



TUGAS AKHIR - KI141502

# APLIKASI      PENGENALAN BERDASARKAN FITUR GABOR

*PALM*

*VEIN*

PETER PANYANANDA  
NRP 5111100112

Dosen Pembimbing I  
Prof.Ir.Handayani Tjandrasa ,M.Sc, Ph.D

Dosen Pembimbing II  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015





**TUGAS AKHIR - KI141502**

# **APLIKASI PENGENALAN *PALM VEIN* BERDASARKAN FITUR GABOR**

**PETER PANYANANDA**  
**NRP 5111100112**

**Dosen Pembimbing I**  
**Prof.Ir.Handayani Tjandrasa ,M.Sc, Ph.D**

**Dosen Pembimbing II**  
**Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2015**

***[Halaman ini sengaja dikosongkan]***



UNDERGRADUATE THESES - KI141502

## APPLICATION OF PALM VEIN RECOGNITION BASED ON GABOR FEATURES

PETER PANYANANDA  
NRP 5111100112

Supervisor I  
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D

Supervisor II  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2015

***[Halaman ini sengaja dikosongkan]***

# **LEMBAR PENGESAHAN**

## **APLIKASI PENGENALAN *PALM VEIN* BERDASARKAN FITUR GABOR**

### **TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas dan Visi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh  
**PETER PANYANANDA**  
**NRP : 5111 100 112**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Prof.Ir.Handayani Tjandrasa ,M.Sc, Ph.D .....  
NIP: 19490823 197603 2 001 (Pembimbing 1)
2. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. ....  
NIP: 19751220 200112 2 002 (Pembimbing 2)

**SURABAYA**  
**JUNI, 2015**

***[Halaman ini sengaja dikosongkan]***



## **APLIKASI PENGENALAN PALM VEIN BERDASARKAN FITUR GABOR**

**Nama Mahasiswa : PETER PANYANANDA**  
**NRP : 5111100112**  
**Jurusan : Teknik Informatika FTIF-ITS**  
**Dosen Pembimbing 1 : Prof.Ir.Handayani Tjandrasa ,M.Sc,  
Ph.D**  
**Dosen Pembimbing 2 : Dr.Eng. Chastine Fatichah, S.Kom.,  
M.Kom.**

### ***Abstrak***

*Keamanan merupakan salah satu faktor yang penting dalam sistem komputer. Umumnya, keamanan pada sistem komputer berupa teks password, atau menggunakan kartu id yang memiliki barcode. Penggunaan teks password atau kartu id sebagai sistem keamanan memiliki beberapa kekurangan. Keadaan inilah yang menuntut sistem keamanan menjadi lebih berkembang.*

*Berbagai sistem keamanan berdasarkan anggota tubuh manusia atau biometrik telah mulai bermunculan, seperti pengenalan iris bola mata, pengenalan wajah, pengenalan palm vein, pengenalan sidik jari. Identifikasi biometrik ini dapat menggambarkan individu berdasarkan karakteristik yang unik.*

*Berbeda dari identifikasi biometrik yang lain, pengenalan palm vein tidak melakukan kontak secara langsung dengan kulit sehingga tidak terpengaruh dengan kondisi kulit seperti kelembapan dan penyakit kulit yang dapat menurunkan akurasi dari scanner. Selain itu bentuk dari pembuluh tiap individu seratus kali lebih unik jika dibandingkan dengan sidik jari.*

*Tugas akhir ini menggunakan fitur Gabor sebagai ekstraksi fitur dari citra palm vein. Citra palm vein yang*

*digunakan dapat diambil pada website CASIA. Sebelum dilakukan ekstraksi fitur, citra palm vein diambil region of interestnya(ROI). Hasil dari ekstraksi fitur Gabor akan dibentuk vein code-nya, dimana nilai pada vein code terdiri dari 0 dan 1. Similaritas antar dua vein code dihitung dengan menggunakan jarak hamming.*

*Dari uji coba dengan menggunakan fitur Gabor, didapatkan akurasi sebesar 85.67% untuk fitur Gabor dengan 4 orientasi dan 86% untuk fitur Gabor dengan 8 orientasi.*

***Kata kunci: Gabor Filter, Jarak hamming, Palm Vein, Biometrik, Vein Code***

## **APPLICATION OF PALM VEIN RECOGNITION BASED ON GABOR FEATURES**

**Student's Name** : PETER PANYANANDA  
**Student's ID** : 5111100112  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Prof.Ir.Handayani Tjandrasa ,M.Sc,  
Ph.D  
**Second Advisor** : Dr.Eng. Chastine Fatichah, S.Kom.,  
M.Kom.

### ***Abstract***

*Security is one of the factors that are important in a computer system. Generally, the security of the computer system in the form of text password, or use an id card that has a barcode. The use of text as a password or ID card in security system has some drawbacks. This condition requires security system becomes more developed.*

*Various security system based on the human body or biometric have begun to emerge, such as recognition iris of the eyeball, face recognition, palm vein recognition, fingerprint recognition. The biometric identification can describe individuals based on unique characteristics*

*Different from other biometric identification, palm vein recognition does not make contact directly with the skin so it is not affected by skin conditions such as moisture and skin diseases that can degrade the accuracy of the scanner. Besides the shape of each individual vessel hundred times more unique than fingerprints.*

*This final project uses Gabor features as feature extraction of palm vein image. Palm vein images that are used can be retrieved on the website CASIA. Prior to feature extraction, region of interest from palm veins taken. Results of*

*Gabor feature extraction will be established as vein code, where the values in the vein code consists of 0 and 1. The similarity between two vein codes are calculated using the Hamming distance.*

*The trials using Gabor features, obtained an accuracy of 85.67% for Gabor features with four orientation and 86% for Gabor features with eight orientation.*

***Keywords: Gabor Algortihm, Hamming Distance, Palm Vein Biometrics, Vein Code***

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**APLIKASI PENGENALAN *PALM VEIN* BERDASARKAN FITUR GABOR**”.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Tuhan Yang Maha Esa.
2. Papa, Mama, Wimala, Jessica, dan Angelica yang telah memberikan dukungan moral dan material yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Prof.Ir.Handayani Tjandrasa ,M.Sc, Ph.D selaku pembimbing I yang telah membantu, membimbing, dan memotivasi penulis dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing II yang juga telah membantu, membimbing, dan memotivasi kepada penulis dalam mengerjakan Tugas Akhir ini.
5. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.

6. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
7. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis
8. Teman-teman seluruh anggota klub Panther Ijo yang selalu memberikan hiburan, serta saran dan masukan selama pengerjaan Tugas Akhir ini.
9. Teman-teman angkatan 2011 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama empat tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2015

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i> .....	vii
<i>Abstract</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
DAFTAR KODE SUMBER .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi .....	3
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	5
BAB II DASAR TEORI.....	7
2.1 Biometrik.....	7
2.2 <i>Vein matching</i> .....	10
2.3 Dataset <i>Palm Vein</i> .....	12
2.4 Gabor .....	12
2.5 Jarak Hamming.....	15
2.6 Pengolahan Citra .....	15
2.6.1 High-pass Filter .....	16
2.6.2 Low-pass Filter.....	17
2.7 Morfologi.....	17
2.8 <i>Region of Interest</i> .....	20
BAB III DESAIN PERANGKAT LUNAK.....	23
3.1 Desain Metode Secara Umum .....	23
3.2 <i>Preprocessing</i> .....	23
3.2.1 Citra <i>Palm Vein</i> .....	25

3.2.2	Ekstraksi ROI .....	25
3.3	<i>Processing</i> .....	30
3.3.1	Filter Gabor.....	31
3.3.2	<i>Vein Code</i> .....	32
3.4	Pengenalan.....	33
3.4.1	Jarak hamming.....	33
3.5	Desain Grafical User Interface .....	34
BAB IV IMPLEMENTASI.....		37
4.1	Lingkungan Implementasi .....	37
4.2	Implementasi.....	37
4.2.1	Implementasi algoritma <i>inner border tracing</i> .....	37
4.2.2	Implementasi diagram distribusi jarak.....	40
4.2.3	Implementasi Ekstraksi Fitur Gabor .....	41
4.2.4	Implementasi Jarak hamming .....	44
BAB V UJI COBA DAN EVALUASI.....		47
5.1	Lingkungan Uji Coba .....	47
5.2	Data <i>Training</i> dan Data <i>Testing</i> .....	47
5.3	Implementasi Kode Sumber.....	48
5.3.1	Algoritma <i>Inner Border Tracing</i> .....	48
5.3.2	Diagram Distribusi Jarak .....	49
5.3.3	Ekstraksi Fitur Gabor dan <i>Vein Code</i> .....	51
5.4	Skenario dan Evaluasi Pengujian.....	55
5.4.1	Skenario Uji Coba 1 .....	56
5.4.2	Skenario Uji Coba 2 .....	57
5.4.3	Skenario Uji Coba 3 .....	58
5.4.4	Skenario Uji Coba 4 .....	58
5.5	Analisis Hasil Uji Coba .....	62
BAB IV KESIMPULAN DAN SARAN.....		67
6.1	Kesimpulan .....	67
6.2	Saran .....	67
DAFTAR PUSTAKA.....		69
LAMPIRAN .....		71
BIODATA PENULIS.....		97



## DAFTAR GAMBAR

<i>Gambar 2. 1 Gambar contoh diagram logika pada sistem biometrik secara umum .....</i>	<i>10</i>
<i>Gambar 2. 2 Gambar contoh citra dataset CASIA .....</i>	<i>13</i>
<i>Gambar 2. 3 Gambar (a) contoh 2D gabor filter pada spatial domain dan Gambar (b) contoh 2D gabor filer pada frekuensi domain.....</i>	<i>14</i>
<i>Gambar 2. 4 Gambar Ilustrasi dari jarak hamming .....</i>	<i>16</i>
<i>Gambar 2. 5 Gambar contoh citra High-pass Filtering .....</i>	<i>17</i>
<i>Gambar 2. 6 Gambar contoh citra Low-pass Filtering .....</i>	<i>17</i>
<i>Gambar 2. 7 Gambar ilustrasi dari proses dilasi .....</i>	<i>19</i>
<i>Gambar 2. 8 Gambar ilustrasi dari proses erosi .....</i>	<i>19</i>
<i>Gambar 2. 9 Gambar ilustrasi dari proses opening .....</i>	<i>20</i>
<i>Gambar 2. 10 Gambar ilustrasi dari proses closing.....</i>	<i>21</i>
<i>Gambar 3. 1 Gambar Alur dari aplikasi .....</i>	<i>24</i>
<i>Gambar 3. 2 Flowchart ekstraksi ROI .....</i>	<i>25</i>
<i>Gambar 3. 3 Gambar pseudocode algoritma inner border tracing .....</i>	<i>27</i>
<i>Gambar 3. 4 Flowchart algoritma inner border tracing .....</i>	<i>30</i>
<i>Gambar 3. 5 Gambar arah pencarian pada 8-connectivity ...</i>	<i>28</i>
<i>Gambar 3. 6 Gambar arah pencarian pada 4-connectivity ...</i>	<i>29</i>
<i>Gambar 3. 7 Gambar pseudocode fungsi pengambilan titik..</i>	<i>31</i>
<i>Gambar 3. 8 Pseudocode pembentukan citra vein code .....</i>	<i>33</i>
<i>Gambar 3. 9 Gambar pseudocode jarak hamming .....</i>	<i>33</i>
<i>Gambar 3. 10 Gambar desain GUI aplikasi .....</i>	<i>35</i>
<i>Gambar 5. 1 Gambar hasil algoritma inner border tracing .....</i>	<i>48</i>
<i>Gambar 5. 2 Citra input palm vein .....</i>	<i>49</i>
<i>Gambar 5. 3 Gambar diagram Distribusi Jarak.....</i>	<i>50</i>
<i>Gambar 5. 4 Gambar hasil ekstraksi ROI.....</i>	<i>51</i>
<i>Gambar 5. 5 Gambar penonjolan pembuluh pada ROI.....</i>	<i>51</i>
<i>Gambar 5. 6 Gambar hasil metode superposition .....</i>	<i>54</i>
<i>Gambar 5. 7 Gambar Vein Code .....</i>	<i>55</i>
<i>Gambar 5. 8 Grafik hasil akurasi skenario uji coba 3.....</i>	<i>63</i>

<i>Gambar 5. 9 Grafik hasil akurasi skenario uji coba 4 .....</i>	<i>64</i>
<i>Gambar 5. 10 Gambar citra vein code pada nilai lambda 2..</i>	<i>65</i>
<i>Gambar 5. 11 Gambar citra vein code pada nilai lambda 3.6</i>	<i>65</i>
<i>.....</i>	<i>65</i>

## DAFTAR TABEL

<i>Tabel 2- 1 Tipe-tipe biometrik.....</i>	<i>8</i>
<i>Tabel 2- 2 Tabel aturan dilasi dan erosi.....</i>	<i>18</i>
<i>Tabel 5- 1 Tabel filter gabor dan hasil ekstraksi fitur gabor berdasarkan orientasinya.....</i>	<i>52</i>
<i>Tabel 5- 2 Tabel perbedaan pada uji skenario 1 .....</i>	<i>59</i>
<i>Tabel 5- 3 Tabel perbedaan pada uji skenario 2 .....</i>	<i>61</i>
<i>Tabel A- 1 Rekap Hasil Keluaran Program Skenario Uji Coba 1,2,3,4 dengan luasan ROI 150 x 150, jumlah orientasi 4, lambda 3, dan nilai thresholding 0.03 .....</i>	<i>71</i>
<i>Tabel A- 2 Rekap Hasil Keluaran Program Skenario Uji Coba 1 dengan luasan ROI 150 x 250.....</i>	<i>73</i>
<i>Tabel A- 3 Rekap Hasil Keluaran Program Skenario Uji Coba 2 dengan 8 orientasi.....</i>	<i>75</i>
<i>Tabel A- 4 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2.....</i>	<i>77</i>
<i>Tabel A- 5 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2.5.....</i>	<i>79</i>
<i>Tabel A- 6 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2.8.....</i>	<i>81</i>
<i>Tabel A- 7 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 3.5.....</i>	<i>83</i>
<i>Tabel A- 8 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 3.8.....</i>	<i>85</i>
<i>Tabel A- 9 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 4.....</i>	<i>87</i>
<i>Tabel A- 10 Rekap Hasil Keluaran Program Skenario Uji Coba 4 dengan nilai thresholding 0.01.....</i>	<i>89</i>
<i>Tabel A- 11 Rekap Hasil Keluaran Program Skenario Uji Coba 4 dengan nilai thresholding 0.02.....</i>	<i>91</i>
<i>Tabel A- 12 Rekap Hasil Keluaran Program Skenario Uji Coba 4 dengan nilai thresholding 0.04.....</i>	<i>93</i>
<i>Tabel A- 13 Rekap Hasil Keluaran Program Skenario Uji Coba 4 dengan nilai thresholding 0.05.....</i>	<i>95</i>

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

<i>Kode Sumber 4. 1 Kode Sumber algoritma inner border tracing .....</i>	<i>39</i>
<i>Kode Sumber 4. 2 Kode Sumber Pengambilan titik <math>P_1</math> dan <math>P_2</math> .....</i>	<i>41</i>
<i>Kode Sumber 4. 3 Kode Sumber fungsi <math>f_{peak}</math> .....</i>	<i>42</i>
<i>Kode Sumber 4. 4 Kode Sumber ekstraksi fitur gabor .....</i>	<i>43</i>
<i>Kode Sumber 4. 5 Kode Sumber fungsi gabor_fn .....</i>	<i>44</i>
<i>Kode Sumber 4. 6 Kode Sumber pembuatan vein code .....</i>	<i>44</i>
<i>Kode Sumber 4. 7 Kode Sumber Jarak hamming .....</i>	<i>45</i>

*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Keamanan merupakan salah satu faktor yang penting dalam sistem komputer. Umumnya, keamanan pada sistem komputer berupa teks password, atau menggunakan kartu id yang memiliki barcode. Penggunaan teks password atau kartu id sebagai sistem keamanan memiliki beberapa kekurangan. Kekurangan inilah yang menuntut sistem keamanan untuk lebih berkembang. Perkembangan sistem keamanan ini telah menghasilkan sebuah sistem keamanan berdasarkan anggota tubuh manusia atau yang sering disebut biometrik. Salah satu contoh dari biometrik ini adalah pengenalan berdasarkan *palm vein* (pembuluh pada telapak tangan) [1].

