih adanya bulu mata dan kelopak mata yang sedikit tertangkap pada daerah tersebut sehingga perlu dilakukan pengolahan untuk mengatasi *noise*. Pada Gambar 4.5 merupakan gambar diagram alir untuk mengisolasi daerah *zigzag collarette* pada citra mata CASIA.



Gambar 4.5 Diagram Alir Isolasi Daerah Zigzag Collarete

Berdasarkan diagram alir pada Gambar 4.5 maka langkah-langkah mengisolasi daerah *zigzag collarete* adalah sebagai berikut:

1. Lingkaran pada pupil digambar dengan menggunakan parameter pusat (B, A) dan jari-jari pupil (r) untuk mendapatkan koordinat pada lingkaran pupil, dimana x dan y yang didefinisikan sebagai berikut

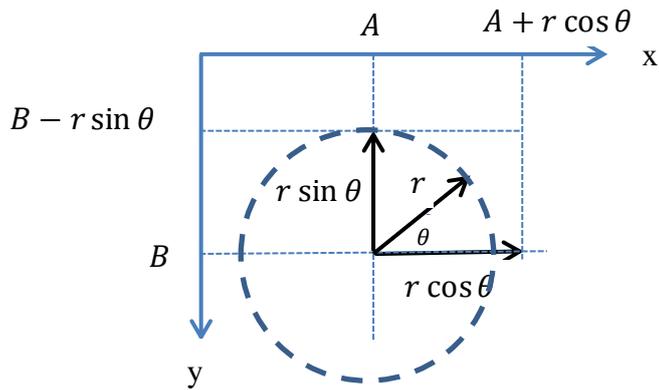
$$x = A + r \cos \theta, 0 \leq \theta \leq 360$$

$$y = B - r \sin \theta, 0 \leq \theta \leq 360$$

Berdasarkan nilai x dan y yang diperoleh, dapat dihitung letak piksel yang mengcover pupil $f(x, y)$ dengan rumus:

$$f(x, y) = 255$$

Nilai 255 dipilih bertujuan agar output gambar yang dihasilkan dapat terlihat dengan jelas perbedaan warna antara pupil dan letak piksel. Dalam koordinat citra bentuk persamaan di atas diilustrasikan pada Gambar 4.6:



Gambar 4.6 Penggambaran Lingkaran dalam Koordinat Citra

2. Dengan cara yang sama seperti pada langkah 1, lingkaran selanjutnya digambar dengan menambahkan jari-jari pupil sebanyak 24 piksel. Ini dilakukan bertujuan untuk menunjukkan batas daerah *zigzag collarette*.
3. Karena pada langkah 1 dan 2 masih terlihat gambar mata secara lengkap maka dilakukan pembatasan dengan hanya mengambil daerah *zigzag collarette* saja. Batas daerah *zigzag collarette* dinyatakan dengan nilai $g(x,y)$ yang berkisar antara

$$A - (r + 24) \leq y \leq A + (r + 24)$$

$$B - (r + 24) \leq x \leq B + (r + 24)$$

4. Hasil yang diperoleh dari langkah 3 masih memiliki *noise* yaitu pupil, bulu mata dan kelopak mata. Untuk mendapatkan daerah yang mengcover pupil, *noise* ini ditutup dengan nilai ambang

$$h(x, y) = \begin{cases} 140, & g(x, y) \leq 100, 200 \leq g(x, y) \leq 255 \\ g(x, y), & \text{lain} \end{cases}$$

Nilai untuk menutup *noise* sebesar 140 didapat dari perhitungan manual rata-rata piksel pada daerah *zigzag collarette*.

4.2.1.4 Rancangan Normalisasi

Rancangan normalisasi adalah rancangan proses mengubah bentuk iris yang diwakili oleh daerah *zigzag collarette* berbentuk lingkaran menjadi bentuk persegi panjang. Tujuan dari proses normalisasi adalah untuk memudahkan proses pemfilteran pada citra iris. Proses normalisasi iris dilakukan dengan memetakan tiap piksel sesuai dengan jari-jari (r) dan sudut (θ) menggunakan transformasi berikut

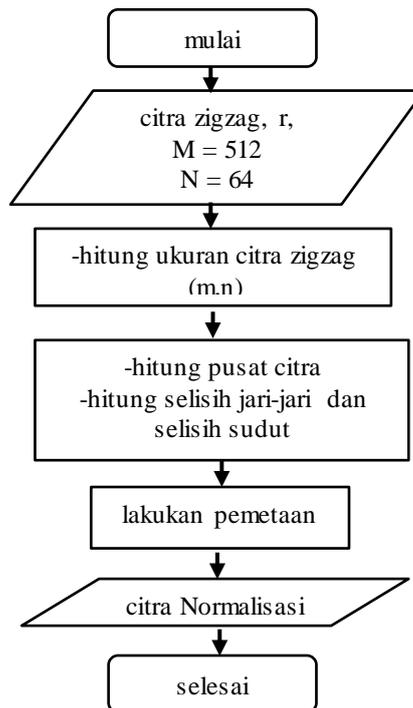
$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta)$$

dengan

$$x(r, \theta) = x_0 + r \cos \theta$$

$$y(r, \theta) = y_0 + r \sin \theta$$

(x_0, y_0) menyatakan pusat koordinat, r menyatakan selisih jari-jari pupil dan jari-jari *zigzag* sedangkan θ menyatakan besar sudut. Berikut ini merupakan diagram alir dari rancangan normalisasi:



Gambar 4.7 Diagram Alir Normalisasi

Penjelasan diagram alir dalam menormalisasi *zigzag collarette* adalah sebagai berikut:

1. Menghitung pusat koordinat citra *zigzag* dengan rumus

$$x_0 = \frac{m + 1}{2}, y_0 = \frac{n + 1}{2}$$

dengan (x_0, y_0) merupakan pusat citra dan (m, n) merupakan ukuran citra *zigzag*

2. Menghitung selisih dari jari-jari pupil dan *zigzag collarette* dengan rumus

$$\Delta r = \frac{r_z - r_p}{M - 1}$$

dengan r_z merupakan jari-jari citra zigzag dan r_p merupakan jari-jari pupil, sedangkan M merupakan panjang dari citra ternormalisasi

3. Menghitung selisih dari sudut dengan rumus

$$\Delta\theta = \frac{2\pi}{N}$$

dengan N merupakan lebar citra ternormalisasi

4. Memetakan tiap piksel sesuai dengan ukuran jari-jari $r_p + \Delta r$ dan ukuran sudut $\Delta\theta$

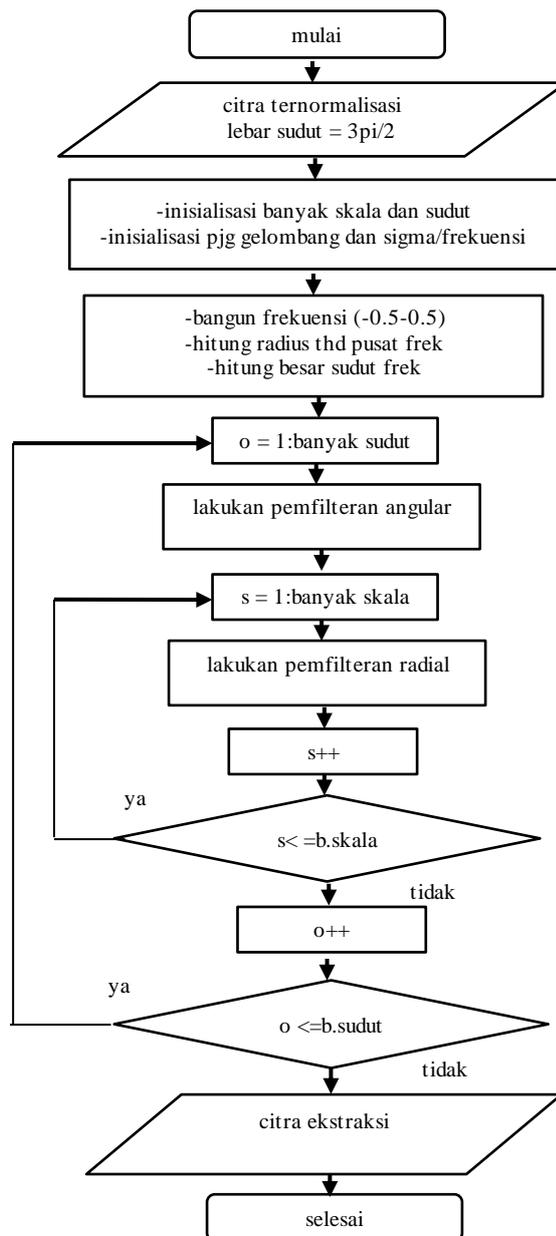
4.2.2 Rancangan Ekstraksi

Rancangan ekstraksi adalah rancangan proses mengekstraksi iris dengan tujuan mendapatkan informasi fitur iris, terdapat dua tahap dalam perancangan ekstraksi yaitu:

1. Melakukan pemfilteran citra dengan filter Log-Gabor pada domain frekuensi
2. Menghitung vektor bobot citra latih dan vektor bobot citra uji dengan *Principal Component Analysis (PCA)*

4.2.2.1 Rancangan Filter Log-Gabor

Rancangan Filter Log-Gabor adalah perancangan proses pemfilteran citra dengan filter Log-Gabor, citra iris yang telah dinormalisasi diubah ke dalam domain frekuensi melalui Transformasi Fourier, transformasi Fourier yang digunakan adalah Transformasi Fourier Cepat atau disebut *Fast Fourier Transform (FFT)*.. Langkah berikutnya adalah membuat matriks filter yang ukurannya sama dengan citra iris ternormalisasi, matriks ini kemudian diisi dengan nilai ternormalisasi dari pusat matriks dengan interval nilai antara -0.5 sampai 0.5 beserta besar sudut terhadap pusat matriks, selanjutnya matriks ini dikonstruksi dengan dua komponen yaitu komponen radial yang berfungsi mengontrol frekuensi filter dan komponen angular yang berfungsi mengontrol arah filter, dua komponen filter ini kemudian dikalikan untuk membentuk matriks filter secara keseluruhan.



