

SKRIPSI
JURUSAN TEKNIK ELEKTRO

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun Oleh:

RIESTA ASTRIAWANTI

NIM. 0810633076-63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG

2012

LEMBAR PERSETUJUAN

**ENHANCEMENT CITRA SIDIK JARI MENGGUNAKAN
METODE GABOR FILTER**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun oleh:

RIESTA ASTRIAWANTI

NIM. 0810633076-63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Muhammad Aswin, MT.

NIP. 19640626 199002 1 001

Waru Djuriatno, ST., MT.

NIP. 19690725 199702 1 001

LEMBAR PENGESAHAN

**ENHANCEMENT CITRA SIDIK JARI MENGGUNAKAN
METODE GABOR FILTER**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*

Disusun oleh:

RIESTA ASTRIAWANTI

NIM. 0810633076-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 10 Agustus 2012

DOSEN PENGUJI

R. Arief Setyawan, ST., MT.

NIP. 19760121 200501 1 001

Adharul Muttaqin, ST., MT.

NIP. 19760121 200501 1 001

Ali Mustofa, ST., MT.

NIP. 19710601 200003 1 001

Mengetahui

Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, MS.

NIP. 19580728 198701 1 001

KATA PENGANTAR

Dengan mengucapkan Alhamdulillah Rabbil'alamin, puji syukur kepada Allah SWT yang telah melimpahkan rahmat, ridho, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Enhancement Citra Sidik Jari Menggunakan Metode Gabor Filter" sebagai salah satu syarat yang diajukan untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.

Penulis menyadari bahwa penulisan skripsi ini tidak akan terselesaikan tanpa adanya bantuan dari berbagai pihak karena itu ucapan terima kasih penulis sampaikan kepada semua pihak yang telah dengan rela meluangkan waktunya untuk membantu penulis dalam proses penulisan skripsi dan juga dengan ikhlas memberikan bimbingan, arahan, masukan, dan semangat sehingga penulis dapat menyelesaikan skripsi ini. Oleh karena itu pada kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Papa Bambang Iwan, Mama Sulastri, Restu Fitriawanti, Reski Tri Setyawan atas segala dukungan dan do'a hingga terselesaikannya skripsi ini
2. Bapak Dr. Ir. Sholeh Hadi Pramono., MS. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
3. Bapak M. Azis Muslim, S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
4. Bapak Moch. Rif'an. ST., MT. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
5. Bapak Ir. Muhammad Aswin, MT. dan Bapak Waru Djuriatno ST., MT. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.
6. Bapak Unggul Wibawa., Ir., M.Sc. selaku dosen penasihat akademik yang telah memberikan nasehat dan arahan dalam proses akademik penulis.
7. Liphoe Eka Pamuncak yang selalu memberi dukungan dan semangat kepada penulis selama proses pengerjaan skripsi ini,
8. Rekan dalam pengerjaan skripsi Karina Seftia, Alvita Arini, Ceri Ahendyarti, Pawitra Suci, Rahma Amalia, Risa Pradita, Mas Adam, Mas

Ferdi, yang telah membantu memberi solusi pada permasalahan yang muncul dan memberi masukan dalam proses pengerjaan skripsi ini.

9. Seluruh teman-teman angkatan 2008 (CONCORDES '08) yang selalu bersama dalam keceriaan.
10. Rekan - Rekan Laboratorium Informatika Dan Komputer.
11. Semua pihak yang telah membantu penyelesaian skripsi ini baik secara langsung dan tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap semoga skripsi ini bermanfaat bagi pembaca dan dapat memberikan sumbangan pikiran bagi pihak-pihak lain khususnya mahasiswa jurusan teknik elektro. Segala kritik dan saran yang penulis terima demi kesempurnaan skripsi ini karena penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna dan masih banyak memiliki kekurangan.

Malang, Agustus 2012

Penyusun

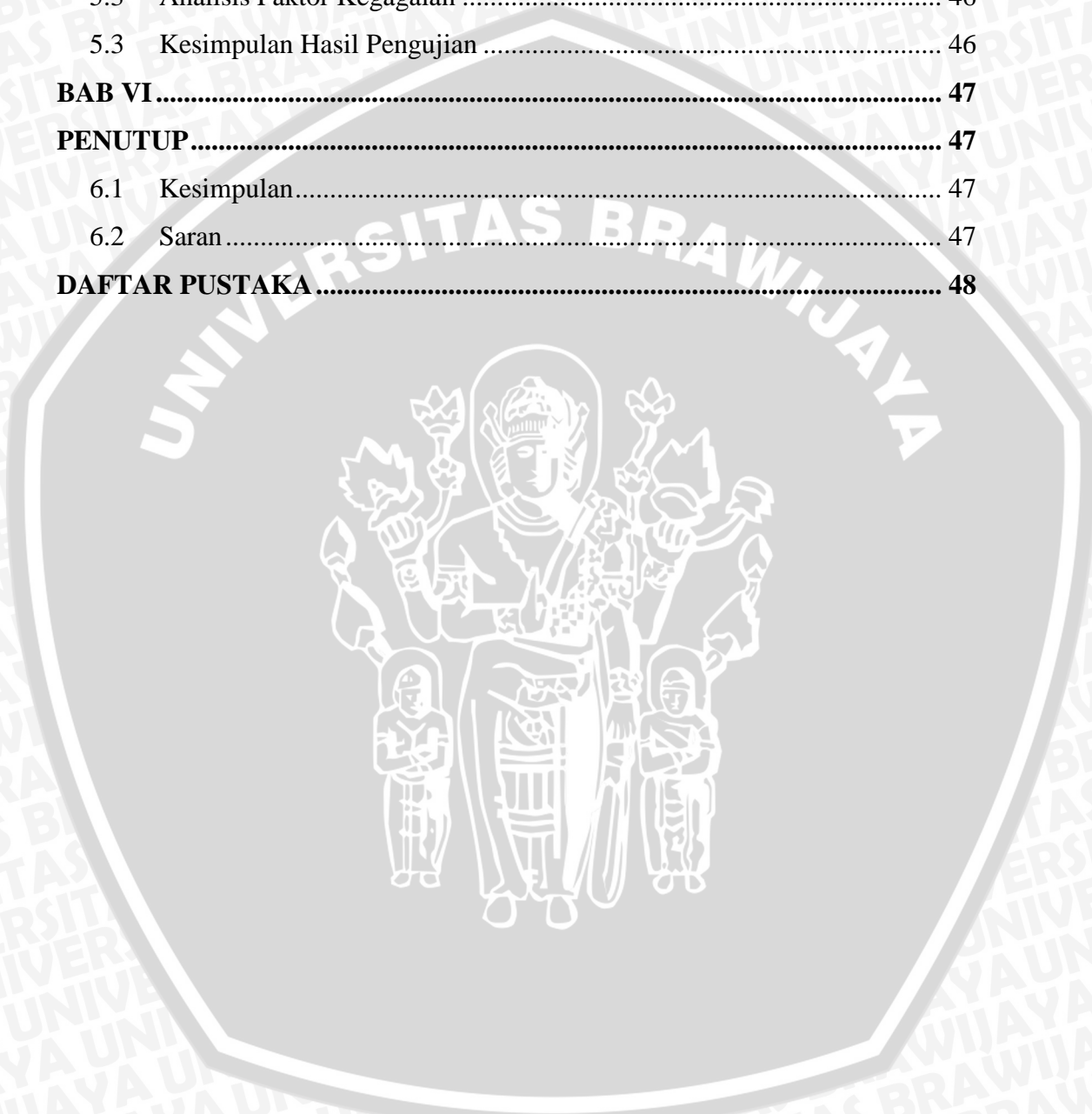


DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	3
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
ABSTRAK	viii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	1
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Sistematika Penulisan.....	3
BAB II	5
TINJAUAN PUSTAKA	5
2.1 Citra	5
2.2 Citra Digital	5
2.3 Pixel dan Resolusi Pixel	6
2.4 Jenis – Jenis Citra	6
2.4.1 Citra Berwarna	7
2.4.2 Citra <i>Grayscale</i>	7
2.4.3 Citra Biner	8
2.5 <i>Bitmap</i>	9
2.5.1 Struktur File <i>Bitmap</i>	9
2.6 Sidik Jari.....	10
2.7 Citra <i>Enhancement</i>	12
2.7.1 Segmentasi	12
2.7.1.1 Metode Otsu.....	13
2.8 Gabor Filter	14
2.9 Operasi Aritmatika Citra	15