Berbeda dengan teknik biometrik seperti *fingerprint*, teknik biometrik pada *palm vein* tidak memerlukan sentuhan secara langsung dengan kamera, sehingga kondisi kulit maupun penyakit kulit yang ada tidak mengganggu proses pengenalan. Selain tidak memerlukan sentuhan secara langsung, proses pengenalan pada biometrik *palm vein* tidak mengganggu kenyamanan dari pengguna, berbeda dari proses pengenalan pada retina yang mengganggu kenyamanan pengguna dengan menyinari mata pada saat proses pengenalan. Sama halnya dengan *fingerprint* dimana sidik jari pengguna selalu tetap, pola dari pembuluh yang dideteksi juga tidak akan mengalami perubahan. Dalam beberapa hal pengenalan melalui *palm vein* lebih baik dibandingkan dengan teknik biometrik lainnya, namun teknik biometrik *palm vein* juga memiliki kekurangan, yaitu tingkat akurasi yang lebih rendah jika dibandingkan dengan teknik biometrik pada retina [2].

Salah satu hal penting dalam pengenalan *palm vein* adalah mengekstraksi pembuluh darah yang terdapat dalam telapak tangan. Sehingga diperlukan proses untuk menonjolkan pembuluh pada citra *palm vein*. Selain proses penonjolan pembuluh, citra-citra *palm vein* umumnya memiliki bentuk serta ukuran dari masing-masing telapak tangan berbeda, sehingga sulit dalam menentukan titik pada ROI citra *palm vein*.

Filter gabor adalah filter yang memiliki frekuensi dan orientasinya menyerupai sistem penglihatan manusia, dan sering digunakan untuk merepresentasikan sebuah tekstur. Oleh karena itu, filter gabor banyak digunakan dalam biometrik salah satu contohnya pengenalan iris [1].

Penentuan wilayah ROI menggunakan diagram distribusi jarak. Setelah mendapatkan ROI pada citra *palm vein*, ROI diproses untuk menonjolkan pembuluh. Setelah dilakukan ekstraksi fitur gabor, hasil ekstraksi direpresentasikan dengan nilai 0 dan 1, untuk mempercepat proses pengenalan dengan metode jarak hamming. Adapun hasil yang diharapkan adalah tingginya tingkat akurasi, sehingga aplikasi ini dapat digunakan oleh masyarakat.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mendapatkan area ROI pada citra *palm vein*?
2. Bagaimana melakukan ekstraksi fitur *palm vein* dengan menggunakan fitur Gabor?
3. Bagaimana mengenali citra *palm vein* dengan jarak hamming?
4. Bagaimana mengevaluasi kinerja metode yang dibangun?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:



1. Data citra yang digunakan menggunakan data citra yang terdapat pada database CASIA.
2. Algoritma yang digunakan menggunakan jarak hamming dan filter gabor.
3. Aplikasi pengenalan merupakan aplikasi berbasis desktop.
4. Aplikasi pengenalan dibangun dengan menggunakan MATLAB.

#### **1.4 Tujuan**

Tugas Akhir ini mempunyai tujuan, yaitu membangun aplikasi pengenalan *palm vein* berdasarkan fitur Gabor.

#### **1.5 Manfaat**

Dengan dibuatnya Tugas Akhir ini maka aplikasi dapat mengenali pengguna sebagai sistem keamanan yang memiliki tingkat akurasi yang tinggi.

#### **1.6 Metodologi**

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.  
Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini, yaitu aplikasi pengenalan *palm vein* dengan menggunakan fitur gabor.
2. Studi literatur  
Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan aplikasi pengenalan *palm vein* berdasarkan fitur Gabor. Informasi dan literatur didapatkan

dari literatur buku dan sumber-sumber informasi lain yang berhubungan.

3. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Pengujian dan evaluasi akan dilakukan dengan menggunakan matlab. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir.

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

## 1.7 Sistematika Penulisan Laporan Tugas Akhir

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

### **Bab I    Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

### **Bab II   Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

### **Bab III   Perancangan Perangkat Lunak**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

### **Bab IV   Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

### **Bab V    Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI Kesimpulan Dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

## **BAB II**

### **DASAR TEORI**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian aplikasi. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Biometrik**

Biometrik berhubungan dengan karakteristik pada manusia. Otentifikasi biometrik digunakan pada bidang komputer sains sebagai bentuk dari identifikasi dan kontrol akses. Biometrik juga digunakan untuk mengidentifikasi individual dalam sebuah grup yang berada dalam pengawasan [3].

Identifikasi biometrik biasanya dikategorikan sebagai karakteristik fisik (*visual*) dan karakteristik perilaku. Karakteristik fisik berhubungan dengan bentuk tubuh, contohnya adalah *fingerprint*, *palm vein*, pengenalan wajah, *palm print*, geometri tangan, pengenalan iris, retina. Karakteristik perilaku menyangkut pola dari tingkah laku seseorang, contohnya gaya berjalan atau pola ketik. *Gambar 2. 1* berisi diagram umum mengenai logika dari sistem biometrik [3]. Selain karakteristik fisik dan karakteristik perilaku juga terdapat *olfactory biometric* yang melakukan pengenalan berdasarkan odor atau bau individu, biometrik kimia yang melakukan proses pengenalan berdasarkan DNA seseorang, dan biometrik audio dimana pengenalan terhadap individu dilakukan berdasarkan input suara yang diberikan. Tabel 2- 1 menunjukkan tipe biometrik beserta kategorinya dan penjelasan singkat [4] :

**Tabel 2- 1 Tipe-tipe biometrik**

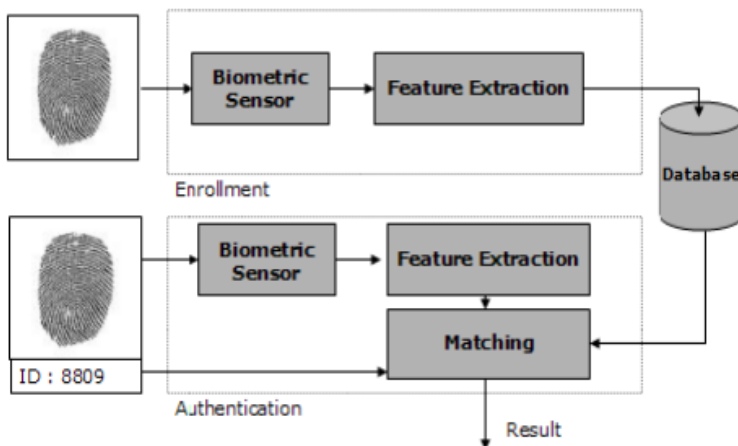
Tipe Biometrik	Kategori	Penjelasan
DNA matching	Biometrik Kimia	Identifikasi individu dengan menggunakan analisis segmen dari DNA.
Pengenalan telinga	Biometrik Fisik	Identifikasi individu berdasarkan bentuk dari telinga.
Pengenalan mata-iris	Biometrik Fisik	Menggunakan fitur yang diekstraksi dari iris untuk mengidentifikasi individu.
Pengenalan mata-retina	Biometrik Fisik	Menggunakan pola pembuluh pada bagian belakang mata untuk pengenalan.
Pengenalan wajah	Biometrik Fisik	Analisis menggunakan fitur wajah atau pola wajah untuk otentifikasi identitas individu.
Pengenalan geometri jari	Biometrik Fisik	Menggunakan geometri 3 dimensi jari untuk menentukan individu.
<i>Gait</i>	Biometrik Perilaku	Menggunakan cara berjalan dari individu dalam menentukan identitasnya.
Pengenalan geometri tangan	Biometrik Fisik	Menggunakan fitur geometri dari tangan, seperti panjang jari, dan lebar tangan.
Odor	<i>Olfactory Biometric</i>	Menggunakan odor dari individu.
Pengenalan tanda	Biometrik Fisik/	Menganalisa cara

tangan	Perilaku	menulis seseorang untuk proses otenifikasi.
Pengenalan ketik	Biometrik Perilaku	Menggunakan karakteristik unik dari ketikan seseorang untuk menentukan identitas.
Pengenalan pembuluh	Biometrik Fisik	Merupakan tipe biometrik yang mengidentifikasi individu melalui pola dari pembuluh yang berada di telapak atau jari.
Verifikasi suara	Biometrik Audio	Metode yang digunakan dalam menentukan identitas dari pemilik suara.
Identifikasi suara	Biometrik Audio	Metode yang digunakan untuk menentukan identitas pemilik suara yang tidak diketahui.

Perangkat biometrik didasarkan pada dua jenis sistem yaitu sistem biometrik multimodal dan sistem biometrik adaptif. Sistem biometrik multimodal menggunakan banyak sensor atau biometrik untuk mengatasi keterbatasan dari sistem biometrik. Sementara sistem biometrik unimodal dibatasi oleh integritas dari karakteristiknya, sistem biometrik multimodal memperoleh beberapa informasi dari tempat yang sama (contohnya beberapa foto pada pemindaian iris atau melakukan pemindaian beberapa kali pada jari yang sama) atau mendapatkan informasi dari biometrik lainnya (contohnya membutuhkan pengenalan suara dengan *fingerprinting*). Sistem biometrik adaptif bertujuan untuk melakukan *auto-update* pada template atau model untuk variasi intra kelas. Keuntungan yang didapatkan pada sistem ini adalah

sistem ini dapat memecahkan masalah keterbatasan jumlah data training dan melacak variansi temporal input data melalui adaptasi [5].

Diagram pada *Gambar 2. 1* mengilustrasikan dua mode pada sistem biometrik yaitu verifikasi dan identifikasi. Verifikasi merupakan mode pada sistem biometrik yang melakukan perbandingan *one to one* dengan data yang terdapat pada database untuk melakukan verifikasi terhadap individu tersebut. Sedangkan identifikasi adalah mode dimana sistem biometrik melakukan perbandingan *one to many* dengan database biometrik dengan usaha untuk membangun identitas dari individu yang tidak diketahui [6].



*Gambar 2. 1 Gambar contoh diagram logika pada sistem biometrik secara umum*

## 2.2 Vein matching

*Vein matching* atau yang disebut *vascular technology* merupakan sebuah teknik dalam identifikasi biometrik melalui analisi dari pola pembuluh darah yang terlihat pada permukaan kulit. Teknik identifikasi ini masih dalam perkembangan dan



belum diakui secara umum dikarenakan teknik ini masih belum dapat diandalkan jika dibandingkan dengan teknik *fingerprinting* [7].

Teknik *fingerprinting* secara umum lebih banyak digunakan oleh kalangan masyarakat. Tetapi teknik *fingerprinting* membutuhkan kontak secara langsung dengan kamera sehingga tingkat kekeringan dari kulit dapat mengganggu tingkat akurasi dari sistem. Penyakit kulit seperti *psoriasis* juga dapat mengganggu dan mengurangi tingkat akurasi dari teknik *fingerprinting*. Berbeda teknik *fingerprinting*, teknik *vein matching* tidak membutuhkan kontak secara langsung dengan kulit, dikarenakan informasi yang dibaca berada di dalam tubuh dan kondisi ataupun penyakit kulit tidak mengurangi tingkat akurasi sistem. Pemindai pembuluh juga bekerja dengan kecepatan yang sangat cepat, kurang dari satu detik. Dibandingkan dengan pemindaian retina yang memiliki tingkat akurasi lebih tinggi, pemindaian retina kurang diminati oleh masyarakat karena sifatnya mengganggu ketika mata diekspos dengan sinar yang tidak diketahui [2].

Pola dari pembuluh darah termasuk unik untuk setiap individu bahkan diantara kembar identik. Telapak tangan(*palm*) memiliki pola pembuluh darah yang luas dan rumit sehingga memiliki banyak fitur yang dapat membedakan identitas setiap pribadi. Oleh karena itu, teknik biometrik *palm vein* merupakan metode yang sangat aman dari otentifikasi karena pola pembuluh terletak pada bawah kulit. Hal ini membuat hampir tidak mungkin untuk dibaca atau disalin oleh orang lain [8].

Citra pola pembuluh darah untuk setiap individu ditangkap dengan menggunakan radiasi sinar NIR(*Near Infrared*). Metode refleksi menerangi telapak dengan menggunakan sinar inframerah dan menangkap cahaya yang dipancarkan oleh region setelah difusi melalui telapak tangan. Hemoglobin yang mengalami proses deoksidasi di dalam pembuluh menyerap sinar inframerah, sehingga mengurangi tingkat refleksi dan menyebabkan pola pembuluh darah tampak sebagai noda hitam [8].

Pembuluh terletak di dalam tubuh dan memiliki fitur-fitur yang berbeda, dan upaya menyalin identitas sangat sulit, sehingga memiliki tingkat keamanan yang tinggi [8]i.

*Palm vein* terletak pada bagian dalam tangan, sehingga *palm vein* dilindungi dan sistem ini tidak rentan terhadap trauma ringan, luka, dll sebaliknya sistem sidik jari lebih rentan terhadap luka. Berikut ini adalah kelebihan dari sistem biometrik *palm vein* [8]:

- a. Sulit untuk disalin
- b. Higenis, dan non-invasif
- c. Memiliki akurasi yang tinggi
- d. Mampu dalam pengenalan *one to one* dan *one to many*

### 2.3 Dataset *Palm Vein*

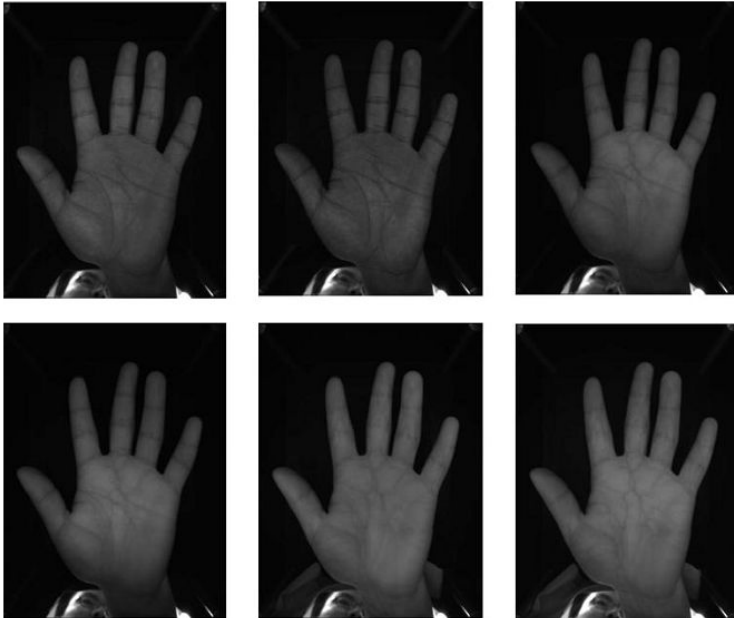
Dataset yang digunakan merupakan citra dari *palmprint multispectral* yang dapat diambil pada situs *biometrics idealtest* [9]. Dataset ini terdiri dari 100 data individu, dimana citra telapak terdiri dari telapak tangan kiri dan telapak tangan kanan, citra ini juga terdiri dari elektronik spektrum berjumlah 460nm, 630 nm, 700nm, 850 nm, 940nm dan WHT(*white light*). Setiap individu memilik 6 citra dimana citra pertama sampai ketiga merupakan citra sesi pertama dan citra keempat sampai citra keenam merupakan citra sesi kedua dengan interval waktu antara dua sesi adalah lebih dari satu bulan. Gambar 2. 2 menunjukkan citra dataset pada individu yang digunakan dimana tiga citra di atas merupakan citra pada sesi pertama sedangkan tiga citra di bawah adalah citra pada sesi kedua.

### 2.4 Gabor

Gabor merupakan filter linear yang sering digunakan untuk deteksi tepi. Frekuensi dan orientasi dari representasi gabor filter menyerupai sistem penglihatan manusia, dan filter ini cocok untuk representasi tekstur. Dalam domain spasial, filter gabor 2D adalah fungsi dari kernel *Gaussian* yang dimodulasi oleh

gelombang sinusoidal. Bentuk dari filter gabor 2D pada bidang spasial dan bidang frekuensi dapat dilihat pada Gambar 2. 3 (a) dan Gambar 2. 3 (b).

Gabor filter memiliki komponen *real* dan komponen *imaginary* yang merepresentasikan arah ortogonal. Kedua komponen dapat dibentuk menjadi sebuah bilangan kompleks atau digunakan secara terpisah. Berikut adalah rumusan gabor filter untuk komponen kompleks (2.1), komponen *real* (2.2), dan komponen *imaginary* (2.3) [10].