Gambar 4.8 Diagram Alir Filter Log-Gabor

Berikut ini penjelasan langkah-langkah yang dilakukan untuk membentuk komponen filter radial beserta nilai parameter yang disarankan oleh (Kovesi, 2001)

1. Menentukan banyak skala filter jika filter yang dibuat lebih dari satu
2. Menentukan nilai pusat frekuensi filter dengan rumus

$$f_0 = \frac{1}{\lambda}$$

dimana λ adalah panjang gelombang dengan nilai 2

3. Menentukan nilai perbandingan dari lebar filter dengan pusat frekuensi $\frac{\sigma}{f_0}$ dengan nilai 0.55, 0.65, dan 0.75
4. Menghitung komponen filter radial dengan rumus

$$G(f) = \exp\left(\frac{-(\log(f / f_0))^2}{2(\log(\sigma / f_0))^2}\right)$$

dengan f adalah frekuensi yang berisi nilai radius ternormalisasi yang dihitung dengan rumus

$$f = \sqrt{x^2 + y^2}$$

dimana nilai x dan y adalah titik ternormalisasi (-0.5-0.5).

Sedangkan langkah-langkah yang dilakukan untuk membentuk komponen filter angular beserta nilai parameter yang disarankan oleh (Kovesi, 2001) adalah sebagai berikut:

1. Menentukan banyak sudut orientasi untuk masing-masing filter
2. Menentukan lebar sudut filter σ_θ dengan nilai $\frac{3}{2}\pi$
3. Menghitung selisih sudut antara sudut orientasi dengan sudut dari tiap-tiap titik ternormalisasi dengan rumus selisih sudut trigonometri

$$\theta - \theta_0 = \tan^{-1}\left(\frac{\sin(\theta - \theta_0)}{\cos(\theta - \theta_0)}\right)$$

4. Menghitung komponen filter angular dengan rumus

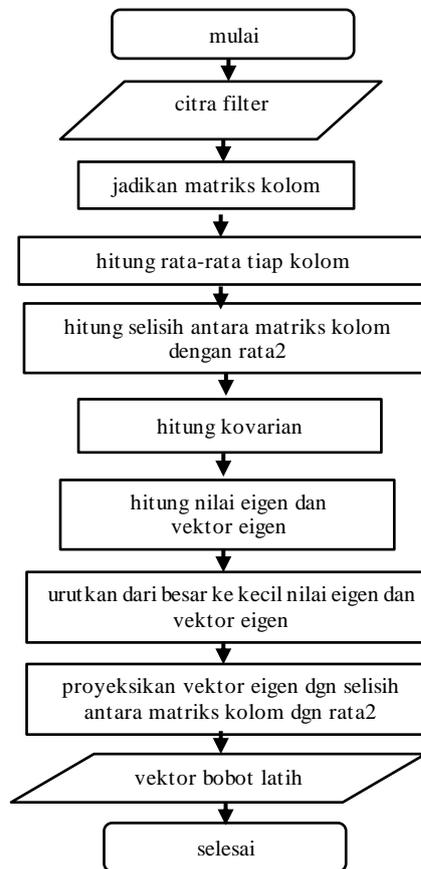
$$G(\theta) = \left(-\frac{(\theta - \theta_0)}{2\sigma_\theta^2}\right)$$

Hasil dari kedua komponen di atas kemudian dikalikan untuk membentuk filter Log-Gabor. Selanjutnya filter Log-Gabor dikalikan dengan citra ternormalisasi yang telah ditransformasi Fourier dan dilakukan invers transformasi Fourier untuk mengubah ke dalam domain spasial. Hasil pemfilteran

citra ternormalisasi adalah nilai absolut dari invers transformasi Fourier atau juga disebut spektrum Fourier.

4.2.2.2 Rancangan PCA

Rancangan PCA terdiri atas tahap pelatihan dan tahap pengujian, pada tahap pelatihan citra masukan yang sudah berupa filter Log-Gabor direduksi dengan PCA menghasilkan vektor eigen, kemudian vektor eigen diambil yang terbesar berdasarkan nilai eigen, selanjutnya dilakukan proyeksi setiap iris terhadap vektor eigen yang dipilih. Hasil proyeksi tiap iris disimpan sebagai data latih. Berikut merupakan diagram alir pelatihan citra filter dengan PCA:



Gambar 4.9 Diagram Alir Tahap Pelatihan PCA

Berikut merupakan penjelasan dari tiap langkah pelatihan PCA:

1. Mengubah bentuk matriks dari tiap-tiap filter menjadi matriks kolom dengan rumus

$$X = \begin{bmatrix} J_{1,1} & J_{1,2} & \dots & J_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{MN,1} & J_{MN,2} & \dots & J_{MN,n} \end{bmatrix}$$

2. Menghitung nilai rata-rata dari tiap-tiap matriks kolom dengan rumus

$$m_k = \frac{1}{n} \sum_{i=1}^n J_{k,i}$$

3. Menghitung selisih matriks rata-rata dengan matriks kolom dengan rumus

$$S_{k,i} = X_{k,i} - m_k$$

4. Menghitung matriks kovarian dengan rumus

$$C = \frac{1}{n-1} S S^T$$

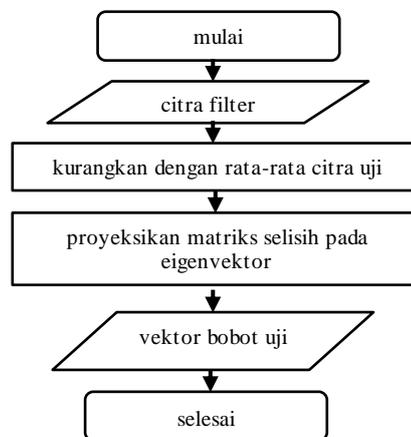
5. Menghitung nilai eigen dan vektor eigen dari matriks kovarian kemudian nilai eigen diurutkan dari terbesar ke terkecil $\lambda_1 > \lambda_2 > \dots \lambda_n$ sedangkan vektor eigen mengikuti posisi nilai eigen setelah diurutkan

$$C v = \lambda v$$

6. Menghitung hasil kali antara transpose vektor kolom eigen (dari nilai eigen terbesar) dengan transpose dari matriks selisih sebagai vektor bobot dengan rumus

$$w = v^T S$$

Sedangkan pada tahap pengujian, citra uji yang sudah berupa filter Log-Gabor dikurangi dengan rata-rata pelatihan, kemudian hasil pengurangan diproyeksikan pada vektor eigen dari pelatihan. Rancangan pengujian terdapat pada diagram alir 4.10 berikut:



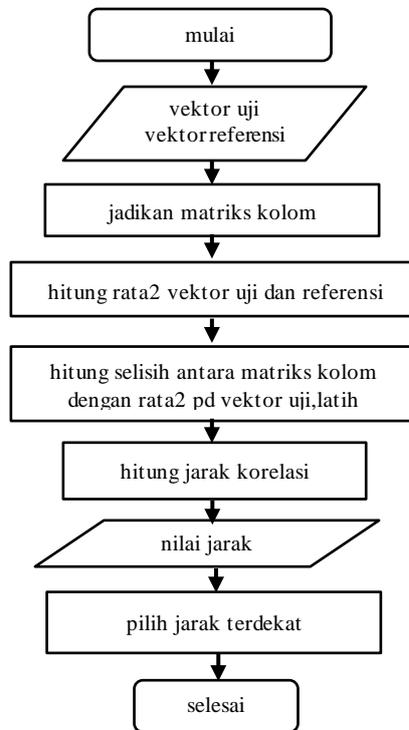
Gambar 4.10 Diagram Alir Tahap Pengujian PCA

4.2.3 Rancangan Pencocokan

Rancangan pencocokan adalah proses merancang pencocokan dengan normalisasi korelasi namun dikarenakan konsep kedekatan maka jarak semakin kecil sehingga perumusan normalisasi korelasi menjadi jarak ternormalisasi korelasi sebagai berikut:

$$d = 1 - \sum_{i=1}^n \sum_{j=1}^m \frac{(p_1(i,j) - \mu_1)(p_2(i,j) - \mu_2)}{nm \sigma_1 \sigma_2}$$

Nilai d adalah jarak ternormalisasi korelasi yang berarti jika vektor uji dan vektor latih memiliki nilai yang minimal maka citra tersebut adalah citra yang sama



Gambar 4.11 Diagram Alir Pencocokan

Penjelasan dari diagram alir pencocokan 4.11 adalah sebagai berikut:

1. Mengubah ke matriks kolom antara citra uji dan citra referensi
2. Menghitung rata-rata citra uji dan citra referensi
3. Menghitung selisih antara matriks kolom dengan rata-rata pada citra uji dan referensi
4. Menghitung jarak dengan rumus normalisasi korelasi
5. Mencari jarak minimal

4.3 Implementasi Proses

Setelah rancangan dibuat maka tahap selanjutnya adalah mengimplementasikan rancangan proses yang telah dibuat ke bahasa pemrograman MATLAB.

4.3.1 Implementasi Pra Pengolahan Citra

Implementasi pra pengolahan citra terdiri dari 4 bagian yaitu

- a. Implementasi deteksi tepi pupil
- b. Implementasi pencarian pusat dan jari-jari
- c. Implementasi isolasi *zigzag collarete*
- d. Implementasi normalisasi

4.3.1.1 Implementasi Deteksi Tepi Pupil

Sesuai dengan rancangan deteksi tepi pupil yang telah dibahas sebelumnya, terdapat beberapa proses yang harus dilakukan untuk mendeteksi tepi pupil sehingga dalam pengimplementasian ke dalam bahasa pemrograman dilakukan pada tiap proses. Berikut ini adalah implementasi deteksi tepi pupil untuk citra CASIA.