2.9.1	Operasi Penjumlahan.....	16
2.9.2	Operasi Invert.....	16
BAB III.....		18
METODE PENELITIAN.....		18
3.1	Studi Literatur.....	18
3.2	Penentuan Spesifikasi Alat.....	18
3.3	Konfigurasi Sistem.....	19
3.5	Pengujian Dan Analisis Sistem.....	20
3.6	Kesimpulan Dan Saran.....	20
BAB IV.....		21
PERANCANGAN DAN IMPLEMENTASI.....		21
4.1	Perancangan Secara Umum.....	21
4.1.1	Blok Sistem Diagram.....	21
4.1.2	Cara Kerja Aplikasi.....	22
4.2	Perancangan Perangkat Lunak.....	22
4.2.1	Perancangan <i>Preprocess</i>	23
4.2.1.1	Perancangan Sub Proses Bagian Konversi Grayscale.....	24
4.2.1.1.1	Perancangan Sub Pengolahan Citra bagian Operasi Bitmap ...	25
4.2.2	Perancangan Proses.....	26
4.2.2.1	Perancangan Sub Proses Bagian Gabor Filter.....	28
4.2.2.2	Perancangan Sub Proses Bagian Penjumlahan Image.....	30
4.2.2.2	Perancangan Sub Proses Bagian Invert Image.....	31
4.3	Implementasi Sistem.....	31
4.3.1	Lingkungan Implementasi.....	32
4.3.2	Implementasi Algoritma.....	32
4.3.2.1	Proses Grayscale.....	32
4.3.2.2	Proses Segmentasi.....	33
4.3.2.3	Proses Gabor Filter.....	34
4.3.2.4	Proses Operasi Aritmatika.....	34
4.4	Implementasi Interface (Antarmuka).....	35
4.4.1	Implementasi Antarmuka Untuk <i>Load</i> Data.....	36
4.4.2	Implementasi Antarmuka Proses Pengolahan Citra.....	36

BAB V.....	38
PENGUJIAN	38
5.1 Pengujian Gabor Filter Dengan 4 Orientasi Sudut Berbeda.....	38
5.2 Pengujian Operasi Aritmatika	42
5.3 Analisis Faktor Kegagalan	46
5.3 Kesimpulan Hasil Pengujian	46
BAB VI.....	47
PENUTUP.....	47
6.1 Kesimpulan.....	47
6.2 Saran.....	47
DAFTAR PUSTAKA	48

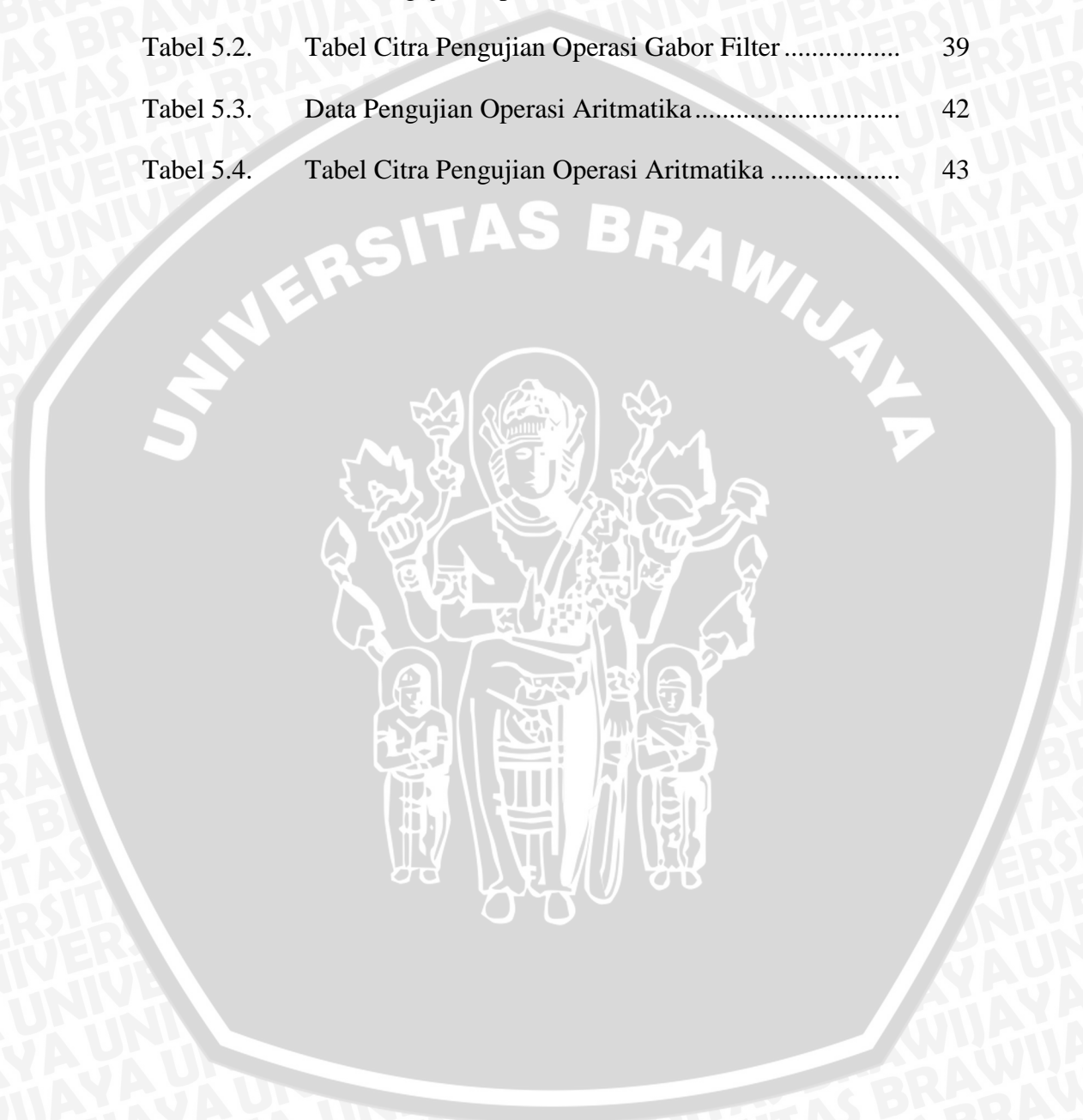


DAFTAR GAMBAR

Gambar 2.1	Citra Digital	5
Gambar 2.2	<i>Image size and resolution</i>	6
Gambar 2.3.	Citra Berwarna	7
Gambar 2.4.	Citra Grayscale.....	8
Gambar 2.5.	Citra Biner.....	9
Gambar 2.6.	Format file <i>bitmap</i> umum	10
Gambar 2.7.	Klasifikasi sidik jari	11
Gambar 2.8.	Metode Otsu.....	14
Gambar 2.9.	Citra hasil gabor filter.....	15
Gambar 3.1.	Diagram Blok Sistem.....	19
Gambar 4.1.	Diagram Blok Sistem Secara Keseluruhan.....	21
Gambar 4.2.	Detail Desain Aplikasi	23
Gambar 4.3.	Flowchart <i>Preprocess</i>	23
Gambar 4.4.	Flowchart Proses Bagian Konversi Grayscale.....	24
Gambar 4.5.	Flowchart Operasi <i>Bitmap</i>	26
Gambar 4.6.	Flowchart <i>Process</i>	27
Gambar 4.7.	Flowchart Proses Gabor Filter	28
Gambar 4.8.	Flowchart Proses Penjumlahan Image	30
Gambar 4.9.	Flowchart Proses Invert Image	31
Gambar 4.10.	Tampilan Utama Program.....	36
Gambar 4.11.	Tampilan Citra Hasil Filterisasi	37