**Gambar 2. 2 Gambar contoh citra dataset CASIA**

*Kompleks*

$$h(x, y, \theta_k, f, \sigma_x, \sigma_y) = \exp \left\{ -\frac{1}{2} \left( \frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_y^2} \right) \right\} \cdot \exp(i2\pi f x_{\theta_k}) \quad (2.1)$$

*Real*

$$h_{Re}(x, y, \theta_k, f, \sigma_x, \sigma_y) = \exp \left\{ -\frac{1}{2} \left( \frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_y^2} \right) \right\} \cdot \cos(2\pi f x_{\theta_k}) \quad (2.2)$$

*Imaginary*

$$h_{Im}(x, y, \theta_k, f, \sigma_x, \sigma_y) = \exp \left\{ -\frac{1}{2} \left( \frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_y^2} \right) \right\} \cdot \sin(2\pi f x_{\theta_k}) \quad (2.3)$$

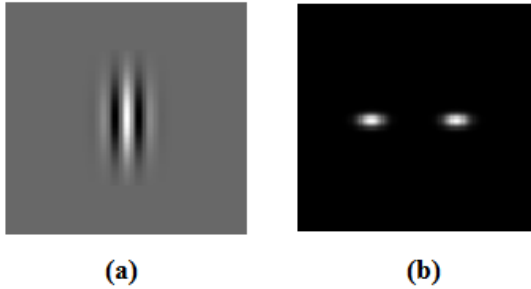
dimana,

$$x_{\theta_k} = x \cos \theta_k + y \sin \theta_k$$

dan

$$y_{\theta_k} = -x \sin \theta_k + y \cos \theta_k$$

Dalam persamaan ini  $f$  merupakan frekuensi pada bidang sinusoidal,  $\theta$  merepresentasikan orientasi dari gabor,  $\sigma_x$  dan  $\sigma_y$  merupakan sigma/ standar deviasi gaussian [1].



**Gambar 2. 3 Gambar (a) contoh 2D gabor filter pada bidang spasial dan Gambar (b) contoh 2D gabor filter pada bidang frekuensi**

## 2.5 Jarak Hamming

Jarak hamming diperkenalkan oleh Richard Hamming melalui paper "*Error detecting and error correcting codes*" pada tahun 1950. Analisis dari hamming digunakan pada beberapa bidang termasuk bidang teori informasi, teori koding, dan kriptografi [11].

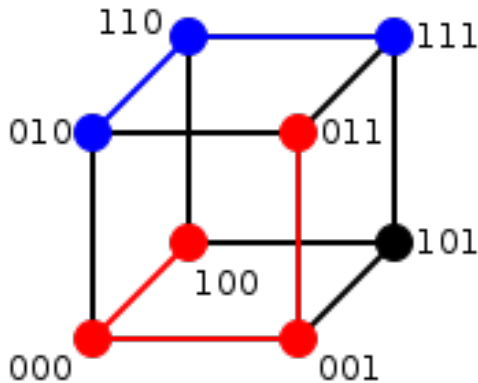
Jarak hamming antara dua string dengan panjang karakter yang sama adalah jumlah posisi dari simbol yang berbeda. Dengan kata lain, jarak hamming mengukur jumlah minimum dari substitusi yang diperlukan untuk mengubah satu string ke yang lain, atau jumlah minimum kesalahan yang bisa mengubah satu string ke yang lain [12]. Contoh dari jarak hamming antara [13]:

- "karolin" and "kathrin" adalah 3.
- "karolin" and "kerstin" adalah 3.
- 1011101 and 1001001 adalah 2.
- 2173896 and 2233796 adalah 3.

Gambar 2. 4 adalah ilustrasi yang menunjukkan sebuah kubus dengan dua contoh jarak(jarak biru dan jarak merah) dimana jarak tersebut dihitung berdasarkan jarak hamming. Jarak biru adalah jarak dari 010-111, sedangkan jarak merah adalah jarak dari 100-011 [14].

## 2.6 Pengolahan Citra

Pengolahan citra adalah bentuk dari pengolahan sinyal dimana inputnya berupa citra atau frame video, dan output citra berupa citra atau sekumpulan karakteristik atau parameter yang berhubungan dengan citra. Sebagian besar teknik pengolahan citra memperlakukan citra sebagai sinyal dua dimensi dan menerapkan teknik pengolahan sinyal standar [15].



**Gambar 2. 4** *Gambar Ilustrasi dari jarak hamming*

Pengolahan citra berhubungan dengan grafika komputer dan visi komputer. Pada grafika komputer, citra diperoleh dari model objek secara fisik, lingkungan, dan pencahayaan. Sedangkan pada visi komputer sering dianggap pengolahan citra level tinggi dan berfungsi untuk menguraikan isi fisik gambar atau urutan gambar [16].

Dalam ilmu modern dan teknologi, pengolahan citra juga mendapat cakupan yang lebih luas karena adanya perkembangan visualisasi ilmiah. Contohnya data *microarray* pada penelitian genetik atau *real-time* multi-aset perdagangan dibidang keuangan [16].

Pada pengolahan citra, *filtering* merupakan salah satu metode yang sering digunakan. *Filtering* citra memungkinkan untuk menerapkan berbagai efek pada citra. *Filtering* pada umumnya digunakan untuk mempertajam citra(*high-pass filter*) atau memperhalus citra(*low-pass filter*).

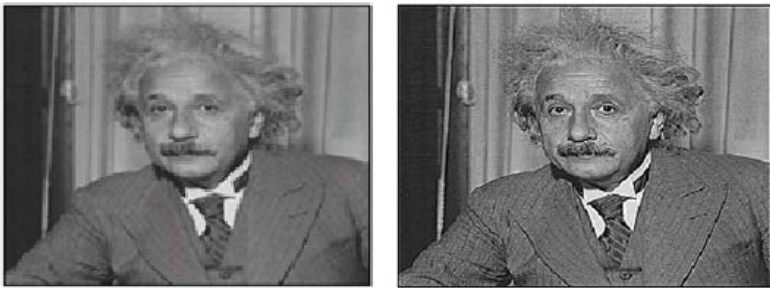
### 2.6.1 High-pass Filter

High-pass filter merupakan filter yang melewatkan signal yang lebih tinggi dari frekuensi *cutoff* sehingga mempertajam

citra yang dihasilkan. Gambar 2. 5 menunjukkan citra yang dipertajam menggunakan *high-pass filter* [17].

### 2.6.2 Low-pass Filter

Low-pass filter adalah filter yang melewatkan signal yang lebih rendah dari frekuensi *cutoff* sehingga membuat citra menjadi *blur*. Gambar 2. 6 menunjukkan citra yang *diblur* dengan menggunakan *low-pass filter* [17].



**Gambar 2. 5** Gambar contoh citra *High-pass Filtering*



**Gambar 2. 6** Gambar contoh citra *Low-pass Filtering*

## 2.7 Morfologi

Morfologi merupakan operasi pengolahan yang memproses citra berdasarkan bentuk. Operasi morfologi menerapkan elemen struktur ke dalam citra input, dan membuat output citra dengan

ukuran yang sama dengan citra input. Dalam operasi morfologi, nilai setiap pixel dalam gambar output didasarkan pada perbandingan pixel pada citra input dengan pixel tetangganya. Dengan memilih ukuran dan bentuk dari tetangga pixel, dapat membentuk operasi morfologi [18].

Operasi yang paling mendasar dalam morfologi adalah dilasi dan erosi. Dilasi menambahkan pixel pada batas-batas objek dalam citra, sedangkan erosi menghilangkan pixel pada batas-batas objek. Jumlah pixel yang ditambahkan atau dihapus dari objek citra tergantung pada ukuran dan bentuk dari elemen struktur yang digunakan untuk memproses gambar. Tabel 2- 2 adalah tabel aturan dalam melakukan dilasi dan erosi [18]:

***Tabel 2- 2 Tabel aturan dilasi dan erosi***

Dilasi	Nilai output pixel adalah nilai maksimum dari semua pixel yang berada pada input pixel tetangganya. Pada citra biner, jika ada pixel yang diset 1, maka output pixel diset menjadi 1.
Erosi	Nilai output pixel adalah nilai minimum dari semua pixel yang berada pada input pixel dan tetangganya. Pada citra biner, jika ada pixel yang diset 0, maka output pixel diset menjadi 0.

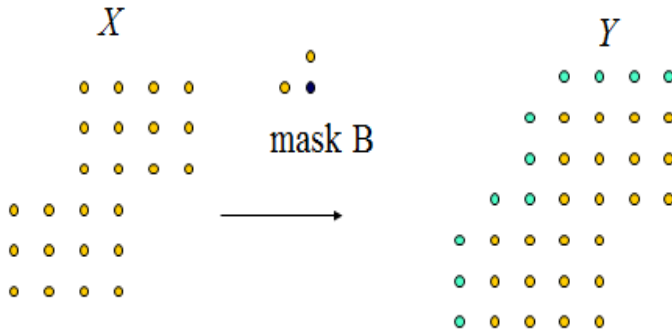
Gambar 2. 7 menunjukkan ilustrasi dari proses dilasi, dimana  $X$  merupakan citra input, mask  $B$  sebagai elemen struktur, dan  $Y$  sebagai citra output dengan pixel biru menunjukkan pixel tambahan akibat dari proses morfologi.

Gambar 2. 8 menunjukkan ilustrasi dari erosi, dengan  $X$  sebagai citra input, mask  $B$  sebagai elemen struktur, dan  $Y$  sebagai citra output dengan pixel putih sebagai pixel yang dihilangkan akibat dari operasi morfologi.

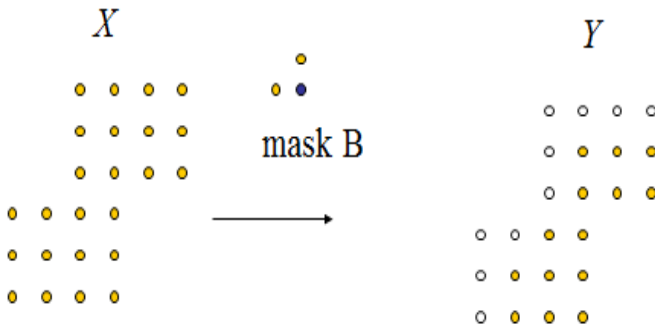
Selain dilasi dan erosi, morfologi juga memiliki dua fungsi lainnya yaitu *opening* dan *closing*. *Opening* merupakan operasi morfologi dimana citra input melakukan proses erosi terlebih dahulu baru kemudian dilakukan proses dilasi. Proses dari operasi *opening* dapat dilihat pada Gambar 2. 9.



Sebaliknya *closing* adalah proses dimana citra input melakukan proses dilasi kemudian dilanjutkan dengan proses erosi. Ilustrasi dari proses *closing* dapat dilihat pada Gambar 2.10.



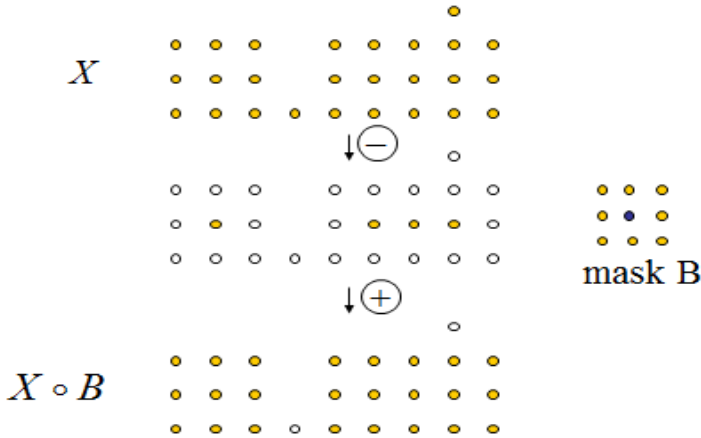
Gambar 2. 7 Gambar ilustrasi dari proses dilasi



Gambar 2. 8 Gambar ilustrasi dari proses erosi

Operasi morfologi ini sangat membantu dalam pengerjaan proses pada citra biner. Beberapa metode yang dibentuk berdasarkan dengan operasi morfologi ini adalah *boundary extraction*, pengisian daerah. Pada *boundary extraction*,

digunakan operasi erosi untuk mendapatkan garis pada tepi objek citra. Pengisian daerah menggunakan operasi dilasi untuk mengisi daerah yang kosong.



Gambar 2. 9 Gambar ilustrasi dari proses opening

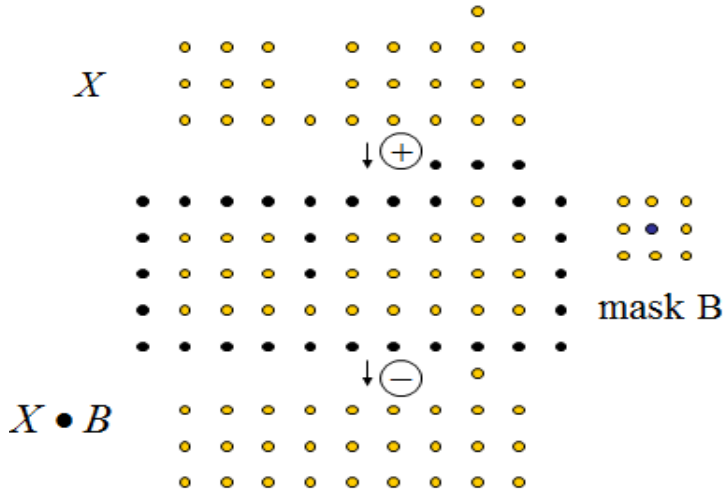
## 2.8 Region of Interest

*Region of interest* atau yang sering disingkat ROI merupakan sebuah subset yang dipilih dari dataset dengan tujuan tertentu. Konsep dari ROI telah banyak digunakan diberbagai bidang. Contohnya pada *medical imaging*, batas dari tumor dapat didefinisikan pada sebuah gambar dengan tujuan untuk mengukur besar tumor. Dalam sistem informasi geografi, sebuah ROI bisa dikatakan sebagai area *polygonal* dari peta 2D. Pada visi komputer, ROI didefinisikan sebagai batas dari sebuah objek [19].

Berikut ini adalah contoh dari ROI berdasarkan dimensinya [19]:

- Dataset 1D : waktu atau interval frekuensi dalam bentuk gelombang.
- Dataset 2D : batas-batas suatu objek pada gambar.

- Dataset 3D : kontur atau permukaan dari sebuah objek volume(atau biasa dikenal dengan nama *Volume of Interest*(VOI)).
- Dataset 4D : outline dari sebuah objek atau interval waktu pada volume waktu



*Gambar 2. 10 Gambar ilustrasi dari proses closing*

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **DESAIN PERANGKAT LUNAK**

Pada bab ini akan dijelaskan perancangan program yang dibuat. Perancangan akan dibagi menjadi tiga proses utama, yaitu:

1. *Preprocessing* untuk mendapatkan ROI(*Region of Interest*) dari citra *palm vein*.
2. Pengekstraksian fitur gabor pada ROI *palm vein* dan pembuatan *vein code* berdasarkan hasil ekstraksi.
3. Pengenalan *vein code* dengan dataset menggunakan algoritma jarak hamming.