Implementasi yang pertama adalah melakukan perubahan ke citra biner, proses ini memerlukan inputan citra iris dan nilai batas ambang sedangkan outputnya adalah citra hitam putih. Pada kode program 1.1 berikut merupakan implementasinya

KODE PROGRAM 1.1-Melakukan perubahan citra ke citra biner

```
function [B] = hitamputih(I,T)
koef = floor(max(max(I))*T);
[baris kolom] = size(I);
B = zeros(baris,kolom);

for i = 1:baris %
    for j = 1:kolom
        if I(i,j)<koef
            B(i,j) = 0;
        else
            B(i,j) = 255;
        end
    end
end
B = logical(B); % Mengubah ke dalam biner
```

Implementasi yang kedua adalah melakukan operator Sobel, proses ini memerlukan inputan citra biner sedangkan outputnya adalah citra tepi Sobel dan arah tepi. Pada kode program 1.2 berikut merupakan implementasinya

KODE PROGRAM 1.2-Melakukan operator Sobel

```
function [G,Theta] = sobel(B)
[baris kolom] = size(B);
G = zeros(baris,kolom);
Theta = zeros(baris,kolom);
for i = 2:baris-1
    for j = 2:kolom-1 % operator sobel dilakukan pd tiap piksel
        gx = (B(i-1,j+1) + 2*B(i,j+1) + B(i+1,j+1) - B(i-1,j-1) -
2*B(i,j-1) - B(i+1,j-1));
        gy = (B(i-1,j-1) + 2*B(i-1,j) + B(i-1,j+1) - B(i+1,j-1) -
2*B(i+1,j) - B(i+1,j+1));
        G(i,j) = sqrt(gx.^2 + gy.^2); % citra tepi
        Theta(i,j) = atan2(gy,gx); % arah tepi (0,180)&(0,-180)
    end
end
```

Implementasi yang ketiga adalah mengelompokkan arah tepi, proses ini memerlukan inputan arah tepi Sobel sedangkan outputnya adalah arah tepi baru yang telah dikelompokkan. Implementasinya pengelompokkan arah pada kode 1.3 berikut

KODE PROGRAM 1.3-Mengelompokkan arah tepi

```
function [ThetaBaru] = kelompokarah(Theta)
[baris kolom] = size(Theta);
ThetaBaru = zeros(baris,kolom);
for i = 1:baris
    for j = 1:kolom
        if Theta(i,j) < 0
            Theta(i,j) = Theta(i,j) + pi;
        end
    end
end
for i = 1:baris
    for j = 1:kolom
        if (Theta(i,j)<pi/8 && Theta(i,j)>=7*pi/8)
            ThetaBaru(i,j) = 0;
        elseif (Theta(i,j)>=pi/8 && Theta(i,j)<3*pi/8)
            ThetaBaru(i,j) = 45;
        elseif (Theta(i,j)>=3*pi/8 && Theta(i,j)<5*pi/8)
            ThetaBaru(i,j) = 90;
        elseif (Theta(i,j)>=5*pi/8 && Theta(i,j)<7*pi/8)
            ThetaBaru(i,j) = 135;
        end
    end
end
```

Implementasi yang keempat adalah proses non maxsup, proses ini memerlukan inputan citra tepi Sobel dan arah baru yang telah dikelompokkan.

Implementasinya pada kode 1.4 berikut:

KODE PROGRAM 1.4-Melakukan proses non max

```
function [Non_max] = nonmaxsup(G,ThetaBaru)
[baris kolom] = size(G);
Non_max = G;
for i = 3:baris-2
    for j = 3:kolom-2
        if ThetaBaru(i,j) == 0
            if (G(i,j)<=G(i,j+1)) || (G(i,j)<=G(i,j-1))
                Non_max(i,j) = 0;
            end
        elseif ThetaBaru(i,j) == 45
            if (G(i,j)<=G(i-1,j+1)) || (G(i,j)<=G(i+1,j-1))
                Non_max(i,j) = 0;
            end
        elseif ThetaBaru(i,j) == 90
            if (G(i,j)<=G(i-1,j)) || (G(i,j)<=G(i+1,j))
                Non_max(i,j) = 0;
            end
        elseif ThetaBaru(i,j) == 135
            if (G(i,j)<=G(i-1,j-1)) || (G(i,j)<=G(i+1,j+1))
                Non_max(i,j) = 0;
            end
        end
    end
end
end
```

Implementasi terakhir deteksi tepi pupil pada CASIA adalah histeresis dengan inputan citra nonmaxsup, output dari proses ini adalah citra deteksi tepi pupil.

KODE PROGRAM 1.5-Melakukan proses histeresis

```
function[BW] = histeresis(Non_max,T1,T2)
h2 = 2;
w2 = 2;
ambang_bawah = T1*max(max(Non_max));
ambang_atas = T2*max(max(Non_max));
Histeresis = Non_max;
[r c] = size(Histeresis);
for i = 1+h2 : r-h2
    for j = 1+w2 : c-w2
        if (Histeresis(i,j) >= ambang_atas)
            Histeresis(i,j) = 255;
        end
        if (Histeresis(i,j) < ambang_atas) && (Histeresis(i,j) >=
ambang_bawah)
            Histeresis(i,j) = 128;
        end
        if ( Histeresis(i,j) < ambang_bawah)
            Histeresis(i,j) = 0;
        end
    end
end
```

```

        End
    end
end
% Penggantian angka 128 menjadi 255 jika 8 tetangga 255
ulang = true;
while ulang
    ulang = false;
    for i = 1+h2 : r-h2
        for j = 1+w2:c-w2
            if (Histeresis(i,j)==128)
                if (Histeresis(i-1,j-1)==255)&&(Histeresis(i-
1,j)==255)&&...
                    (Histeresis(i-1,j+1)==255)&&(Histeresis(i,j-
1)==255)&&...
                        (Histeresis(i,j+1)==255)&&(Histeresis(i+1,j-
1)==255)&&...
                            (Histeresis(i+1,j)==255)&&(Histeresis(i+1,j+1)==255);
                                Histeresis(i,j) = 255;
                                    ulang = true;
                                        end
                                            end
                                                end
                                                    end
                                                        end
BW = Histeresis;

```

4.3.1.2 Implementasi Pencarian Pusat dan Jari-jari

Implementasi pencarian pusat dan jari-jari pada citra CASIA memiliki inputan berupa parameter radius pencarian dan memori penyimpanan matriks akumulator. Pencarian pusat dan jari-jari dilakukan dengan metode Transformasi Hough seperti pada kode program 1.6 berikut:

KODE PROGRAM 1.6-Melakukan Transformasi Hough

```

[y,x]=find(BW); % Mencari koordinat tepi yang
tidak nol
[sy,sx]=size(BW); % Ukuran matriks tepi sy =
banyak baris, sx = banyak kolom

totalpix = length(x); % Banyak tepi yang tidak nol

HM = zeros(sy,sx,24); % Membuat matriks Hough dengan
ukuran sy dan sx dengan r = 24
R = 28:51; % Membuat ukuran jari-jari =
28:50
R2 = R.^2; % Jari-jari pangkat 2 sesuai
rumus lingkaran
sz = sy*sx;

for cnt = 1:totalpix % Hanya piksel yang tidak nol
yang diproses
    for cntR = 1:24 % Jari-jari yang dibuat mulai
r = 1, r = 2,...r = 24

```

```

        b = 1:sy; % Maksimal prediksi pusat b
sesuai dengan banyaknya baris maksimal pada citra
        a = (round(x(cnt) - sqrt(R2(cntR) - (y(cnt) - b).^2))); %
Membuat lingkaran buatan
        b = b(imag(a)==0 & a>0); % Menelusuri baris b yang
memenuhi syarat lingkaran
        a = a(imag(a)==0 & a>0); % Menelusuri kolom a yang
memenuhi syarat lingkaran
        ind = sub2ind([sy,sx],b,a); % Mendapatkan urutan dari a
dan b
        HM(sz*(cntR-1)+ind) = HM(sz*(cntR-1)+ind) + 1; % Voting
sesuai vektor dengan isi 3 dimensi cntR = 1 (dimensi 1), cntR = 2
(dimensi 2)...dst
    end
end

for cnt = 1:24
    H(cnt) = max(max(HM(:, :, cnt))); % Mencari maksimal voting
pada tiap-tiap Hough matriks sebanyak 24 dimensi
end

[maxval, maxind] = max(H); % Mencari voting maksimal
pada matriks H
[y,x] = find(HM(:, :, maxind) == maxval); % Menemukan koordinat dari
Hough matriks yang banyak dilewati
r = maxind + 27; % maxind merupakan jari-
jari dimana voting pada koordinat maksimum
B = y(1); % Koordinat Pusat
A = x(1);
imshow(I); hold on;
plot(mean(A), mean(B), 'xr')
text(mean(A), mean(B), num2str(r), 'color', 'green')

```

Inputan pada kode program diatas adalah citra deteksi tepi pupil sedangkan outputnya adalah koordinat pusat pupil dan jari-jari pupil, selanjutnya parameter lingkaran berupa koordinat pusat pupil dan jari-jari digunakan sebagai inputan pada pengisolasian *zigzag collarette*.

