



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

Pengenalan Palm Vein Menggunakan Metode Ekstraksi Fitur Local Binary Patterns by Neighborhoods (nLBP)

DELIA RISTI NARESWARI
NRP 05111440000033

Dosen Pembimbing
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

Pengenalan Palm Vein Menggunakan Metode Ekstraksi Fitur Local Binary Patterns by Neighborhood (nLBP)

DELIA RISTI NARESWARI
NRP 05111440000033

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

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

***PALM VEIN RECOGNITION USING LOCAL BINARY
PATTERNS BY NEIGHBORHOODS (nLBP)
FEATURE EXTRACTION METHOD***

DELIA RISTI NARESWARI
NRP 05111440000033

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

Supervisor II
Dini Adni Navastara, S.Kom., M.Sc

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Pengenalan Palm Vein Menggunakan Metode Ekstraksi Fitur Local Binary Patterns by Neighborhoods (nLBP)

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

DELIA RISTI NARESWARI
NRP : 05111440000033

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom
NIP. 19751220 200112 2 002



Dini Adni Navastara, S.Kom., M.Sc.
NIP. 19851017 201504 2 001

(pembimbing 2)

SURABAYA
JUNI 2018

[Halaman ini sengaja dikosongkan]

Pengenalan Palm Vein Menggunakan Metode Ekstraksi Fitur Local Binary Patterns by Neighborhoods (nLBP)

Nama Mahasiswa : Delia Risti Nareswari
NRP : 05111440000033
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRAK

Saat ini, informasi sudah menjadi hal yang sangat bernilai dalam aspek kehidupan sehingga proteksi terhadap keamanan informasi menjadi begitu penting. Teknologi autentikasi seperti pasangan user ID dan password telah digunakan di berbagai sistem. Namun, teknologi tersebut dirasakan memiliki banyak kekurangan karena sangat rentan untuk dicuri, dilupakan, dan dipalsukan sehingga dibutuhkan teknologi keamanan lain yang lebih aman. Salah satu keamanan yang dianggap lebih aman dan akurat adalah teknologi autentikasi biometrik. Selama ini sudah terdapat beberapa penelitian autentikasi berbasis karakteristik biometrik yang sudah dikembangkan, salah satunya yaitu autentikasi biometrik berdasarkan pola pembuluh darah vena telapak tangan (palm vein). Dalam pengenalan palm vein, akan dilakukan proses mendapatkan ciri dari pembuluh darah tersebut yaitu dengan ekstraksi fitur. Proses ekstraksi fitur ini sangat berpengaruh dalam tahap pengenalan palm vein.

Tugas Akhir ini mengusulkan penggunaan metode Local Binary Patterns by Neighborhoods (nLBP) untuk pengenalan palm vein. Proses pengenalan palm vein terdiri dari 3 proses utama yaitu preprocessing yang bertujuan mendapatkan Region of Interest (ROI) pada citra pembuluh darah telapak tangan serta

memperbaiki citra agar terlihat pola pembuluh darah. Proses kedua yaitu ekstraksi fitur untuk mendapatkan fitur atau ciri dari pola pembuluh darah. Proses terakhir adalah melakukan klasifikasi menggunakan Support Vector Machine (SVM).

Data yang digunakan sebanyak 600 data citra palm vein dengan jumlah 100 kelas. Hasil akurasi uji coba pengenalan palm vein menggunakan metode ekstraksi fitur nLBP yaitu sebesar 86.66% dengan parameter distance metode nLBP yaitu 4, gamma sebesar 0.001, nilai penalty coefficient (C) sebesar 2, dan menggunakan kernel RBF. Sedangkan untuk pengenalan palm vein menggunakan metode LBP menghasilkan akurasi sebesar 81.66%. Dari hasil uji coba dapat disimpulkan bahwa metode yang diusulkan nLBP menghasilkan akurasi lebih baik dalam pengenalan palm vein dibandingkan LBP.

Kata kunci: palm vein, pengenalan palm vein, pola pembuluh darah, nLBP, klasifikasi SVM.

PALM VEIN RECOGNITION USING LOCAL BINARY PATTERNS BY NEIGHBORHOODS (nLBP) FEATURE EXTRACTION METHOD

Student Name : Delia Risti Nareswari
Student ID : 05111440000033
Major : Informatics Department Faculty of ICT
1st Supervisor : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
2nd Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRACT

Nowadays, information has become very valuable in the aspects of life so that protection of information security becomes so important. Authentication technologies such as user ID and password have been used in various systems. However, the technology is felt to have many shortcomings because it is very vulnerable to be stolen, forgotten, and falsified so that more secure security technology is required. One of the security that is considered more secure and accurate is the biometric authentication technology. So far, there have been some authentication studies based on biometric characteristics that have been developed, one of which is biometric authentication based on palm vein pattern. In the palm vein recognition, the process to get the characteristics of the blood vessel is called feature extraction. The feature extraction process is very influential in palm vein recognition.

This undergraduate thesis proposes Local Binary Patterns by Neighborhoods (nLBP) feature extraction method for the palm vein recognition. The process of palm vein recognition consists of three main processes of preprocessing aimed at obtaining Region of Interest (ROI) on the palm vein image and improving the image to be visible patterns of blood vessels. The second process is feature extraction to get features or characteristics of the blood

vessel pattern. The last process is to classify image using the Support Vector Machine (SVM).

The data used as much as 600 data palm vein image with the number of classes are 100. The result of accuracy of palm vein recognition test using nLBP feature extraction method is 86.66% with distance parameter of nLBP method is 4, gamma is 0.001, penalty coefficient (C) value is 2, and using RBF kernel. As for the palm vein recognition using LBP method produces an accuracy of 81.66%. From the experimental results, it can be concluded that the proposed method nLBP produces better accuracy in palm vein recognition than LBP.

Keywords: palm vein, palm vein recognition, palm vein pattern, nLBP, SVM classifier

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Pengenalan Palm Vein Menggunakan Metode Ekstraksi Fitur Local Binary Patterns by Neighborhoods (nLBP)”**.

Pengerjaan tugas akhir ini menjadi suatu pengalaman yang baik bagi penulis. Penulis dapat memperoleh banyak pengalaman yang berharga dalam memperdalam dan meningkatkan keilmuan dalam bidang informatika selama perkuliahan di Teknik Informatika ITS.

Tugas akhir ini selesai karena tidak lepas dari bantuan dan dukungan berbagai pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT, karena atas izin-Nya lah penulis dapat menyelesaikan Tugas Akhir dengan baik.
2. Orangtua penulis (Darmanto dan Riris Ratna) yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Dr.Eng. Chastine Fatichah, S.Kom, M.Kom. dan Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
4. Saudara kandung (Rinda Ayu dan Gama Putra) serta seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.
5. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku kepala Laboratorium KCV yang bersedia meminjamkan salah satu komputer untuk pengerjaan tugas akhir.
6. Dhanar Prayoga, S.Kom., yang selalu menemani, menghibur, dan memberikan semangat kepada penulis.

7. Teman-teman di Laboratorium KCV: para admin yang telah banyak membantu memfasilitasi penulis dalam pengerjaan, *user* TA lainnya dan semua *user* KCV yang sudah menemani penulis.
8. Mahasiswa angkatan 2014 lain yang sama-sama berjuang dari semester awal hingga akhir yang banyak memberi bantuan, semangat serta pelajaran hidup bagi penulis.
9. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

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

Surabaya, Juni 2018

Delia Risti Nareswari

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Tugas Akhir.....	3
1.5. Manfaat Tugas Akhir.....	3
1.6. Metodologi	3
1.7. Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1. Pengenalan <i>Palm Vein</i>	7
2.2. <i>Local Binary Patterns by Neighborhoods (nLBP)</i>	7
2.3. <i>Region Of Interest (ROI)</i>	9
2.4. <i>Median Filter</i>	10
2.5. <i>Adaptive Histogram Equalization</i>	10
2.6. <i>Adaptive Noise Removal</i>	11
2.7. Anisotropic Diffusion Filter	13
2.8. Operasi Morfologi	14
2.8.1. Dilasi	15
2.8.2. Erosi	15
2.8.3. <i>Opening</i>	16
2.8.4. <i>Closing</i>	16
2.9. Klasifikasi <i>Support Vector Machine (SVM)</i>	17
BAB III PERANCANGAN PERANGKAT LUNAK	21
3.1. Data	21

3.1.1. Data Masukan.....	21
3.1.2. Data Proses.....	23
3.1.3. Data Keluaran.....	23
3.2. Desain Sistem Secara Umum	24
3.3. Akuisisi Citra <i>Palm Vein</i>	24
3.4. Perancangan <i>Preprocessing</i>	25
3.4.1. ROI Detection.....	27
3.4.2. Perbaikan Citra	28
3.4.3. Penonjolan Pembuluh Darah Telapak Tangan	28
3.5. Perancangan Ekstraksi Fitur.....	30
3.6. Perancangan Klasifikasi	31
BAB IV IMPLEMENTASI.....	35
4.1. Lingkungan Implementasi.....	35
4.2. Implementasi	35
4.2.1. Implementasi Akuisisi Citra <i>Palm Vein</i>	35
4.2.2. Implementasi <i>Preprocessing</i>	36
4.2.3. Implementasi Ekstraksi Fitur.....	42
4.2.4. Implementasi Klasifikasi <i>Support Vector Machine</i> (SVM).....	44
BAB V PENGUJIAN DAN EVALUASI	47
5.1. Lingkungan Pengujian.....	47
5.2. Data Uji Coba.....	47
5.3. Hasil Uji Coba Pada Setiap Tahap Sistem	48
5.3.1. Preprocessing.....	48
5.3.2. Ekstraksi Fitur	50
5.4. Skenario Uji Coba	50
5.4.1. Skenario Uji Coba Parameter <i>Distance</i>	51
5.4.2. Skenario Uji Coba Parameter Gamma	52
5.4.3. Skenario Uji Coba Parameter C	53
5.4.4. Skenario Uji Coba Parameter Kernel SVM.....	53
5.4.5. Skenario Uji Coba Perbandingan Metode	54
5.5. Analisis Hasil Uji Coba.....	55
BAB VI KESIMPULAN DAN SARAN.....	57
6.1. Kesimpulan.....	57
6.2. Saran.....	57

DAFTAR PUSTAKA.....	59
LAMPIRAN.....	61
BIODATA PENULIS.....	65

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Dataset palm vein	7
Gambar 2.2 Proses perbandingan dalam nLBP	8
Gambar 2.3 Contoh Penggunaan Median Filter[8] (a) citra awal; (b) citra hasil median filter	11
Gambar 2.4 Contoh penggunaan <i>adaptive histogram equalization</i> [9] (a) citra awal; (b) citra hasil <i>adaptive histogram equalization</i>	12
Gambar 2.5 Citra awal sebelum dilakukan <i>Adaptive Noise Removal</i> [11]	13
Gambar 2.6 Citra hasil <i>Adaptive Noise Removal</i> [11]	13
Gambar 2.7 Citra awal sebelum dilakukan ADF[12]	14
Gambar 2.8 Citra hasil ADF[12]	14
Gambar 2.9 Citra awal sebelum dilakukan <i>opening</i> [14]	16
Gambar 2.10 Citra hasil morfologi <i>opening</i> [14]	16
Gambar 2.11 Citra awal sebelum dilakukan <i>closing</i> [15]	17
Gambar 2.12 Citra hasil morfologi <i>closing</i> [15]	17
Gambar 2.13 Contoh bidang pemisah (<i>hyperplane</i>)[16]	18
Gambar 2.14 Ilustrasi <i>Multi Class SVM</i>	20
Gambar 3.1 Citra <i>Palm Vein</i>	22
Gambar 3.2 Diagram alir sistem pengenalan <i>palm vein</i>	25
Gambar 3.3 Diagram Alir Akuisisi Citra <i>Palm Vein</i>	26
Gambar 3.4 Diagram alir dari tahap <i>preprocessing</i>	27
Gambar 3.5 Citra hasil proses ROI	28
Gambar 3.6 Diagram Alir ROI Detection[17]	29
Gambar 3.7 Hasil Perbaikan Citra (a)Median Filter (b)Adaptive Histeq (c)Noise Removal	29
Gambar 3.8 Citra hasil penonjolan pembuluh darah (a)ADF (b)Convert (c) <i>Closing</i> (d) <i>Substract</i> (e)Adjust	30
Gambar 3.9 Diagram Alir Proses Ekstraksi Fitur nLBP	32
Gambar 3.10 Diagram Alir Proses Klasifikasi	33
Gambar 5.1 Sampel hasil <i>preprocessing</i> pada citra <i>palm vein</i>	50

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Data Proses	23
Tabel 4.1. Lingkungan implementasi perangkat keras dan lunak.....	35
Tabel 5.1 Lingkungan pengujian perangkat keras dan lunak	47
Tabel 5.2 Persentase Akurasi Fitur nLBP dengan Variasi Nilai <i>Distance</i> pada Ekstraksi Fitur nLBP	51
Tabel 5.3 Persentase Akurasi Fitur nLBP dengan Variasi Nilai Gamma	52
Tabel 5.4 Persentase Akurasi Fitur nLBP dengan Variasi Nilai C	53
Tabel 5.5 Persentase Akurasi Fitur nLBP dengan Variasi Kernel	54
Tabel 5.6 Persentase Akurasi nLBP dan LBP	54

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tahap Akuisisi Citra	36
Kode Sumber 4.2 Implementasi ROI <i>Detection</i>	40
Kode Sumber 4.3 Implementasi Pencarian <i>fpeak</i> pada proses ROI <i>Detection</i>	41
Kode Sumber 4.4 Implementasi perbaikan citra	41
Kode Sumber 4.5 Implementasi penonjolan pembuluh darah telapak tangan	42
Kode Sumber 4.6 Implementasi Ekstraksi Fitur nLBP	44
Kode Sumber 4.7 Implementasi klasifikasi SVM	45

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1. Latar Belakang

Saat ini, informasi sudah menjadi hal yang sangat bernilai dalam aspek kehidupan sehingga proteksi terhadap keamanan informasi menjadi begitu penting. Teknologi otentifikasi seperti pasangan *user ID* dan *password* telah digunakan di berbagai sistem. Namun, teknologi tersebut dirasakan memiliki banyak kekurangan karena sangat rentan untuk dicuri, dilupakan, dan dipalsukan sehingga dibutuhkan teknologi keamanan lain yang lebih aman. Salah satu keamanan yang dianggap lebih aman dan akurat adalah teknologi biometrika. Sistem teknologi biometrik bekerja atas dasar ciri-ciri fisik seseorang. Selama ini sudah ada beberapa penelitian otentifikasi berbasis karakteristik biometrik yang sudah dikembangkan dan diterapkan, diantaranya identifikasi biometrik melalui sidik jari, wajah, iris mata, suara, *finger-print*, dan lain-lain. Namun, metode-metode tersebut masih belum sempurna dalam semua skenario aplikasi. Dengan adanya hal tersebut, maka berkembanglah identifikasi biometrik berdasarkan pola pembuluh darah vena telapak tangan (*palm vein*) yang berada di bawah permukaan kulit[1].