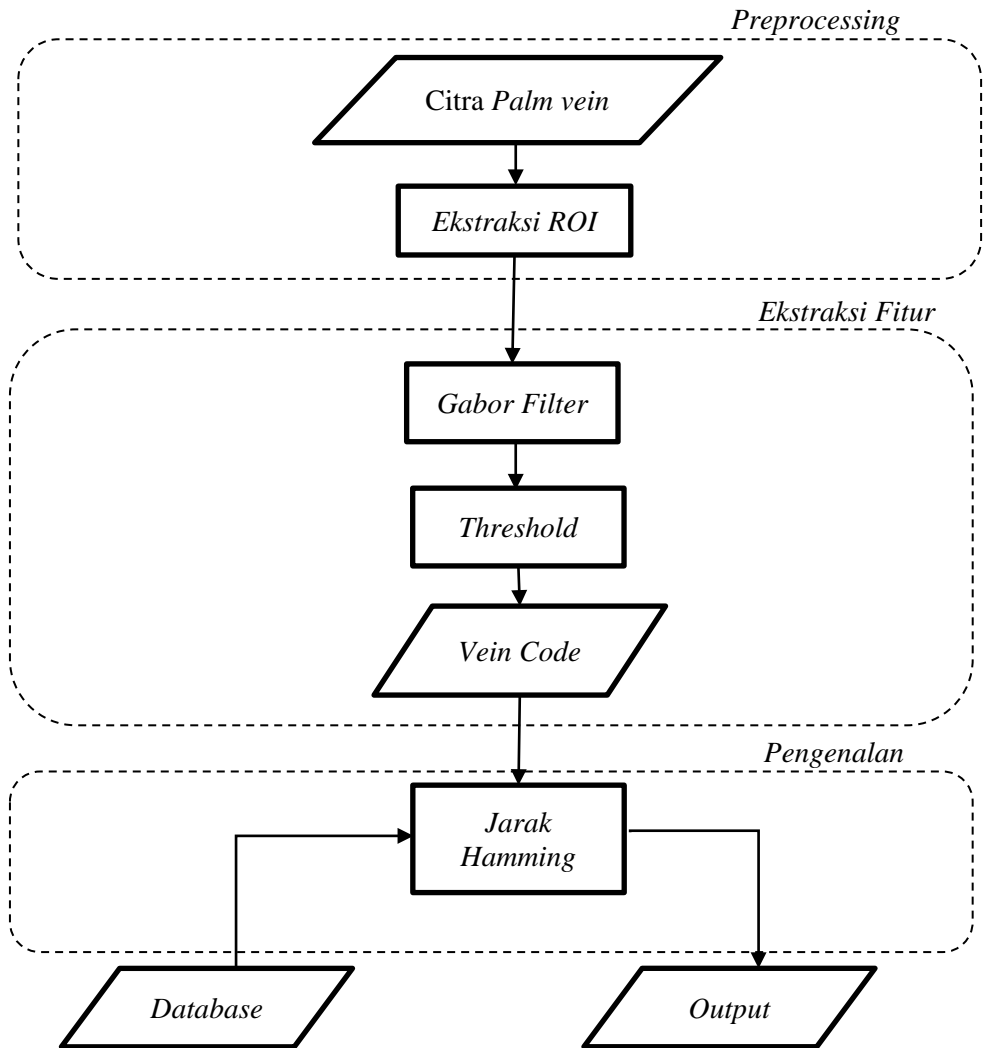
Pada bab ini akan dijelaskan gambaran umum setiap program utama dalam *flowchart* selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode*.

#### **3.1 Desain Metode Secara Umum**

Metode ini terdiri dari tahap *preprocessing*, tahap ekstraksi fitur, dan tahap pengenalan. Tahap *preprocessing* dimulai dengan memasukkan citra *palm vein* untuk mendapatkan ROI pada citra tersebut. Kemudian tahap selanjutnya adalah ekstraksi fitur dengan menggunakan filter gabor dan melakukan *thresholding* untuk mendapatkan *vein code*. Tahap yang terakhir adalah tahap pengenalan dari *vein code* dari citra yang dimasukkan dengan *vein code* yang ada pada database dengan menggunakan metode jarak hamming. Diagram alur dari proses aplikasi ditunjukkan pada Gambar 3. 1.

#### **3.2 *Preprocessing***

Dalam mengawali proses pengerjaan aplikasi, maka harus disiapkan terlebih dahulu data yang akan digunakan dalam proses pengerjaan. *Preprocessing* adalah tahapan yang penting untuk dilakukan. Pada Tugas Akhir kali ini, tahap *preprocessing* yang dilakukan adalah pengambilan ROI.

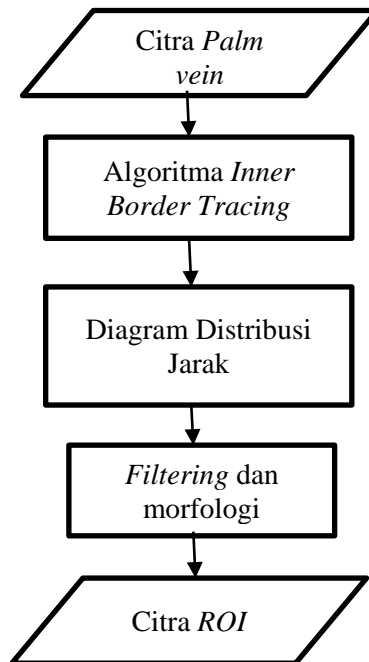


*Gambar 3. 1 Gambar Alur dari aplikasi*

### 3.2.1 Citra *Palm Vein*

Citra *palm vein* yang digunakan berasal dari website CASIA. Dataset yang digunakan berjumlah 100 orang, dimana masing-masing orang memiliki enam citra dan setiap orang memiliki dua sesi, dimana kedua sesi tersebut memiliki rentang waktu lebih dari satu bulan. Untuk sesi pertama terdiri dari citra pertama sampai citra ketiga, sedangkan sesi kedua terdiri dari citra keempat, kelima, dan keenam. Citra yang digunakan adalah citra yang memiliki elektronik spektrum berjumlah 940 nm. Citra *palm vein* yang digunakan pada Tugas Akhir ini adalah citra *palm vein* pada tangan kiri.

### 3.2.2 Ekstraksi ROI



Gambar 3. 2 Flowchart ekstraksi ROI

Untuk meningkatkan akurasi verifikasi, diperlukannya perbandingan fitur *palm vein* yang diambil pada daerah yang sama. Daerah yang dimaksud adalah ROI (*Region of Interest*). Oleh karena itu, penting untuk menempatkan ROI dalam posisi yang sama walaupun pada citra yang berbeda untuk memastikan stabilitas dari fitur *palm vein* [1]. Namun sulit untuk menempatkan ROI pada posisi yang sama, hal ini dikarenakan bentuk dan ukuran setiap citra yang berbeda. Flowchart untuk proses ekstraksi ROI dapat dilihat pada Gambar 3. 2.

Sebelum dilakukan ekstraksi ROI, citra input terlebih dahulu perlu dilakukan pencarian batas objek (telapak tangan) dengan menggunakan algoritma *inner border tracing*. Pseudocode untuk algoritma *inner border tracing* dapat dilihat pada Gambar 3. 3.

Algoritma *inner border tracing* merupakan algoritma yang digunakan untuk membentuk citra border dari suatu objek. Algoritma ini memiliki dua tipe yaitu *4-connectivity* dan *8-connectivity*. Perbedaan pada dua tipe ini adalah *4-connectivity* hanya mencari pixel tetangga dari 4 arah, sedangkan *8-connectivity* mencari pixel tetangga dari 8 arah. Pada Tugas Akhir ini, algoritma *inner border tracing* menggunakan *8-connectivity*.

Variabel *startCoor* pada pseudocode di atas merupakan koordinat dari pixel objek yang pertama ditemukan dalam hal ini adalah pixel border dari citra *palm vein*. Koordinat pixel dari *startCoor* ditemukan dengan cara mencari satu persatu pixel dari pojok kiri atas citra ke pojok kanan bawah sampai menemukan pixel objek. Pada algoritman dengan *4-connectivity* nilai *dir* diinisialisasi dengan 0, sedangkan pada *8-connectivity* nilai *dir* diinisialisasi dengan 7.

Masukan	Citra input biner <i>palm vein</i>
Keluaran	Citra border <i>palm vein</i>



```

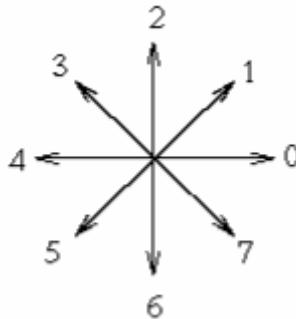
1. startCoor = koodinat pixel border objek
2. dir = 7 jika menggunakan 8-connectivity
3. pixel sekarang = startCoor
4. flag = 0
5. while flag tidak sama dengan 1
6.   if dir adalah bilangan ganjil
7.     Mencari nilai arah dengan rumus
       mod(dir+7,8)
8.   else
9.     Mencari nilai arah dengan rumus
       mod(dir+6,8)
10.  Mencari posisi pixel tetangga berdasarkan
     nilai arah
11.  Update nilai dir sesuai posisi pixel
     tetangga
12.  If nilai pixel tetangga == nilai pixel
     sekarang
13.    Output(pixel sekarang) = 1
14.    Pixel sekarang = pixel tetangga
15.  Else
16.    Mencari pixel tetangga yang lainnya
17.    Kembali ke step 11
18.  If pixel sekarang = startCoor
19.    flag = 1

```

**Gambar 3. 3 Gambar pseudocode algoritma inner border tracing**

Pembentukan *inner border* akan selesai jika nilai koordinat dari pixel sekarang sama dengan nilai dari startCoor. Pixel tetangga yang dimaksud adalah salah satu pixel dari 3x3 *neighborhood* dimana pusat dari *neighborhood* tersebut adalah pixel sekarang. Variabel *dir* adalah variabel yang digunakan dalam mencari posisi pixel tetangga. Nilai arah dari pencarian koordinat pixel tetangga dilakukan dengan menggunakan rumusan  $(dir+7) \bmod 8$ , jika nilai dari arah pencarian sebelumnya merupakan bilangan genap, dan rumusan  $(dir+6) \bmod 8$  jika nilai arah pencarian sebelumnya merupakan bilangan ganjil. Jika menggunakan 4-connectivity, maka rumusan untuk mencari nilai arah pixel tetangga yang digunakan adalah  $(dir+3) \bmod 4$ . Jika nilai dari pixel tetangga tidak sama dengan nilai dari pixel sekarang, maka perlu mencari arah yang baru. Dalam mencari

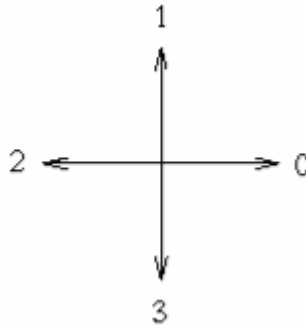
nilai arah baru hanya perlu mengikuti arah yang berlawanan dengan arah jarum jam. Arah pencarian pixel tetangga untuk algoritma 8-connectivity dapat dilihat pada Gambar 3. 5, dimana nilai 0 sampai 7 merupakan nilai *dir* yang menandakan posisi dari pixel tetangga. Sedangkan arah pencarian pixel tetangga untuk 4-connectivity dapat dilihat pada Gambar 3. 6. Output merupakan citra yang memiliki ukuran yang sama dengan citra input dan citra ini diinisialisasi dengan nilai 0. *Flowchart* untuk algoritma *inner border tracing* dapat dilihat pada Gambar 3. 4.



**Gambar 3. 4 Gambar arah pencarian pada 8-connectivity**

Diagram distribusi jarak merupakan diagram yang digunakan untuk menghitung jarak pada setiap pixel border. Diagram ini menggambarkan jarak dari tiap pixel border menuju ke pixel pada bagian tengah bawah citra. Diagram ini memiliki bentuk yang sama dengan telapak tangan, dimana diagram ini memiliki lima puncak dan empat lembah dimana tiap puncak mewakili jari pada citra input. Namun diagram ini memiliki perbedaan dengan citra input, yaitu jari paling kiri pada diagram adalah jari kelingking sedangkan pada citra input, jari paling kiri adalah ibu jari. Perbedaan ini disebabkan karena proses pada algoritma *inner border tracing* memiliki arah yang berlawanan dengan arah dari jarum jam sehingga jari yang dideteksi pertama kali adalah jari kelingking. Dengan adanya diagram ini akan memudahkan dalam pengambilan titik  $P_1$  dan  $P_2$ .  $P_1$  adalah titik

lembah yang berada diantara jari kelingking dan jari manis, sedangkan  $P_2$  adalah titik lembah yang berada diantara jari telunjuk dan jari tengah. Titik  $P_1$  dan  $P_2$  ini nantinya akan digunakan sebagai titik acuan dalam melakukan ekstraksi ROI. Pseudocode untuk pengambilan titik pada diagram distribusi jarak dapat dilihat pada Gambar 3. 7.



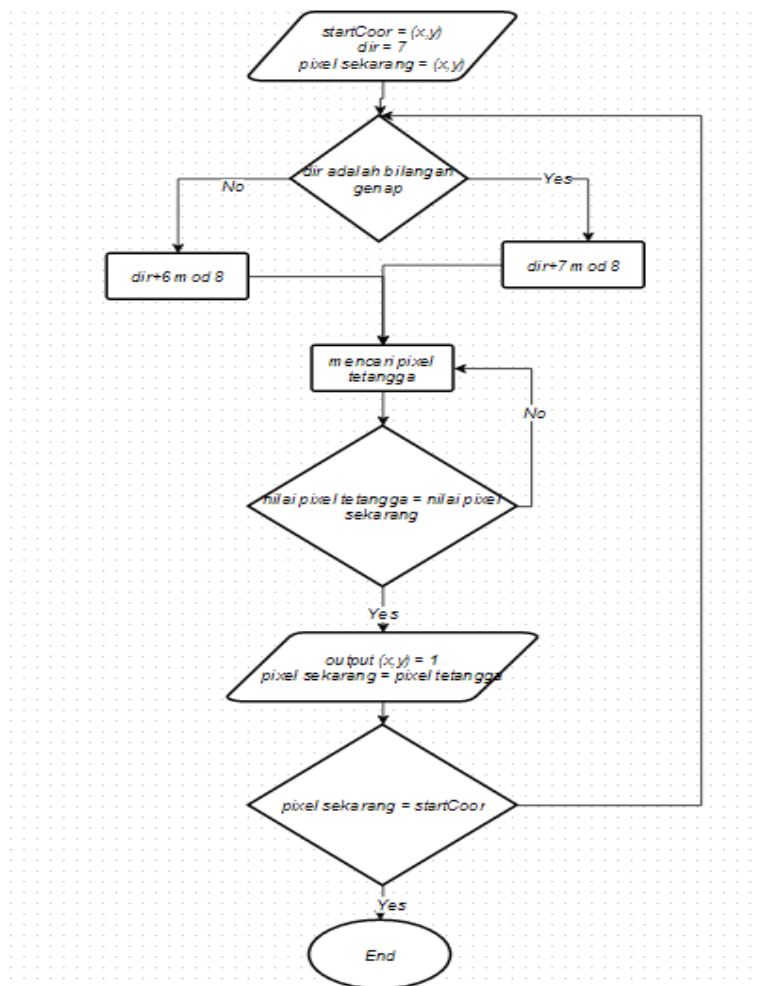
**Gambar 3. 5 Gambar arah pencarian pada 4-connectivity**

Nilai *top* dan *bottom* pada pseudocode Gambar 3. 7 merupakan nilai batas atas dan nilai batas bawah dengan rentang nilai yang berdasarkan pada variabel *sensivitas*. Nilai maksimum dan minimum dicari berdasarkan rentangan nilai tersebut, jika nilai sumbu y pada indeks ke-i merupakan nilai maksimum, maka nilai dari *outputMax* telah didapatkan, begitu pula dengan nilai *outputMin*. Setelah mendapatkan list posisi pixel yang menandakan puncak dan lembah dari diagram distribusi jarak, maka titik  $P_1$  dan  $P_2$  dengan mudah didapatkan. Setelah mendapatkan titik  $P_1$  dan  $P_2$ , maka ekstraksi ROI dilakukan dengan membentuk persegi berdasarkan dari jarak antara  $P_1$  dan  $P_2$ .

Setelah mendapatkan citra ROI, kemudian citra ROI tersebut dilakukan filtering dan operasi morfologi untuk membuat garis pembuluh pada citra tampak lebih jelas.

### 3.3 Processing

Pada bagian ini dijelaskan rumus dari filter gabor dan pembuatan *vein code* yang digunakan.



Gambar 3. 6 Flowchart algoritma inner border tracing

Masukan	Nilai sumbu x, nilai sumbu y, sensitivitas
Keluaran	Posisi nilai puncak
<pre> 1. Data = nilai dari sumbu x dan sumbu y 2. S = sensitivitas 3. numP = 1 4. For i=1 : size(x) 5.     mencari nilai top dan bottom sesuai dengan        sensitivitas 6.     if y(i) == max y dari top sampai bottom 7.         outputMax = data(i); 8.     if y(i) == min y dari top sampai bottom 9.         outputMin = data(i); </pre>	

**Gambar 3. 7 Gambar pseudocode fungsi pengambilan titik**

### 3.3.1 Filter Gabor

*Palm vein* merupakan pola dari pembuluh darah yang terletak di bawah kulit. Setiap orang memiliki struktur dan posisi dari pembuluh yang unik. Struktur dan posisi pembuluh ini tidak mengalami perubahan. Karena *palm vein* mengandung jaringan pembuluh yang menyerupai seperti karakteristik garis, *palm vein* dapat dilihat sebagai citra yang bertekstur. 2D filter gabor sering digunakan untuk mengekstraksi dari citra yang bertekstur [1]. Oleh karena pada Tugas Akhir ini akan menggunakan gabor filter sebagai ekstraksi fitur. Pengekstraksian fitur dengan filter gabor pada pola pembuluh sudah banyak dilakukan contohnya seperti Daugman(1993) [20] yang menggunakan 2D filter gabor untuk pengenalan iris dan Jen Chun Lee(2012) yang menggunakan filter gabor untuk pengenalan *palm vein* [1].