4.3.1.3 Implementasi Isolasi Zigzag Collarette

Implementasi isolasi *zigzag collarette* pada citra CASIA memiliki beberapa rancangan pada tiap proses. Implementasi yang pertama adalah menggambar lingkaran pupil, inputan pada proses ini adalah citra mata asli dengan jari-jari dan koordinat pusat lingkaran pupil sedangkan outputnya adalah citra mata dengan gambar lingkaran pada tepi pupil, berikut ini merupakan kode programnya

KODE PROGRAM 1.7-Menggambar lingkaran pupil

```
I = imread('111.bmp');
B = 135; % Koordinat pusat lingkaran
A = 182;
r = 38; % Jari-jari lingkaran
n = 2000;
theta = (2*pi)/n; % Selisih sudut
sudut = theta:theta:2*pi; % Interval sudut
y = B - r*sin(sudut); % Koordinat baris lingkaran
x = A + r*cos(sudut); % Koordinat kolom lingkaran
for i=1:n
    H(round(y(i)),round(x(i))) = 255; %Batas lingkaran
pupilberwarna putih
end
bts_p = H;
figure(1), imshow(bts_p);
```

Implementasi yang kedua adalah menggambar batas lingkaran *zigzag collarette*, inputan pada proses ini adalah citra mata asli dengan lingkaran pada pupil sedangkan outputnya adalah citra mata asli dengan dua gambar lingkaran yang merupakan daerah *zigzag collarete*, berikut ini merupakan kode programnya

KODE PROGRAM 1.8-Menggambar lingkaran *zigzag collarette*

```
r2 = r + 24; % Batas zigzag collarette
y = B - r2*sin(sudut); % Koordinat baris lingkaran
x = A + r2*cos(sudut); % Koordinat kolom lingkaran
for i=1:n
    H(round(y(i)),round(x(i)))=255;
end
figure(2), imshow(H);
```

Implementasi yang ketiga adalah menampilkan piksel yang membatasi daerah *zigzag collarette*, inputan pada proses ini adalah citra mata asli sedangkan outputnya adalah citra mata dengan ukuran panjang dan lebar yang sama dan telah terbatas hanya daerah *zigzag collarette*, berikut ini merupakan kode programnya

KODE PROGRAM 1.9-Menampilkan hanya bagian zigzag

```
% Menampilkan hanya bagian zigzag dan pupil
box = 2*r2;
G = K((B-box/2):(B+box/2),(A-box/2):(A+box/2));
figure(3), imshow(G)
```

Implementasi yang keempat adalah menutup *noise* berupa pupil, kelopak mata dan bulu mata, inputan pada proses ini adalah citra mata yang terbatas hanya daerah *zigzag collarette*, sedangkan outputnya adalah citra mata dengan pentupan

noise sehingga hanya terlihat bagian *zigzag collarette*, berikut ini merupakan kode programnya.

KODE PROGRAM 1.10-Menutup pupil, bulu mata, dan kelopak mata

```
% Menutup pupil
[tinggi lebar] = size(G);
for i = 1:tinggi
    for j = 1:lebar
        if G(i,j)<=100
            G(i,j) = 140;
        elseif G(i,j) > 200 && G(i,j) <= 255
            G(i,j) = 140;
        end
    end
end
bts_z = G;
figure(4), imshow(bts_z);
```

Output dari proses ini berupa citra yang telah terisolasi daerah *zigzag collarette*, citra ini digunakan sebagai inputan untuk menormalisasi citra.

4.3.1.4 Implementasi Normalisasi

Implementasi normalisasi pada citra mata CASIA adalah inputan citra mata yang telah terisolasi daerah *zigzag collarette* kemudian parameter jari-jari pupil, serta panjang dan lebar normalisasi yang akan dibuat sedangkan output dari proses ini adalah citra persegi panjang dengan ukuran panjang dan lebar dari inputan, berikut ini merupakan kode programnya

KODE PROGRAM 1.11-Menormalisasi iris

```
function [M] = normalisasi(A,r,lebar,panjang)
r2 = r + 24;

% Menampilkan normalisasi
imR = A;
M = lebar;
N = panjang;
rMax = 1;
rMin = r/r2;

[Mr Nr] = size(imR); % ukuran citra persegi panjang
Om = (Mr+1)/2; % Koordinat pusat citra
On = (Nr+1)/2;
sx = (Mr-1)/2; % Faktor skala
sy = (Nr-1)/2;

imP = zeros(M,N); % Inisialisasi panjang dan lebar
delR = (rMax - rMin)/(M-1); % Selisih dari jari-jari [0,lebar-1] dalam range [rMin,rMax] ==> satuan
```

```

delT = 2*pi/N; % Selisih dari sudut
[0:2pi/panjang:2pi-2pi/panjang]

% Proses dilakukan dari tiap-tiap radius dan tiap-tiap sudut
for ri = 1:M
    for ti = 1:N
        r = rMin + (ri - 1)*delR; % Mulai proses untuk r
        t = (ti - 1)*delT; % Mulai proses untuk t
        x = r*cos(t); % Ambil titik x
        y = r*sin(t); % Ambil titik y
        xR = x*sx + Om;
        yR = y*sy + On;
        xR = ceil(xR);
        yR = ceil(yR);
        imP (ri, ti) = imR(xR, yR); % Proses pemetaan
    end
end
P = uint8(imP);
M = P;

```

Output dari proses ini yaitu berupa citra ternormalisasi, proses berikutnya adalah ekstraksi terhadap citra ternormalisasi

4.3.2 Implementasi Ekstraksi

Sesuai dengan rancangan ekstraksi yang telah dibuat maka untuk implementasi ekstraksi terdapat dua tahapan yaitu implementasi filter Log-Gabor dan implementasi PCA. Berikut ini dijelaskan mengenai implementasinya.

4.3.2.1 Implementasi Filter Log-Gabor

Implementasi ekstraksi Filter Log-Gabor pada citra mata CASIA yaitu mengekstrak citra normalisasi dengan pemfilteran. Inputan dari proses ini terdiri atas citra ternormalisasi, banyak skala, banyak orientasi, panjang gelombang, perbandingan lebar filter dengan pusat frekuensi, lebar sudut filter yang telah dirancang sebelumnya, sedangkan outputnya adalah citra ekstraksi. Berikut ini merupakan kode programnya.

KODE PROGRAM 2.1-Mengekstraksi dengan Filter Log-Gabor

```

function [G] = gaborconvolve1(im, nscale, norient, minWaveLength,
mult, sigmaOnf, dThetaOnSigma)
if ~isa(im, 'double')
    im = double(im);
end
[rows cols] = size(im);
imagefft = fft2(im); % Transformasi Fourier
EO = cell(nscale, norient); % Alokasi memori

```

```

F = EO; % Alokasi memori absolut
[x,y] = meshgrid( [-cols/2:(cols/2-1)]/cols, [-rows/2:(rows/2-1)]/rows); % Bangun grid untuk memenuhi image
radius = sqrt(x.^2 + y.^2); % Hitung jari-jari dari pusat frekuensi
radius(round(rows/2+1),round(cols/2+1)) = 1; % Frekuensi pada pusat adalah 1
theta = atan2(-y,x); % Hitung sudut tiap frekuensi
sintheta = sin(theta); % Hitung sin
costheta = cos(theta); % Hitung cos
clear x;
clear y;
clear theta;

thetaSigma = pi/norient/dThetaOnSigma; % Perhitungan bandwidth angular
for o = 1:norient,

    angl = (o-1)*pi/norient; % Membagi sudut sesuai banyaknya norient
    wavelength = minWaveLength; % Frekuensi maksimum ditentukan oleh panjang gelombang terkecil
    ds = sintheta * cos(angl) - costheta * sin(angl);
    dc = costheta * cos(angl) + sintheta * sin(angl);
    dtheta = abs(atan2(ds,dc));
    spread = exp((-dtheta.^2) / (2 * thetaSigma^2));
    for s = 1:nscale,
        fo = 1.0/wavelength;
        logGabor = exp((-log(radius/fo)).^2) / (2 * log(sigmaOnf)^2);
        logGabor(round(rows/2+1),round(cols/2+1)) = 0;
        filter = fftshift(logGabor .* spread);
        EO{s,o} = ifft2(imagefft .* filter);
        F{s,o} = abs(EO{s,o}); % Hitung nilai absolut
        wavelength = wavelength * mult;
    end
end
G = cell2mat(F); % Jadikan satu matriks

```

4.3.2.2 Implementasi PCA

Implementasi PCA pada citra mata CASIA dilakukan pada seluruh data mata, output pada proses ini adalah vektor bobot referensi. Berikut ini merupakan kode pelatihan PCA

KODE PROGRAM 2.2-Proses PCA

```

jpg = 256;
lbr = 32;
image_dims = [lbr jpg];
%Buat data training
data1 = xlsread('database.xlsx','datapca','B1:B90');
index_data = xlsread('database.xlsx','datapca','C1:C90');
b_data = length(data1);

for i = 1:b_data

```

```

img = num2str(data1(i));
img_read1 = [img, '.bmp-filterlog.jpg'];
gmb_mata = imread(img_read1);
gmb_mata = imresize(gmb_mata, [lbr pjg]);
f(:,i) = gmb_mata(:);
end
f = double(f);
% Langkah 1 dan 2: mencari rata-rata dan selisih antara rata-rata
dgn data
images = f;
num_images = b_data;
mean_mata = mean(images, 2);
shifted_images = images - repmat(mean_mata, 1, num_images);
% % Langkah 3 dan 4: hitung nilai eigen dan vektor eigen
[evectors, score, evalues] = princomp(images');
% % Langkah 5: pilih nilai eigen yang signifikan atau bisa disebut
PC
num_eigen = 90;
evectors = evectors(:, 1:num_eigen);
% % Langkah 6: proyeksi citra ke subruang vektor eigen
features = evectors' * shifted_images;

```

Sedangkan implementasi PCA pada tahap pengujian adalah sebagai berikut:

```

% %Buat data uji
[test,data2] = xlsread('database.xlsx','datapca','H179');
img_read1 = num2str(test);
%img_read1 = [data2{1},'.bmp-filterlog.jpg'];
img_read1 = [img_read1, '.bmp-filterlog.jpg'];
input_image = imread(img_read1);
input_image = imresize(input_image, image_dims);
input_image = double(input_image);
% hitung kesamaan antara data uji dengan referensi dgn jarak
korelasi
feature_vec = evectors' * (input_image(:) - mean_mata);

```

4.3.3 Implementasi Pencocokan

Implementasi pencocokan pada citra mata CASIA dilakukan dengan menggunakan normalisasi korelasi, inputan pada proses ini adalah vektor uji dengan vektor bobot referensi sedangkan outputnya berupa indeks minimal, berikut ini merupakan kode programnya:

KODE PROGRAM 3.1-Perhitungan normalisasi korelasi

```

skor = arrayfun(@(n)
pdist2(features(:,n)', feature_vec', 'correlation'), 1:num_images);
% % cari pencocokan yang paling minimal
[match_score, match_ix] = min(skor);
% % tampilkan hasilnya
figure, imshow([uint8(input_image) reshape(images(:,match_ix),
image_dims)]);
title(index_data(match_ix));

```

“Halaman ini sengaja dikosongkan”

BAB V

HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi tentang hasil penelitian dan pembahasan dari tiap tahapan yang dilakukan untuk mengenali pola iris mata yang terdiri atas pra pengolahan citra, ekstraksi ciri dan pencocokan, selanjutnya dilakukan uji akurasi dan analisa kinerja filter Log-Gabor dan *PCA* dalam mengenali individu melalui pola iris mata. Database yang digunakan untuk menguji hasil adalah database mata *CASIA*. Database ini didapatkan dengan mendownload di website <http://www.cbsr.ia.ac.cn/IrisDatabase>.