Setiap manusia memiliki pola pembuluh darah telapak tangan yang berbeda, walaupun bagi orang yang kembar identik sekalipun. Pola pembuluh darah setiap individu tidak akan sama dan tidak akan berubah sehingga pasti sulit untuk dipalsukan. Dalam pengenalan pembuluh darah setiap individu, akan dilakukan proses mendapatkan ciri dari pembuluh darah tersebut dengan ekstraksi fitur. Proses ekstraksi fitur ini sangat berpengaruh untuk mendapatkan hasil keakuratan yang bagus.

Terdapat beberapa penelitian sebelumnya yang membahas metode ekstraksi fitur untuk pengenalan *palm vein*. Diantaranya, pada tahun 2011, L. Mirmohamadsadeghi dan A. Drygajlo menawarkan metode ekstraksi fitur tekstur dari citra *palm vein* menggunakan *Local Binary Pattern* (LBP) dengan hasil akurasi 93%. Hasil percobaan menunjukkan bahwa vena yang tidak jelas dapat menjadi masalah dalam pengenalan *palm vein* [2]. Selanjutnya, B. A. Rosdi, C. W. Shing, dan S. A. Suandi melakukan percobaan terhadap pengenalan *finger vein* menggunakan pengembangan dari LBP yang disebut *Local Line Binary Pattern* (LLBP) untuk mengatasi permasalahan vena yang tidak jelas[3]. Penelitian-penelitian sebelumnya juga banyak yang menggunakan pengembangan metode ekstraksi fitur LBP.

Terdapat penelitian lain yang baru dari pengembangan LBP yaitu penelitian oleh Yilmaz Kaya *et al.* yang menawarkan metode *Local Binary Patterns by Neighborhoods* (nLBP) dan *Directional Local Binary Patterns* (dLBP) untuk mendapatkan pendekatan pencarian tekstur baru pada data citra *Brodatz, Butterfly and Kylberg texture*. Hasil akurasi yang didapatkan adalah 99.64% [4]. Karena hasil metode nLBP yang akurat pada dataset tekstur tersebut, maka pada tugas akhir ini mengusulkan penggunaan metode *Local Binary Patterns by Neighborhoods* (nLBP) untuk pengenalan *palm vein*.

Aplikasi pengenalan *palm vein* ini dibagi menjadi 3 proses utama yaitu *preprocessing* yang bertujuan mendapatkan fokus citra pada pembuluh darah telapak tangan dan memperbaiki citra agar terlihat pola pembuluh darah. *Preprocessing* terdiri dari *Region of Interest* (ROI) *Detection*, perbaikan citra, dan penonjolan pembuluh darah telapak tangan. Proses kedua adalah ekstraksi fitur untuk mendapatkan fitur/ciri dari pola pembuluh darah. Proses ekstraksi fitur ini menggunakan metode nLBP. Proses terakhir adalah melakukan pelatihan atau *training* dengan SVM untuk mendapatkan model SVM dan melakukan pengujian atau *testing* untuk pengenalan *palm vein* yang kemudian diukur performanya menggunakan akurasi.

1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur *palm vein* menggunakan metode *Local Binary Patterns by Neighborhoods* (nLBP)?
2. Bagaimana melakukan pengenalan *palm vein* menggunakan metode nLBP dan *Support Vector Machine* (SVM)?
3. Bagaimana mengukur kinerja sistem pengenalan *palm vein*?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain sebagai berikut:

1. Implementasi akan dilakukan menggunakan Matlab dan Python.
2. Dataset yang digunakan untuk percobaan adalah data citra yang terdapat pada database *CASIA Multi-Spectral Palmprint Image Database V1.0* (CASIA-MS-PalmprintV1) untuk *palm vein*.

1.4. Tujuan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini adalah untuk membangun aplikasi pengenalan *palm vein* berdasarkan citra *palm* menggunakan metode ekstraksi fitur *Local Binary Patterns by Neighborhoods* (nLBP).

1.5. Manfaat Tugas Akhir

Manfaat yang diperoleh dari pembuatan tugas akhir ini adalah dapat membantu proses pengenalan *palm vein* dan memberi opsi/metode baru pada otentikasi biometrik dalam meningkatkan sistem keamanan informasi.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir. Selain itu dijabarkan tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Subbab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada tahap ini dilakukan pencarian literatur berupa paper atau jurnal yang bisa digunakan sebagai referensi untuk pengerjaan Tugas Akhir. Literatur yang dipelajari pada pengerjaan Tugas Akhir ini berasal dari jurnal ilmiah yang diambil dari berbagai sumber di internet, beserta berbagai literatur online tambahan terkait *ROI detection*, ekstraksi fitur pada *palm vein* menggunakan metode nLBP, metode klasifikasi *Support Vector Machine* (SVM) pada *palm vein*. Studi literatur yang digunakan sebagai acuan bersumber dari *paper* yang berasal dari jurnal internasional bereputasi yaitu *sciencedirect* dengan judul “*Two novel local binary pattern descriptor for texture analysis*”.

3. Perancangan Perangkat Lunak

Pada tahap ini berisi 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 tahap ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Kemudian dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahap ini yaitu merealisasikan

rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat untuk mengetahui kemampuan algoritma yang digunakan, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul pada aplikasi yang dibuat.

6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7. Sistematika Penulisan

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

Bab II Tinjauan Pustaka

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 penjelasan mengenai desain, perancangan, bahan, dan pemodelan proses yang digunakan dalam

Tugas Akhir ini yang direpresentasikan dengan Kode Sumber.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi. Pembangunan aplikasi menggunakan MATLAB sesuai permasalahan dan batasan yang telah dijabarkan pada Bab I.

Bab V Pengujian dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang berisi kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah atau kode-kode sumber yang penting pada sistem.

BAB II

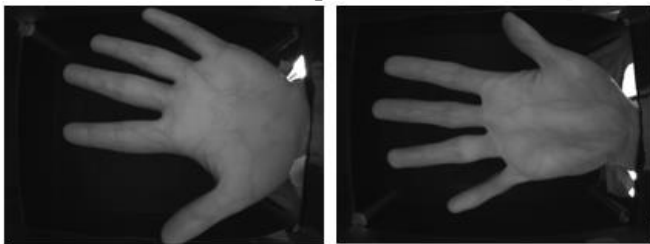
DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai ekstraksi fitur *Local Binary Patterns by Neighborhoods*, *preprocessing* pada *palm vein*, dan klasifikasi SVM.

2.1. Pengenalan *Palm Vein*

Pengenalan *palm vein* merupakan metode otentikasi biometrik yang mengenali pola berdasarkan pola pembuluh darah vena pada telapak tangan manusia. *Palm vein* terletak pada bagian dalam tangan sehingga tidak rentan terhadap luka, keringat, dan lain-lain. Teknik biometrik *palm vein* termasuk metode otentikasi yang aman karena pola pembuluh darah terletak di bawah kulit. Hal ini membuat polanya tidak mungkin disalin atau dipalsukan.

Proses akuisisi *palm vein* menggunakan sensor yang terdiri dari *illumination device* yang menghasilkan sinar inframerah yang akan menyinari telapak tangan dan *capturing device* berupa kamera digital tanpa filter inframerah atau kamera *digital noise reduction* (DNR)[5]. Pada Gambar 2.1 merupakan contoh dataset *palm vein*.



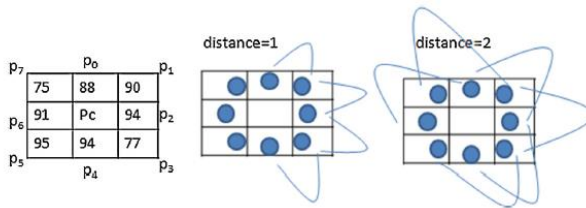
Gambar 2.1 Dataset *palm vein*

2.2. *Local Binary Patterns by Neighborhoods* (nLBP)

nLBP[4] merupakan metode pengembangan dari LBP (*Local Binary Pattern*). Metode ini bergantung pada hubungan 8 tetangga,

yang dinotasikan $P = \{P0, P1, P2, P3, P4, P5, P6, P7\}$, satu sama lain di sekitar piksel. Pada metode nLBP, perbandingan dilakukan di dalam *neighbor* atau tetangganya sendiri yang ditentukan dengan parameter *distance*/jarak, d . Jadi nilai masing-masing piksel dibandingkan dengan piksel tetangga di sebelahnya (sesuai dengan parameter d) dan menghasilkan nilai 1 atau 0, sedangkan proses lain sama seperti pada LBP.

Tahap-tahap metode nLBP adalah menentukan parameter *distance*, d . Karena terdapat 8-neighborhoods, maka range *distance* adalah 1 sampai dengan 7. Kemudian membentuk matriks window 3x3. Setelah itu dilakukan perhitungan nilai *Pixel center* (P_c) dengan cara membandingkan nilai masing-masing piksel dibandingkan dengan piksel tetangga di sebelahnya (sesuai dengan parameter d) dan menghasilkan nilai 1 atau 0. Perhitungan nilai P_c dapat dilihat pada persamaan 2.1 dan 2.2 Gambar 2.2 merupakan gambaran proses pembandingan dalam nLBP dengan parameter d .



Gambar 2.2 Proses pembandingan dalam nLBP

Gambar 2.2 menunjukkan contoh bagaimana nLBP menghasilkan pola biner dari *distance* yang berbeda.

$$P_c = \{S(P0 > P1), S(P1 > P2), S(P2 > P3), S(P3 > P4), S(P4 > P5), S(P5 > P6), S(P6 > P7), S(P7 > P0)\} \quad (2.1)$$

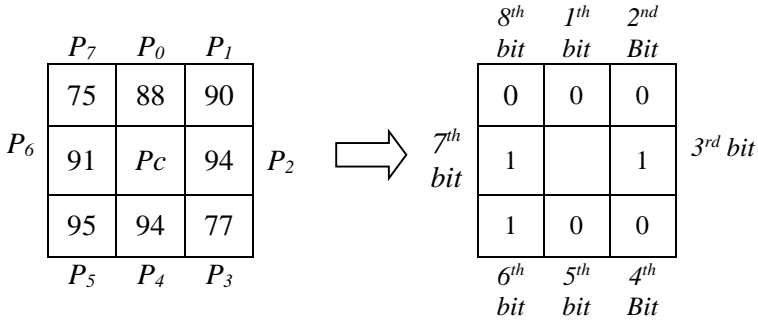
Dimana S menunjukkan perbandingan dan dihitung dengan :

$$S(P_i > P_j) = \begin{cases} 1, & P_i > P_j \\ 0, & P_i \leq P_j \end{cases} \quad (2.2)$$

Kemudian, nilai P_c yang sudah didapatkan dalam bentuk biner, dikonversi nilainya ke decimal.

Sebagai contoh, perhitungan P_c untuk parameter $d=1$.

• Untuk $d=1$



$P_c = \{S(88 > 90), S(90 > 94), S(94 > 77), S(77 > 94), S(94 > 95), S(95 > 91), S(91 > 75), S(75 > 88)\}$

Lalu didapatkan $P_c = \{0, 0, 1, 0, 0, 1, 1, 0\}$.

Nilai decimal dari $(0, 0, 1, 0, 0, 1, 1, 0)_2$ adalah 38.

Jadi nilai P_c adalah 38.

Nilai P_c sangat bergantung pada $distance(d)$. Nilai P_c yang didapatkan itulah yang menjadi nilai fitur.

2.3. Region Of Interest (ROI)

Region of interest (ROI) merupakan bagian dari sebuah gambar yang dibutuhkan untuk identifikasi dengan tujuan tertentu[6]. Konsep dari ROI telah banyak digunakan diberbagai bidang. Berikut adalah contoh ROI berdasarkan dimensinya[7]:

- Dataset 1D : berupa waktu atau interval frekuensi dalam bentuk gelombang.
- Dataset 2D : berupa batas-batas suatu objek pada gambar.
- Dataset 3D : kontur atau permukaan dari sebuah objek volume(atau biasanya dikenal sebagai *Volume of Interest* (VOI)).
- Dataset 4D : outline dari sebuah objek atau interval waktu pada volume waktu.

Macam-macam bentuk ROI 2D yaitu circle, ellipse, rectangle, dan polygonal. Pengambilan ROI dalam tugas akhir ini dilakukan

menggunakan algoritma *inner border tracing* yang akan menghasilkan fokus citra pada pembuluh darah telapak tangan.

2.4. Median Filter

Median filter merupakan teknik peningkatan kualitas citra dalam hal penghilangan *noise*. Metode ini termasuk dalam *non-linear filtering*[8]. Tujuan dari *median filter* ini adalah mereduksi *noise* yang terdapat pada citra dengan *blurring* sehingga akan memperhalus gambar atau gambar akan terlihat lebih *smooth*.

Median filter bekerja dengan mengganti nilai suatu pixel pada citra asal dengan nilai median dari piksel tersebut dan lingkungan tetangganya. Nilai median dicari dengan melakukan pengurutan nilai piksel dari nilai terkecil hingga terbesar dari titik-titik pada matriks yang sudah ditentukan, lalu dicari dan ditentukan nilai tengahnya. Bentuk persamaan dari Median Filter dapat dilihat dalam persamaan 2.3.

$$\text{Median} = \begin{cases} \frac{1}{2}(X_{(n/2)} + X_{(1+n/2)}), & \text{if } n \text{ genap} \\ X_{(n+1/2)}, & \text{if } n \text{ ganjil} \end{cases} \quad (2.3)$$

Dimana X adalah elemen data dan n adalah ukuran himpunan data. Apabila n merupakan bilangan genap, maka nilai Median Filter merupakan setengah dari penjumlahan elemen data pada urutan $n/2$ dan elemen data pada urutan $1 + n/2$. Sedangkan apabila n merupakan bilangan ganjil, maka nilai Median Filter merupakan elemen data pada urutan $n + 1/2$. Gambar 2.3 merupakan contoh penggunaan *median filter*.

2.5. Adaptive Histogram Equalization

Adaptive Histogram Equalization (AHE) merupakan salah satu teknik untuk meningkatkan kontras pada citra. Metode ini cocok untuk memperbaiki kontras lokal dan meningkatkan definisi tepi pada setiap wilayah citra.