Rumusan yang digunakan dalam membentuk 2D filter gabor dapat dilihat pada rumusan (3.1), dimana untuk mendapat nilai  $\sigma$ , digunakan rumusan (3.2) .

$$g(x, y, \gamma, \psi, \lambda, \theta, \sigma) = \exp\left(-\frac{1}{2} * \left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right) * \cos\left(\frac{2\pi * x'}{\lambda} + \psi\right) \quad (3.1)$$

dimana,

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta, \\ y' &= -x \sin \theta + y \cos \theta, \\ \sigma(b) &= \frac{\lambda}{\pi} * \sqrt{\frac{\log 2}{2}} * \frac{2^b + 1}{2^b - 1}, \\ \sigma_x &= \sigma, \\ \sigma_y &= \sigma / \gamma \end{aligned} \quad (3.2)$$

Pada rumusan (4),  $\lambda$  merupakan panjang gelombang sinusoidal,  $\theta$  merepresentasikan orientasi dari gabor,  $\psi$  adalah fase offset,  $\sigma$  merupakan standar deviasi gaussian, dan  $\gamma$  adalah rasio aspek spasial. Nilai  $\sigma$  dicari dengan menggunakan  $b$ , dimana  $b$  merupakan nilai *bandwidth*. Nilai  $b$  yang digunakan pada rumusan di atas adalah 1.

### 3.3.2 Vein Code

*Vein code* merupakan hasil dari ekstraksi fitur yang telah dilakukan *thresholding* sehingga nilai dari *vein code* adalah 0 dan 1. Tujuan dibuatnya *vein code* adalah untuk mempercepat proses pengenalan, karena nilai pixel pada *vein code* hanya terdiri dari 0 dan 1. *Vein code* diproses dengan aturan sebagai berikut :

- $V(x,y) = 1$ , jika  $G(x,y) \geq 0.3$
- $V(x,y) = 0$ , jika  $G(x,y) < 0.3$

$V(x,y)$  merupakan nilai pixel *vein code* pada posisi  $x$  dan  $y$ .  $G(x,y)$  adalah nilai pixel dari hasil ekstraksi fitur pada posisi  $x$  dan  $y$ . Pseudocode yang digunakan dalam membuat vein code dapat dilihat pada Gambar 3. 8.

Masukan	Citra hasil ekstraksi fitur gabor
Keluaran	Citra vein code
1. Mencari koordinat dari nilai pixel pada citra hasil ekstraksi fitur gabor $\geq 0.03$ 2. Memberi nilai 1 pada citra vein code sesuai dengan koordinat yang sudah ditemukan	

**Gambar 3. 8 Pseudocode pembentukan citra vein code**

### 3.4 Pengenalan

Pada bagian pengenalan akan membahas mengenai pseudocode dari jarak hamming.

#### 3.4.1 Jarak hamming

Jarak hamming digunakan untuk mengukur satu persatu similaritas dari *vein code* input dengan *vein code* yang terdapat pada database. Pengukuran similaritas ini juga dilakukan oleh Daugman(1993) pada pengenalan dengan iris [20]. Algoritma jarak hamming dapat dilihat pada Gambar 3. 9.

Masukan	<i>Vein code input</i> dan <i>vein code database</i>
Keluaran	Nilai jarak hamming
1. Mencari nilai XOR dari tiap pixel untuk <i>vein code input</i> dan <i>vein code database</i> 2. Menghitung distance berdasarkan jumlah dari nilai XOR	

**Gambar 3. 9 Gambar pseudocode jarak hamming**

*Exclusive OR* atau yang disingkat XOR merupakan salah satu dari operasi *logical* dengan aturan sebagai berikut:

- Output akan bernilai 0, jika kedua input bernilai sama.

- Output akan bernilai 1, jika kedua input memiliki nilai yang berbeda

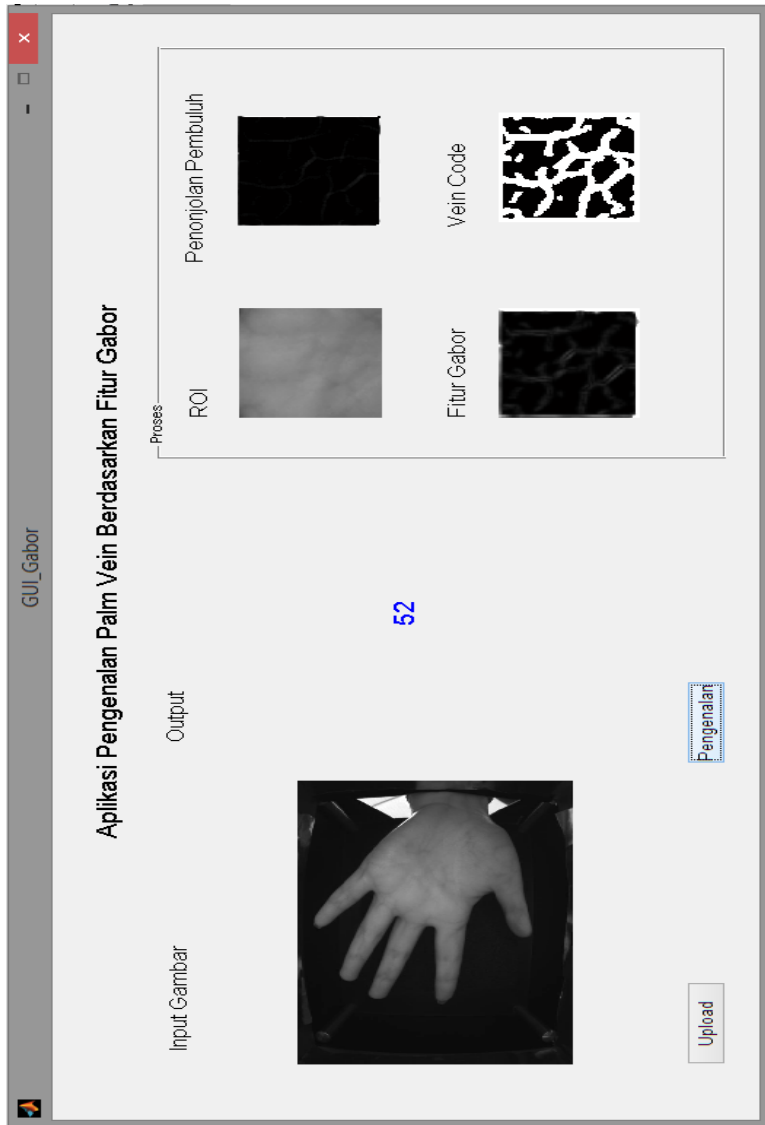
Karena karakteristik ini, XOR digunakan dalam proses jarak hamming sehingga citra input yang menyerupai citra pada database memiliki jarak similaritas yang paling kecil dibandingkan dengan nilai lainnya. Jarak hamming akan bernilai 0 jika citra input yang dimasukkan sama dengan citra yang ada pada database.

### 3.5 Desain Grafical User Interface

*Grafical user interface*(GUI) pada aplikasi pengenalan *palm vein* berdasarkan fitur gabor terdiri dari lima gambar, satu output berupa teks dan dua tombol untuk memulai proses pengenalan. Gambar GUI aplikasi dapat dilihat pada Gambar 3.10. Berikut adalah penjelasan singkat mengenai komponen-komponen yang terdapat pada GUI :

- Gambar input, yang menampilkan gambar yang diinput oleh user.
- Gambar ROI, menampilkan hasil dari proses ekstraksi ROI.
- Gambar ROI-Process menampilkan hasil dari proses *filtering* dan penonjolan pembuluh pada citra.
- Gambar Fitur Gabor, menunjukkan hasil citra yang sudah diekstraksi fitur dengan menggunakan fitur gabor.
- Gambar Vein Code, menampilkan citra ekstraksi fitur yang terdiri dari 0 dan 1.
- Tombol Upload berfungsi untuk memasukkan citra input yang ingin dikenali.
- Tombol Pengenalan berfungsi untuk melakukan proses pengenalan citra *palm vein*.
- Output teks merupakan teks yang mendefinisikan indeks citra berdasarkan dari hasil proses pengenalan.





Gambar 3. 10 Gambar desain GUI aplikasi

*[Halaman ini sengaja dikosongkan]*

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah *Matlab 7.6.0 (R2008a)* yang diinstal pada sistem operasi *Windows 8*.

### **4.2 Implementasi**

Pada subbab ini akan dijelaskan implementasi yang digunakan pada setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan program. Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program Tugas Akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

#### **4.2.1 Implementasi algoritma *inner border tracing***

Algoritma *Inner border tracing* digunakan untuk membentuk citra baru dimana citra baru tersebut merupakan citra yang menampilkan batas(*border*) dari citra input. Selain untuk menampilkan batas objek, algoritma ini juga digunakan untuk menyimpan list dari pixel *border* yang dibentuk. List pixel ini nantinya akan digunakan dalam pengambilan dua titik pada ROI citra palm vein. Kode sumber untuk algoritma *inner border tracing* dapat dilihat pada Kode Sumber 4. 1.

1	se = strel('disk',17);
2	imgbw = imopen(imgbw_in,se);
3	ctrImg=zeros(size(imgbw,1),size(imgbw,2));
4	[r c] = find(imgbw==1,1);
5	ctrImg(r,c) = 1;
6	coor = [r,c];
7	startCoor = [r,c];
8	maksCoorR = size(img,1);
9	maksCoorC = size(img,2);
10	coorHalfWrist(1,1) = maksCoorR;
11	coorHalfWrist(1,2) = maksCoorC/2;
12	count = 1;
13	dir = 7;
14	indikator = 1;
15	while indikator == 1
16	if mod(dir,2)==0
17	startDir = mod(dir+7,8);
18	else
19	startDir = mod(dir+6,8);
20	end
21	currentDir = startDir;
22	for i=1:8
23	if currentDir==8
24	currentDir = 0;
25	End
26	switch (currentDir)
27	case 0
28	posisi = [0,1];
29	dir = 0;
30	case 1
31	posisi = [-1,1];
32	dir = 1;
33	case 2
34	posisi = [-1,0];
35	dir = 2;
36	case 3
37	posisi = [-1,-1];
38	dir = 3;

39	<code>case 4</code>
40	<code>posisi = [0,-1];</code>
41	<code>dir = 4;</code>
42	<code>case 5</code>
43	<code>posisi = [1,-1];</code>
44	<code>dir = 5;</code>
45	<code>case 6</code>
46	<code>posisi = [1,0];</code>
47	<code>dir = 6;</code>
48	<code>case 7</code>
49	<code>posisi = [1,1];</code>
50	<code>dir = 7;</code>
51	<code>end</code>
52	<code>newCoor = coor + posisi;</code>
53	<code>If newCoor(1,1)&gt;0 &amp;&amp; newCoor(1,2) &gt; 0 &amp;&amp; newCoor(1,1) &lt; maksCoorR &amp;&amp; newCoor(1,2) &lt; maksCoorC</code>
54	<code>if imgbw(newCoor(1,1),newCoor(1,2))== 1</code>
55	<code>ctrImg(newCoor(1,1),newCoor(1,2))=1;</code>
56	<code>coor = newCoor;</code>
63	<code>direction(count,1) = dir;</code>
65	<code>break;</code>
66	<code>else</code>
67	<code>currentDir = currentDir + 1;</code>
68	<code>end</code>
69	<code>else</code>
70	<code>currentDir = currentDir + 1;</code>
71	<code>end</code>
72	<code>end</code>
73	<code>if startCoor == coor</code>
74	<code>indikator = 0;</code>
75	<code>end</code>
76	<code>end</code>

**Kode Sumber 4. 1 Kode Sumber algoritma inner border tracing**

Sebelum dilakukan algoritma *inner border tracing*, citra dilakukan proses *opening*, proses ini bertujuan untuk menghilangkan noise yang terdapat pada citra *palm vein*. Proses

*opening* dapat dilihat pada baris ke 1 dan baris ke 2 pada Kode Sumber 4. 1. Baris ke 4 berfungsi untuk mencari nilai koordinat pixel objek yang pertama ditemukan. Algoritma *Inner border tracing* pada Tugas Akhir ini menggunakan *8-connectivity* sehingga variabel *dir* diinisialisasi dengan nilai 7. Baris 23 sampai baris 53 berfungsi dalam menentukan posisi dari pixel tetangga berdasarkan nilai *dir* yang didapatkan. Algoritma utama terletak dari baris 15 sampai baris 76.

#### 4.2.2 Implementasi diagram distribusi jarak

Diagram distribusi jarak digunakan untuk pengambilan titik  $P_1$  dan titik  $P_2$ , dimana  $P_1$  adalah titik lembah yang berada diantara jari kelingking dan jari manis, sedangkan  $P_2$  adalah titik lembah yang berada diantara jari telunjuk dan jari tengah. Kode sumber dalam pengambilan titik  $P_1$  dan titik  $P_2$  dapat dilihat pada Kode Sumber 4. 2 .

Baris 1 sampai baris 4 berfungsi untuk mengatur bentuk diagram, dimana koordinat pertama pada diagram merupakan koordinat pixel yang memiliki jarak yang paling kecil. Baris 5 sampai baris 7 berfungsi untuk menghilangkan noise yang berada dibagian bawah citra. Cara menghilangkan noise tersebut adalah dengan melakukan *thresholding* dimana pixel yang memiliki posisi sumbu y diatas 550, akan memiliki jarak bernilai 0. Pada baris 8 berfungsi untuk memanggil fungsi *fpeak*, kode fungsi *fpeak* dapat dilihat pada Kode Sumber 4. 3. Baris 9 sampai baris 20 dimaksudkan untuk menghapus duplikasi nilai puncak. Setelah mendapatkan titik  $P_1$  dan  $P_2$ , maka langkah selanjutnya adalah mengekstraksi ROI.

1	<code>[r c]=find(coorDis(:,3)==min(coorDis(:,3)));</code>
2	<code>newCoorDis=zeros(size(coorDis,1),size(coorDis,2),size(coorDis,3));</code>
3	<code>newCoorDis(1:size(coorDis,1)-r+1,; )=coorDis(r:size(coorDis,1),: );</code>
4	<code>newCoorDis(size(coorDis,1)-</code>

	<code>r+2:size(coorDis),: )=coorDis(1:r-1,: );</code>
5	<code>zeroDis = 550;</code>
6	<code>[r c] = find(newCoorDis(:,1)&gt;zeroDis);</code>
7	<code>newCoorDis(r,3) = 0;</code>
8	<code>output = fpeak(sumbuX,newCoorDis(:,3),50);</code>
9	<code>for i=1:size(output,1)-1</code>
10	<code>    if output(i,2)==output(i+1,2)</code>
11	<code>        output(i,3) = 1;</code>
12	<code>    else</code>
13	<code>        output(i,3) = 0;</code>
14	<code>    end</code>
15	<code>end</code>
16	<code>improveOut = find(output(:,3)==1);</code>
17	<code>improveOut = sort(improveOut,'descend');</code>
18	<code>for i=1:size(improveOut)</code>
19	<code>    output(improveOut(i,1),: )=[];</code>
20	<code>end</code>
21	<code>pixelP1 = output(3,1);</code>
22	<code>pixelP2 = output(7,1);</code>

**Kode Sumber 4. 2 Kode Sumber Pengambilan titik  $P_1$  dan  $P_2$**

### 4.2.3 Implementasi Ekstraksi Fitur Gabor

Ekstraksi fitur gabor digunakan untuk melakukan ekstraksi fitur dengan menggunakan filter gabor. Proses tersebut diimplementasikan pada Kode Sumber 4. 4.