5.1 Hasil Pra Pengolahan Citra Iris Mata

Proses pra pengolahan citra yang dilakukan terdiri atas empat tahapan yaitu:

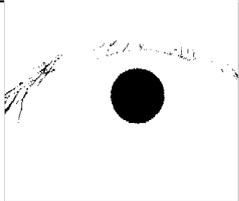
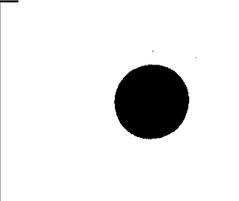
- a. Mendeteksi tepi pupil pada masing-masing database citra
- b. Mencari pusat dan jari-jari pupil dengan Transformasi Hough Lingkaran
- c. Mengisolasi daerah *zigzag collarete*
- d. Mengubah bentuk lingkaran ke persegi panjang atau normalisasi iris

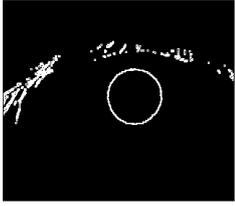
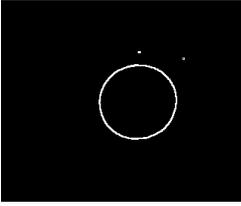
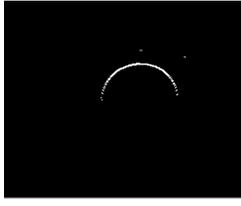
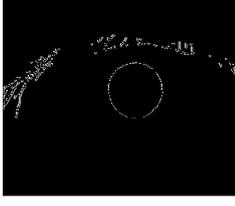
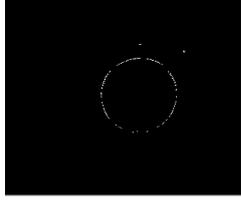
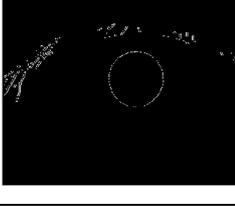
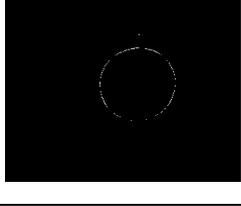
Hasil yang diperoleh dari keempat tahapan dilakukan pada citra mata *CASIA* sebagai berikut:

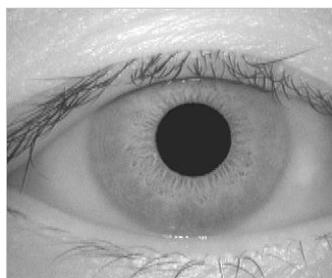
5.1.1 Hasil Deteksi Tepi Pupil

Pada penelitian ini deteksi tepi pupil berhasil mendeteksi tepi-tepi di sekitar pupil. Tabel 5.1 merupakan hasil deteksi tepi pupil pada tiap-tiap proses pada dari citra mata *CASIA*.

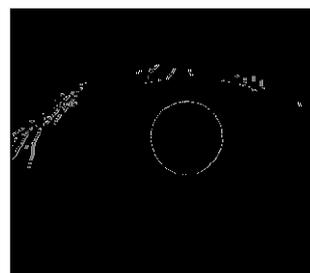
Tabel 5.1 Proses Deteksi Tepi Pupil dari Citra Mata *CASIA*

Proses	111.bmp	211.bmp	Keterangan
Perubahan ke citra biner			Mendeteksi awal daerah pupil

Operator Sobel			Mendeteksi tepi pupil, tepi masih terlihat tebal
Pengelompokkan Arah			Pengelompokkan arah pupil $0^\circ, 45^\circ, 90^\circ, 135^\circ$
Nonmaxsup			Meredam piksel yang non-max, tepi lebih tipis
Histeresis			Mempertegas daerah tepi



(a) Citra Asli



(b) Hasil Deteksi Tepi Pupil

Gambar 5.1 Hasil Deteksi Tepi Pupil dari Citra Mata CASIA

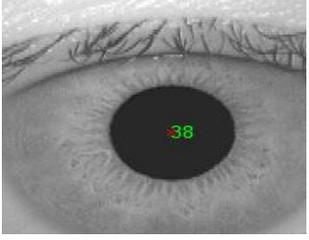
Gambar (a) merupakan citra asli keabuan dari database citra mata CASIA terlihat bahwa terjadi perbedaan warna mencolok antara pupil dengan iris, pada gambar (b) merupakan hasil deteksi tepi pupil terlihat tepi-tepi yang membatasi pupil dan iris, tepi-tepi yang dihasilkan masih belum bisa menggambarkan secara keseluruhan batas antara iris dan pupil sehingga perlu

dilakukan pengolahan citra yang mana bisa menggambarkan batas keseluruhan antara iris dan pupil.

5.1.2 Hasil Pencarian Pusat dan Jari-jari Pupil

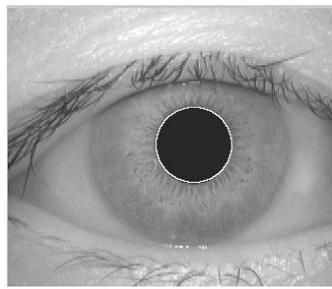
Pada penelitian ini algoritma Transformasi Hough Lingkaran berhasil menemukan koordinat pusat pupil dan jari-jari pupil, seperti pada Tabel 5.2 berikut:

Tabel 5.2 Pusat dan Jari-jari Pupil dari Citra Mata CASIA '111.bmp','211.bmp'

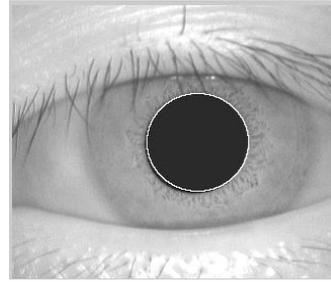
Nama Mata	Gambar Mata	Koordinat Pusat Pupil	Jari-jari
111.bmp		135,182	38
211.bmp		141,183	50

Pada Tabel 5.2 jari-jari lingkaran dari citra mata CASIA dengan kode '111.bmp' ditandai dengan tanda hijau di pusat pupil tertulis 38 sedangkan pada citra mata CASIA '211.bmp' tertulis 50.

Dari parameter yang didapat yaitu jari-jari dan koordinat pusat pupil, penggambaran lingkaran dapat dilakukan dengan memberi nilai 255 (putih) pada setiap piksel (x,y) yang dilewati agar terlihat batas tepi pupil. Pada penelitian ini telah berhasil menggambarkan lingkaran pada citra asli pada citra mata CASIA berikut merupakan hasilnya pada Gambar 5.2.



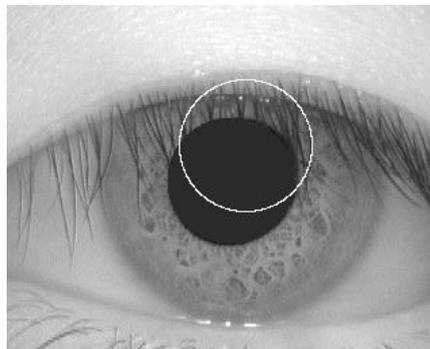
(a) Citra Hough '111.bmp'
 $r = 38$, Pusat (135,182)



(b) Citra Hough '211.bmp'
 $r = 50$, Pusat (141,183)

Gambar 5.2 Penggambaran Lingkaran dalam Citra Asli

Pada tahap ini beberapa citra tidak berhasil menggambarkan bentuk lingkaran hough yaitu citra '413.bmp' yang disebabkan karena banyaknya bulu mata yang terdeteksi oleh deteksi tepi Canny sehingga lingkaran hough menangkap titik maksimal pada bulu mata bukan pada tepi pupil. Berikut ini pada Gambar 5.3 merupakan penggambaran citra lingkaran yang tidak berhasil.

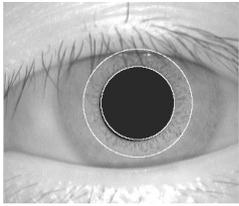
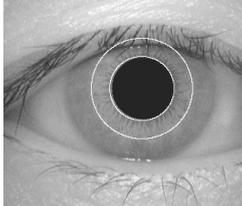
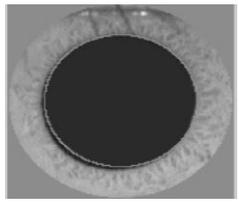
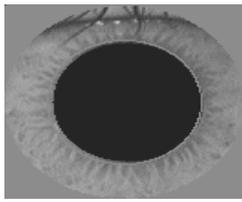
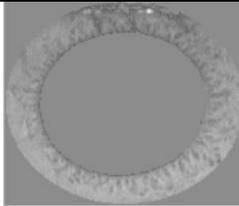
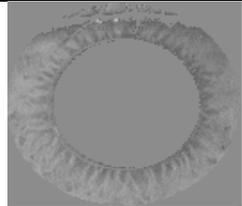
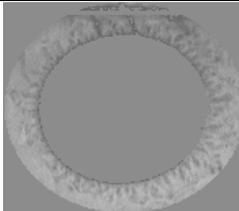
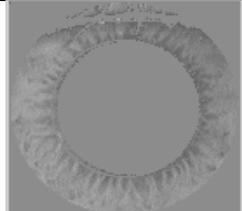


Gambar 5.3 Penggambaran Lingkaran yang Tidak Berhasil dalam Citra Asli ('413.bmp')

5.1.3 Hasil Pengisolasian Daerah *Zigzag Collarette*

Pada Tabel 5.3 merupakan hasil penelitian dari pemrosesan untuk mengisolasi daerah *zigzag collarette*. Hasil yang diperoleh menunjukkan bahwa daerah *zigzag collarette* telah terisolasi dengan cukup baik, beberapa noise seperti kelopak mata dan bulu mata sudah tertutup dengan nilai 140.