Gambar 2.3 Contoh Penggunaan Median Filter[8] (a) citra awal; (b) citra hasil median filter

AHE ini dapat memperbaiki citra dengan kontras lebih terang atau lebih gelap dengan cara mengubah setiap piksel dengan fungsi transformasi.

Variasi dari AHE adalah *Contrast Limited Adaptive Histogram Equalization* (CLAHE). CLAHE beroperasi pada wilayah/*region* kecil dalam gambar, yang disebut *tiles*[9]. *Adaptive histogram equalization* menghitung fungsi pengubahan kontras setiap *tiles* secara individual/satu per satu. Setiap kontras *tiles* akan ditingkatkan sehingga histogram dari wilayah *output* akan sesuai dengan histogram yang ditentukan oleh nilai distribusi. Lalu setiap *neighboring tiles* akan digabung menggunakan interpolasi bilinear untuk menghilangkan batas-batas yang diinduksi secara artifisial. Kontras pada daerah homogen dapat dibatasi untuk menghindari amplifikasi *noise* yang mungkin terdapat pada gambar. Contoh penggunaan *adaptive histogram equalization* dapat dilihat pada gambar 2.4.

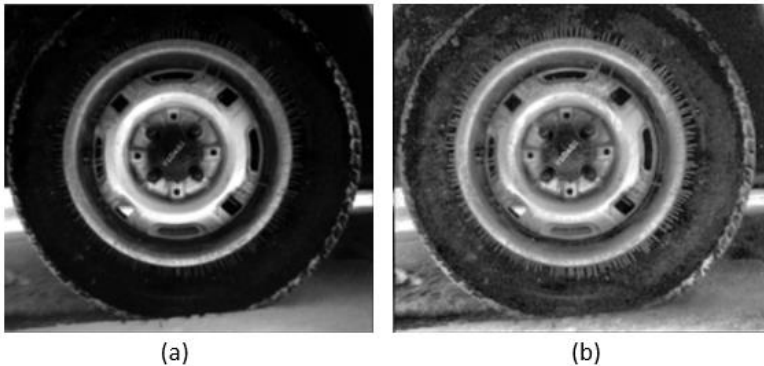
2.6. Adaptive Noise Removal

Adaptive Noise Removal merupakan teknik *filtering* yang digunakan untuk mengurangi *noise* dengan mengganti nilai pixel tetangga dengan nilai baru yang didapat nilai ketetanggaannya. Filter ini bertujuan untuk memperbaiki citra yang awalnya terdapat

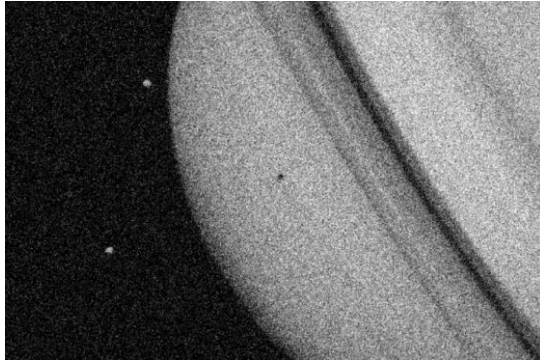
titik-titik *noise* pada citra, lalu *noise* tersebut menjadi berkurang dan citra terlihat lebih jelas. Algoritma dari teknik filter *adaptive noise removal* yaitu[10]:

- Membentuk graf pixel tetangga dengan cara menentukan piksel-piksel tetangga mana saja yang akan mempengaruhi perubahan nilai untuk piksel aktual.
- Memberi bobot nilai berdasarkan perbedaan jarak antar piksel yang difilter dengan piksel-piksel tetangga. Bobot piksel-piksel tetangga dihitung dengan fungsi *Gaussian*.
- Mengganti nilai piksel dengan rata-rata nilai.
- Filter ini akan memilih piksel tetangga yang relevan. Ukuran dan bentuk piksel tetangga akan mempengaruhi perubahan citra.

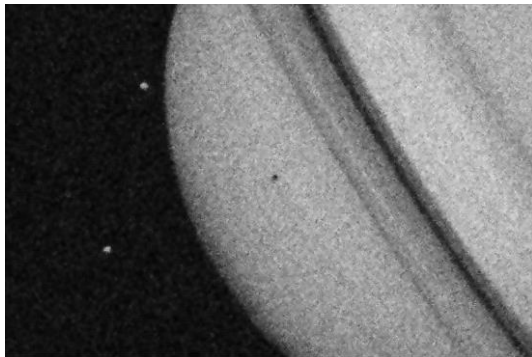
Penggunaan filter *adaptive noise removal* dapat dilihat pada gambar 2.5 dan 2.6.



Gambar 2.4 Contoh penggunaan *adaptive histogram equalization*[9]
(a) citra awal; (b) citra hasil *adaptive histogram equalization*



Gambar 2.5 Citra awal sebelum dilakukan *Adaptive Noise Removal*[11]



Gambar 2.6 Citra hasil *Adaptive Noise Removal*[11]

2.7. Anisotropic Diffusion Filter

Anisotropic Diffusion Filter merupakan teknik untuk mengurangi *noise* pada gambar tanpa menghilangkan bagian penting dari konten gambar, seperti garis, tepi, atau bagian lain yang menginterpretasikan gambar[12]. Contoh penggunaan *anisotropic diffusion filter* dapat dilihat pada gambar 2.7 dan 2.8.



Gambar 2.7 Citra awal sebelum dilakukan ADF[12]



Gambar 2.8 Citra hasil ADF[12]

2.8. Operasi Morfologi

Operasi morfologi memiliki peranan yang penting dalam pengolahan citra digital. Prinsip dari operasi morfologi adalah mengekstrak komponen-komponen citra yang berguna dalam representasi dan deskripsi bentuk seperti ekstrak boundary dari suatu region.

Pada metode operasi morfologi sering digunakan *structuring element*[13]. *Structuring element* merupakan kumpulan pola spesifik berupa koordinat angka diskrit. Representasi koordinat

angka yang biasa digunakan adalah koordinat cartesian. Terdapat dua operasi morfologi dasar yaitu dilasi dan erosi. Operasi morfologi dilasi dan rotasi juga dapat dikombinasikan membentuk operasi morfologi *opening* dan *closing*. Operasi morfologi dikenakan pada objek yang terdapat pada citra biner. Objek citra direpresentasikan dengan nilai biner 1 sedangkan *background* bernilai biner 0.

2.8.1. Dilasi

Dilasi merupakan operasi morfologi yang membuat sebuah objek menjadi meluas atau menebal sesuai dengan bentuk [13] dan ukuran dari *structuring element* yang digunakan. Jika suatu citra X dikenai *structuring element* Y dengan β sebagai anggota dari Y maka notasi untuk dilasi ditunjukkan pada persamaan 2.4.

$$X \oplus Y = \bigcup_{\beta \in Y} X + \beta \quad (2.4)$$

Pada operasi dilasi, setiap piksel citra X yang berada tepat pada pusat piksel *structuring element* Y maka tetangga dari piksel tersebut mengikuti jumlah dan pola dari ketetanggaan pusat piksel *structuring element* Y .

2.8.2. Erosi

Erosi merupakan operasi morfologi yang membuat sebuah objek menyusut atau menipis [13]. Objek akan menyusut atau menipis sesuai dengan bentuk dan ukuran dari *structuring element*. Jika suatu citra X dikenai *structuring element* Y dengan β sebagai anggota dari Y maka notasi untuk dilasi ditunjukkan pada persamaan 2.5.

$$X \ominus Y = \bigcap_{\beta \in Y} X - \beta \quad (2.5)$$

Pada operasi erosi, setiap piksel citra X yang berada tepat pada pusat piksel *structuring element* Y maka tetangga dari piksel tersebut dihapus sesuai jumlah dan pola dari ketetanggaan pusat piksel *structuring element* Y .

2.8.3. *Opening*

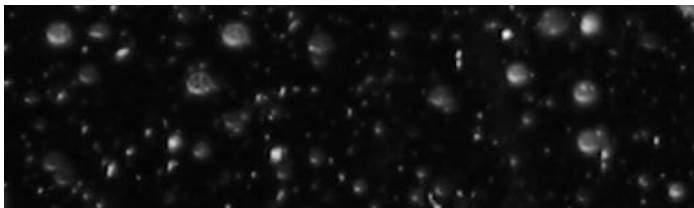
Operasi morfologi *opening* umumnya digunakan untuk memperhalus kontur dari sebuah objek, memutuskan garis tipis yang menghubungkan dua region besar dan menghilangkan tonjolan/objek kecil. Pada operasi morfologi *opening* dilakukan erosi terlebih dahulu, lalu dilakukan dilasi[14].

Contoh penggunaan operasi *opening* dengan *structuring element* ditunjukkan pada Gambar 2.9 dan 2.10.

2.8.4. *Closing*

Operasi morfologi *closing* digunakan untuk menghaluskan bagian dari kontur dari sebuah objek dan menyatukan dua objek yang terputus. *Closing* juga digunakan untuk menghilangkan lubang kecil pada objek dengan mengisi celah yang terdapat dalam kontur. Pada operasi morfologi *closing* dilakukan dilasi terlebih dahulu, lalu dilakukan erosi[15].

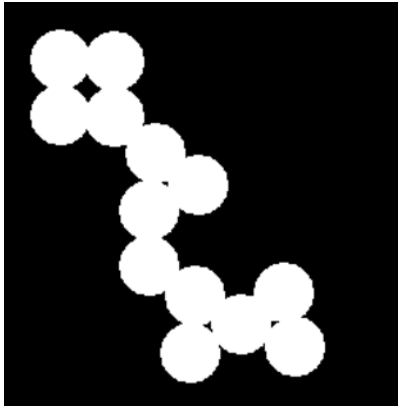
Contoh penggunaan operasi *closing* ditunjukkan seperti pada Gambar 2.11 dan 2.12.



Gambar 2.9 Citra awal sebelum dilakukan *opening*[14]



Gambar 2.10 Citra hasil morfologi *opening*[14]



Gambar 2.11 Citra awal sebelum dilakukan *closing*[15]

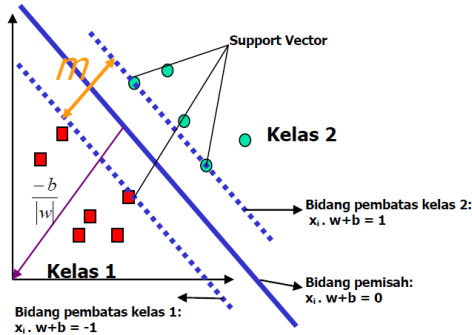


Gambar 2.12 Citra hasil morfologi *closing*[15]

2.9. Klasifikasi *Support Vector Machine* (SVM)

Support Vector Machine (SVM)[16] merupakan teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM merupakan metode yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* (bidang pemisah) klasifier yang optimal yang dapat memisahkan dua set data dari dua kelas berbeda. Bidang pemisah atau *hyperplane* terbaik tidak hanya dapat memisahkan data tetapi juga

memiliki margin paling besar. *Hyperplane* pemisah terbaik antara dua kelas dapat ditemukan dengan mengukur *margin* dari *hyperplane* tersebut. *Margin* adalah jarak antara *hyperplane* dengan pola (*support vector*) terdekat dari masing-masing kelas. Contoh bidang pemisah (*hyperplane*) yang dapat memisahkan dua buah kelas dapat dilihat pada Gambar 2.13.



Gambar 2.13 Contoh bidang pemisah (*hyperplane*)[16]

SVM memiliki prinsip dasar *linier classifier* namun SVM telah dikembangkan agar dapat bekerja pada problem *non-linear* dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi. Menurut Satosa (2007) *hyperplane* klasifikasi linier SVM dinotasikan:

$$f(x) = \mathbf{w}^T \mathbf{x} + b \quad (2.6)$$

sehingga menurut Vapnik dan Cortes (1995) diperoleh persamaan berikut

$$[(\mathbf{w}^T \cdot \mathbf{x}_i) + b] \geq 1 \text{ untuk } y_i = +1 \quad (2.7)$$

$$[(\mathbf{w}^T \cdot \mathbf{x}_i) + b] \leq -1 \text{ untuk } y_i = -1$$

Dengan, \mathbf{w}^T adalah normal bidang, b adalah posisi bidang relatif terhadap pusat koordinat, \mathbf{x}_i adalah himpunan data training, $i = 1, 2, \dots, n$ dan $y_i =$ label kelas dari \mathbf{x}_i .

Dalam mencari *hyperplane* terbaik dapat digunakan metode *Quadratic Programming* (QP) Problem. Solusi untuk

mengoptimasi penemuan Vapnik (1995) yaitu menggunakan fungsi *Lagrange* sebagai berikut:

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^n \alpha_i \{y_i [(w^T \cdot x_i) + b] - 1\} \quad (2.8)$$

Dengan α_i = pengganda fungsi *Lagrange* dan $i=1,2,\dots,n$

Lalu terdapat penyempurnaan-penyempurnaan lain hingga didapatkan persamaan *dual problem* L_d berikut oleh Hestie *et al*(2001).

$$L_d = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.9)$$

Jadi persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan 2.10.

$$\max_{\alpha} L_d = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.10)$$

Dengan batasan, $\alpha_i \geq 0$, $i=1,2,\dots,n$ dan $\sum_{i=1}^n \alpha_i y_i = 0$

Dengan demikian, diperoleh nilai α_i yang akan digunakan untuk menemukan w . Data yang memiliki nilai $\alpha_i > 0$ adalah *support vector* sedangkan sisanya memiliki nilai $\alpha_i = 0$. Maka fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Setelah solusi permasalahan *quadratic programming* ditemukan (nilai α_i), maka kelas dari data pengujian x dapat ditentukan berdasarkan nilai dari fungsi keputusan pada persamaan 2.11.

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i x_i \cdot x_d + b \quad (2.11)$$

x_i adalah *support vector*, ns adalah jumlah *support vector*, dan x_d adalah data yang akan diklasifikasikan.