DAFTAR TABEL

Tabel 5.1	Data Pengujian Operasi Gabor Filter.....	38
Tabel 5.2.	Tabel Citra Pengujian Operasi Gabor Filter	39
Tabel 5.3.	Data Pengujian Operasi Aritmatika.....	42
Tabel 5.4.	Tabel Citra Pengujian Operasi Aritmatika	43



ABSTRAK

RIESTA ASTRIAWANTI, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, 2012, "*Enhancement Citra Sidik Jari Menggunakan Metode Gabor Filter*", Dosen Pembimbing: Ir. Muhammad Aswin, MT. dan Waru Djuriatno, ST., MT.

Sistem identifikasi pada sidik jari manusia mulai banyak digunakan. Untuk menemukan titik ciri dari sidik jari seseorang maka harus dilakukan proses ekstraksi titik ciri yang dapat membedakan antara satu citra sidik jari dengan citra sidik jari yang lain. Namun terkadang citra yang di ambil masih memiliki kualitas yang kurang baik sehingga harus dilakukan *enhancement* pada citra input sidik jari untuk dapat menemukan titik ciri yang benar. Kualitas sidik jari memberikan pengaruh penting terhadap proses ekstraksi titik ciri agar dapat dilakukan proses identifikasi dengan hasil yang benar. Gabor filter merupakan salah satu teknik filterisasi untuk menghilangkan *noise* pada citra sidik jari menggunakan parameter frekuensi dan orientasi yang terdapat pada sidik jari.

Aplikasi ini mampu memperbaiki citra sidik jari pada orientasi sudut 0° , 45° , 90° , dan 135° dimana setelah dilakukan proses gabor filter akan terbentuk 4 citra baru hasil filterisasi, kemudian dilakukan operasi aritmatika untuk mendapatkan 1 citra akhir yang selanjutnya dapat diekstraksi titik ciri untuk proses identifikasi. Kondisi *ridge*, *valley*, dan *background* pada citra sangat mempengaruhi keberhasilan aplikasi agar proses filterisasi dapat mempertegas garis *ridge* dan menghilangkan *background* pada citra.

Kata kunci: *enhancement*, gabor filter, sidik jari.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Biometrika merupakan pengembangan dari metode dasar identifikasi yang menggunakan karakteristik alami manusia sebagai basisnya. Di dalam dunia teknologi informasi, biometrika relevan dengan teknologi yang digunakan untuk menganalisa fisik dan perilaku manusia. Salah satu bentuk karakteristik fisik yang dapat dipakai sebagai indikator dalam identifikasi personal adalah sidik jari.

Sidik jari adalah gurat-gurat yang terdapat di kulit ujung jari. Sidik jari memiliki karakteristik berupa gabungan dari pola bukit (*ridge*) dan lembah (*valley*). Sidik jari manusia digunakan untuk keperluan identifikasi karena tidak ada dua manusia yang memiliki sidik jari tepat sama. Pada saat ini sidik jari menjadi teknologi yang dirasa cukup akurat, aman, mudah, dan nyaman untuk dipakai sebagai identifikasi bila dibandingkan dengan sistem biometrik yang lainnya.

Sebelum sistem biometrika, pengenalan diri dilakukan dengan menggunakan metode konvensional seperti penggunaan PIN dan password. Penggunaan PIN atau password dapat menimbulkan beberapa permasalahan, antara lain adanya faktor lupa dan suatu waktu dapat diketahui oleh orang lain yang tidak berwenang. Kelemahan-kelemahan yang terdapat pada sistem tradisional tersebut dapat diatasi dengan penggunaan sistem biometrika. Sistem biometrika merupakan sistem pengenalan individu melalui ciri bagian tubuh atau perilakunya. Oleh sebab itu, sistem biometrika ini tidak dipengaruhi faktor lupa dan digunakan oleh orang lain. Salah satu biometrika yang sering digunakan dalam proses identifikasi yaitu sidik jari. Kendala utama pada proses pengenalan sidik jari yaitu terkadang citra dari sidik jari yang diambil memiliki kualitas yang rendah sehingga memerlukan *enhancement* pada citra agar tingkat kesalahan pada saat identifikasi berkurang.

Dalam tugas akhir ini akan digunakan teknik *enhancement* sidik jari berbasis gabor filter. Gabor filter ini melakukan pemfilteran pada citra sidik jari dengan memperjelas pola bukit (*ridge*) dan lembah (*valley*) pada citra sidik jari.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan secara jelas diatas, maka rumusan masalah ditekankan pada:

1. Bagaimana cara melakukan proses pengolahan input citra sidik jari sehingga untuk selanjutnya siap untuk diekstraksi dan direpresentasikan titik cirinya.
2. Bagaimana cara melakukan pemfilteran pada citra sidik jari menggunakan teknik gabor filter

1.3 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan program ini antara lain:

1. Citra sidik jari diambil dari hasil scanning sidik jari yang diberi tinta.
2. Ridge dan valley terlihat jelas dan tidak putus.
3. Tidak termasuk ekstraksi ciri sidik jari.
4. Orientasi dan Frekuensi lokal ridge sudah ditetapkan.

1.4 Tujuan

Tujuan tugas akhir (skripsi) ini adalah merancang dan membangun suatu aplikasi *enhancement* citra sidik jari sehingga untuk selanjutnya dapat diekstraksi titik cirinya.

1.5 Manfaat

Diharapkan manfaat yang dapat diperoleh melalui pengerjaan skripsi ini adalah :

a) Bagi Penyusun

1. Diharapkan dapat merancang aplikasi *enhancement* citra sidik jari menggunakan metode gabor filter menggunakan bahasa pemrograman C#.

2. Memperoleh pemahaman mengenai kelebihan serta kekurangan perangkat lunak yang telah dibuat.
3. Memahami bahwa sistem yang telah dibuat adalah hasil pengembangan daya pikir manusia, sebagai sumber daya terpenting dalam pembangunan sistem.
4. Menambah wawasan ilmu pengetahuan yang telah dipelajari sebelumnya, dan serta sebagai pelatihan berpikir kritis dalam menyelesaikan suatu masalah yang dihadapi.
5. Menerapkan ilmu yang telah didapat dalam perkuliahan.
6. Memberikan sumbangan bagi perkembangan ilmu pengetahuan khususnya dalam bidang *Information Technology (IT)*.

b) Bagi Pengguna

1. Memberikan kemudahan pengguna dalam melakukan perbaikan citra sidik jari agar selanjutnya dapat menghasilkan ekstraksi titik ciri yang benar pada citra sidik jari dalam proses identifikasi.