Tabel 5.3 Proses Isolasi Daerah Zigzag Collarette dari Citra Mata CASIA dengan nomor '111.bmp' dan '121.bmp'

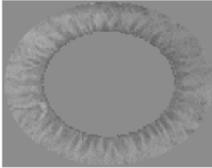
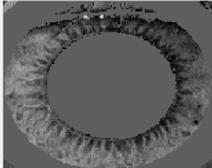
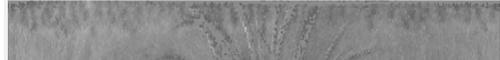
Proses	111.bmp	121.bmp	Keterangan
Gambar batas zigzag			Menggambar batas lingkaran zigzag <i>collarette</i>
Kotak Pembatas			Membatasi daerah zigzag
Tutup Pupil dan Bulu Mata			Menutup noise pupil dan bulu mata
Tutup Kelopak Mata			Menutup kelopak mata

Proses pengisolasian daerah *zigzag collarette* masih belum menggambarkan sepenuhnya daerah iris karena masih terdapat citra *mask* yang digunakan untuk menutup *noise* di luar *zigzag collarette*, sehingga diperlukan pengolahan lanjutan untuk mensegmentasi hanya bagian iris.

5.1.4 Hasil Normalisasi Iris

Hasil penelitian menunjukkan bahwa proses normalisasi berjalan dengan baik dan dapat memetakan tiap piksel *zigzag collarette* ke bentuk persegi panjang. Pada Tabel 5.4 merupakan hasil normalisasi *zigzag collarette* pada citra mata CASIA '111.bmp', '211.bmp'.

Tabel 5.4 Hasil Normalisasi Iris

Citra Mata	Isolasi zigzag collarete	Normalisasi Iris
<p style="text-align: center;">CASIA 111.bmp</p>		
<p style="text-align: center;">CASIA 211.bmp</p>		

5.2 Hasil Ekstraksi Citra Iris Mata

Hasil ekstraksi yang dilakukan terdiri atas:

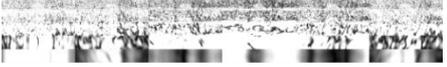
1. Hasil pemfilteran citra dengan filter Log-Gabor
2. Hasil *Principal Component Analysis (PCA)*

5.2.1 Hasil Filter Log-Gabor

Hasil penelitian yang diperoleh menunjukkan bahwa proses filter Log-Gabor dapat berjalan dengan baik, citra filter yang dihasilkan menyerupai tekstur visual. Pada Tabel 5.5 merupakan hasil pemfilteran citra iris dengan filter Log-Gabor pada domain frekuensi.

Tabel 5.5 Hasil Pemfilteran Iris

Citra Mata	Normalisasi Iris	Filter Log-Gabor
<p style="text-align: center;">CASIA 121.bmp</p>		

Citra Mata	Normalisasi Iris	Filter Log-Gabor
CASIA 211.bmp		

5.2.2 Hasil *Principle Component Analysis (PCA)*

Hasil penelitian PCA terdiri atas vektor bobot pelatihan dan vektor bobot pengujian, vektor bobot pelatihan merupakan kumpulan vektor yang mewakili dimensi citra itu sendiri, vektor ini didapatkan dengan mengalikan vektor eigen yang terpilih dengan selisih citra antara citra rata-rata latih dengan citra asli. Berikut pada Tabel 5.6 merupakan hasil vektor bobot pada citra mata CASIA.

Tabel 5.6 Hasil Vektor Bobot Pelatihan dengan 10 Vektor Eigen

Vektor Bobot 1	Vektor Bobot 2	Vektor Bobot 3	Vektor Bobot 4	Vektor Bobot 5	...
-657.057	-650.424	-673.697	2953.279	759.8578	...
1867.001	651.5869	1372.915	-891.863	238.8916	...
496.5394	722.126	305.948	-388.68	2015.781	...
-433.197	-334.683	-345.024	127.944	1101.508	...
294.0376	-193.841	-315.223	1747.612	-207.839	...
395.5381	943.4668	251.2902	443.1719	78.016	...
455.9704	-745.383	-420.432	-1386.43	1083.286	...
801.4418	-304.743	-16.5552	1442.065	-656.845	...
-649.33	55.16853	762.1528	1150.089	-756.42	...
674.3945	73.25764	564.3568	123.0537	-1644.42	...

Pada Tabel 5.6 tiap-tiap kolom menyatakan vektor bobot pada masing-masing citra, vektor bobot ini disimpan untuk kemudian dicari jarak terhadap vektor bobot uji, vektor bobot uji didapat dengan mengalikan antara transpose vektor eigen dengan selisih antara citra uji dengan citra rata-rata latih. Pada Tabel 5.7 merupakan contoh hasil perhitungan jarak vektor bobot uji dengan vektor bobot latih

Tabel 5.7 Hasil Perhitungan Vektor Bobot Uji dengan Vektor Bobot Latih

Jarak	Vektor Bobot 1	Vektor Bobot 2	Vektor Bobot 3	Vektor Bobot 4	Vektor Bobot 5	...
Vektor Bobot Uji	0.281876	0.281999	0.616601	1.645031	0.717289	...

Pada Tabel 5.7 menunjukkan bahwa vektor bobot uji memiliki nilai minimal terhadap vektor bobot satu yang berarti citra uji sama dengan citra yang diwakili oleh vektor bobot satu.

Pada pengujian filter Log-Gabor PCA terhadap citra mata CASIA dilakukan pengujian terhadap 210 citra mata dari 30 orang yang berbeda. Pada proses pelatihan menggunakan 90 citra masing-masing 3 citra dari 30 orang yang merupakan pengambilan pada sesi pertama, selanjutnya pada proses pengujian menggunakan 120 citra masing-masing 4 citra dari 30 orang yang merupakan bagian dari citra yang dilatih. Pengujian dilakukan dengan mengambil semua vektor bobot pelatihan kemudian dilakukan pencocokan terhadap vektor bobot uji untuk mengetahui besar akurasi. Pengambilan vektor eigen yang mana menghasilkan vektor bobot pada tiap-tiap citra mempengaruhi akurasi pengenalan pada tiap iris. Pada Tabel 5.8 merupakan hubungan antara jumlah vektor eigen yang diambil dengan hasil akurasi pengenalan pada tiap iris

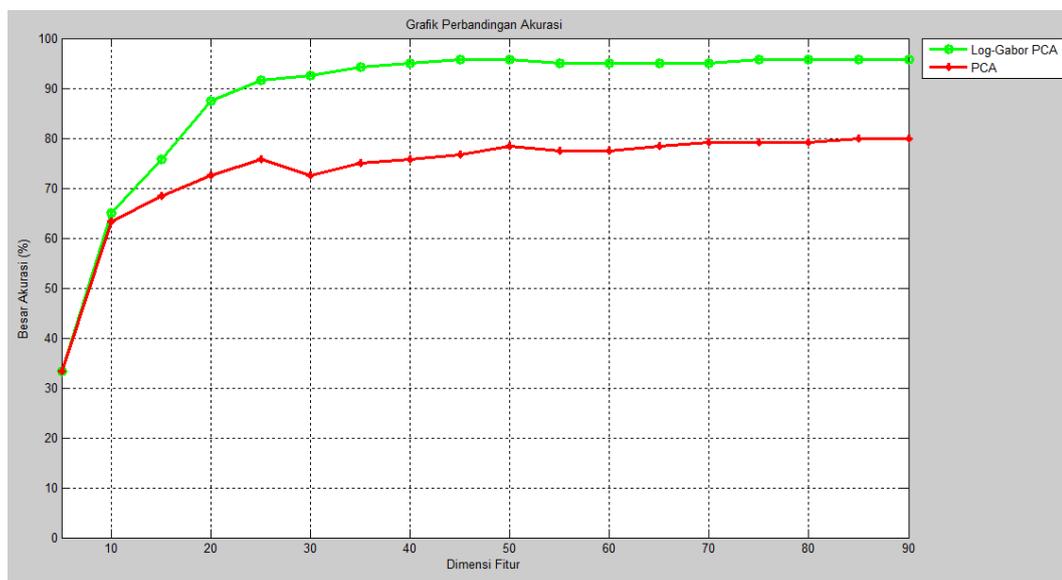
Tabel 5.8 Hasil Akurasi Terhadap Jumlah Vektor Eigen

No	Jumlah Vektor Eigen	Akurasi
1	5	33.33%
2	10	65%
3	15	75.83%
4	25	91.67%
5	45	95.83%

Pada Tabel 5.8 menunjukkan bahwa banyak vektor bobot yang diambil mempengaruhi hasil akurasi, penggunaan vektor eigen yang signifikan dengan nilai eigen terbesar menghasilkan akurasi yang optimal, hal ini terlihat seperti

pada Tabel 5.8, penggunaan 45 vektor eigen saja menghasilkan besar akurasi yang mencapai 95.83%.