Selain itu, SVM dapat bekerja pada data *non-linier* dengan menggunakan pendekatan kernel. Fungsi kernel yaitu untuk

memetakan dimensi awal (dimensi yang lebih rendah) himpunan data ke dimensi baru (dimensi yang relatif lebih tinggi). Macam-macam fungsi kernel diantaranya:

1. Kernel Linear

$$K(x_i, x_j) = x_i^T x_j \quad (2.12)$$

2. Kernel Gaussian Radial Basic Function (RBF)

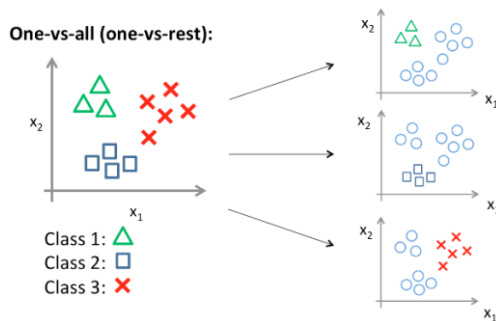
$$K(x_i, x_j) = (\gamma \cdot x_i^T x_j + r)^d, \gamma > 0 \quad (2.13)$$

3. Kernel Polynomial

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (2.14)$$

x_i dan x_j adalah pasangan dua data training. Parameter γ , c , $d > 0$ merupakan konstanta. Fungsi kernel yang harus digunakan untuk substitusi *dot product* di *feature space* bergantung pada data karena fungsi kernel ini akan menentukan fitur baru yang akan dicari *hyperplane*-nya.

Seiring berjalannya waktu, pengembangan SVM dilakukan sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas. Metode ini disebut *Multi Class SVM*. Proses pemodelan pada *Multi Class SVM* membutuhkan data pelatihan yang lebih banyak dibandingkan dengan dua kelas. Tahapan metode *Multi Class* ini yaitu membuat model atau melakukan pemisahan setiap kelas dengan kelas lain yang dilakukan sebanyak jumlah kelas. Kemudian untuk memprediksi kelas dari data masukan, maka dibandingkan terhadap setiap model yang sudah dibuat. Ilustrasi *Multi Class SVM* dapat dilihat pada Gambar 2.14.



Gambar 2.14 Ilustrasi *Multi Class SVM*

BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini dijelaskan mengenai rancangan sistem perangkat lunak yang akan diimplementasikan. Perancangan yang dijelaskan meliputi perancangan sistem, data dan proses. Data yang dimaksud adalah data yang akan diolah dalam perangkat lunak baik sebagai pembelajaran maupun pengujian sehingga tujuan Tugas Akhir ini bisa tercapai. Proses adalah tahap-tahap yang ada dalam sistem sebagai pengolah data meliputi *preprocessing* dan ekstraksi fitur *Local Binary Pattern by Neighborhoods* (nLBP).

3.1. Data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga akan menghasilkan data keluaran yang diharapkan.

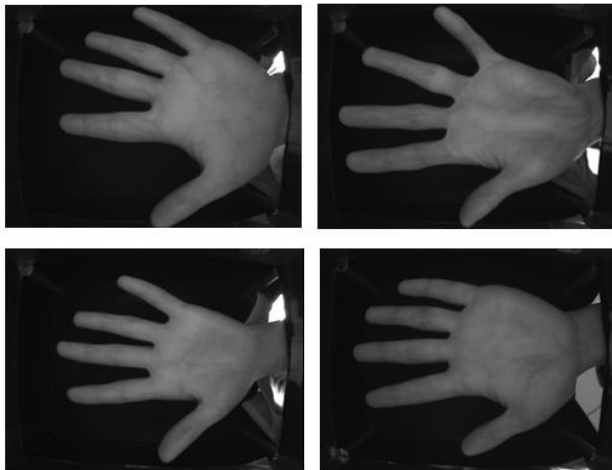
Data yang digunakan pada proses pengenalan *palm vein* ini dibagi menjadi tiga macam yaitu data masukan, data proses, dan data keluaran. Data masukan adalah data input dari pengguna perangkat lunak. Data proses adalah data yang digunakan dan dihasilkan ketika tahap-tahap pengenalan *palm vein* dilakukan. Data keluaran adalah data hasil dari proses pengenalan *palm vein*. Penjelasan masing-masing data sebagai berikut.

3.1.1. Data Masukan

Data masukan adalah data yang digunakan sebagai masukan dalam pengoperasian sistem. Data yang digunakan yaitu citra *palm vein* yang bersumber dari *database CASIA Multi-Spectral Palmprint Image Database V1.0* (CASIA-MS-PalmprintV1). Database ini berisi 600 citra *palm vein* yang diambil dari 100 orang, masing-masing orang memiliki enam citra. Jadi pada database ini terdiri dari 100 kelas, setiap kelas terdapat 6 citra. Sehingga total citra yaitu 600 citra *palm vein*.

Tiap kelas pada database *palm vein* diambil dalam dua sesi dengan interval lebih dari satu bulan. Masing-masing sesi terdiri dari tiga citra. Citra *palm vein* yang digunakan pada Tugas Akhir ini adalah citra *palm vein* pada tangan kiri.

Contoh format nama file citra *palm vein* pada database CASIA adalah 001_1_940_01.jpg. Pada format file tersebut, 001 menunjukkan *palm vein* kelas orang pertama, huruf l menunjukkan bahwa itu adalah telapak tangan sebelah kiri (*left*). Angka 940 menunjukkan electromagnetic spectrum. Angka 01 menunjukkan citra diambil pada sesi pertama. Contoh citra *palm vein* dari database CASIA ditunjukkan pada Gambar 3.1



Gambar 3.1 Citra *Palm Vein*

Database ini akan dibagi menjadi dua, yaitu data latih dan data uji. Data latih digunakan untuk melatih sistem agar keluarannya sesuai dengan yang diharapkan. Sedangkan data uji digunakan untuk menguji sistem setelah dilakukan proses pelatihan atau *training*. Jumlah data latih dan data uji akan dibagi dengan perbandingan yang sama antar kelasnya.

3.1.2. Data Proses

Data proses adalah data yang digunakan selama proses berjalan. Data proses ditunjukkan dalam Tabel 3.1.

Tabel 3.1 Data Proses

No.	Nama data	Keterangan
1.	Data citra <i>palm vein</i>	Data dari database yang akan digunakan pada tahap <i>preprocessing</i> untuk mendapatkan ROI. Data ini bertipe jpg.
2.	Data citra hasil <i>preprocessing</i>	Data yang dihasilkan dari <i>preprocessing</i> . Data ini digunakan sebagai masukan dari tahap ekstraksi fitur. Data ini bertipe .jpg
3.	Data vektor fitur nLBP	Data yang dihasilkan dari tahap ekstraksi fitur.

Pemrosesan akan dilakukan dengan menggunakan data masukan yang berupa data citra *palm vein* bertipe jpg. Data citra ini akan diolah hingga menghasilkan nilai-nilai fitur yang akan digunakan sebagai data latih dan uji dalam proses klasifikasi. Data latih digunakan sebagai data belajar yang akan membentuk model pada tahap pelatihan, sedangkan data uji digunakan sebagai masukan untuk pencocokan atau uji coba. Data latih dan data uji nantinya akan diproses hingga menjadi vektor fitur *palm vein*. Data vektor fitur ini akan disimpan dalam file berekstensi .txt.

3.1.3. Data Keluaran

Data keluaran dari sistem ini adalah data hasil dari proses-proses yang sudah dilakukan. Data hasil dari ekstraksi fitur akan diproses menggunakan klasifikasi *Support Vector Machine*. Hasil dari proses klasifikasi adalah prediksi kelas pada masing-masing data uji dan nilai evaluasinya.

3.2. Desain Sistem Secara Umum

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi pengenalan *palm vein*. Untuk dapat menggunakan aplikasi pengenalan *palm vein*, pengguna harus memiliki data masukan.

Pada aplikasi pengenalan *palm vein*, terdapat tiga tahapan utama yaitu *preprocessing*, ekstraksi fitur, dan klasifikasi.

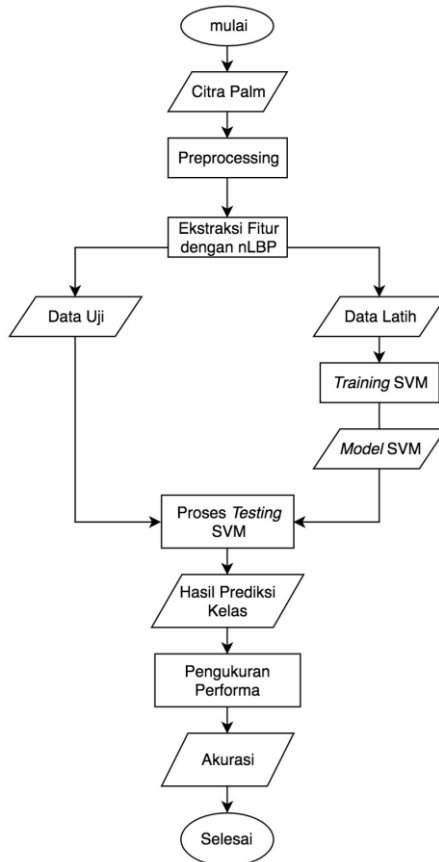
Tahapan *preprocessing* merupakan tahap awal yang dilakukan untuk perbaikan citra dan pengambilan fokus bagian pembuluh darah citra. Data masukan dalam tahap ini yaitu citra *palm vein*. Tahap ini terdiri dari proses ROI, median filter, adaptive histogram equalization, adaptive noise removal, anisotropic diffusion filter, image closing, subtract image, dan adjusting image. Hasil tahap ini yaitu fokus citra pada pembuluh darah telapak tangan.

Selanjutnya hasil dari proses *preprocessing* akan menjadi data masukan pada tahap ekstraksi fitur. Tahap ekstraksi fitur dilakukan untuk mendapatkan nilai representatif dari setiap piksel yang telah melalui proses *preprocessing*. Pada tahap ini digunakan metode ekstraksi fitur *Local Binary Patterns by Neighborhood*. Hasil dari ekstraksi fitur tersebut akan digunakan pada metode klasifikasi *Support Vector Machine*. Hasil akhir dari klasifikasi adalah nilai evaluasi yang menjadi tolok ukur keakuratan proses klasifikasi. Diagram alir dari sistem ditunjukkan pada Gambar 3.2.

3.3. Akuisisi Citra *Palm Vein*

Seperti yang telah dijelaskan pada bagian perancangan data, citra *palm vein* akan digunakan sebagai masukan dalam proses pengenalan *palm vein*. Citra *palm vein* diasumsikan telah disimpan di *disk* komputer. Dalam tugas akhir ini, citra *palm vein* yang digunakan berasal dari *database* CASIA versi 1. Diagram alir proses akuisisi citra *palm vein* dapat dilihat pada Gambar 3.3. Tahap awal dari proses akuisisi citra ini adalah memilih sebuah citra *palm vein* dari *disk*. Selanjutnya citra tersebut akan dipanggil

ke dalam sistem. Terakhir, citra *palm vein* yang terpilih akan ditampilkan di dalam sistem.

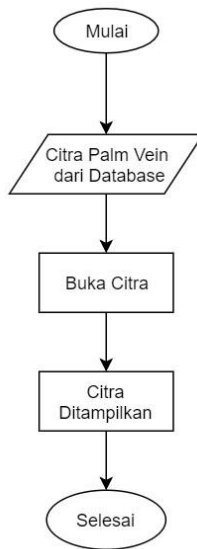


Gambar 3.2 Diagram alir sistem pengenalan *palm vein*

3.4. Perancangan *Preprocessing*

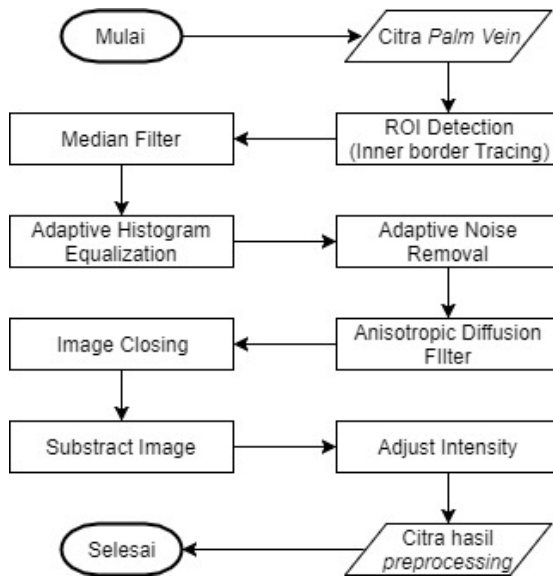
Tahap awal yang dilakukan setelah akuisisi citra adalah mempersiapkan data terlebih dahulu dengan *preprocessing*. Tahap ini bertujuan untuk memperbaiki citra dan mengambil fokus bagian pembuluh darah citra. *Preprocessing* dilakukan melalui beberapa tahap yaitu proses pengambilan ROI, perbaikan citra, dan

penonjolan pembuluh darah telapak tangan. Diagram alir dari proses *preprocessing* ditunjukkan pada Gambar 3.4.



Gambar 3.3 Diagram Alir Akuisisi Citra *Palm Vein*

Berdasarkan diagram alir pada Gambar 3.4, tahap awal dari *preprocessing* adalah pengambilan ROI menjadi bagian utama yaitu bagian terluas dari telapak tangan menggunakan algoritma *inner border tracing*. Setelah itu, dilakukan perbaikan citra menggunakan *median filter*, *adaptive histogram equalization*, dan *adaptive noise removal*. Untuk proses penonjolan pembuluh darah telapak tangan dilakukan *anisotropic diffusion filter*, morfologi *closing*, *subtract image*, dan *adjust intensity*. Hasil dari *preprocessing* ini akan menjadi data masukan untuk proses selanjutnya. Masing-masing tahap akan dijelaskan lebih lanjut pada subbab ini.



Gambar 3.4 Diagram alir dari tahap *preprocessing*

3.4.1. ROI Detection

Pengambilan ROI dilakukan menggunakan algoritma *inner border tracing* untuk mendapatkan bagian utama dari telapak tangan. Diagram alir proses pengambilan ROI dari citra *palm vein* ditunjukkan pada Gambar 3.6.

Algoritma *inner border tracing* menggunakan 8-connectivity yang berarti akan mencari piksel tetangga dari 8 arah[17]. Inisialisasi untuk mencari posisi piksel tetangga pada 8-connectivity adalah 7. Untuk mendapatkan nilai arah dari piksel tetangga, apabila nilai dari arah pencarian sebelumnya adalah bilangan genap maka menggunakan rumus $(\text{posisi piksel} + 7) \bmod 8$. Sedangkan apabila nilai dari pencarian sebelumnya adalah bilangan ganjil, maka menggunakan rumus $(\text{posisi piksel} + 6) \bmod 8$. Lalu, untuk mencari nilai arah baru hanya perlu mengikuti arah berlawanan dengan jarum jam. Setelah mencari koordinat piksel, dilanjutkan menghitung jarak setiap pixel border menuju piksel bagian tengah bawah citra menggunakan diagram distribusi jarak.