1	<code>Data = sortrows([x,y]);</code>
2	<code>for i=1:rx</code>
3	<code>    if i-s&lt;1</code>
4	<code>        top = 1;</code>
5	<code>    else</code>
6	<code>        top = i-s;</code>
7	<code>    end</code>
8	<code>    y = Data(:,2);</code>
9	<code>    if i+s&gt;length(y)</code>
10	<code>        bottom = length(y);</code>

11	<code>else</code>
12	<code>bottom = i + s;</code>
13	<code>end</code>
14	<code>tP=(y(i)==max(y(top:bottom)));</code>
15	<code>bP=(y(i)==min(y(top:bottom)));</code>
16	<code>if tP==1    bP==1</code>
17	<code>isP = Data(i,: );</code>
18	<code>else</code>
19	<code>isP = [nan,nan];</code>
20	<code>end</code>
21	<code>if sum(isnan(isP))== 0</code>
22	<code>output(numP,: ) = isP;</code>
23	<code>numP = numP + 1;</code>
24	<code>end</code>

**Kode Sumber 4. 3 Kode Sumber fungsi fpeak**

1	<code>fiturImg2 = medfilt2(fiturImg,[10 10]);</code>
2	<code>fiturImg3=adapthisteq(fiturImg2,'clipLimit' ,0.08,'Distribution','rayleigh');</code>
3	<code>fiturImg4 = wiener2(fiturImg3,[12 12]);</code>
4	<code>fiturImg5 = anisodiff2D(fiturImg4, 10, 1/7, 20, 1);</code>
5	<code>fiturImg6=uint8(round(fiturImg5-1));</code>
6	<code>se = strel('disk',3);</code>
7	<code>fiturImg7 = imclose(fiturImg6,se);</code>
8	<code>fiturImg8 = imsubtract(fiturImg7,fiturImg6);</code>
9	<code>fiturImg9 = imadjust(fiturImg8);</code>
10	<code>lambda = 3;</code>
11	<code>psi = [0 pi/2];</code>
12	<code>gamma = 0.5;</code>
13	<code>bw = 1;</code>
14	<code>N = 4;</code>
15	<code>fiturImg_n = im2double(fiturImg9);</code>
16	<code>img_out_re = zeros(size(fiturImg_n,1), size(fiturImg_n,2), N);</code>
17	<code>for j=1:N</code>
18	<code>theta = (j-1)*pi/N;</code>



19	<code>gb_re=gabor_fn(bw,gamma,psi(1),lambda,theta )+ li * gabor_fn(bw,gamma,psi(2),lambda,theta);</code>
20	<code>img_out_re(:, :, j) = imfilter(fiturImg_n, gb_re, 'symmetric');</code>
21	<code>end</code>
22	<code>img_out_disp_re = sum(abs(img_out_re).^2, 3).^0.5;</code>

**Kode Sumber 4. 4 Kode Sumber ekstraksi fitur gabor**

Sebelum dilakukan ekstraksi fitur gabor, hasil dari ekstraksi ROI dilakukan proses filter untuk menghilangkan noise dan memberi penonjolan pada pembuluh sehingga pembuluh tampak lebih jelas. Proses *filtering* dan penonjolan pembuluh, dapat dilihat baris 1 sampai baris 9 pada Kode Sumber 4. 4. Proses *filtering* yang digunakan adalah median filter, wiener filter, anisodiff, dan adapthisteq. Proses penonjolan yang digunakan adalah closing, imsubtract, dan imadjust. Pada baris 10 sampai baris 14 merupakan parameter yang digunakan untuk membuat filter gabor. Pembuatan filter gabor dilakukan dengan pemanggilan fungsi *gabor\_fn*. Kode sumber untuk *gabor\_fn* dapat dilihat pada Kode Sumber 4. 5. Untuk menyatukan 4 orientasi yang berbeda pada fitur gabor, Tugas Akhir ini menggunakan metode *superposition* yang terletak pada baris 22. Langkah selanjutnya adalah membuat *vein code* berdasarkan hasil ekstraksi fitur yang telah dilakukan. Kode Sumber 4. 6 berisi kode yang digunakan dalam pembuatan *vein code*.

1	<code>function</code>
2	<code>gb=gabor_fn(bw,gamma,psi,lambda,theta)</code>
3	<code>sigma_x = sigma;</code>
4	<code>sigma_y = sigma/gamma;</code>
5	<code>sz = 60;</code>
6	<code>[x y]=meshgrid(-(sz/2): (sz/2));</code>
7	<code>x_theta=x*cos(theta)+y*sin(theta);</code>

8	<code>y_theta=-x*sin(theta)+y*cos(theta);</code>
9	<code>gb=exp(- 0.5*(x_theta.^2/sigma_x^2+y_theta.^2/sigma_ y^2)).*cos(2*pi/lambda*x_theta+psi);</code>

**Kode Sumber 4. 5 Kode Sumber fungsi gabor\_fn**

Dalam membuat filter gabor, fungsi ini mengikuti rumusan (4) yang berada pada subab 3.3.1. Begitu juga dengan pencarian nilai pada variabel *sigma* mengikuti rumusan (5). Variabel *sz* adalah variabel yang digunakan untuk mendefinisikan ukuran dari jendela dari filter gabor.

1	<code>Vr = zeros(size(img_out_disp_re,1), size(img_out_disp_re,2));</code>
2	<code>findRe = find(img_out_disp_re&gt;=0.03);</code>
3	<code>for i=1:size(findRe,1)</code>
4	<code>    Vr(findRe(i)) = 1;</code>
5	<code>end</code>

**Kode Sumber 4. 6 Kode Sumber pembuatan vein code**

*Vr* merupakan variabel zeros yang memiliki nilai 0 untuk setiap pixelnya dan berukuran sama dengan citra hasil ekstraksi fitur. Fungsi *find* pada matlab berfungsi untuk menemukan indeks yang memiliki nilai lebih dari sama dengan 0.03. Langkah selanjutnya adalah memberi nilai 1 ke setiap pixel yang ditemukan oleh fungsi *find*.

#### 4.2.4 Implementasi Jarak hamming

Jarak hamming digunakan dalam melakukan pengenalan terhadap *vein code* input dengan *vein code* pada database. Kode sumber untuk implementasi jarak hamming dapat dilihat pada Kode Sumber 4. 7.

1	<code>for i = 1:50</code>
2	<code>    for j = 1:3</code>
3	<code>        hammDist=xor(cur_vi,dataset_vi(:, :, i, j));</code>

4	<code>valHammDist(i,j) = sum(sum(hammDist));</code>
5	<code>if(valHammDist(i,j) &lt; minDist)</code>
6	<code>minDist = valHammDist(i,j);</code>
7	<code>minVr(m,1) = i;</code>
8	<code>minVr(m,2) = j;</code>
9	<code>end</code>
10	<code>hammDist = [];</code>
11	<code>end</code>
12	<code>end</code>

***Kode Sumber 4. 7 Kode Sumber Jarak hamming***

*I* merupakan jumlah individu yang terdapat pada database. *J* adalah banyaknya citra sesi pertama. Jarak hamming dilakukan dengan cara mengukur citra dengan menggunakan operasi logical XOR. Langkah selanjutnya adalah melakukan penjumlahan dimana jarak yang paling kecil merupakan hasil dari aplikasi pengenalan ini.

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kebenaran dan uji kinerja serta analisa setiap pengujian.

#### **5.1 Lingkungan Uji Coba**

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji aplikasi pengenalan *palm vein* berdasarkan fitur gabor pada Tugas Akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
  - a. Prosesor: Intel® Core™ i3-2350M CPU @ 2.30GHz
  - b. *Memory*(RAM): 4,00 GB
  - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
  - a. Sistem operasi: *Windows 8Professional*.
  - b. Perangkat pengembang: *Matlab 7.6.0 (R2008a)*.

#### **5.2 Data Training dan Data Testing**

Pembagian data training dan data testing dilakukan berdasarkan dengan sesi pengambilan citra *palm vein*. Untuk sesi pertama citra *palm vein* akan digunakan sebagai data training pada Tugas Akhir ini. Sedangkan sesi kedua akan dijadikan sebagai data testing. Jumlah data training dan data testing adalah sama, yaitu masing-masing berjumlah tiga citra. Citra yang digunakan sebagai data training dan data testing adalah citra yang sudah diekstraksi fitur gabornya dan berbentuk *vein code*.

### 5.3 Implementasi Kode Sumber

Pada subab ini berisi penjelasan mengenai hasil citra yang dihasilkan oleh kode sumber yang telah dibuat pada bab sebelumnya.

#### 5.3.1 Algoritma *Inner Border Tracing*

Algoritma ini digunakan untuk mendapatkan pixel border dari suatu objek pada citra. Gambar hasil dari algoritma *inner border tracing* yang diimplementasikan pada citra *palm vein* dapat dilihat pada Gambar 5. 1. Untuk citra input yang digunakan dapat dilihat pada Gambar 5. 2



**Gambar 5. 1** Gambar hasil algoritma *inner border tracing*

Pada Gambar 5. 1 dapat dilihat bahwa masih terdapat *noise* walaupun citra telah dilakukan proses *opening*. Untuk

menghilangkan noise ini, digunakan proses *thresholding* pada posisi pixel terhadap sumbu Y dalam kode sumber pengambilan titik ROI.

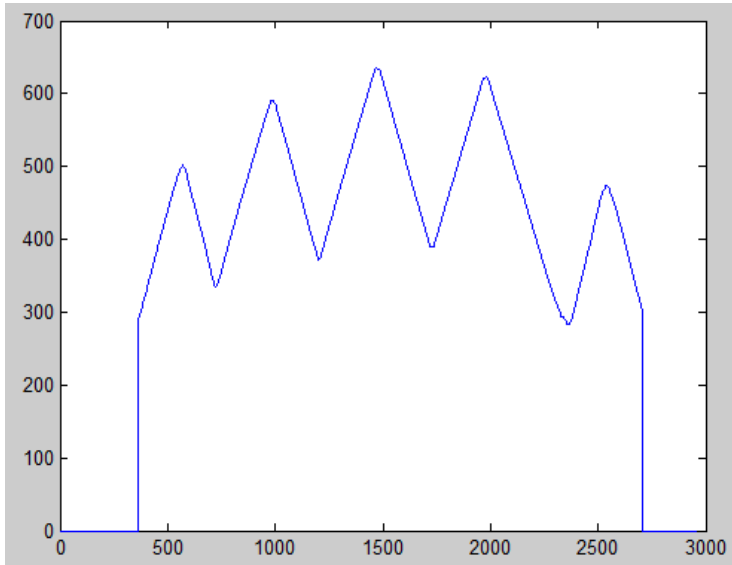


*Gambar 5. 2 Citra input palm vein*

### **5.3.2 Diagram Distribusi Jarak**

Diagram ini berfungsi dalam pengambilan titik yang diperlukan dalam ekstraksi ROI. Diagram distribusi jarak pada

aplikasi ini dapat dilihat pada gambar Gambar 5. 3 sedangkan hasil ekstraksi ROI dapat dilihat pada Gambar 5. 4.



**Gambar 5. 3 Gambar diagram Distribusi Jarak**

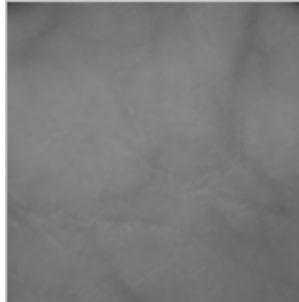
Sumbu X pada diagram diatas merupakan indeks dari tiap pixel pada citra *palm vein*, sedangkan sumbu Y adalah jarak dari pixel tersebut ke pixel pada bagian tengah bawah citra. Dapat dilihat diagram ini menyerupai citra telapak tangan karena terdiri dari lima puncak yang merepresentasikan jari pada citra.

Gambar 5. 4 merupakan citra hasil ekstraksi ROI berdasarkan pengambilan titik pada diagram distribusi jarak. Garis gelap pada citra ROI ini merupakan gambar dari pembuluh darah. Namun garis pembuluh ini masih terlihat samar, oleh karena itu diperlukan proses yang dapat membuat garis pembuluh terlihat dengan lebih jelas. Gambar 5. 5 merupakan gambar dari hasil proses tersebut.

Pada Gambar 5. 5 garis pembuluh lebih terlihat jelas jika dibandingkan dengan sebelumnya. Garis yang berwarna putih



merepresentasikan garis pembuluh sedangkan warna hitam disekelilingnya merupakan kulit pada telapak tangan. Gambar 5. 5 pada buku Tugas Akhir ini telah dilakukan thresholding sehingga garis pembuluh terlihat lebih jelas.



*Gambar 5. 4 Gambar hasil ekstraksi ROI*

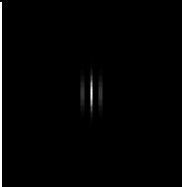
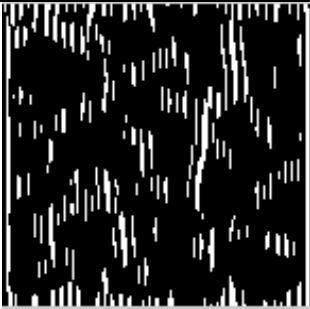
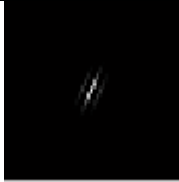





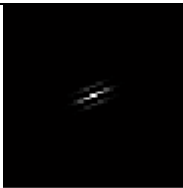

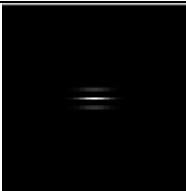

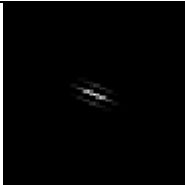
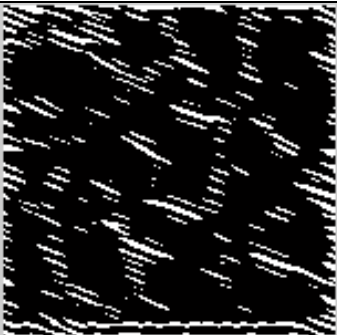
*Gambar 5. 5 Gambar penonjolan pembuluh pada ROI*




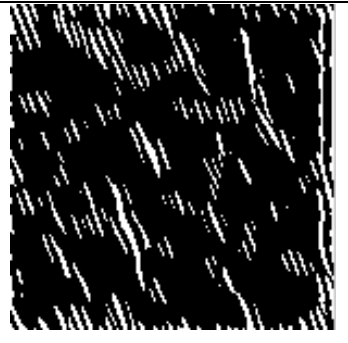
### **5.3.3 Ekstraksi Fitur Gabor dan Vein Code**

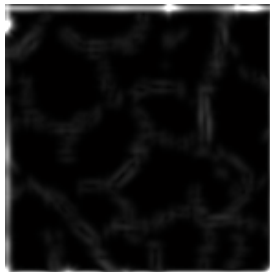
Ekstraksi fitur gabor dilakukan dengan melakukan konvolusi pada citra terhadap filter gabor. Tabel 5- 1 menunjukkan filter gabor pada orientasi  $0^\circ$ ,  $22.5^\circ$ ,  $45^\circ$ ,  $67.5^\circ$ ,  $90^\circ$ ,  $112.5^\circ$ ,  $135^\circ$ ,  $157.5^\circ$  beserta dengan citra hasil ekstraksi fitur untuk setiap orientasinya. Untuk citra hasil ekstraksi fitur berdasarkan orientasi telah dilakukan thresholding sehingga hasil tampak lebih jelas dibandingkan dengan aslinya.

***Tabel 5- 1 Tabel filter gabor dan hasil ekstraksi fitur gabor berdasarkan orientasinya***

Orientasi	Filter gabor	Hasil Ekstraksi Fitur berdasarkan orientasi
0°		
22.5°		
45°		

67.5°		
90°		
112.5°		

135°		
157.5°		



*Gambar 5. 6 Gambar hasil metode superposition*

Dapat dilihat pada Tabel 5- 1, citra hasil ekstraksi fitur berdasarkan orientasinya terdiri dari garis putus-putus berwarna

putih dimana garis putus-putus ini mengikuti arah atau orientasi dari filter gabor. Untuk percobaan dengan jumlah 4 orientasi digunakan  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ , sedangkan untuk percobaan dengan jumlah 8 orientasi digunakan  $0^\circ$ ,  $22.5^\circ$ ,  $45^\circ$ ,  $67.5^\circ$ ,  $90^\circ$ ,  $112.5^\circ$ ,  $135^\circ$ ,  $157.5^\circ$ .

Gambar 5. 6 menunjukkan gambar hasil dari ekstraksi fitur yang sudah digabung dengan menggunakan metode *superposition*. Pada gambar dapat dilihat bahwa garis pembuluh yang tadinya tipis melebar akibat dari ekstraksi fitur. Setelah dilakukan ekstraksi fitur, citra akan dibuat bentuk *vein code*-nya. *Vein code* pada citra *palm vein* dapat dilihat pada Gambar 5. 7.

#### 5.4 Skenario dan Evaluasi Pengujian

Uji coba ini dilakukan untuk menguji apakah fungsionalitas program telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. Uji coba akan didasarkan pada beberapa skenario untuk menguji kesesuaian dan kinerja aplikasi.

Skenario pengujian terdiri dari 3 pengujian yaitu:

1. Skenario perhitungan akurasi berdasarkan dengan luasan ROI.
2. Skenario perhitungan akurasi dengan perbedaan orientasi.
3. Skenario perhitungan akurasi berdasarkan variansi dari variabel lamda.
4. Skenario perhitungan akurasi berdasarkan variansi dari nilai *thresholding*.