Selanjutnya dilakukan analisa terhadap penggunaan metode Filter Log-Gabor PCA dengan metode PCA saja pada pengenalan iris pada database CASIA, analisa dilakukan dengan membandingkan akurasi dan penggunaan jumlah vektor eigen. Pada Gambar 5.4 merupakan perbandingan metode filter Log-Gabor PCA dan PCA saja



Gambar 5.4 Perbandingan Metode Filter Log-Gabor PCA dengan PCA

Pada Gambar 5.4 garis hijau merupakan metode filter Log-Gabor PCA sedangkan garis merah merupakan metode PCA, pada saat penggunaan 10 vektor eigen kedua metode menunjukkan akurasi yang sama namun setelah penggunaan 20 vektor eigen akurasi yang didapat pada filter Log-Gabor PCA sudah mencapai 80% sedangkan PCA masih berada di bawah 80 %, akurasi maksimal yang didapat pada metode PCA mencapai 80% sedangkan filter Log-Gabor PCA sudah mencapai 95 %. Hal ini menunjukkan bahwa metode filter Log-Gabor PCA sangat baik karena akurasi maksimal yang didapat mencapai 95 %.

Analisa selanjutnya adalah mengetahui akurasi dan waktu kinerja dari beberapa metode, pada penelitian ini dibandingkan antara metode filter Log-Gabor saja, PCA saja dan gabungan antara filter Log-Gabor dan PCA.

Tabel 5.9 Hasil Akurasi Terhadap Berbagai Metode

No	Metode	Akurasi
1	PCA	76.67 %
2	Filter Log-Gabor	95.83 %
3	Filter Log-Gabor dan PCA	95.83 %

Pada Tabel 5.9 akurasi tertinggi mencapai 95.83% oleh filter Log-Gabor dan gabungan metode filter Log-Gabor dan PCA dengan 45 vektor eigen, hal ini disebabkan karena tekstur yang dihasilkan melalui filter Log-Gabor dapat menangkap informasi citra secara lokal sehingga dapat membedakan antara citra satu dengan citra yang lain.

Tabel 5.10 Waktu Kinerja Pengenalan Terhadap Berbagai Metode

No	Metode	Waktu Rata-rata
1	PCA	1.88433
2	Filter Log-Gabor	2.37785
3	Filter Log-Gabor dan PCA	2.15064

Pada Tabel 5.10 waktu pengenalan rata-rata terbaik didapat oleh metode PCA, hal ini disebabkan karena pada proses PCA terjadi pengurangan dimensi fitur pada citra ternormalisasi saat proses pencocokan, fitur-fitur yang diwakili vektor eigen terpilih saja yang dipakai sehingga waktu pengenalan menjadi lebih efisien, sedangkan pada proses filter Log-Gabor menggunakan semua informasi tekstur yang ada pada citra sehingga pada saat proses pencocokan, dimensi yang digunakan sebanyak ukuran piksel citra hal ini menyebabkan waktu pencocokan lebih lama daripada metode PCA, sedangkan pada gabungan metode filter Log-Gabor dan PCA didapat waktu kinerja berada diantara metode PCA saja dan filter Log-Gabor saja, hal ini disebabkan proses yang terjadi menggunakan filter Log-Gabor menghasilkan semua informasi tekstur, namun pada saat proses pencocokan terjadi pengurangan dimensi sehingga waktu total menjadi lebih cepat daripada dengan menggunakan filter Log-Gabor saja.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil uji coba dengan data CASIA diperoleh kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil melakukan prapengolahan citra yaitu pendeteksian tepi pupil dengan menggunakan metode deteksi tepi *Canny* dan Transformasi Hough dengan ketepatan 90%, pengisolasian daerah *zigzag collarette*, penutupan bulu mata dengan nilai sebesar 140 piksel.
2. Batas ambang yang diperoleh agar pendeteksian tepi pupil mencapai ketepatan maksimal adalah $T_1 = 0.7$ dan $T_2 = 0.8$
3. Gabungan metode filter Log-Gabor dan PCA menghasilkan akurasi pengenalan dari citra iris mencapai 95.83% dengan menggunakan 45 vektor eigen, sedangkan waktu yang dibutuhkan untuk pengenalan lebih cepat 0.12 detik daripada pengenalan dengan metode filter Log-Gabor.

6.2 Saran

Berdasarkan hasil penelitian, beberapa hal yang dapat dijadikan sebagai bahan masukan untuk pengembangan lebih lanjut:

1. Pada tahap akuisisi perlu dilakukan pengujian dengan citra mata berwarna sehingga pengembangan aplikasi pengenalan iris menjadi lebih luas.
2. Pada tahap prapengolahan citra masih ada bagian iris yang tertutup bulu mata sehingga perlu dilakukan penelitian lanjut terhadap berbagai metode segmentasi seperti operasi morfologi, Transformasi Hough Parabola agar bagian iris terbebas dari *noise* seperti bulu mata dan kelopak mata.
3. Pada saat proses pencocokan hanya sebatas dilakukan terhadap citra iris baru yang merupakan bagian dari citra yang sudah dilatih, sehingga untuk penelitian selanjutnya dapat digunakan citra iris baru yang bukan merupakan bagian dari citra yang sudah dilatih.

DAFTAR PUSTAKA

- Adler F.H. (1965), "*Physiology of the Eye: Clinical Application*". The CV. Mosby Company, London.
- Aggrawal, H. dan Raman Maini.(2004).“Study and Comparison of Various Image Edge Detection Techniques”. International Journal Image Processing (IJIP), Volume (3): Issue (1).
- Ali,MH Jafar, Aboul Ella Hassanenian, “An Iris Recognition System To Enhance E-Security Environment Based On Wavelet Theory”, AMO Advanced Modelling and Optimization ,Volume 5 Number 2, 2003.
- Bank ATM by Iris.< <http://www.youtube.com/watch?v=NDi3UKrWybU>>.
- CASIA iris image database.<<http://www.cbsr.ia.ac.cn/IrisDatabase>>.
- Chowhan, S.S, G. N Shindhe. (2009).” Evaluation of Statistical Feature Encoding Techniques on Iris Images”. ICCES, vol.9, no.1, pp.67-73
- Daugman, J. (2004).”How Iris Recognition Works”. IEEE Transactions On Circuits And Systems For Video Technology, Vol. 14, No. 1, January 2004
- Duda, Richard O, Peter E. Hart. (1972)” Use of the Hough Transformation To Detect Lines and Curves in Pictures, Communication of the ACM, Vol 15, No 1, January 1972.
- Field, D.J. (1987).“Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells”.Journal of the Optical Society of America A, Vol. 4, No. 12, pp 2379-2394, December 1987.
- Gonzalez, R.C., dan R.E.Woods. (2009). "*Digital Image Processing Using MATLAB 2nd Edition*". Prentice Hall, New Jersey.
- Jain,Anil K., Sharath Pankanti dan Salil Prabhakar. (2003),“Biometric Recognition: Security and Privacy Concerns”. IEEE Security & Privacy March 2003.
- Kadir, Abdul, Adhi Susanto. (2011).”*Teori dan Aplikasi Pengolahan Citra Digital*”, Penerbit Andi, Yogyakarta.
- Kovesi, Peter. (2001), MATLAB Functions for Computer Vision and Image Analysis <http://www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html>

- Ma, Li, Yunhong Wang dan Tieniu Tan. (2002), "Iris Recognition Based on Multichannel Gabor Filter". ACCV2002: The 5th Asian Conference on Computer Vision, 23--25 January 2002.
- Ma, Li, Yunhong Wang dan Tieniu Tan. (2002), "Iris Recognition Using Circular Symmetric Filter". Proceedings of the 16th International Conference on Pattern Recognition, Vol. II, pp. 414–417, 2002.
- Masek, L. (2003), "Recognition of Human Iris Pattern for Biometric Identification". <http://www.csse.uwa.edu.au/pk/student/project/libor>.
- Murmu, Rabindra Kumar. (2009), "*Image Segmentation Using Hough Transform*". National Institute of Technology, Rourkella.
- Patil, Pravin S. S.R Kolhe. (2012), "The Comparison of Iris Recognition using Principal Component Analysis, Log-Gabor and Gabor Wavelets". International Journal of Computer Applications Vol 43-No 1. April 2012
- Putra, Darma. (2009), "*Konsep Dasar, Teknik Analisis Citra dan Tahapan Membangun Aplikasi Sistem Biometrika*". Penerbit Andi, Yogyakarta.
- Rai, Himansu. (2014), "Iris recognition using combined support vector machine and Hamming distance approach". Expert Systems with Applications 41 (2014) 588–593
- Wildes, R.(1997), "Iris recognition: an emerging biometric technology". Proc. IEEE **85**(9), 1348–1363 (1997)
- Yao, Peng, J. Li, X. Ye, Z. Zhuang dan B. Li. (2006), "Analysis and Improvement of An Iris Identification Algorithm" Proceeding of the 18th ICPR vol. 4, pp. 362–365. IEEE, Piscataway, NJ
- Yao, Peng, J. Li, X. Ye, Z. Zhuang dan B. Li (2006), "Iris Recognition Algorithm Using Modified Log-Gabor Filters," ICPR 2006 - 18th Intl. Conf. on Pattern Recognition, vol.4, pp.461-464
- Yu, Qingwen, Xiaoyu Tan dan Hong Wang. (2015), "A Face Recognition Method Based on Total Variation Minimization and Log-Gabor Filter". International Conference on Electromechanical Control Technology and Transportation (ICECTT 2015)

LAMPIRAN A HASIL IDENTIFIKASI PENGENALAN IRIS

Nama Mata	HASIL IDENTIFIKASI		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
121	1	1	1
122	1	1	1
123	1	1	1
124	1	1	1
221	0	1	1
222	1	1	1
223	1	1	1
224	1	1	1
321	1	1	1
322	1	1	1
323	0	1	1
324	0	1	1
421	0	0	1
422	0	1	1
423	0	1	1
424	1	1	1
521	1	1	1
522	0	1	1
523	1	1	1
524	1	1	1
621	1	1	1
622	0	1	1
623	1	1	1
624	0	1	1
721	0	0	1
722	1	1	1
723	1	1	1
724	1	1	1
821	1	1	1
822	1	1	1
823	1	0	0
824	1	1	1
921	1	1	1
922	1	1	1
923	1	1	1
924	1	1	1
1021	1	1	1
1022	0	1	1
1023	1	1	1
1024	1	1	1
1121	1	1	1
1122	1	1	1
1123	1	1	1