Dari sini, list dari border yang dibentuk akan disimpan. List piksel inilah yang nantinya akan digunakan dalam pengambilan dua titik pada ROI yaitu titik P1 dan P2. P1 adalah lembah yang berada diantara jari kelingking dan jari manis, dan P2 adalah titik tembah yang berada diantara jari tengah dan jari telunjuk. ROI akan didapatkan dari titik P1 dan P2 yang berbentuk persegi.

Hasil pengambilan ROI ditunjukkan pada Gambar 3.5.



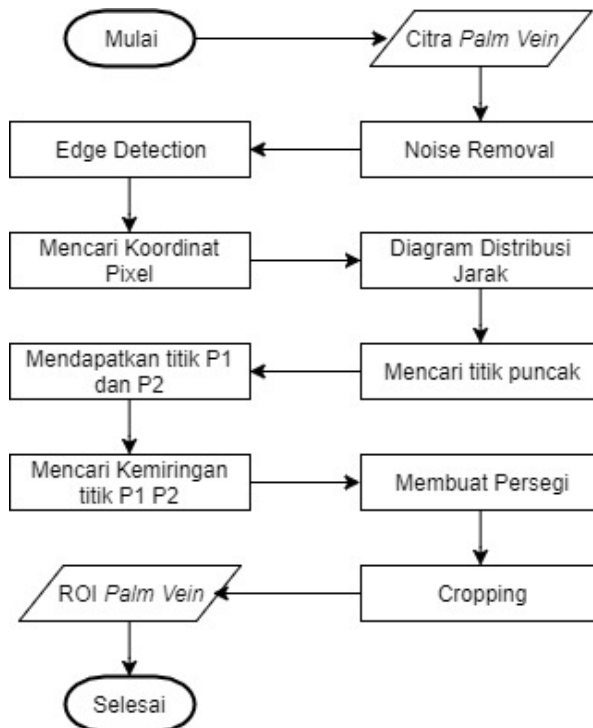
Gambar 3.5 Citra hasil proses ROI

3.4.2. Perbaikan Citra

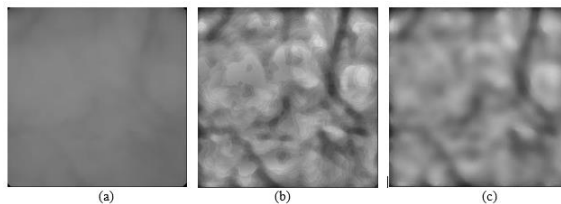
Setelah mendapatkan ROI *palm vein*, selanjutnya adalah melakukan perbaikan citra agar dapat diolah dan digunakan pada tahap selanjutnya. Tahap perbaikan citra ini seperti penghilangan noise dan enhancement menggunakan median filter, adaptive histogram equalization, dan adaptive noise removal. Hasil perbaikan citra ditunjukkan pada Gambar 3.7.

3.4.3. Penonjolan Pembuluh Darah Telapak Tangan

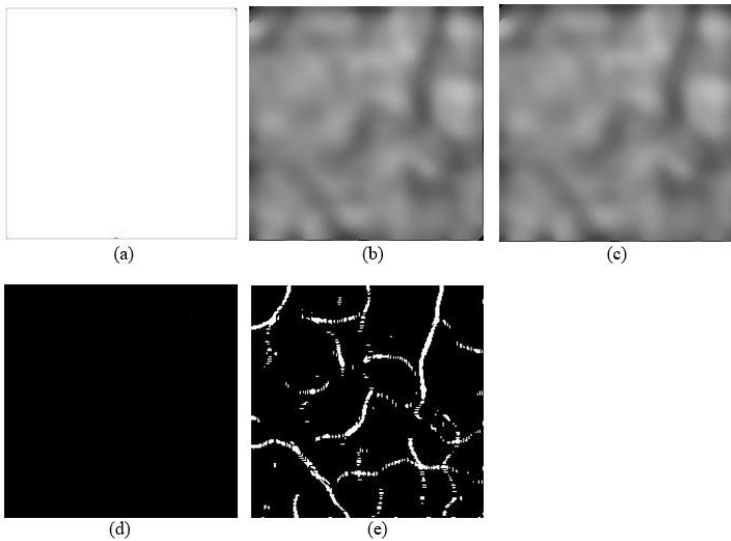
Setelah citra diperbaiki, selanjutnya adalah proses penonjolan pembuluh darah telapak tangan. Pada tahap ini dilakukan pemisahan objek pembuluh darah dengan telapak tangan yaitu dengan *anisotropic diffusion filter*, morfologi closing, serta subtract image hasil dari *anisotropic diffusion filter* dengan citra hasil morfologi *closing*. Hasil penonjolan pembuluh darah telapak tangan ditunjukkan pada Gambar 3.8.



Gambar 3.6 Diagram Alir ROI Detection[17]



Gambar 3.7 Hasil Perbaikan Citra
(a)Median Filter (b)Adaptive Histeq (c)Noise Removal



Gambar 3.8 Citra hasil penonjolan pembuluh darah (a)ADF (b)Convert (c)Closing (d)Subtract (e)Adjust

3.5. Perancangan Ekstraksi Fitur

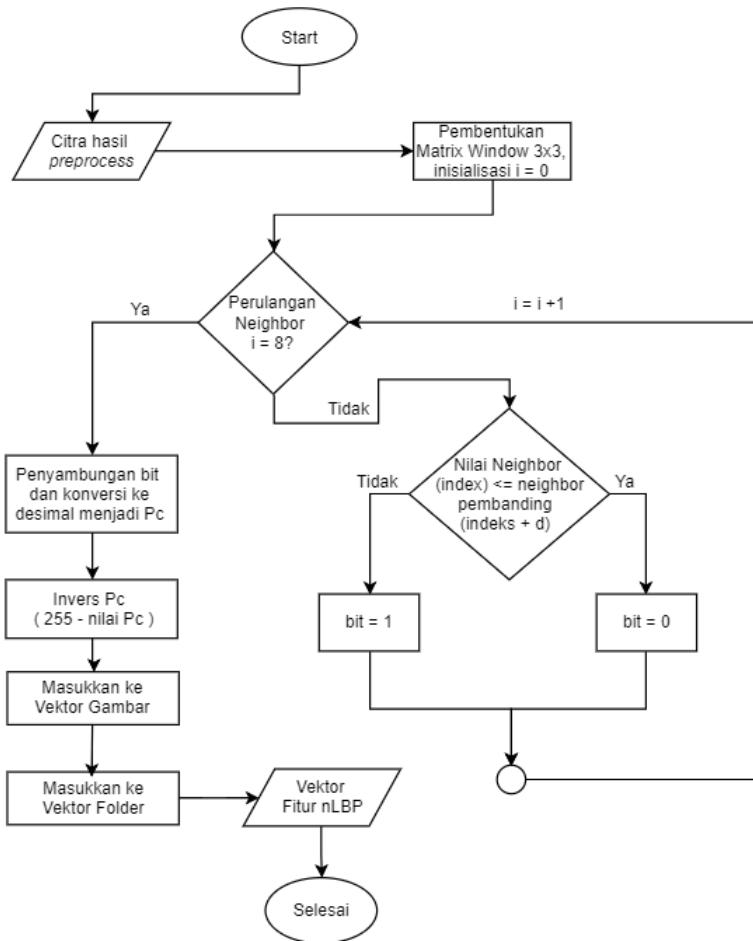
Ekstraksi fitur *palm vein* merupakan tahap untuk mendapatkan nilai representatif dari setiap piksel yang telah melalui proses *preprocessing*. Data masukan tahap ini adalah hasil dari *preprocessing*. Metode ekstraksi fitur yang digunakan yaitu *Local Binary Patterns by Neighborhoods* (nLBP). Sebelum masuk ke tahap ekstraksi fitur, dilakukan *thresholding* terlebih dahulu. Pada Tugas Akhir ini menggunakan *threshold* 60 yang berarti titik yang bernilai lebih dari 60 akan menjadi 255 (putih) dan bernilai kurang dari 60 akan menjadi 0 (hitam). Setelah itu, masuk ke proses nLBP.

Pada awal proses nLBP, *distance* atau jarak antar piksel tetangga ditentukan terlebih dahulu. Karena metode ini menggunakan ukuran matriks window 3x3, maka terdapat 8 buah *neighbor*. Setiap *neighbor* mempunyai tujuh *neighbor* lain yang dapat digunakan untuk menghitung nilai nLBP. Maka range

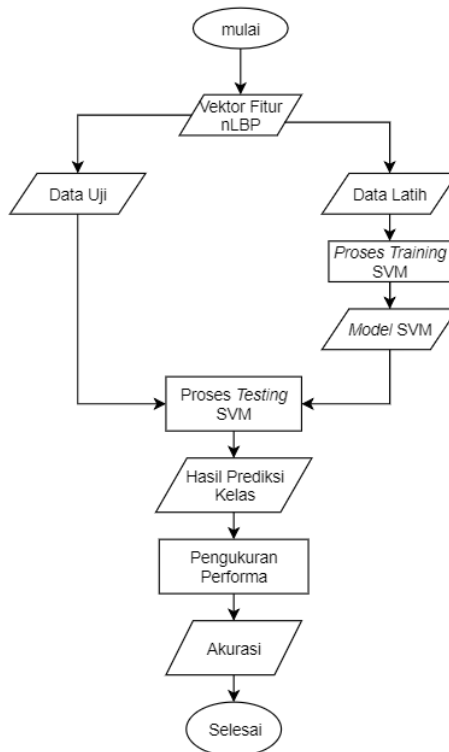
distance adalah 1 sampai 7. Setelah itu, membuat window matriks 3x3. Lalu melakukan perhitungan nilai *Pixel center* (P_c) dengan membandingkan nilai setiap piksel tetangganya dengan piksel tetangga lainnya (sesuai *distance*) dan mengambil nilai 1 atau 0. Setelah itu akan didapatkan nilai P_c dalam bentuk biner. Selanjutnya dilakukan konversi nilai ke bentuk decimal. Lalu nilai inilah yang disimpan menjadi nilai P_c . Kemudian dilakukan invers pada semua nilai P_c dengan cara 255 dikurangi nilai P_c . Nilai hasil invers P_c diubah ke bentuk vektor satu baris lalu menjadi vektor gambar. Selanjutnya vektor-vektor gambar dimasukkan ke vektor folder agar hasil vektor fitur semua gambar menjadi 1 file. Setelah itu hasil vektor fitur semua gambar disimpan dalam sebuah file berekstensi .txt. Hasil inilah yang akan digunakan untuk tahap klasifikasi. Diagram alir ekstraksi fitur nLBP ditunjukkan pada Gambar 3.9.

3.6. Perancangan Klasifikasi

Klasifikasi dilakukan dengan mengolah data hasil dari proses ekstraksi fitur. Data masukan dari klasifikasi ini adalah hasil ekstraksi fitur dalam bentuk vektor fitur yang disimpan dalam file berekstensi .txt. Untuk setiap data latih dan uji dijadikan vektor fitur. Kemudian dilakukan tahap verifikasi menggunakan metode klasifikasi *Support Vector Machine* (SVM). Klasifikasi SVM dilakukan dengan menentukan kernel dan parameter-parameternya. Hasil akhir dari proses ini adalah prediksi kelas dari hasil tahap klasifikasi. Diagram alir dari tahap klasifikasi ini ditunjukkan pada Gambar 3.10.



Gambar 3.9 Diagram Alir Proses Ekstraksi Fitur nLBP



Gambar 3.10 Diagram Alir Proses Klasifikasi

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari yang dilakukan berdasarkan perancangan sistem yang telah dibahas pada bab sebelumnya. Implementasi ini menjelaskan mengenai proses pengenalan *palm vein* dengan menampilkan kode sumber yang digunakan.

4.1. Lingkungan Implementasi

Objek citra yang akan diolah pada implementasi tugas akhir ini adalah citra *palm vein* bertipe jpg. Tabel 4.1 menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan.

Tabel 4.1. Lingkungan implementasi perangkat keras dan lunak

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel ® Core(TM) i7-7500U CPU @ 2.70GHz (4 CPUs), ~2.9GHz
Perangkat lunak	Sistem Operasi : Windows 10 Pro 64-bit Perangkat Pengembang: Matlab R2017b, Python Spyder Perangkat Pembantu: Sublime Text, draw.io

4.2. Implementasi

Pada subbab ini menjelaskan implementasi proses yang sudah dijelaskan pada Bab 3. Pada bagian implementasi ini akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam Tugas Akhir ini dan disertai dengan kode sumber.

4.2.1. Implementasi Akuisisi Citra *Palm Vein*

Tahap awal yang dilakukan pada proses pengenalan *palm vein* adalah akuisisi citra. Akuisisi citra adalah proses untuk menyiapkan dan mengambil data citra dari *disk*. Berikut adalah

implementasi akuisisi citra *palm vein* yang ditunjukkan oleh Kode Sumber 4.1.

1	<code>datacitra = dir('path');</code>
2	<code>for i=1:length(datacitra)</code>
3	<code> tmp = strcat('path',datacitra(i).name);</code>
4	<code> img_input = imread(tmp);</code>
5	<code> %code ekstraksi fitur</code>
6	<code>End</code>

Kode Sumber 4.1 Implementasi Tahap Akuisisi Citra

Baris ke- 1 adalah menyimpan dataset citra yang berada dalam folder sesuai direktori pada variabel `datacitra`. Baris ke- 2 adalah perulangan sebanyak citra yang ada pada folder. Baris ke- 3, menyimpan data pada variabel `tmp`. Baris ke- 4 adalah membaca citra *palm vein*.

4.2.2. Implementasi *Preprocessing*

Setelah citra *palm vein* dibaca di dalam sistem, maka selanjutnya adalah melakukan tahap *preprocessing*. Tujuan dari tahap ini yaitu memperbaiki citra dan mengambil bagian pembuluh darah citra. Tahap *preprocessing* yang dilakukan yaitu *ROI detection*, perbaikan citra, dan penonjolan pola pembuluh darah telapak tangan. Penjelasan dari masing-masing tahapan adalah sebagai berikut.