*Gambar 5. 7 Gambar Vein Code*

#### 5.4.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah mengenai perbedaan luasan ROI, dimana perbedaan ini dimaksudkan untuk melihat apakah jumlah garis pembuluh yang terdeteksi pada luasan 150 x 150 dan 150 x 250 memiliki hasil yang sama, dimana pada luasan ROI 150 x 250 memiliki jumlah garis pembuluh yang lebih banyak dibandingkan pada luasan ROI 150 x 150. Penulis menggunakan 100 data individu, dimana setiap individunya terdiri dari enam citra yang dibagi menjadi dua untuk data training dan data testing. Luasan ROI yang akan dibandingkan adalah 150 x 150 dan 150 x 250.

Setelah dilakukan percobaan, maka harus dilakukan evaluasi untuk melihat sejauh mana kesesuaian aplikasi dengan data uji yang disediakan. Untuk menentukan kesesuaian keluaran dengan aplikasi dan analisa yang telah dibuat diperlukan pengujian akurasi. Pengujian akurasi dapat dilakukan dengan melihat kesesuaian hasil keluaran pada program dengan hasil *actual class* nya. Pada Tabel 5- 2 dijelaskan perbedaan pada luasan 150 x 150 dengan 150 x 250 pada dua data citra. Pada Tabel 5- 2, data citra yang digunakan adalah data citra individu ke-2 dan citra pada individu ke-3. Hasil yang didapatkan pada citra individu ke-2 adalah pada luasan 150 x 150 gagal dalam melakukan pengenalan dimana aplikasi mengenali citra input sebagai citra pada individu ke-54 sedangkan luasan 150 x 250 berhasil dalam melakukan proses pengenalan. Sebaliknya pada citra individu ke-3, hasil yang didapatkan adalah percobaan dengan luasan 150 x 150 berhasil mengenali citra input, dan percobaan dengan luasan ROI 150 x 250 gagal dalam mengenali citra input.

Pada akhir percobaan, didapatkan bahwa hasil akurasi dengan membandingkan dua luasan ROI adalah sama, yaitu 85.67%. Hasil skenario uji coba 1 secara lengkap dapat dilihat pada LAMPIRAN.

#### 5.4.2 Skenario Uji Coba 2

Skenario uji coba 2 mengenai perbedaan jumlah orientasi, skenario ini dimaksudkan untuk melihat apakah perbedaan jumlah orientasi pada pembuatan filter gabor akan berpengaruh terhadap tingkat akurasi yang akan dihasilkan, dimana semakin banyak orientasinya maka semakin detail hasil ekstraksi fitur gabor yang dihasilkan. Skenario uji coba 2 merupakan skenario percobaan dengan dua orientasi yang memiliki jumlah yang berbeda, dimana percobaan 1 memiliki 4 dengan orientasi yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ , sedangkan percobaan 2 memiliki jumlah 8 orientasi yaitu  $0^\circ$ ;  $22.5^\circ$ ,  $45^\circ$ ,  $67.5^\circ$ ,  $90^\circ$ ,  $112.5^\circ$ ,  $135^\circ$ ,  $157.5^\circ$ . Untuk luasan ROI yang digunakan pada kedua percobaan adalah  $150 \times 150$ .

Setelah dilakukan percobaan, maka harus dilakukan evaluasi untuk melihat sejauh mana kesesuaian aplikasi dengan data uji yang disediakan. Untuk menentukan kesesuaian keluaran dengan aplikasi dan analisa yang telah dibuat diperlukan pengujian akurasi. Pengujian akurasi dapat dilakukan dengan melihat kesesuaian hasil keluaran pada program dengan hasil *actual class* nya. Tabel perbedaan antara percobaan dengan 4 orientasi dan percobaan dengan 8 orientasi dapat dilihat pada Tabel 5- 3. Citra input yang digunakan pada Tabel 5- 3 adalah citra pada individu ke-13 dan citra pada individu ke-62. Hasil yang didapatkan dengan citra individu ke-13 sebagai citra input adalah percobaan dengan 4 orientasi mengenali citra input sebagai citra pada individu ke-56, dan percobaan dengan 8 orientasi berhasil mengenali citra sebagai citra individu ke-13. Untuk citra input pada individu ke-62, hasil yang didapatkan adalah pada percobaan 4 orientasi berhasil mengenali citra input sedangkan untuk percobaan dengan 8 orientasi mengenali citra input sebagai citra pada individu ke-34.

Pada akhir proses percobaan, didapatkan bahwa hasil akurasi dengan membandingkan dua jumlah orientasi yang berbeda adalah 85.67% untuk jumlah orientasi 4 dan 86% untuk jumlah orientasi 8. Hasil skenario uji coba 2 secara lengkap dapat dilihat pada LAMPIRAN.

### 5.4.3 Skenario Uji Coba 3

Skenario uji coba 3 adalah skenario yang berdasarkan pada variansi nilai  $\lambda$ . Skenario uji coba 3 dilakukan untuk melihat tingkat akurasi yang dihasilkan berdasarkan variansi dari variabel nilai  $\lambda$ , dimana semakin besar nilai  $\lambda$  maka semakin besar juga filter gabor yang dihasilkan. Variansi nilai  $\lambda$  yang digunakan karena nilai  $\lambda$  berpengaruh pada hasil ekstraksi fitur. Semakin besar nilai  $\lambda$  maka lebar garis pada pembuluh semakin melebar. Nilai  $\lambda$  maksimum yang digunakan adalah 4, sedangkan nilai  $\lambda$  minimumnya adalah 2.

Setelah dilakukan percobaan, maka harus dilakukan evaluasi untuk melihat sejauh mana kesesuaian aplikasi dengan data uji yang disediakan. Untuk menentukan kesesuaian keluaran dengan aplikasi dan analisa yang telah dibuat diperlukan pengujian akurasi. Pengujian akurasi dapat dilakukan dengan melihat kesesuaian hasil keluaran pada program dengan hasil *actual class* nya. Grafik hasil akurasi skenario uji coba 3 dapat dilihat pada Gambar 5. 8.



Setelah proses percobaan dijalankan, didapatkan hasil akurasi terendah adalah 66.33% terletak pada nilai  $\lambda$  2, dan hasil akurasi tertinggi adalah 86% terletak pada nilai  $\lambda$  3.5. Hasil skenario uji coba 3 secara lengkap dapat dilihat pada LAMPIRAN.



### 5.4.4 Skenario Uji Coba 4

Skenario uji coba 4 adalah variansi pada nilai *thresholding* pada saat pembuatan citra *vein code*. Skenario ini dilakukan untuk melihat perbedaan yang dihasilkan berdasarkan nilai pembatas pada proses *thresholding*. Variansi nilai *thresholding* yang diujikan adalah 0.01, 0.02, 0.03, 0.04, dan 0.05, dimana semakin kecil nilai *thresholding* maka semakin lebar pembuluh pada citra *vein code* yang akan dihasilkan.







*Tabel 5- 2 Tabel perbedaan pada uji skenario 1*

Individu (Kelas Sebenarnya)	Luasan ROI	Prediksi Kelas
2	Luasan ROI 150 x 150 	54
2	Luasan ROI 150 x 250 	2

3	<div>Luasan ROI 150 x 150</div> 	3
3	<div>Luasan ROI 150 x 250</div> 	31

*Tabel 5- 3 Tabel perbedaan pada uji skenario 2*

Individu (Kelas Sebenarnya)	Jumlah orientasi	Prediksi Kelas
13	4 orientasi 	56
13	8 orientasi 	13
62	4 orientasi 	62

62	8 orientasi	34
		

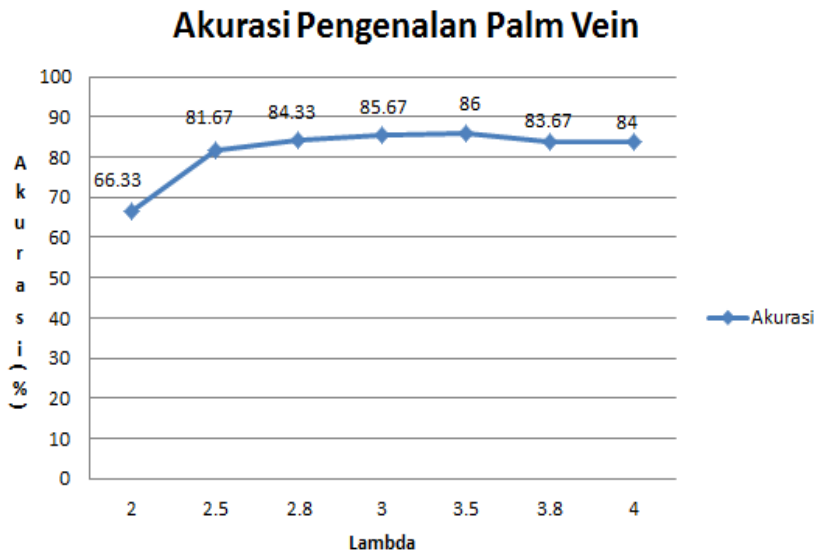
Setelah dilakukan percobaan, maka harus dilakukan evaluasi untuk melihat sejauh mana kesesuaian aplikasi dengan data uji yang disediakan. Untuk menentukan kesesuaian keluaran dengan aplikasi dan analisa yang telah dibuat diperlukan pengujian akurasi. Pengujian akurasi dapat dilakukan dengan melihat kesesuaian hasil keluaran pada program dengan hasil *actual class* nya. Grafik hasil akurasi skenario uji coba 4 dapat dilihat pada Gambar 5. 9.

Hasil yang didapatkan pada skenario uji coba ini yaitu nilai akurasi terendah didapatkan pada nilai thresholding 0.05 dengan tingkat akurasi sebesar 77.67%, sedangkan tingkat akurasi terbaik diperoleh pada nilai thresholding 0.02 sebesar 86%. Hasil skenario uji coba 4 secara lengkap dapat dilihat pada LAMPIRAN.

## 5.5 Analisis Hasil Uji Coba

Pada skenario uji coba 1 didapatkan bahwa hasil akurasi yang didapatkan adalah sama yaitu 85.67% namun pada prosesnya, kedua percobaan baik luasan ROI 150 x 150 dan luasan ROI 150 x 250 memiliki hasil yang berbeda. Tabel 5- 2 dengan citra pada individu ke-2 sebagai citra input menunjukkan bahwa luasan ROI 150 x 150 gagal dalam mengenali citra input sedangkan ROI 150 x 250 berhasil dalam mengenali citra tersebut

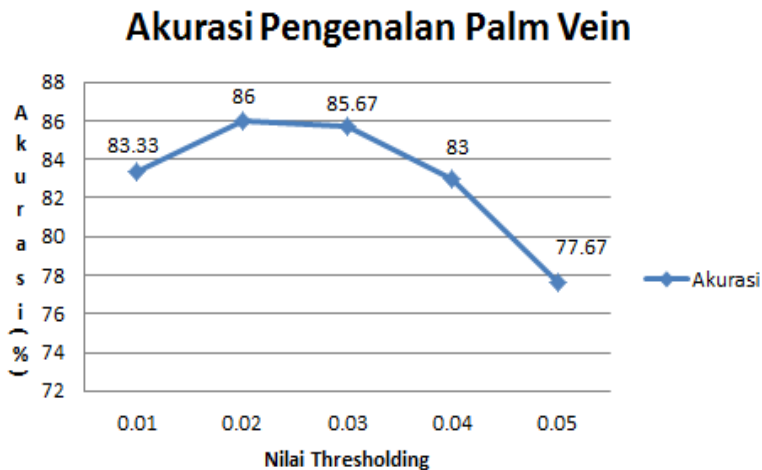
sebagai citra pada individu ke-2. Pada citra pada individu ke-3 sebagai citra input, hasil yang didapatkan adalah percobaan dengan luasan ROI 150 x 150 berhasil dalam mengenali citra input, sedangkan percobaan luasan ROI 150 x 250 gagal dalam mengenali citra input. Luasan ROI 150 x 150 gagal dalam mengenali citra input dikarenakan masih terdapat noise pada garis pembuluh. Noise yang dimaksudkan adalah titik-titik hitam yang terdapat pada pembuluh. Sedangkan luasan ROI 150 x 250 gagal dalam proses pengenalan dikarenakan pada beberapa citra *palm vein* terdapat garis tangan yang gelap dimana garis tangan ini dideteksi sebagai garis pembuluh. Perbedaan yang tampak pada kedua percobaan adalah pada *vein code* dengan luas 150 x 250, garis pembuluh tampak lebih lebar dan titik-titik hitam yang masih terdapat pada luasan 150 x 150 telah menghilang.



**Gambar 5. 8** Grafik hasil akurasi skenario uji coba 3

Pada skenario uji coba 2 didapatkan hasil akurasi untuk percobaan dengan 4 orientasi adalah 85.67% dan untuk percobaan

dengan 8 orientasi adalah 86%. Perbedaan pada kedua percobaan dapat dilihat pada citra *vein code* yang ditampilkan pada Tabel 5-3. Pada percobaan dengan 8 orientasi, *border* atau garis tepi pada pembuluh tampak lebih halus jika dibandingkan dengan percobaan 4 orientasi dan pembuluh pada 8 orientasi juga tampak lebih lebar. Hasil yang didapatkan dengan citra individu ke-13 sebagai citra input adalah percobaan dengan 4 orientasi gagal dalam mengenali citra input sedangkan percobaan dengan 8 orientasi berhasil mengenali citra tersebut sebagai citra pada individu ke-13. Hasil yang didapatkan dengan citra individu ke-62 sebagai citra input adalah percobaan dengan 4 orientasi berhasil dalam mengenali citra input sebagai citra pada individu ke-62, sedangkan percobaan dengan 8 orientasi gagal dalam mengenali citra input. Percobaan dengan 4 orientasi mengalami kegagalan dalam proses pengenalan disebabkan karena terdapat noise berupa titik hitam pada pembuluh. Percobaan dengan 8 orientasi gagal dalam mengenali citra input dikarenakan lebarnya pembuluh membuat pembuluh-pembuluh yang terpisah menjadi satu.



**Gambar 5. 9** Grafik hasil akurasi skenario uji coba 4



*Gambar 5. 10 Gambar citra vein code pada nilai lambda 2*



*Gambar 5. 11 Gambar citra vein code pada nilai lambda 3.6*

Nilai lambda yang digunakan dalam uji coba 3 ini digunakan dalam pembuatan filter gabor. Variansi nilai yang diujikan dimulai dari 2 sampai 4. Hasil yang didapatkan pada uji coba ini adalah nilai lambda 2 memiliki hasil akurasi terendah yaitu 66.33% sedangkan untuk hasil akurasi tertinggi yaitu 86% didapatkan pada nilai lambda 3.5. Nilai lambda 2 memiliki akurasi yang rendah dikarenakan citra *vein code* yang dihasilkan memiliki titik-titik yang lebih banyak dan banyak pembuluh yang terputus jika dibandingkan dengan citra *vein code* pada nilai lambda 3.6. Gambar 5. 10 dan Gambar 5. 11 menunjukkan citra *vein code* pada nilai lambda 2 dan 3.6.

Variansi nilai *thresholding* pada uji coba 4 digunakan dalam pembentukan citra *vein code*, dimana variansi yang diujikan adalah 0.01, 0.02, 0.03, 0.04, dan 0.05. Semakin rendah nilai *thresholding* maka semakin tebal garis pembuluh yang

dihasilkan Hasil akurasi terendah yang didapatkan sebesar 77.67% pada nilai *thresholding* 0.05, dan hasil tertinggi yang didapatkan adalah 86% pada nilai *thresholding* 0.02. Hasil tertinggi didapatkan pada nilai 0.02 karena citra *vein code* yang dihasilkan lebih menyerupai citra hasil ekstraksi fitur gabor jika dibandingkan dengan yang lainnya, sedangkan pada nilai *thresholding* 0.05 didapatkan hasil terendah karena garis pembuluh lebih tipis dan semakin banyak titik-titik hitam pada garis pembuluh



## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

#### **6.1 Kesimpulan**

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

1. Aplikasi pengenalan citra palm vein berdasarkan fitur gabor mampu menghasilkan pengenalan dengan tingkat akurasi sebesar 85.67%.
2. Berdasarkan skenario ujicoba yang telah dilakukan, perbedaan luasan ROI dan perbedaan jumlah orientasi tidak berpengaruh secara signifikan terhadap tingkat akurasi semula.
3. Skenario uji coba dengan variansi nilai lambda, tingkat akurasi terendah yaitu 66.33% didapatkan pada nilai lambda 2 dan tingkat akurasi terbaik didapatkan pada nilai lambda 3.5 yaitu sebesar 86%.
4. Pada skenario uji coba dengan variansi nilai thresholding, didapatkan akurasi terendah adalah 77.67% pada nilai thresholding 0.05 dan tingkat akurasi terbaik didapatkan pada nilai thresholding 0.02 yaitu 86%.