Nama Mata	HASIL IDENTIFIKASI		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
1124	1	1	1
1221	1	1	0
1222	1	1	1
1223	1	1	1
1224	1	1	1
1321	1	1	1
1322	1	1	1
1323	1	1	1
1324	0	1	1
1421	1	1	1
1422	1	1	1
1423	1	1	1
1424	1	1	1
1521	1	1	1
1522	1	1	1
1523	1	1	1
1524	1	1	1
1621	1	1	1
1622	1	1	1
1623	1	1	1
1624	1	1	1
1721	1	1	1
1722	1	1	1
1723	1	1	1
1724	1	1	1
1821	0	0	1
1822	1	1	1
1823	1	1	1
1824	1	1	1
1921	1	1	1
1922	1	1	1
1923	1	1	1
1924	0	1	1
2021	1	1	1
2022	1	1	1
2023	0	0	0
2024	1	1	1
2121	0	1	1
2122	0	1	1
2123	1	1	1
2124	0	1	1
2221	1	1	1
2222	1	1	1
2223	0	1	0
2224	0	1	1
2321	1	1	1

Nama Mata	HASIL IDENTIFIKASI		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
2322	1	1	1
2323	1	1	1
2324	1	1	1
2421	1	1	1
2422	1	1	1
2423	1	1	1
2424	0	1	1
2521	1	1	1
2522	0	1	1
2523	1	1	1
2524	1	1	1
2621	1	1	1
2622	1	1	1
2623	1	1	1
2624	0	1	1
2721	1	1	1
2722	0	1	1
2723	1	1	1
2724	1	1	1
2821	1	1	1
2822	1	1	1
2823	0	1	1
2824	0	1	1
2921	0	1	0
2922	1	1	1
2923	1	1	1
2924	1	1	1
3021	0	1	1
3022	1	1	1
3023	1	1	1
3024	1	1	1
Akurasi	76.67%	95%	95%

Keterangan: 1 menyatakan dikenali benar, 0 menyatakan dikenali salah

“Halaman ini sengaja dikosongkan”

LAMPIRAN B WAKTU RATA-RATA PENGENALAN IRIS

Nama Mata	WAKTU		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
121	2.4603114	3.371066	2.945693
122	0.8642797	1.406349	1.233346
123	1.099131	1.625844	1.445023
124	0.9817478	1.523991	1.329166
221	0.860821	1.392627	1.222566
222	0.8952541	1.41833	1.247054
223	0.8461777	1.397482	1.223632
224	0.8792355	1.381036	1.2354
321	1.7657487	2.246814	2.031281
322	2.2124317	2.645915	2.414853
323	2.0714599	2.524305	2.335801
324	1.9761213	2.442968	2.267449
421	4.6147365	5.056616	4.793979
422	4.3590117	4.877158	4.536189
423	3.4247672	3.801422	3.562232
424	5.0412684	5.400374	5.028484
521	2.9180204	3.539411	3.118981
522	2.6540259	3.263146	2.875959
523	2.3057809	2.818072	2.542808
524	2.7479651	3.306669	2.969333
621	0.7359187	1.305374	1.093951
622	0.733449	1.324281	1.115905
623	0.692956	1.249867	1.096326
624	0.7277163	1.255125	1.109546
721	1.8357999	2.486565	2.142009
722	1.5665039	2.147834	1.866942
723	1.6577746	2.235999	1.955295
724	1.4781088	2.050492	1.789063
821	2.5721566	3.178317	2.793413
822	2.4477835	2.989351	2.703414
823	1.7272949	2.276337	2.035172
824	1.7500174	2.314802	2.028384
921	0.7890393	1.326922	1.155171
922	0.757492	1.292144	1.129596
923	0.7124417	1.265263	1.095994
924	0.7242824	1.264418	1.091487
1021	2.0522752	2.605429	2.340108
1022	1.7666559	2.307982	2.088909
1023	2.1054598	2.631774	2.422236
1024	2.0841924	2.636136	2.412486
1121	5.5677481	6.028935	5.602614
1122	5.3868317	5.775247	5.330187
1123	5.4417179	5.750372	5.297447

Nama Mata	WAKTU		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
1124	5.1896608	5.546913	5.082996
1221	2.521003	2.948043	2.675967
1222	2.0542078	2.485638	2.247524
1223	2.7967909	3.269865	2.969255
1224	2.9349153	3.36493	3.034176
1321	0.7855308	1.275699	1.109993
1322	0.8145137	1.331277	1.150238
1323	0.842394	1.316827	1.134549
1324	1.0185658	1.503767	1.330665
1421	0.7374007	1.264835	1.069589
1422	0.71734	1.229833	1.058039
1423	0.6839496	1.208263	1.037234
1424	0.7290197	1.265077	1.068932
1521	3.543463	4.023434	3.607415
1522	1.824949	2.30902	2.085334
1523	2.6110352	3.073438	2.797479
1524	2.5529335	3.045782	2.738089
1621	1.0490166	1.542666	1.364203
1622	1.0546168	1.542322	1.374934
1623	0.9682496	1.451494	1.308907
1624	0.9327082	1.400896	1.285334
1721	0.6647535	1.140518	1.03763
1722	0.7966393	1.303579	1.167018
1723	0.8369371	1.316509	1.200078
1724	0.8650468	1.364955	1.255456
1821	0.884951	1.375128	1.231416
1822	1.1569632	1.601554	1.501808
1823	0.8836276	1.365434	1.237219
1824	0.8191773	1.304253	1.180559
1921	1.8670637	2.304558	2.118587
1922	1.2402044	1.685053	1.546381
1923	1.1145529	1.582296	1.433437
1924	1.0788917	1.53889	1.400015
2021	1.9281526	2.367511	2.153469
2022	1.5364158	1.966165	1.804619
2023	2.0560614	2.518545	2.306668
2024	1.6936465	2.17949	1.974343
2121	1.2758169	1.747488	1.572356
2122	1.4282125	1.850783	1.711144
2123	1.5252451	1.956232	1.790501
2124	1.3139225	1.769651	1.622621
2221	2.8344191	3.244202	2.994288
2222	2.2104691	2.655401	2.437842
2223	3.1714772	3.593178	3.311884
2224	1.4881196	1.953695	1.780852
2321	5.6679632	6.036741	5.654916

Nama Mata	WAKTU		
	PCA	FILTER LOG-GABOR	FILTER LOG-GABOR PCA
2322	5.9753322	6.296085	5.885801
2323	6.2997554	6.601333	6.136964
2324	6.4152991	6.700717	6.182487
2421	0.9824147	1.471463	1.306817
2422	0.7790191	1.265974	1.114867
2423	0.7892697	1.292453	1.127265
2424	0.9403953	1.403084	1.245891
2521	1.0981948	1.570285	1.365245
2522	1.9305763	2.411598	2.154563
2523	0.8749925	1.377718	1.209302
2524	1.2429527	1.728324	1.566306
2621	2.5708103	3.071647	2.776447
2622	1.9398083	2.456138	2.204065
2623	2.1720719	2.802742	2.438611
2624	1.9769983	2.528908	2.222313
2721	1.2782467	1.808472	1.629281
2722	1.2294798	1.741673	1.580919
2723	1.5681035	2.107931	1.928006
2724	1.5901745	2.10209	1.872632
2821	0.8986772	1.407983	1.237702
2822	0.8343757	1.357058	1.191807
2823	0.6604754	1.158471	1.035628
2824	0.6201029	1.109845	0.99697
2921	1.1483315	1.635871	1.502542
2922	1.204401	1.679311	1.512662
2923	1.0587757	1.53023	1.389625
2924	1.0670446	1.54912	1.425599
3021	1.9037828	2.377153	2.170453
3022	3.7562861	4.242735	3.914905
3023	3.1782976	3.655241	3.298887
3024	1.1423661	1.640223	1.442348
Waktu Rata-Rata	1.884331	2.377852	2.150648

“Halaman ini sengaja dikosongkan”

BIODATA PENULIS



Lahir pada tahun 1988 di kota kecil Sidoarjo, tepatnya 25 Agustus penulis memulai awal kehidupan di dunia. Selama hidupnya, penulis telah menyelesaikan pendidikan formalnya, yaitu SD Negeri Sidokumpul 1 Sidoarjo, SLTP Negeri 6 Sidoarjo, SMAK Untung Suropati Sidoarjo, dan akhirnya diterima sebagai mahasiswa program Strata I Matematika Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2006 melalui SPMB. Penulis lulus Strata I pada tahun 2011. Pada tahun 2014 melanjutkan Strata II Matematika Institut Teknologi Sepuluh Nopember Surabaya. Penulis akhirnya berhasil menyelesaikan studi pada awal tahun 2017 ☺. Penulis yang mempunyai nama lengkap M. Rizky Faundra selalu membawa motto hidup “Sebaik-baiknya manusia bermanfaat bagi orang lain” agar kita selalu membantu sekuat tenaga dan pikiran untuk kebaikan terhadap sesama. Penulis mempunyai hobby yaitu olahraga, mengajar (ngelesi anak SD, SMP, SMA). Dengan nilai kesederhanaan, penulis ingin berbagi pengalaman dan ilmu pengetahuan dengan pembaca yang berminat pada penulis atau hal lain di email rizkyfaundra@gmail.com karena penulis tertarik dalam berbagi pengalaman dan hal-hal yang baru terutama tentang biometrika, biomedik... Semoga para pembaca dapat memberikan saran dan kritik yang membangun penulis agar bisa menjadi pribadi yang menarik, unik, dan bersahaja. Terima kasih...