4.2.2.1. Implementasi *ROI Detection*

Pada tahap ini akan dilakukan pengambilan bagian telapak tangan dengan algoritma *inner border tracing*[17]. Implementasi *ROI Detection* ditunjukkan pada Kode Sumber 4.2. Pertama-tama dilakukan proses *opening* pada citra untuk menghilangkan *noise* pada citra. Proses *opening* dapat dilihat pada baris ke- 1 dan 2. Lalu entukan nilai koordinat piksel objek yang pertama ditemukan pada baris ke- 3. Pada baris ke- 5 sampai 10 digunakan untuk mendapatkan nilai arah dari piksel tetangga. Apabila nilai dari arah pencarian sebelumnya adalah bilangan genap maka menggunakan

rumus $(\text{posisi pixel} + 7) \bmod 8$. Sedangkan apabila nilai dari pencarian sebelumnya adalah bilangan ganjil, maka menggunakan rumus $(\text{posisi pixel} + 6) \bmod 8$. Pada baris ke- 11 sampai 65 berfungsi untuk menentukan posisi dari piksel tetangga berdasarkan nilai *dir* yang didapat dan pencarian nilai arah menggunakan *8-connectivity* serta arah yang berlawanan dengan jarum jam.

Pada baris ke- 66 sampai 72 berfungsi untuk mengatur bentuk diagram distribusi jarak. Koordinat petama pada diagram adalah koordinat piksel yang memiliki jarak terkecil. Dari diagram distribusi jarak inilah akan diambil titik P_1 dan P_2 . Titik P_1 adalah titik lembah yang berada diantara jari manis danelingking, sedangkan titik P_2 adalah titik lembah yang berada diantara jari telunjuk dan jari tengah. Pada baris ke- 73 dan 74 digunakan untuk menghilangkan *noise* dengan cara melakukan *thresholding* terhadap piksel yang mempunyai posisi sumbu y diatas 550 maka memiliki jarak bernilai 0. Pada baris ke- 75 digunakan untuk memanggil fungsi *fpeak*. Fungsi *fpeak* ditunjukkan pada Kode Sumber 4.3.

Pada baris ke- 76 sampai 82 berfungsi untuk menghapus duplikasi nilai puncak. Pada baris ke- 83 sampai 91 digunakan untuk mendapatkan titik P_1 dan P_2 . Pada baris ke- 92 sampai 94 digunakan untuk mendapatkan titik P_3 dan P_4 . Pada baris ke- 95 digunakan untuk melakukan *cropping* dari titik yang telah ditentukan menjadi persegi. Pada baris terakhir digunakan untuk mengubah ukuran citra hasil *cropping* menjadi 150x150.

1	<code>se = strel('disk',17);</code>
2	<code>imgbw = imopen(imgbw_in,se);</code>
3	<code>startCoor = [r,c];</code>
4	<code>while indikator == 1</code>
5	<code> if mod(dir,2)==0</code>
6	<code> startDir = mod(dir+7,8);</code>
7	<code> Else</code>
8	<code> startDir = mod(dir+6,8);</code>
9	<code> End</code>
10	<code> currentDir = startDir;</code>

11	<code>for i=1:8</code>
12	<code> if currentDir==8</code>
13	<code> currentDir = 0;</code>
14	<code> End</code>
15	<code> switch (currentDir)</code>
16	<code> case 0</code>
17	<code> posisi = [0,1];</code>
18	<code> dir = 0;</code>
19	<code> case 1</code>
20	<code> posisi = [-1,1];</code>
21	<code> dir = 1;</code>
22	<code> case 2</code>
23	<code> posisi = [-1,0];</code>
24	<code> dir = 2;</code>
25	<code> case 3</code>
26	<code> posisi = [-1,-1];</code>
27	<code> dir = 3;</code>
28	<code> case 4</code>
29	<code> posisi = [0,-1];</code>
30	<code> dir = 4;</code>
31	<code> case 5</code>
32	<code> posisi = [1,-1];</code>
33	<code> dir = 5;</code>
34	<code> case 6</code>
35	<code> posisi = [1,0];</code>
36	<code> dir = 6;</code>
37	<code> case 7</code>
38	<code> posisi = [1,1];</code>
39	<code> dir = 7;</code>
40	<code> End</code>
41	<code> newCoor = coor + posisi;</code>
	<code> if newCoor(1,1) > 0 &&</code>
42	<code> newCoor(1,2) > 0 && newCoor(1,1) <</code>
	<code> maksCoorR && newCoor(1,2) < maksCoorC</code>
	<code> if</code>
43	<code> imgbw(newCoor(1,1),newCoor(1,2))== 1</code>
44	<code> ctrImg(newCoor(1,1),newCoor(1,2))=1;</code>

45	coor = newCoor;
46	a = (newCoor(1,1) - coorHalfWrist(1,1))^2;
47	b = (newCoor(1,2) - coorHalfWrist(1,2))^2;
48	distance = sqrt(a + b);
49	coorDis(count,1) = newCoor(1,1);
50	coorDis(count,2) = newCoor(1,2);
51	coorDis(count,3) = distance;
52	direction(count,1) = dir;
53	count= count + 1;
54	break;
55	Else
56	currentDir = currentDir + 1;
57	End
58	Else
59	currentDir = currentDir + 1;
60	End
61	End
62	if startCoor == coor
63	indikator = 0;
64	End
65	End
66	for i = 1:size(coorDis(:,3))
67	sumbuX(1,i) = i;
68	End
69	[r c] = find(coorDis(:,3) == min(coorDis(:,3)));
70	newCoorDis = zeros(size(coorDis,1),size(coorDis,2),si ze(coorDis,3));

71	<code>newCoorDis(1:size(coorDis,1)-r+1,:) = coorDis(r:size(coorDis,1),:);</code>
72	<code>newCoorDis(size(coorDis,1)- r+2:size(coorDis),:) = coorDis(1:r-1,:);</code>
73	<code>[r c] = find(newCoorDis(:,1)>zeroDis);</code>
74	<code>newCoorDis(r,3) = 0;</code>
75	<code>output = fpeak(sumbuX,newCoorDis(:,3),50);</code>
76	<code>for i=1:size(output,1)-1</code>
77	<code> if output(i,2)==output(i+1,2)</code>
78	<code> output(i,3) = 1;</code>
79	<code> Else</code>
80	<code> output(i,3) = 0;</code>
81	<code> End</code>
82	<code>End</code>
83	<code>temp = zeros(size(img,1), size(img,2));</code>
84	<code>temp(coorP1(1,1), coorP1(1,2)) = 1;</code>
85	<code>temp(coorP2(1,1), coorP2(1,2)) = 1;</code>
86	<code>tempRotate = imrotate(temp, 90- deg,'bilinear','crop');</code>
87	<code>[i, j] = find(tempRotate);</code>
88	<code>newCoorP1(1,1) = i(1);</code>
89	<code>newCoorP1(1,2) = j(1);</code>
90	<code>newCoorP2(1,1) = i(size(i,1));</code>
91	<code>newCoorP2(1,2) = j(size(j,1));</code>
92	<code>a = (newCoorP1(1,1)-newCoorP2(1,1))^2;</code>
93	<code>b = (newCoorP1(1,2)-newCoorP2(1,2))^2;</code>
94	<code>distance = sqrt(a + b);</code>
95	<code>fiturI = imcrop(newImg, [newCoorP1(1,2), newCoorP1(1,1), distance , distance]);</code>
96	<code>fiturImg = imresize(fiturI,[150 150]);</code>

Kode Sumber 4.2 Implementasi ROI Detection

4.2.2.2. Implementasi Pencarian *fpeak* pada proses ROI Detection

Proses pencarian *fpeak* ini berfungsi untuk mencari titik puncak dari setiap jari, kemudian akan menghasilkan keluaran

yang digunakan untuk menentukan titik P_1 dan P_2 . Implementasi pencarian *fpeak* ditunjukkan pada Kode Sumber 4.3.

1	Data = sortrows([x,y]);
2	for i=1:rx
3	if i-s<1
4	top = 1;
5	else
6	top = i-s;
7	End
8	y = Data(:,2);
9	if i+s>length(y)
10	bottom = length(y);
11	Else
12	bottom = i + s;
13	End
14	if sum(isnan(isP))==0
15	output(numP,:) = isP; %#ok
16	numP = numP + 1;
17	End
18	End

Kode Sumber 4.3 Implementasi Pencarian *fpeak* pada proses ROI Detection

4.2.2.3. Implementasi Perbaikan Citra

Setelah mendapatkan ROI *palm vein*, maka selanjutnya dilakukan perbaikan citra yang nantinya digunakan untuk menonjolkan atau mendapatkan pola pembuluh darah telapak tangan. Pada baris pertama dilakukan *median filter*. Pada baris ke-2 dilakukan *adaptive histogram equalization*. Lalu pada baris terakhir dilakukan *adaptive noise removal* menggunakan wiener. Implementasi perbaikan citra ditunjukkan pada Kode Sumber 4.4.

1	img_med = medfilt2(b,[10 10]);
2	img_adapt = adapthisteq(img_med,'clipLimit',0.08,'Distribution','rayleigh');
3	img_anr = wiener2(img_adapt,[12 12]);

Kode Sumber 4.4 Implementasi perbaikan citra

4.2.2.4. Implementasi Penonjolan Pembuluh Darah Telapak Tangan

Setelah citra diperbaiki, maka selanjutnya yaitu menonjolkan pola pembuluh darah. Implementasi penonjolan pembuluh darah telapak tangan ditunjukkan pada Kode Sumber 4.5. Pada baris ke- 1 digunakan untuk menonjolkan pola pembuluh darah. Pada baris ke- 2 digunakan untuk mengubah nilai double menjadi tipe gambar. Pada baris ke- 3 dan 4 dilakukan proses *closing*. Pada baris ke- 5 dilakukan pengurangan citra hasil penonjolan pembuluh darah dengan hasil *closing*. Pada baris ke- 6 dilakukan *adjust intensity* yang bertujuan untuk memperbaiki intensitas citra hasil pengurangan.

1	<code>img_adf = anisodiff2D(img_anr, 50, 1/7, 20, 1);</code>
2	<code>img_conv= uint8(round(img_adf-1));</code>
3	<code>se = strel('disk',3);</code>
4	<code>img_close = imclose(img_conv,se);</code>
5	<code>imsub = imsubtract(img_close,img_conv);</code>
6	<code>img_adj = imadjust(imsub);</code>

Kode Sumber 4.5 Implementasi penonjolan pembuluh darah telapak tangan

4.2.3. Implementasi Ekstraksi Fitur

Pada tahap ekstraksi fitur ini menggunakan metode nLBP. Sebelum dilakukan proses ekstraksi fitur, dilakukan *thresholding* terlebih dahulu. Setelah itu masuk metode nLBP. Pada baris ke- 1 digunakan untuk menentukan parameter nLBP yaitu *distance* (*d*) antar piksel tetangga. Range nilai *distance* adalah 1 sampai 7 karena menggunakan 8-*neighbors*. Selanjutnya pada baris ke- 4 digunakan untuk membuat matriks window 3x3. Pada baris ke- 5 digunakan untuk mengubah matriks ke bentuk vektor. Hal ini digunakan untuk proses perhitungan nilai nLBP atau *Pc*. Pada baris ke- 6 sampai 12 dilakukan proses perhitungan nilai *Pc* pada setiap piksel dengan membandingkan nilai piksel tertentu dengan piksel tetangga yang sesuai dengan parameter *distance*. Selanjutnya pada

baris ke- 13 sampai 18 digunakan untuk memasukkan hasil perhitungan P_c ke dalam matriks 3×3 . Pada baris ke- 19 digunakan untuk mengambil nilai 1 atau 0 pada setiap *Pixel center* (P_c) matriks. Pada baris ke- 20 digunakan untuk menghitung nilai decimal dari P_c . Selanjutnya dilakukan proses invers nilai P_c setiap piksel yang dapat dilihat pada baris ke- 21. Lalu, pada baris ke- 22 nilai hasil invers P_c dimasukkan ke vektor gambar. Pada baris ke- 23 digunakan untuk memasukkan vektor gambar ke vektor folder agar hasil vektor fitur semua gambar menjadi 1 file. Setelah itu hasil vektor fitur semua gambar disimpan ke dalam file berekstensi .txt. Implementasi proses nLBP ditunjukkan pada Kode Sumber 4.6.

1	D = 1;
2	for i = 1:row max
3	for j = 1:col max
4	A = Input_Im(i:i+L-1, j:j+L-1);
5	vec_A = [A(1,1:3) A(2,3) A(3,3) A(3,2) A(3,1) A(2,1)];
6	for p=1:size(vec_A, 2)
7	index = mod(p+D,8);
8	if index == 0
9	index = 8;
10	End
11	res_A(1,p) = vec_A(1,p)+ 1 - vec_A(1,index);
12	End
13	A(1,1:3) = res_A(1:3);
14	A(2,3) = res_A(4);
15	A(3,3) = res_A(5);
16	A(3,2) = res_A(6);
17	A(3,1) = res_A(7);
18	A(2,1) = res_A(8);
19	A(A>0) = 1;
20	LBP_Im(i,j) = A(C,L) + A(L,L)*2 + A(L,C)*4 + A(L,1)*8 + A(C,1)*16 + A(1,1)*32 + A(1,C)*64 + A(1,L)*128;

21	<code>invers_resnorm = 255 - LBP_Im ;</code>
22	<code>result_vec = invers_resnorm(:)';</code>
23	<code>vec_mat(z,:) = result_vec;</code>
24	<code>End</code>
25	<code>End</code>

Kode Sumber 4.6 Implementasi Ekstraksi Fitur nLBP

4.2.4. Implementasi Klasifikasi *Support Vector Machine* (SVM)

Pada tahap klasifikasi ini memanfaatkan library dari python yaitu Scikit-learn. Masukan dari tahap klasifikasi ini adalah vektor fitur hasil ekstraksi fitur dalam bentuk file bertipe .txt. Pada baris ke- 1 dilakukan pembagian data latih dan data uji menggunakan StratifiedKFold. Parameter `n_splits` adalah 6, artinya pengujian dilakukan sebanyak enam kali dengan data uji sebanyak 6 data dari masing-masing kelas (dalam 1 kali pengujian menggunakan 1 data uji dari masing-masing kelas), dan sisanya menjadi data latih. Pada baris ke- 2 sampai 8 adalah fungsi klasifikasi SVM dengan parameter-parameternya. Pada baris ke 9 adalah perhitungan akurasi yang menjadi hasil akhir klasifikasi. Implementasi proses klasifikasi SVM ditunjukkan pada Kode Sumber 4.6.