#### **6.2 Saran**

Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Untuk meningkatkan nilai akurasi, dapat dicoba untuk mengubah parameter gabor filter yang digunakan, seperti lambda dan frekuensi.

2. Dapat menggunakan metode lain selain gabor sebagai metode dalam melakukan ekstraksi fitur.
3. Melakukan proses lain dalam melakukan penonjolan pembuluh sehingga objek pembuluh lebih tampak.

## DAFTAR PUSTAKA

- [1] Jen-Chun Lee, "A novel biometric system based on palm vein image," *Pattern Recognition Letters*, vol. 33, pp. 1520-1528, April 2012.
- [2] Feutech Engineers. [Online].  
[http://feutechengineers.com/Access-Control-Details.php?p\\_id=36](http://feutechengineers.com/Access-Control-Details.php?p_id=36)
- [3] International Conferences. [Online].  
<http://www.conferenceseries.com/biometric.php>
- [4] Biometrics Institute. [Online].  
<http://www.biometricsinstitute.org/pages/types-of-biometrics.html>
- [5] International Conferences. [Online].  
<http://www.conferenceseries.com/biometric-devices.php>
- [6] Pallavi Chauhan, "Steps In Fingerprint Enhancement Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 7, July 2014.
- [7] Manpriya kaur. (2014, December) A Research Study of Different Biometrics and Basic Analysis. [Online].  
<https://prezi.com/dbgljvrjsiwy/a-research-study-of-different-biometrics-and-basic-analysis/>
- [8] Biometric Newsportal. [Online].  
[http://www.biometricnewsportal.com/palm\\_biometrics.asp](http://www.biometricnewsportal.com/palm_biometrics.asp)
- [9] Biometrics Ideal Test. [Online].  
<http://biometrics.idealtest.org/dbDetailForUser.do?id=6>
- [10] Jianzhu Cui, Jing Li Lizhuang Liu, *Advances in Mechanical and Electronic Engineering*, Sally Lin David Jin, Ed.: Springer, 2012.
- [11] World Ebook Library. [Online].  
[http://www.worldebooklibrary.com/articles/Hamming\\_distance](http://www.worldebooklibrary.com/articles/Hamming_distance)

- [12] Cram101 Textbook Reviews, *e-Study Guide for Number Theory: An Introduction to Mathematics*, textbook by W.A. Coppel: *Mathematics, Mathematics*, 2nd ed.: Cram101 Textbook Reviews, 2012, 2012.
- [13] (2014, July) Loop and Break. [Online].  
<http://www.loopandbreak.com/hamming-distance/>
- [14] (2008, September) Liupq's Blog. [Online].  
<https://liupq.wordpress.com/2008/09/11/hamming-distance/>
- [15] Byeong-Ho KANG, "A Review on Image and Video processing," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 2, no. 2, April 2007.
- [16] Mrunalini P Moon Rashmi A Jain, "Image Processing And Pattern Generation," *International Journal of Engineering Research and*, vol. 2, no. 5, pp. 1888-1893, October 2012.
- [17] Academia. [Online].  
<http://www.academia.edu/9609004/High>
- [18] MathWorks. [Online].  
<http://www.mathworks.com/help/images/morphology-fundamentals-dilation-and-erosion.html>
- [19] Revolvly. [Online].  
[http://www.revolvly.com/main/index.php?s=Region%2520of%2520Interest&item\\_type=topic](http://www.revolvly.com/main/index.php?s=Region%2520of%2520Interest&item_type=topic)
- [20] J.G. Daugman, "High confidence visual recognition of persons by a test of," *IEEE Tran*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [21] (2008, July) Matlab Server. [Online].  
[http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection\\_params.html](http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection_params.html)
- [22] MathWorks. [Online].  
[http://www.mathworks.com/matlabcentral/fileexchange/23253-gabor-filter/content/Gabor%20Filter/gabor\\_fn.m](http://www.mathworks.com/matlabcentral/fileexchange/23253-gabor-filter/content/Gabor%20Filter/gabor_fn.m)

## LAMPIRAN

**Tabel A- 1 Rekap Hasil Keluaran Program Skenario Uji Coba 1,2,3,4 dengan luasan ROI 150 x 150, jumlah orientasi 4, lambda 3, dan nilai thresholding 0.03**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	5
6	58	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	60
22	22	22	86

23	23	23	23
24	24	24	24
25	25	25	25
26	26	53	52
27	27	27	27
28	20	28	28
29	26	62	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46

47	47	47	47
48	48	48	34
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	80	24
58	58	58	58
59	61	59	59
60	60	60	60
61	61	22	35
62	62	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	96
67	94	67	53
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74

75	75	75	75
76	39	8	70
77	77	77	77
78	78	78	78
79	79	79	79
80	80	80	8
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	50	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	4	99	99
100	100	100	100
Total		84	91
Akurasi (%)		85.67	

**Tabel A- 2 Rekap Hasil Keluaran Program Skenario Uji Coba 1  
dengan luasan ROI 150 x 250**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	2	2	2
3	31	3	3
4	4	4	98
5	5	5	77
6	23	6	23
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	30	56
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	56	87	26
22	22	22	28
23	23	23	23
24	24	24	24
25	25	25	25

26	26	14	26
27	27	27	27
28	28	28	28
29	29	72	29
30	30	34	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	32	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	96	48	79
49	49	49	49
50	50	50	50
51	92	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	34
56	56	56	56
57	55	80	80
58	58	58	58
59	59	59	59
60	60	95	60
61	61	22	46
62	62	94	89
63	63	63	63
64	64	64	64
65	65	65	65
66	39	66	5
67	73	67	73
68	68	68	68
69	69	69	69
70	70	70	70
71	99	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	76	70	76
77	77	77	77

78	78	78	78
79	79	79	79
80	80	80	80
81	15	81	81
82	82	82	82
83	31	31	74
84	84	84	84
85	46	85	35
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	70	96	96
97	97	97	97
98	98	98	98
99	60	99	99
100	100	100	100
Total	85	89	83
Akurasi (%)		85.67	



**Tabel A- 3 Rekap Hasil Keluaran Program Skenario Uji Coba 2 dengan 8 orientasi**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	5
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	39
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	46	28	28
29	26	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	80	24
58	58	58	58
59	61	59	59
60	60	60	60
61	61	22	46
62	34	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	38
67	53	67	53
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	61	8	8
77	77	77	77

78	78	78	78
79	79	79	35
80	80	80	80
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	50	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	36	99	99
100	100	100	100
Total	83	92	83
Akurasi (%)		86	

**Tabel A- 4 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	34
2	20	2	2
3	20	3	20
4	4	4	62
5	5	5	97
6	20	6	84
7	7	7	7
8	20	8	8
9	96	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	5
14	14	14	14
15	15	20	20
16	16	16	16
17	17	17	17
18	18	18	18
19	19	20	19
20	20	20	20
21	94	20	20
22	96	20	80
23	23	23	23
24	24	84	24
25	20	25	25

26	26	20	4
27	27	27	27
28	87	28	28
29	82	62	29
30	30	34	96
31	96	20	31
32	32	32	32
33	33	33	33
34	34	56	34
35	35	35	12
36	94	34	36
37	20	37	37
38	38	38	38
39	27	27	96
40	40	83	65
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	55	45
46	46	46	46
47	47	96	96
48	96	96	20
49	49	49	49
50	50	50	50
51	28	97	51
52	52	20	52

53	53	53	53
54	54	54	54
55	55	55	26
56	4	56	56
57	20	80	56
58	58	58	58
59	20	20	20
60	60	84	60
61	61	79	96
62	34	8	34
63	63	63	20
64	64	64	64
65	65	65	72
66	4	96	96
67	94	67	96
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	97
74	20	72	74
75	75	20	75
76	20	20	70
77	77	77	77

78	78	78	78
79	56	20	96
80	80	80	8
81	16	81	81
82	82	82	82
83	56	20	56
84	84	84	84
85	56	15	28
86	86	86	86
87	96	87	4
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	60
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	96	98	20
99	65	4	4
100	100	100	100
Total	68	68	63
Akurasi (%)		66.33	

**Tabel A- 5 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2.5**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	97
6	75	6	24
7	7	7	7
8	8	8	8
9	8	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	56
14	14	14	14
15	15	15	44
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	86	20	60
22	22	22	20
23	23	23	23
24	24	24	24
25	25	25	25

26	26	1	62
27	27	27	27
28	20	28	28
29	82	62	29
30	30	60	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	60
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	96
48	48	48	34
49	49	49	49
50	50	50	50
51	28	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	55	80	36
58	58	58	58
59	20	59	59
60	60	84	60
61	61	79	96
62	34	8	27
63	63	63	63
64	64	64	64
65	65	65	65
66	11	66	96
67	94	67	96
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	70	8	70
77	77	77	77

78	78	78	78
79	79	79	96
80	80	80	8
81	16	81	81
82	82	82	82
83	83	83	72
84	84	84	84
85	56	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	30	98	98
99	65	4	4
100	100	100	100
Total	82	88	75
Akurasi (%)		81.67	

**Tabel A- 6 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 2.8**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	5
6	75	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	56
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	60
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	20	28	28
29	26	62	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	76
48	48	48	34
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	80	24
58	58	58	58
59	39	59	59
60	60	60	60
61	61	22	35
62	34	8	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	96
67	94	67	94
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	39	8	70
77	77	77	77

78	78	78	78
79	79	79	79
80	80	80	8
81	16	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	50	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	84	99	99
100	100	100	100
Total	83	90	80
Akurasi (%)		84.33	



**Tabel A- 7 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 3.5**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	16
5	5	5	93
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	71
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	39
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	46	28	28
29	26	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	5	24
58	58	58	58
59	59	59	59
60	60	60	60
61	61	22	46
62	74	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	38
67	53	67	53
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	39	8	8
77	77	77	77

78	78	78	78
79	79	79	79
80	80	80	80
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	46	15	74
86	86	86	86
87	87	87	50
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	4	99	99
100	100	100	100
Total	85	92	81
Akurasi (%)		86	

**Tabel A- 8 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 3.8**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	93	5	93
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	66
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	39
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	46	28	71
29	26	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	92
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	90
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	92	88	92
58	58	58	58
59	59	59	59
60	60	49	60
61	61	22	46
62	23	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	19
67	53	67	53
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	61	8	8
77	77	77	77

78	78	78	78
79	79	79	35
80	80	80	80
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	46	15	74
86	86	86	86
87	87	87	50
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	69	99	99
100	100	100	100
Total	83	91	77
Akurasi (%)		83.67	

**Tabel A- 9 Rekap Hasil Keluaran Program Skenario Uji Coba 3 dengan lambda 4**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	47	2	2
3	3	3	3
4	4	4	16
5	93	5	93
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	19
14	14	14	14
15	15	15	88
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	99	67	68
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	46	28	71
29	29	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	92
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	90
49	49	49	49
50	50	50	50
51	23	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	34
56	56	56	56
57	92	38	92
58	58	58	58
59	59	59	59
60	60	49	60
61	61	61	46
62	23	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	19
67	53	67	82
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	61	8	8
77	77	77	77

78	78	78	78
79	79	79	35
80	80	80	80
81	74	81	81
82	82	82	82
83	83	66	66
84	84	84	84
85	46	15	74
86	86	86	86
87	87	87	50
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	46
97	97	97	97
98	5	98	98
99	99	99	99
100	100	100	100
Total	85	91	76
Akurasi (%)		84	

**Tabel A- 10 Rekap Hasil Keluaran Program Skenario Uji Coba 4  
dengan nilai thresholding 0.01**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	47	2	2
3	3	3	97
4	4	4	16
5	5	5	93
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	88
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	67	68
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	31	28	71
29	26	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	92
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	90
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	98	56	56
57	92	5	5
58	58	58	58
59	59	59	59
60	60	49	60
61	61	22	46
62	23	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	19
67	53	67	82
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	74
73	73	73	73
74	74	74	74
75	75	75	75
76	39	8	8
77	77	77	77

78	78	78	78
79	79	79	35
80	80	80	80
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	46	15	74
86	86	86	86
87	87	87	50
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	46
97	97	97	97
98	5	98	98
99	99	99	99
100	100	100	100
Total	84	91	75
Akurasi (%)		83.33	



**Tabel A- 11 Rekap Hasil Keluaran Program Skenario Uji Coba 4  
dengan nilai thresholding 0.02**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	93
6	39	6	24
7	7	7	7
8	8	8	8
9	9	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	26	63
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	52
27	27	27	27
28	31	28	28
29	26	17	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	9	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	80	24
58	58	58	58
59	61	59	59
60	60	60	60
61	61	22	46
62	74	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	6	66	38
67	53	67	53
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	61	8	8
77	77	77	77

78	78	78	78
79	79	79	79
80	80	80	80
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	50	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	36	99	99
100	100	100	100
Total	83	92	83
Akurasi (%)		86	

**Tabel A- 12 Rekap Hasil Keluaran Program Skenario Uji Coba 4 dengan nilai thresholding 0.04**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	54	2	2
3	3	3	3
4	4	4	16
5	5	5	97
6	58	6	24
7	7	7	7
8	8	8	8
9	18	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	56
14	14	14	14
15	15	15	44
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	39	20	60
22	22	22	86
23	23	23	23
24	24	24	24
25	25	25	25

26	26	53	62
27	27	27	27
28	28	28	28
29	82	62	29
30	30	29	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	76
48	48	48	34
49	49	49	49
50	50	50	50
51	28	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	56	56	56
57	24	80	24
58	58	58	58
59	20	59	59
60	60	84	60
61	61	79	35
62	34	62	27
63	63	63	63
64	64	64	64
65	65	65	65
66	11	66	96
67	94	67	94
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	39	8	70
77	77	77	77

78	78	78	78
79	79	79	79
80	80	80	96
81	74	81	81
82	82	82	82
83	83	83	66
84	84	84	84
85	56	15	74
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	5	98	98
99	84	4	4
100	100	100	100
Total	83	89	77
Akurasi (%)		83	

**Tabel A- 13 Rekap Hasil Keluaran Program Skenario Uji Coba 4  
dengan nilai thresholding 0.05**

Kelas Citra	Indeks Citra ke - i		
	i=4	i=5	i=6
1	1	1	1
2	96	2	2
3	3	3	3
4	4	4	16
5	5	5	97
6	58	6	24
7	7	7	7
8	8	8	8
9	54	9	9
10	96	10	10
11	11	11	11
12	12	12	12
13	13	56	56
14	14	14	14
15	15	15	44
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	86	20	60
22	96	22	20
23	23	23	23
24	24	84	24
25	25	25	25

26	26	1	4
27	27	27	27
28	87	28	28
29	82	62	29
30	30	34	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	1	36	36
37	20	37	37
38	38	38	38
39	39	39	96
40	40	40	40
41	41	41	41
42	42	42	15
43	43	43	43
44	44	44	44
45	45	55	45
46	46	46	46
47	47	47	96
48	48	96	20
49	49	49	49
50	50	50	50
51	87	51	51
52	52	52	52

53	53	53	53
54	54	54	54
55	55	55	9
56	4	56	56
57	20	15	24
58	58	58	58
59	20	59	59
60	60	84	60
61	61	79	35
62	34	8	62
63	63	63	63
64	64	64	64
65	65	65	65
66	20	96	96
67	94	67	96
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	20	8	70
77	77	77	77

78	78	78	78
79	79	94	96
80	80	80	96
81	16	81	81
82	82	82	82
83	20	83	56
84	84	84	84
85	56	15	74
86	86	86	86
87	96	87	4
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	70
95	95	95	95
96	96	96	96
97	97	97	97
98	62	98	97
99	84	4	4
100	100	100	100
Total	76	83	74
Akurasi (%)		77.67	

## BIODATA PENULIS



Peter Panyananda Handira, lahir di Banjarmasin, pada tanggal 18 Februari 1993. Penulis menempuh pendidikan mulai dari SDK III Wr.Soepratman (1999-2005), SMP Santo Fransiskus Assisi (2005-2008), SMA Kolese Santo Yusup (2008-2011) dan S1 Teknik Informatika ITS (2011-2015). Selama masa kuliah, penulis pernah mengikuti kepanitian dalam kejuaraan basket 3C sebagai anggota dari seksi

perlengkapan

Selama kuliah di teknik informatika ITS. Penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Komunikasi dengan penulis dapat melalui email: **peter.if.112@gmail.com**