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini adalah sebagai berikut :

BAB I Pendahuluan

Menjelaskan latar belakang, rumusan masalah, ruang lingkup, tujuan dan sistematika penulisan skripsi

BAB II Dasar Teori

Menjelaskan kajian pustaka dan dasar teori yang digunakan

BAB III Metodologi

Menjelaskan metode yang digunakan dalam pengerjaan skripsi

BAB IV Perancangan dan Implementasi

Menjelaskan langkah langkah perancangan aplikasi sistem enhancement sidik jari beserta hasil implementasinya

BAB V Pengujian

Menjelaskan langkah-langkah pengujian dari sistem yang telah dibuat dan membahas hasil pengujiannya.

BAB VI Kesimpulan dan Saran

Berisi Kesimpulan dan Saran



BAB II

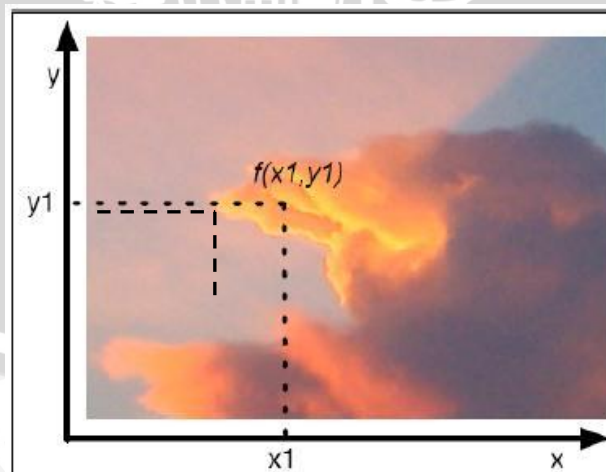
TINJAUAN PUSTAKA

2.1 Citra

Citra adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi *continue* dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi obyek, obyek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat – alat optik, misalnya mata pada manusia, kamera, *scanner*, dan sebagainya, sehingga bayangan objek yang disebut citra terekam.

2.2 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut (T Suyono dkk, 2009).. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*).

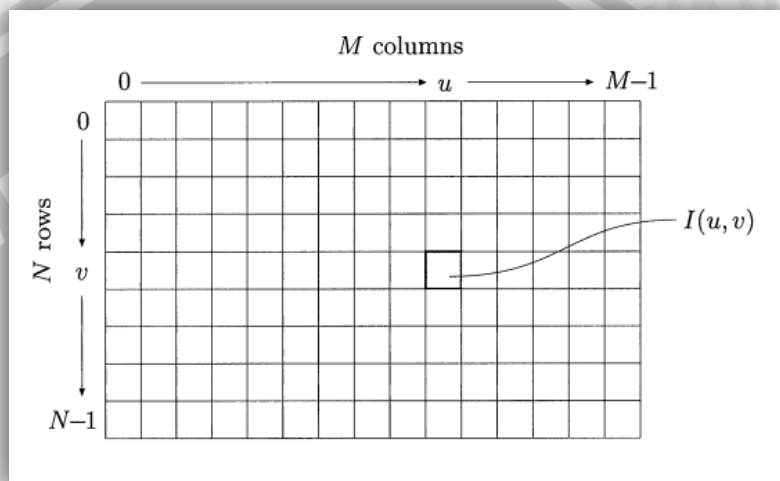


Gambar 2.1 Citra digital

Sumber : <http://digilib.itelkom.ac.id>

2.3 Pixel Dan Resolusi Pixel

Piksel merupakan satuan komponen terkecil yang menentukan ukuran dari suatu gambar (Gonzales & Woods, 2002). Resolusi citra menyatakan ukuran bit dari suatu citra dalam satuan piksel *image*. Untuk memudahkan, kita asumsikan dengan gambar segiempat dengan panjang M (dalam kolom) dan tinggi N (dalam baris) dari matrix I , seperti pada Gambar 2.2.



Gambar 2.2 Image size and resolution

Sumber: Wilhem Burger., Mark James Burger. Digital Images Processing. 2008

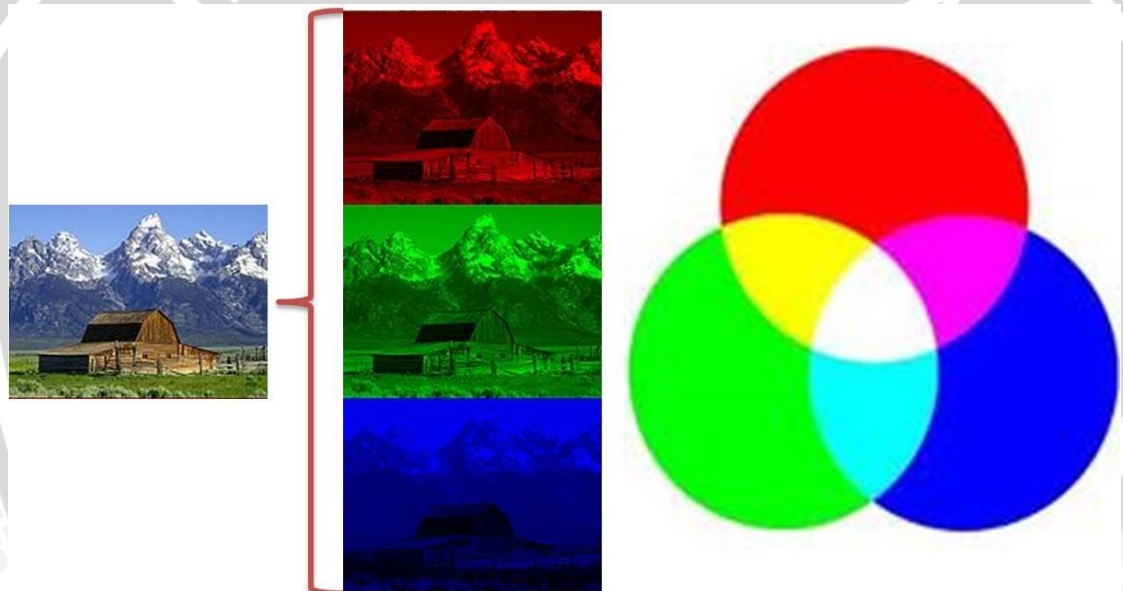
Banyaknya piksel persatuan panjang dikenal dengan satuan *dot per inch* (dpi). Semakin banyak titik per incinya, kualitas citra semakin baik dan detail, namun semakin banyak pula memori (bit) yang diperlukan untuk menyimpan citra tersebut.

2.4 Jenis - Jenis Citra

Informasi penting dari suatu citra adalah informasi mengenai intensitas cahaya yang tertangkap, ini erat hubungannya dengan warna. Suatu citra digital dapat dikelompokkan berdasarkan format intensitas penyusunnya yaitu citra berwarna, citra skala keabuan (*grayscale*), dan citra biner (*monokrom*).

2.4.1 Citra Berwarna

Citra berwarna merupakan kombinasi dari tiga warna dasar yaitu merah, hijau, dan biru. Citra ini biasanya disebut citra RGB. Kombinasi dari tiga warna RGB menghasilkan warna yang khas untuk piksel yang bersangkutan. Setiap komponen warna memiliki intensitas sendiri dengan nilai minimum 0 dan nilai maksimal 225 dan memiliki panjang 8 bit, hal ini menyebabkan setiap pixel pada citra RGB membutuhkan 3 *byte* di dalam media penyimpanan. Pada citra RGB mempunyai range warna $(2^8)(2^8)(2^8) = 2^{24}$, sehingga RGB mempunyai 16.777.216 kombinasi warna. Dalam windows citra berwarna dikenal dengan istilah *true color*.