1	<code>skf = StratifiedKFold(n_splits=6, shuffle=True, random_state = 0)</code>
2	<code>clf = svm.SVC (C=1.0, kernel='linear', tol=0.001, max_iter=-1)</code>
3	<code>clf = svm.SVC (C=1.0, kernel='rbf', gamma=0.001, tol=0.001, max_iter=-1)</code>
4	<code>clf = svm.SVC (C=1.0, kernel='poly', degree=2, gamma=0.001, tol=0.001, max_iter=-1)</code>
6	<code>for train_indices, test_indices in skf.split(Datafitur,Class):</code>
7	<code>clf.fit(Datafitur[train_indices], Class[train_indices])</code>

8	<code>Class_Pred</code> <code>=</code> <code>clf.predict(Datafitur[test_indices])</code>
9	<code>akurasi</code> <code>+=</code> <code>accuracy_score(Class[test_indices],</code> <code>Class_Pred)</code>

Kode Sumber 4.7 Implementasi klasifikasi SVM

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini dibahas tentang hasil uji coba dan evaluasi pada program yang telah selesai diimplementasi pada tugas akhir ini. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan pengujian yang telah ditentukan.

5.1. Lingkungan Pengujian

Dalam proses pengujian sistem, dibutuhkan suatu lingkungan uji coba yang sesuai dengan standar kebutuhan. Lingkungan uji coba ini mencakup perangkat keras dan perangkat lunak. Spesifikasi perangkat keras dan lunak yang digunakan dalam implementasi ini ditunjukkan pada Tabel.

Tabel 5.1 Lingkungan pengujian perangkat keras dan lunak

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel ® Core(TM) i7-7500U CPU @ 2.70GHz (4 CPUs), ~2.9GHz
Perangkat lunak	Sistem Operasi : Windows 10 Pro 64-bit Perangkat Pengembang: Matlab R2017b, Python Spyder 3.2.4, Scikit-learn 0.19.1 Perangkat Pembantu: Sublime Text

5.2. Data Uji Coba

Data yang digunakan sebagai masukan dalam uji coba pengenalan *palm vein* adalah citra *palm vein* yang bersumber dari database *CASIA Multi-Spectral Palmprint Image Database V1.0* (*CASIA-MS-PalmprintV1*). Database ini berisi 600 citra *palm vein* yang diambil dari 100 orang, masing-masing orang memiliki enam citra.

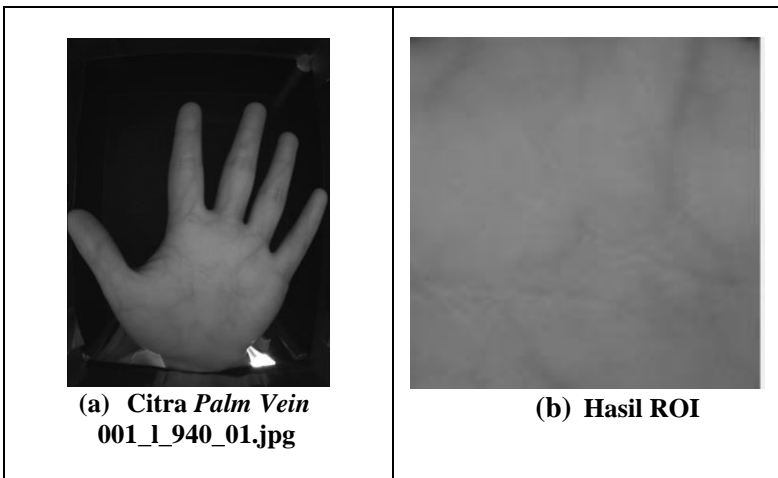
Dalam melakukan pengujian di tahap klasifikasi, citra *palm vein* dibagi menjadi data latih dan data uji. Pembagian data ini menggunakan *Cross Validation* yang dibagi secara rata setiap kelasnya yang sering disebut *Stratified*. Data di-*split* menjadi 6 bagian untuk setiap kelasnya. Setiap kelas terdapat 6 data. Jumlah kelas pada *database* ini sebanyak 100 kelas. Sehingga didapatkan setiap kelas diambil satu data sebagai data uji sedangkan sisanya digunakan sebagai data latih. Karena *split*-nya adalah 6, maka pengujian dilakukan sebanyak 6 kali dengan data latih berjumlah $5 \times 100 = 500$ data dan data uji berjumlah $1 \times 100 = 100$ data.



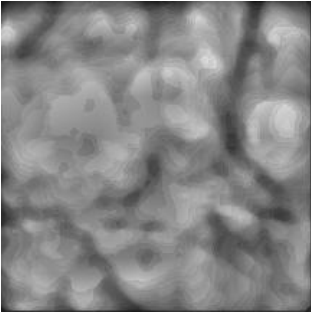
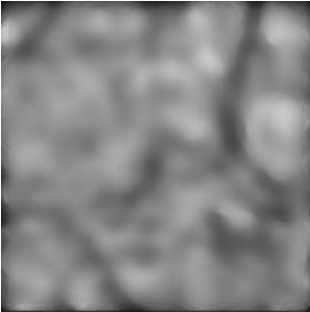
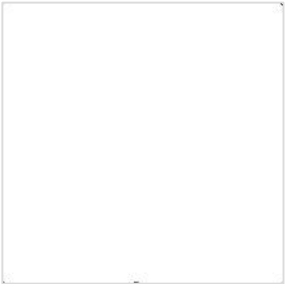
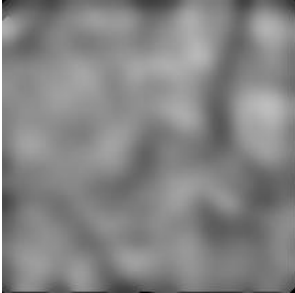
5.3. Hasil Uji Coba Pada Setiap Tahap Sistem

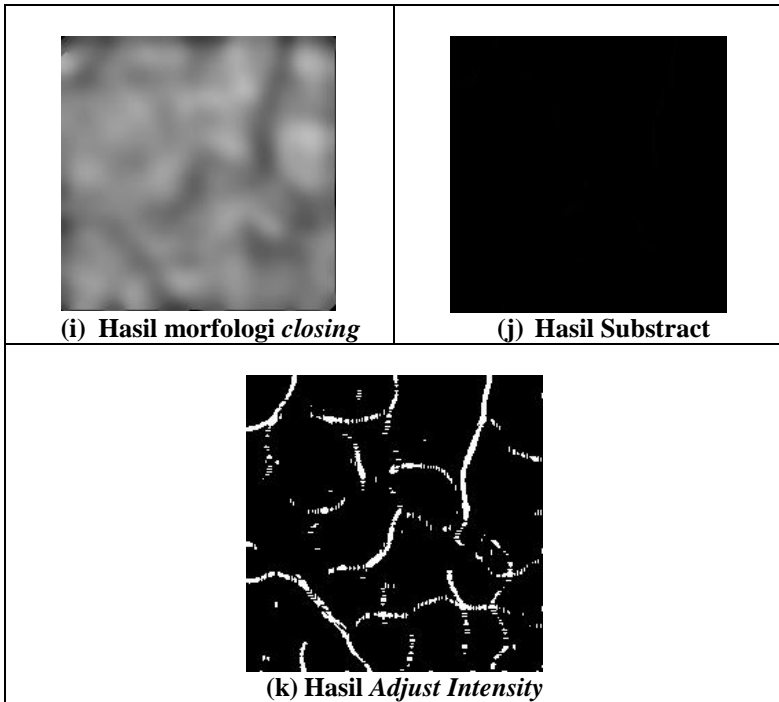
Pada subbab ini akan dijabarkan mengenai hasil uji coba setiap tahap pada sistem yaitu *preprocessing* dan ekstraksi fitur.

5.3.1. Preprocessing

Pada tahap *preprocessing*, citra *palm vein* akan diolah untuk mendapatkan hasil fokus citra pada pembuluh darah telapak tangan. Contoh sampel langkah-langkah untuk mendapatkan hasil fokus citra ditunjukkan pada Gambar 5.1.



	
(c) Hasil Resize	(d) Hasil <i>Median Filter</i>
	
(e) Hasil <i>Adaptive Histogram Equalization</i>	(f) Hasil <i>Adaptive Noise Removal</i>
	
(g) Hasil <i>Anisotropic Diffusion Filter</i>	(h) Hasil <i>Convert</i>



Gambar 5.1 Sampel hasil *preprocessing* pada citra *palm vein*

5.3.2. Ekstraksi Fitur

Ekstraksi fitur yang digunakan dalam uji coba ini menggunakan *Local Binary Patterns by Neighborhoods* (nLBP). Langkah-langkah ekstraksi fitur dengan metode nLBP telah dijelaskan pada subbab 3.5. Hasil ekstraksi fitur disimpan dalam file .txt yang nantinya digunakan sebagai data masukan pada proses klasifikasi.

5.4. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan menggunakan data latih dan data uji. Pada uji coba ini, implementasi algoritma ekstraksi fitur pada klasifikasi

SVM akan diuji kinerja/performanya pada masing-masing skenario.

Skenario uji coba terdiri dari 5 macam yaitu:

1. Perhitungan performa hasil akurasi dengan variasi parameter *distance* (d) pada metode ekstraksi fitur nLBP. Parameter *distance* (d) yang akan diuji yaitu 1, 2, 3, 4, 5, 6, dan 7
2. Perhitungan performa hasil akurasi dengan variasi parameter nilai gamma pada SVM
3. Perhitungan performa hasil akurasi dengan variasi parameter nilai C (*Penalty coefficient*) pada SVM
4. Perhitungan performa hasil akurasi dengan variasi parameter kernel pada SVM. Variasi kernel yang diuji yaitu Linear, Radial Base Function (RBF) atau Gaussian Kernel, dan Polynomial.
5. Perbandingan performa hasil akurasi dari metode ekstraksi fitur LBP dan nLBP

5.4.1. Skenario Uji Coba Parameter *Distance*

Skenario uji coba ini adalah perhitungan performa hasil akurasi dengan variasi parameter *distance* (d) pada metode ekstraksi fitur nLBP. Skenario dicoba dengan menggunakan beberapa variasi *distance* (d), yaitu 1, 2, 3, 4, 5, 6, dan 7. Parameter *default* yang digunakan dalam skenario ini adalah *Penalty coefficient* ($C = 1$), $\gamma(\text{auto}=0.00003543)$, dan kernel RBF. Hasil skenario uji coba parameter *distance* ditunjukkan pada Tabel 5.2.

Tabel 5.2 Persentase Akurasi Fitur nLBP dengan Variasi Nilai *Distance* pada Ekstraksi Fitur nLBP

<i>Distance</i> (d)	Akurasi (%)
1	75.66
2	75.00
3	75.66
4	76.16
5	75.66
6	75.66
7	75.66

Berdasarkan hasil performa akurasi pada Tabel 5.2, ekstraksi fitur nLBP dengan menggunakan parameter *distance* (d) bernilai 4 menghasilkan nilai akurasi tertinggi yaitu 76.16%. Sedangkan *distance* lainnya cenderung menghasilkan hasil akurasi yang sama. Dari uji coba tersebut, didapatkan bahwa parameter *distance* (d) yang terbaik adalah 4.

5.4.2. Skenario Uji Coba Parameter Gamma

Skenario uji coba ini adalah variasi parameter nilai gamma pada SVM. Skenario dicoba dengan menggunakan beberapa variasi nilai gamma yaitu $\text{auto}(0.00003543)$, 0.0001, 0.001, 0.01, 0.1, dan 1. Parameter *default* yang digunakan dalam skenario ini adalah *distance* ($d = 4$), *Penalty coefficient* ($C = 1$), dan kernel RBF. Hasil skenario uji coba parameter gamma ditunjukkan pada Tabel 5.3.

Tabel 5.3 Persentase Akurasi Fitur nLBP dengan Variasi Nilai Gamma

Gamma	Akurasi (%)
0.00003543	76.16
0.0001	76.66
0.001	84.83
0.01	43.00
0.1	68.33
1	1.33

Berdasarkan hasil performa akurasi pada Tabel 5.3, klasifikasi dengan menggunakan parameter gamma sebesar 0.001 menghasilkan nilai akurasi tertinggi yaitu 84.83%. Sedangkan akurasi terendah terjadi ketika nilai gamma 1. Sehingga dapat diamati bahwa semakin besar nilai gamma maka akurasi akan menurun. Dari uji coba tersebut, didapatkan bahwa parameter gamma yang terbaik adalah 0.001.

5.4.3. Skenario Uji Coba Parameter C

Skenario uji coba ini adalah perhitungan performa hasil akurasi dengan variasi parameter nilai C (*Penalty coefficient*) pada SVM. Skenario dicoba dengan menggunakan beberapa variasi C yaitu 1, 2, 10, 30, 50, 70 dan 100. Parameter *default* yang digunakan dalam skenario ini adalah *distance* ($d = 4$), $\gamma(0.001)$, dan kernel RBF. Hasil skenario uji coba parameter C ditunjukkan pada Tabel 5.4.

Tabel 5.4 Persentase Akurasi Fitur nLBP dengan Variasi Nilai C

C	Akurasi (%)
1	84.83
2	86.66
10	86.66
30	86.66
50	86.66
70	86.66
100	86.66

Berdasarkan hasil performa akurasi pada Tabel 5.2, klasifikasi dengan menggunakan nilai parameter C yaitu 2, 10, 30, 50, 70 dan 100 menghasilkan nilai akurasi tertinggi yang sama yaitu 86.66%. Parameter C (*Penalty coefficient*) adalah sebuah parameter yang mengontrol besarnya toleransi eror. Apabila nilai C semakin besar, maka *penalty*/toleransi terhadap eror akan semakin besar. Dan begitu pula sebaliknya. Oleh karena itu, dipilihlah parameter C terbaik adalah 2.

5.4.4. Skenario Uji Coba Parameter Kernel SVM

Skenario uji coba ini adalah perhitungan performa hasil akurasi dengan variasi parameter kernel pada SVM. Skenario dicoba dengan menggunakan beberapa variasi kernel yaitu linear, RBF, dan polynomial. Parameter *default* yang digunakan dalam skenario ini adalah *distance* ($d = 4$), *Penalty coefficient* ($C = 2$), dan $\gamma(0.001)$. Hasil skenario uji coba parameter kernel SVM ditunjukkan pada Tabel 5.5.

Tabel 5.5 Persentase Akurasi Fitur nLBP dengan Variasi Kernel

Kernel	Akurasi (%)
Linear	84.83
RBF	86.66
Polynomial	36.66

Berdasarkan hasil performa akurasi pada Tabel 5.5, klasifikasi dengan menggunakan parameter kernel RBF menghasilkan nilai akurasi tertinggi yaitu 86.66%. Sedangkan kernel linear menghasilkan akurasi 84.83%. Akurasi terendah sebesar 36.66% diperoleh pada kernel polynomial. Dari uji coba tersebut, didapatkan bahwa kernel terbaik adalah Radial Base Function (RBF).