Gambar 2.3 Citra berwarna

Sumber : <http://digilib.ittelkom.ac.id>

2.4.2 Citra Grayscale

Citra *grayscale* merupakan citra digital dimana unsur RGB (Red Green Blue) hanya digunakan untuk menentukan tingkat intensitas pada suatu citra. Pada citra *grayscale* warna yang dimiliki adalah hitam, keabuan, putih. Tingkat

keabuan merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati warna putih. Pada citra grayscale 8 bit, masing – masing komponen memiliki nilai minimum 0 dan nilai maksimal 255 dan memiliki panjang 8 bit. Untuk mendapatkan citra *grayscale* dari citra warna dapat dicari menggunakan rumus berikut :

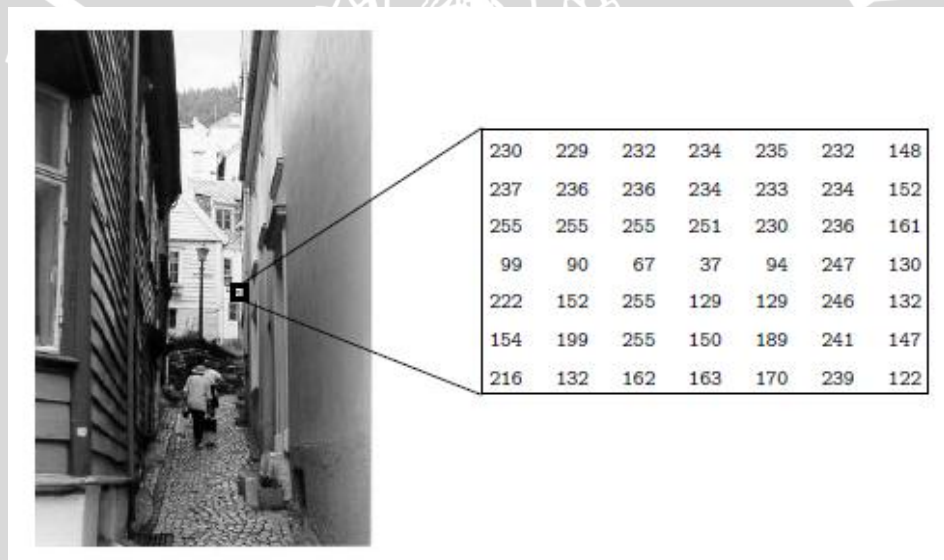
$$Y = RGB = grayscale = (wr \times red) + (wg \times green) + (wb \times blue)$$

dimana w_r , w_b , dan w_g bernilai :

$$w_r = 0,299$$

$$w_b = 0,587$$

$$w_g = 0,114$$



Gambar 2.4 Citra Grayscale

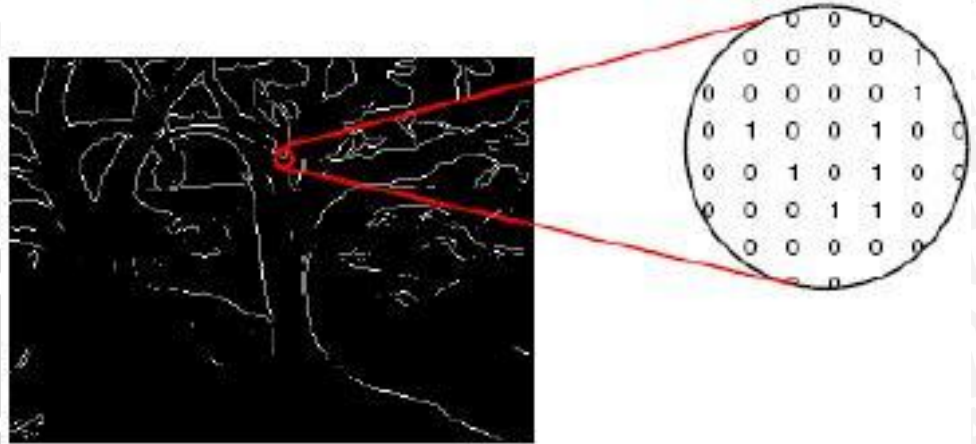
Sumber : Alasdair McAndrew. An Introduction to Digital Image

Processing with Matlab. 2004

2.4.3 Citra Biner

Citra biner disebut juga citra monokrom hanya memiliki dua nilai yaitu 1 atau 0 yang jika di implementasikan kedalam 8-bit nilai yang digunakan adalah

0 dan 225. Biasanya citra biner ini digunakan pada pengolahan citra seperti segmentasi.



Gambar 2.5 Citra Biner

Sumber : <http://digilib.itelkom.ac.id>

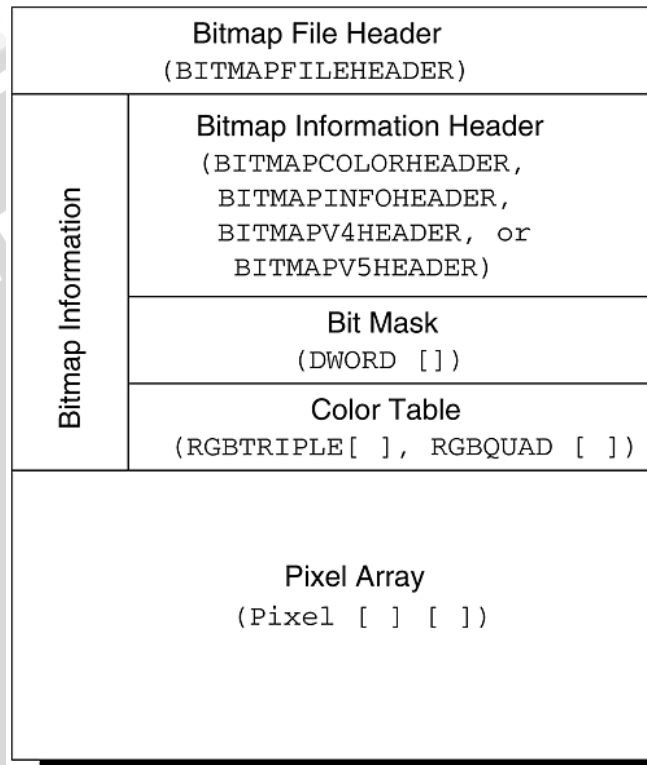
2.5 *Bitmap*

Bitmap atau raster merupakan gambar digital yang direpresentasikan dalam bentuk matriks atau dipetakan dengan menggunakan bilangan biner ataupun bilangan lain. Citra ini memiliki kelebihan untuk memanipulasi warna dari sebuah gambar serta menyimpan data kode citra secara digital dan lengkap (cara penyimpanan per piksel). Gambar *bitmap* dengan resolusi (jumlah piksel setiap satuan ukur) besar, akan terlihat lebih halus dibandingkan dengan yang memiliki resolusi rendah. Resolusi gambar *bitmap* dinyatakan dalam satuan *dot per inch* (dpi) atau *pixel per inch* (ppi).

2.5.1 Struktur File *Bitmap*

Struktur penyimpanan file *bitmap* pada umumnya terdiri dari 4 blok data yaitu: *BMP header*, *Bit Information (DIB header)*, *Color Palette*, dan *Bitmap Data*. *BMP header* berisi informasi umum dari citra *bitmap*. Blok ini berada pada bagian awal file citra dan digunakan untuk mengidentifikasi citra. Beberapa aplikasi pengolah citra akan membaca blok ini untuk memastikan bahwa citra

tersebut berformat *bitmap* dan tidak dalam kondisi rusak. Bit information berisi informasi detail dari citra *bitmap*, yang akan digunakan untuk menampilkan citra pada layar. *Color palette* berisi informasi warna yang digunakan untuk indeks warna *bitmap*, dan *bitmap data* berisi data citra yang sebenarnya, piksel per piksel. Model ruang warna yang digunakan pada citra *bitmap* adalah RGB (red, green, dan blue).



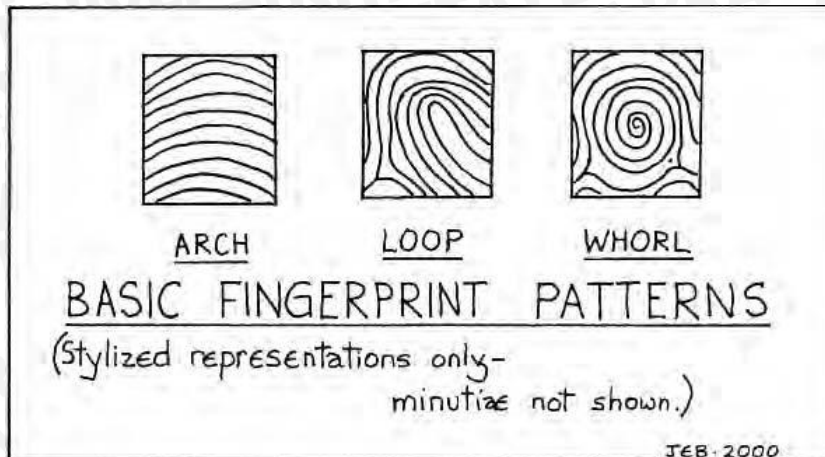
Gambar 2.6 Format file *bitmap* umum

Sumber: Feng Yuan, Windows Graphics Programming Win 32. 2000

2.6 Sidik Jari

Sidik jari adalah suatu bentuk pola garis (ridge) pada permukaan sebuah ujung jari. Sebuah sidik jari berkualitas baik dapat dibedakan berdasarkan pola – pola dan ciri – ciri (feature) yang menyediakan ekstraksi feature yang bermanfaat untuk pencocokan sidik jari.

Berdasarkan pola garis (ridge) dan lembah (valley), sidik jari dapat dibedakan menjadi tiga kelas utama yaitu, : *Arch*, *Loop*, dan *Whorl*.



Gambar 2.7 Klasifikasi sidik jari

Sumber : Ruud Bolle. Automatic Fingerprint Recognition Systems. 2004

Menurut Prabhakar Dari klasifikasi ini dapat dibagi menjadi beberapa subklasifikasi, yaitu :

1. *Arch* merupakan bentuk pokok sidik jari yang semua garis – garisnya datang dari satu sisi lukisan, mengalir atau cenderung mengalir ke sisi yang lain dari lukisan itu, dengan bergelombang naik ditengah – tengah, *Arch* terdiri dari :
 - a. *Plain Arch* adalah bentuk pokok sidik jari dimana garis – garis datang dari sisi lukisan yang satu mengalir ke arah sisi yang lain, dengan sedikit bergelombang naik ditengah.
 - b. *Tented arch* adalah bentuk pokok sidik jari yang memiliki garis tegak (*upthrust*), sudut(*angel*) atau dua dan tiga ketentuan *loop*.

Dari beberapa populasi *arch* mempunyai presentasi sebesar 5%.

2. *Loop* adalah bentuk pokok sidik jari dimana satu garis atau lebih datang dari satu sisi lukisan, mlereng, menyentuh atau melintasi suatu garis bayangan yang ditarik antara *delta* dan *core*, berhenti atau cenderung berhenti kearah sisi semula. *Loop* dibagi menjadi *left loop*, *right loop*, dan *double loop*. Berbeda dengan *arch*, jumlah individu yang mempunyai klasifikasi *loop* sangat besar yaitu sebesar 60%.

3. *Whorl* adalah bentuk pokok sidik jari, mempunyai dua delta dan setidaknya satu garis melingkar di dalam pattern area, terletak diantara kedua delta. Pada klasifikasi ini jumlah presentasi individu sebesar 35%.

2.7 Citra Enhancement

Definisi Citra *Enhancement* adalah berasal dari kata *enhancement* yang artinya mempertinggi atau meningkatkan, jadi citra *enhancement* memiliki arti secara utuh adalah mempertinggi atau meningkatkan kualitas citra dengan metode - metode tertentu. Proses – proses dalam citra *enhancement* terdiri dari sekumpulan teknik yang dilakukan untuk meningkatkan tampilan visual dari sebuah citra, atau untuk mengkonversikan citra ke dalam bentuk yang lebih cocok untuk analisa mata manusia atau analisa mesin. Citra *Enhancement* digunakan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter – parameter citra. Dengan cara ini, maka ciri – ciri khusus yang terdapat di dalam citra dapat ditonjolkan.

2.7.1 Segmentasi

Tahap awal dari proses *enhancement* citra sidik jari adalah segmentasi citra. Segmentasi bertujuan untuk membagi suatu citra menjadi beberapa daerah (*region*). Segmentasi memiliki tujuan menemukan karakteristik khusus yang dimiliki oleh suatu citra. Salah satu teknik yang sering digunakan untuk segmentasi adalah teknik pengambangan (*Thresholding*). Proses *thresholding* akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Citra hasil *thresholding* dapat didefinisikan sebagaimana persamaan dibawah ini:

$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) \geq T \\ 0 & \text{jika } f(x, y) < T \end{cases} \quad (2.1)$$

Dengan $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peranan penting dalam proses pengambangan. Kualitas hasil citra biner sangat bergantung pada nilai T yang digunakan. Jika di implementasikan pada citra 8 – bit nilai yang digunakan adalah 0(hitam) dan 255(putih).

2.7.1.1 Metode Otsu

Metode Otsu merupakan metode yang digunakan untuk menghitung nilai ambang (T) secara otomatis berdasarkan citra masukan. Tujuan dari metode otsu adalah membagi histogram citra gray level ke dalam dua daerah yang berbeda secara otomatis tanpa membutuhkan bantuan user untuk memasukkan nilai ambang. Pendekatan yang dilakukan oleh metode otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis Diskriminan akan memaksimalkan variabel tersebut agar dapat membagi objek dengan latar belakang.

Nilai ambang yang dicari pada suatu citra *grayscale* dilambangkan dengan k . Nilai k ini berkisar antara 1 sampai dengan L sedangkan L memiliki nilai maksimum 255. Probabilitas untuk pixel i dapat dinyatakan dengan persamaan :

$$p_i = \frac{n_i}{N} \quad (2.2)$$

Dengan n_i menyatakan jumlah pixel dengan tingkat keabuan l dan N menyatakan banyaknya pixel pada citra.