5.4.5. Skenario Uji Coba Perbandingan Metode

Skenario uji coba ini adalah perbandingan performa hasil akurasi dari metode ekstraksi fitur LBP dan nLBP. nLBP yang digunakan adalah nLBP *non-uniform* dan LBP yang digunakan adalah LBP *non-uniform*. Parameter *default* yang digunakan dalam skenario ini adalah *Penalty coefficient* ($C = 2$), $\gamma(0.001)$, dan kernel RBF. Hasil skenario uji coba ini ditunjukkan pada Tabel 5.6.

Tabel 5.6 Persentase Akurasi nLBP dan LBP

Metode	Akurasi (%)
nLBP(d=4)	86.66
LBP	81.66

Berdasarkan hasil performa akurasi pada Tabel 5.6, metode nLBP menghasilkan akurasi lebih tinggi dari metode LBP yaitu sebesar 86.66%. Sedangkan metode LBP menghasilkan akurasi sebesar 81.66%. Dari uji coba tersebut, didapatkan bahwa metode ekstraksi fitur nLBP lebih baik daripada metode LBP dalam pengenalan *palm vein*.

5.5. Analisis Hasil Uji Coba

Berdasarkan hasil skenario-skenario uji coba yang telah dilakukan, dapat diketahui bahwa tingkat ketepatan dalam pengenalan *palm vein* sudah baik.

Pada skenario uji coba parameter *distance*, dilakukan perhitungan performa hasil akurasi dengan variasi parameter *distance* (d) pada metode ekstraksi fitur nLBP. Hasil performa yang didapatkan tidak terlalu memiliki perbedaan yang signifikan antar *distance*-nya. Hasil akurasi tertinggi didapatkan ketika parameter *distance* (d) bernilai 4. Tidak adanya perbedaan yang signifikan ini dapat terjadi karena banyak fitur yang tidak terlalu memberikan pengaruh besar terhadap hasil klasifikasi.

Skenario uji coba ke- 2 adalah perhitungan performa hasil akurasi dengan variasi parameter nilai gamma pada SVM. Dari hasil performa yang didapatkan, dapat dilihat bahwa semakin besar nilai gamma maka akurasi semakin menurun. Hasil akurasi tertinggi didapatkan ketika parameter gamma bernilai 0.001.

Skenario uji coba ke- 3 adalah perhitungan performa hasil akurasi dengan variasi parameter nilai C (*Penalty coefficient*) pada SVM. Dari hasil performa yang didapatkan, dapat dilihat bahwa hasilnya tidak terlalu memiliki perbedaan yang signifikan. Parameter C (*Penalty coefficient*) merupakan parameter yang mengontrol besar toleransi eror. Apabila nilai C semakin besar, maka *penalty*/toleransi terhadap eror akan semakin besar. Sedangkan apabila nilai C semakin kecil, maka *penalty*/toleransi terhadap eror juga akan semakin kecil. Hasil akurasi tertinggi didapatkan ketika C bernilai 2, 10, 30, 50, 70, dan 100. Oleh karena semakin kecil nilai C maka *penalty* eror semakin kecil, maka nilai C terbaik adalah 2.

Skenario uji coba ke- 4 adalah perhitungan performa hasil akurasi dengan variasi parameter kernel pada SVM. Dari hasil performa yang didapatkan, dapat dilihat bahwa kernel RBF menghasilkan akurasi tertinggi. Sedangkan paling rendah yaitu kernel polynomial dengan *range* yang cukup jauh. Hal ini dapat terjadi karena data yang tidak cocok dengan penggunaan kernel

polynomial atau kurangnya eksplorasi parameter untuk kernel polynomial.

Skenario uji coba ke- 5 adalah perbandingan performa hasil akurasi dari metode ekstraksi fitur LBP dan nLBP. Pada uji coba ini, metode nLBP dibandingkan dengan metode LBP tradisional. Hasil uji coba ini adalah metode nLBP menghasilkan performa yang lebih tinggi dibandingkan metode LBP tradisional. Sehingga didapatkan bahwa metode nLBP lebih unggul untuk pengenalan *palm vein*.

Dari uji coba yang dilakukan, terdapat kelas yang sering salah ketika diklasifikasi yaitu kelas 6. Kesalahan dalam klasifikasi atau pengenalan ini dapat terjadi karena kurang jelasnya pola pembuluh darah dan masih ada noise yang tersisa. Sehingga ketika diklasifikasikan ternyata mirip dengan pola pembuluh darah kelas lain.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang diperoleh berdasarkan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dari hasil pengamatan yang diperoleh dari uji coba dan evaluasi yang dilakukan, maka dapat disimpulkan sebagai berikut.

1. Parameter *distance* (d) terbaik untuk metode ekstraksi fitur nLBP adalah 4.
2. Parameter-parameter SVM yang terbaik adalah nilai γ 0.001, nilai C (*Penalty coefficient*) 2, dan kernel Radial Base Function (RBF).
3. Metode ekstraksi fitur *Local Binary Patterns by Neighborhoods* (nLBP) menghasilkan akurasi lebih besar dari metode LBP yaitu 86.66%.

6.2. Saran

Saran-saran yang dapat diberikan dalam pengujian metode ekstraksi fitur nLBP pada *palm vein* adalah sebagai berikut:

1. Pada penelitian selanjutnya dapat dilakukan perbaikan tahap *preprocessing* untuk memperjelas pola pembuluh darah pada *palm* agar menghasilkan data yang berkualitas tinggi sehingga dapat menghasilkan performa yang lebih optimal pada pengenalan *palm vein*.
2. Pada penelitian selanjutnya dapat dilakukan juga eksplorasi penggunaan parameter klasifikasi SVM.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] J. Y. Sari, C. Fatchah, dan N. Suciati, "*LOCAL LINE BINARY PATTERN FOR FEATURE EXTRACTION ON PALM VEIN RECOGNITION*," *J. Ilmu Komput. Dan Inf.*, vol. 8, no. 2, hlm. 111–118, Aug. 2015.
- [2] L. Mirmohamadsadeghi dan A. Drygajlo, "Palm vein recognition with local texture patterns," *IET Biometrics*, vol. 3, no. 4, hlm. 198–206, Jan 2014.
- [3] B. A. Rosdi, C. W. Shing, dan S. A. Suandi, "Finger Vein Recognition Using Local Line Binary Pattern," *Sensors*, vol. 11, no. 12, hlm. 11357–11371, Nov 2011.
- [4] Y. Kaya, Ö. F. Ertuğrul, dan R. Tekin, "Two novel local binary pattern descriptors for texture analysis," *Applied Soft Computing*, vol. 34, no. Supplement C, hlm. 728–735, Sep 2015.
- [5] M. Watanabe, T. Endoh, M. Shiohara, dan S. Sasaki, "Palm vein authentication technology and its applications," hlm. 2, Sep 2005.
- [6] "ROI-Based Processing - MATLAB & Simulink." [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/roi-based-processing.html>. [Diakses: 10-Jul-2018].
- [7] "Region of interest." [Daring]. Tersedia pada: <https://www.revolvy.com/main/index.php?s=Region+of+interest>. [Diakses: 06-Jun-2018].
- [8] "2-D median filtering - MATLAB medfilt2." [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/medfilt2.html>. [Diakses: 06-Jun-2018].
- [9] "Contrast-limited adaptive histogram equalization (CLAHE) - MATLAB adapthisteq." [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/adapthisteq.html>. [Diakses: 05-Jan-2018].
- [10] A. Rachmad, "Pengolahan Citra Digital Menggunakan Teknik Filtering Adaptive Noise Removal Pada Gambar Bernoise," *Prosiding Seminar Nasional Teknoin*, Jan 2008.

- [11] “2-D adaptive noise-removal filtering - MATLAB wiener2.” [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/wiener2.html>. [Diakses: 30-Apr-2018].
- [12] “Anisotropic diffusion filtering of images - MATLAB imdiffusefilt.” [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/imdiffusefilt.html>. [Diakses: 06-Jun-2018].
- [13] M. Ikhsan, “MORFOLOGI DAN DETEKSI TEPI PADA CITRA DIGITAL SEBAGAI OBJECT COUNTING DALAM MENGHITUNG JUMLAH SAMPEL BAKTERI PADA CAWAN PETRI.”
- [14] “Morphologically open image - MATLAB imopen.” [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/imopen.html>. [Diakses: 05-Jan-2018].
- [15] “Morphologically close image - MATLAB imclose.” [Daring]. Tersedia pada: <https://www.mathworks.com/help/images/ref/imclose.html>. [Diakses: 05-Jan-2018].
- [16] C.-W. Hsu, C.-C. Chang, dan C.-J. Lin, “A Practical Guide to Support Vector Classification,” hlm. 16, Mei 2016.
- [17] P. Peter, "Aplikasi Pengenalan Palm Vein Berdasarkan Fitur Gabor," Institut Teknologi Sepuluh Nopember, 2017.

LAMPIRAN

Lampiran 1. Hasil prediksi nLBP (Label Kelas) dengan parameter $d=4$, $C=2$, $\gamma=0.001$, kernel RBF.

No	Nama Data Uji	Aktual	Prediksi
1	001_1_940_05.jpg	1	1
2	002_1_940_06.jpg	2	2
3	003_1_940_01.jpg	3	3
4	004_1_940_01.jpg	4	60
5	005_1_940_01.jpg	5	5
6	006_1_940_06.jpg	6	21
7	007_1_940_04.jpg	7	7
8	008_1_940_01.jpg	8	8
9	009_1_940_01.jpg	9	9
10	010_1_940_03.jpg	10	10
11	011_1_940_05.jpg	11	11
12	012_1_940_05.jpg	12	12
13	013_1_940_01.jpg	13	21
14	014_1_940_04.jpg	14	14
15	015_1_940_01.jpg	15	85
16	016_1_940_03.jpg	16	16
17	017_1_940_05.jpg	17	17
18	018_1_940_05.jpg	18	18
19	019_1_940_03.jpg	19	19
20	020_1_940_04.jpg	20	20
21	021_1_940_01.jpg	21	21
22	022_1_940_01.jpg	22	22
23	023_1_940_01.jpg	23	23
24	024_1_940_03.jpg	24	24
25	025_1_940_03.jpg	25	25
26	026_1_940_04.jpg	26	26
27	027_1_940_02.jpg	27	27
28	028_1_940_04.jpg	28	28
29	029_1_940_01.jpg	29	29
30	030_1_940_04.jpg	30	30
31	031_1_940_01.jpg	31	21
32	032_1_940_03.jpg	32	32

33	033_1_940_01.jpg	33	33
34	034_1_940_03.jpg	34	34
35	035_1_940_01.jpg	35	35
36	036_1_940_05.jpg	36	62
37	037_1_940_01.jpg	37	37
38	038_1_940_06.jpg	38	38
39	039_1_940_05.jpg	39	39
40	040_1_940_06.jpg	40	40
41	041_1_940_01.jpg	41	38
42	042_1_940_02.jpg	42	42
43	043_1_940_04.jpg	43	43
44	044_1_940_04.jpg	44	44
45	045_1_940_03.jpg	45	45
46	046_1_940_01.jpg	46	46
47	047_1_940_06.jpg	47	94
48	048_1_940_06.jpg	48	66
49	049_1_940_02.jpg	49	49
50	050_1_940_05.jpg	50	50
51	051_1_940_05.jpg	51	21
52	052_1_940_01.jpg	52	52
53	053_1_940_05.jpg	53	53
54	054_1_940_06.jpg	54	54
55	055_1_940_02.jpg	55	80
56	056_1_940_04.jpg	56	56
57	057_1_940_04.jpg	57	57
58	058_1_940_06.jpg	58	58
59	059_1_940_06.jpg	59	59
60	060_1_940_06.jpg	60	60
61	061_1_940_03.jpg	61	61
62	062_1_940_02.jpg	62	62
63	063_1_940_06.jpg	63	63
64	064_1_940_04.jpg	64	64
65	065_1_940_03.jpg	65	55
66	066_1_940_05.jpg	66	66
67	067_1_940_04.jpg	67	67
68	068_1_940_01.jpg	68	68
69	069_1_940_01.jpg	69	69

70	070_1_940_01.jpg	70	70
71	071_1_940_05.jpg	71	71
72	072_1_940_03.jpg	72	72
73	073_1_940_06.jpg	73	73
74	074_1_940_05.jpg	74	74
75	075_1_940_05.jpg	75	75
76	076_1_940_06.jpg	76	76
77	077_1_940_03.jpg	77	77
78	078_1_940_03.jpg	78	78
79	079_1_940_01.jpg	79	79
80	080_1_940_04.jpg	80	80
81	081_1_940_05.jpg	81	81
82	082_1_940_01.jpg	82	82
83	083_1_940_01.jpg	83	83
84	084_1_940_05.jpg	84	84
85	085_1_940_01.jpg	85	62
86	086_1_940_05.jpg	86	86
87	087_1_940_02.jpg	87	87
88	088_1_940_04.jpg	88	88
89	089_1_940_04.jpg	89	89
90	090_1_940_01.jpg	90	90
91	091_1_940_05.jpg	91	91
92	092_1_940_02.jpg	92	92
93	093_1_940_01.jpg	93	93
94	094_1_940_04.jpg	94	94
95	095_1_940_02.jpg	95	95
96	096_1_940_02.jpg	96	96
97	097_1_940_04.jpg	97	97
98	098_1_940_02.jpg	98	98
99	099_1_940_05.jpg	99	99
100	100_1_940_03.jpg	100	100

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, **Delia Risti Nareswari**, lahir di Surabaya, 24 Januari 1996. Penulis menempuh pendidikan sekolah dasar di SD Negeri Manukan Wetan 1, Surabaya. Melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 1 Surabaya dan selanjutnya di SMA Negeri 5 Surabaya. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama kuliah, penulis aktif menjadi asisten praktikum Komputasi Numerik dan aktif dalam organisasi tingkat jurusan dan fakultas, serta kepanitiaan acara. Diantaranya penulis berpartisipasi sebagai anggota *Departement of Internal Affairs* BEM FTIf (2015-2016), sekretaris *Departement of Internal Affairs* BEM FTIf (2016-2017) anggota Departemen Dalam Negeri HMTC (2015-2016), anggota sie Dana dan Usaha Schematics 2015 dan 2016.

Penulis juga mengikuti kegiatan pelatihan, diantaranya berpartisipasi sebagai peserta aktif LKMM Pra Tingkat Dasar FTIf 2014 dan peserta aktif LKMM Tingkat Dasar HMTC 2015.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas Visi (KCV). Penulis dapat dihubungi melalui email: deliaaristi@gmail.com