Pada nilai momen kumulatif ke nol, momen kumulatif ke satu, dan nilai rata-rata dapat dinyatakan pada persamaan di bawah ini :

$$\omega(k) = \sum_{i=1}^k p_i \quad (2.3)$$

$$\mu(k) = \sum_{i=1}^k i \cdot p_i \quad (2.4)$$

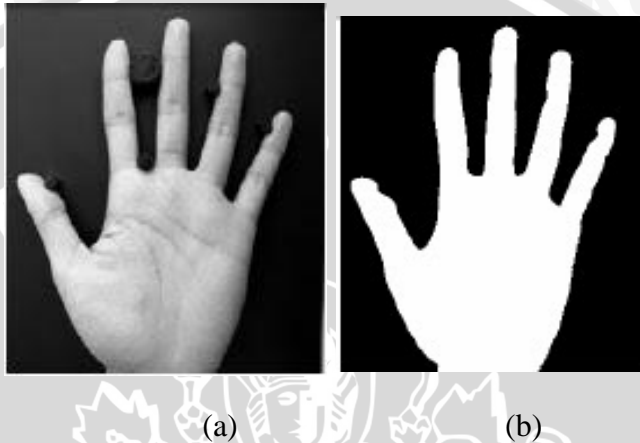
$$\mu_T(k) = \sum_{i=1}^L i \cdot p_i \quad (2.5)$$

Sedangkan untuk nilai ambang k dapat ditentukan dengan memaksimumkan persamaan :

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \sigma_B^2(k) \quad (2.6)$$

dengan :

$$\sigma_B^2(k^*) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (2.7)$$



Gambar 2.8 Metode Otsu : (a) Citra grayscale (b) Citra hasil metode otsu

Sumber : <http://primaliaprima.blogspot.com>

2.8 Gabor Filter

Gabor filter merupakan salah satu metode pada proses perbaikan citra. Gabor filter merupakan bagian dari metode filtering domain spasial, pengertian domain spasial yaitu merujuk pada ruang dari citra. Metode gabor filter ini pada prinsipnya diterapkan pada suatu citra sidik jari yang ternormalisasikan, dikonvolusi dengan n buah filter gabor dengan arah orientasi yang berbeda. Sehingga pada akhirnya terbentuk n buah citra baru terfilterisasi. Dimana masing – masing arah orientasi dibentuk dari rumus berikut:

$$G(x, y, f, \theta) = \exp \left\{ \frac{-1}{2} \left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right) \right\} \cos(2 \pi f x') \quad (2.8)$$

$$x' = x \sin \theta + y \cos \theta \quad (2.9)$$

$$y' = x \cos \theta - y \sin \theta \quad (2.10)$$

Dimana θ adalah orientasi gabor filter, f adalah frekuensi spatial dari ridge,serta δ_x dan δ_y adalah konstanta dari Gaussian. Frekuensi spatial yang digunakan yaitu $(1 / K)$ dimana $k = 10$. Pada algoritma originalnya Hong. dkk, memberikan nilai tetapan untuk masing – masing δ_x dan δ_y yaitu 4.00 dan 4.00.

Untuk mendapatkan citra *enhancement* digunakan penerapan gabor filter G pada persamaan berikut:

$$E(i, j) = \sum_{u=-\frac{W_x}{2}}^{\frac{W_x}{2}} \sum_{v=-\frac{W_y}{2}}^{\frac{W_y}{2}} G(u, v, O(i, j), F(i, j))N(i - u, j - v) \quad (2.11)$$

Dimana $O(i,j)$ adalah orientasi citra, $F(i,j)$ adalah frekuensi ridge citra, N adalah citra sidik jari ternormalisasi, W_x dan W_y adalah lebar dan tinggi dari mask gabor filter.



Gambar 2.9 Citra hasil gabor filter

Sumber : Raymond Thai.

2.9 Operasi Aritmatika Citra

Image arithmetic atau operasi aritmatika citra merupakan proses pengolahan citra memanfaatkan operator-operator aritmatika yang ada terhadap dua buah citra *input* atau lebih. Proses operasi aritmatika citra dijalankan dengan melakukan pengolahan pada pixel per pixel. Setiap nilai pixel dari citra *output* bergantung pada nilai pixel citra *input* pada posisi yang sama. Oleh karena itu pada operasi arimatika citra ini lebih baik menggunakan resolusi citra yang sama agar didapatkan hasil yang baik.

Pada operasi aritmatika citra juga dapat menggunakan sebuah nilai konstanta untuk diproses ke seluruh pixel citra *input*. Macam-macam operasi arimatika pada citra :

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Invert

2.9.1 Operasi Penjumlahan

Pada operasi penjumlahan ini citra *output* dihasilkan dari pengurangan citra *input* yang pertama di jumlah dengan citra *input* yang berikutnya. Persamaan operasi penjumlahan dapat dituliskan seperti di bawah ini :

$$o(x, y) = u_1(x, y) + u_2(x, y) \quad (2.15)$$

Selain digunakan untuk mengurangi citra dengan citra operasi penjumlahan juga dapat dilakukan dengan proses pengurangan citra dengan sebuah konstanta dimana konstanta ini akan mengurangi setiap pixel pada citra dan persamaannya seperti di bawah ini :

$$o(x, y) = u(x, y) + K \quad (2.16)$$

2.9.2 Operasi Invert

Invers atau logika NOT pada pengolahan citra akan memberikan keluaran hasil berupa kebalikan dari citra input. Logika NOT dapat dilihat dari perbedaan antara AND dengan NAND, atau OR dengan NOR Kebalikan yang ditunjukkan oleh masing – masing contoh diatas erupakan kebalikan NOT. Penggunaan dalam citra biner sangat sederhana. Apabila citra input bernilai 1 maka keluaran akan bernilai 0, sebaliknya input bernilai 0 akan menghasilkan keluaran bernilai 1. Jika diimplementasikan pada citra 8 bit maka, input citra bernila 0 maka akan menghasilkan keluaran bernilai 255, sebaliknya apabila inpuy bernilai 255 maka akan menghasilkan keluaran bernilai 0.

Penggunaan NOT pada citra berwarna akan menghasilkan bentuk negatif dari citra tersebut. Rumus yang digunakan dalam operasi negatif untuk citra berwarna adalah :

$$o(x,y) = 255 - u(x,y) \quad (2.17)$$



BAB III

METODE PENELITIAN

Dalam penyusunan skripsi ini, dirancang suatu Perancangan Aplikasi sistem perbaikan sidik jari menggunakan metode gabor filter. Adapun langkah-langkah yang akan dikerjakan meliputi sebagai berikut :

1. Studi Literatur
2. Penentuan Spesifikasi Alat
3. Kofigurasi Sistem
4. Pengujian dan Analisis sistem
5. Kesimpulan dan Saran

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Membaca dan mempelajari buku – buku yang berhubungan dengan *image processing* dan *biometrics*.
2. Mempelajari metode *enhancement* citra sidik jari.
3. Mempelajari teknik – teknik dasar pemrograman dengan menggunakan Microsoft Visual Studio .NET

3.2 Penentuan Spesifikasi Alat

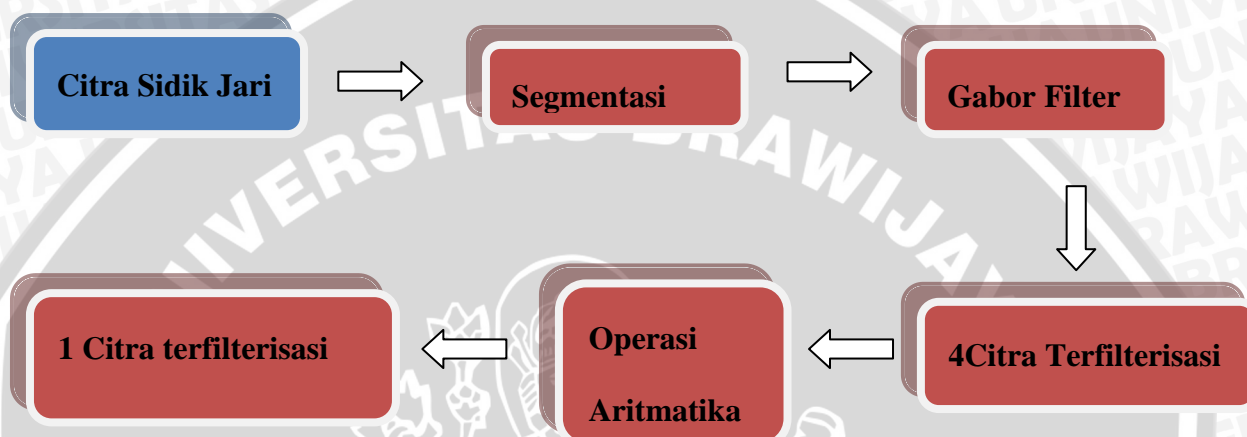
Untuk menentukan perangkat yang akan digunakan untuk menunjang proses pembuatan aplikasi sistem perbaikan sidik jari menggunakan metode gabor filter, antara lain:

- a. Perangkat Keras :
 1. 1 unit laptop dengan spesifikasi Intel(R) Core(TM) 2 Duo CPU, VGA NVIDIA GeForce G 105M Total Memory 2288 MB.
 2. Scanner canon mp 160.

- b. Perangkat Lunak :
1. *Operating system* Windows 7
 2. *Microsoft Visual Studio 2010.NET*

3.3 Konfigurasi Sistem

Sistem *enhancement* sidik jari ini terdiri dari beberapa komponen yang dapat digambarkan dengan konfigurasi seperti berikut :



Gambar 3.1. Diagram Blok sistem

Keterangan:

1. Citra Sidik Jari : Citra yang akan diamati didalam sistem.
2. Segmentasi : Proses pemisahan bagian wilayah dalam frame yang mengandung *ridges* dan *valleys* dari wilayah latar belakang.
3. Gabor Filter : Gabor filter merupakan filtering menggunakan *local ridge orientation* dan *ridge frequency* untuk menghilangkan noise pada citra.
4. 4 Citra Terfilterisasi : Citra output hasil proses gabor filter dengan 4 orientasi berbeda
5. Operasi Aritmatika : proses ini dilakukan untuk menggabungkan 4 citra terfilterisasi menjadi 1 citra terfilterisasi
6. 1 Citra Terfilterisasi : Citra output hasil operasi aritmatika.

3.4 Pengujian dan Analisis Sistem

Pada tahap ini pengujian menggunakan 10 citra yang akan di uji hal ini dilakukan untuk memastikan bahwa sistem yang telah dirancang memiliki tingkat kesalahan atau error yang sangat kecil dan dengan akurasi yang tinggi. Agar dapat diketahui apakah sistem yang telah dirancang dapat bekerja dengan baik sesuai dengan perancangan yang telah dibuat sebelumnya maka akan dilakukan beberapa pengujian yang berangkaian atau berurutan. Tahapan rangkaian pengujian adalah sebagai berikut :

1. Pengujian gabor filter dengan 4 orientasi berbeda.
2. Pengujian operasi aritmatika untuk mendapatkan 1 citra terfilterisasi.

3.5 Kesimpulan Dan Saran

Pada bagian ini akan dilakukan penarikan kesimpulan yang dihasilkan dari hasil pengujian dan analisis terhadap aplikasi *enhancement* citra sidik jari menggunakan metode gabor filter. Tahapan selanjutnya adalah membuat saran yang berguna sebagai masukan pada penelitian untuk menyempurnakan kekurangan yang ada sehingga di dapat perbaikan sistem yang lebih baik pada penelitian berikutnya.



BAB IV

PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan menjelaskan perancangan dan implementasi aplikasi perbaikan citra sidik jari dan terdapat beberapa tahap. Tahap pertama merupakan tahap pre proses dimana proses ini untuk menyiapkan citra yang akan diolah dan tahap kedua merupakan proses perbaikan citra sidik jari.

4.1 Perancangan Secara Umum

Tahap awal yang dibutuhkan adalah membuat perancangan aplikasi secara global dimana tahap awal ini berfungsi sebagai acuan dalam proses pembuatan aplikasi yang akan dibuat. Perancangan ini didahului dengan pendefinisian kegiatan pelaku atau *user* dalam menggunakan program perbaikan citra sidik jari dengan pengolahan citra serta perangkat yang digunakan meliputi blok sistem diagram dan cara kerja aplikasi.

4.1.1 Blok Sistem Diagram

Pada sistem perbaikan citra sidik jari terdiri dari beberapa langkah yang dapat digambarkan menjadi blok diagram dengan model seperti pada gambar 4.1.



Gambar 4.1 Diagram Blok Sistem Secara Keseluruhan

Sumber : Perancangan

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

1. Citra sidik jari digunakan sebagai input sistem.
2. Melakukan image processing pada citra sidik jari.
3. Citra sidik jari yang telah melewati proses pengolahan citra pada sistem akan menghasilkan suatu hasil citra sidik jari yang telah terfilterisasi untuk selanjutnya dapat diproses untuk pendeteksian titik minutiae pada citra sidik jari.

4.1.2 Cara Kerja Aplikasi

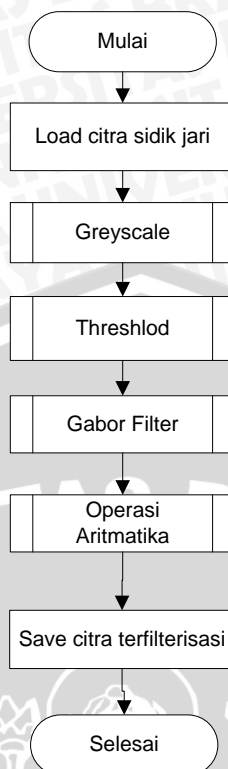
Aplikasi perbaikan citra sidik jari menggunakan metode gabor filter memiliki cara kerja yang dimulai dengan pengambilan citra sidik jari menggunakan scanner. Teknik pengambilan citra yang digunakan yaitu user meletakkan sidik jari yang telah diberi tinta pada sebuah kertas, kemudian kertas tersebut di scan menggunakan scanner yang telah di atur pada 600Dpi. Selanjutnya me-load citra yang kemudian dilakukan proses pengolahan citra dan proses filter menggunakan metode gabor filter.

4.2 Perancangan Perangkat Lunak

Pada bagian perancangan ini perangkat lunak yang akan dibuat menggunakan bahasa pemrograman *Microsoft Visual Studio C#.NET 2008* dan sistem yang digunakan untuk membangun perangkat lunak ini dirancang dengan spesifikasi mampu melakukan hal-hal sebagai berikut :

1. Mengakses data citra yang telah tersimpan di dalam harddisk komputer.
2. Melakukan proses perubahan dari citra berwarna menjadi citra *grayscale*.
3. Melakukan proses threshold pada citra secara otomatis.
4. Melakukan proses filterisasi pada objek.

Sedangkan untuk detail desain aplikasi secara umum akan ditunjukkan pada gambar 4.2.

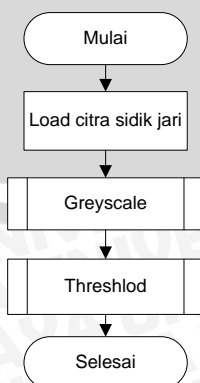


Gambar 4.2 Detail Desain Aplikasi

Sumber : Perancangan

4.2.1 Perancangan *Preprocess*

Citra sidik jari yang telah di-capture menggunakan scanner dan disimpan di dalam harddisk komputer di load, kemudian citra tersebut diubah menjadi citra grayscale. Setelah itu dilakukan proses pencarian nilai threshold untuk merubah citra grayscale menjadi citra biner. Flowchart perancangan *preprocess* seperti pada gambar 4.3.

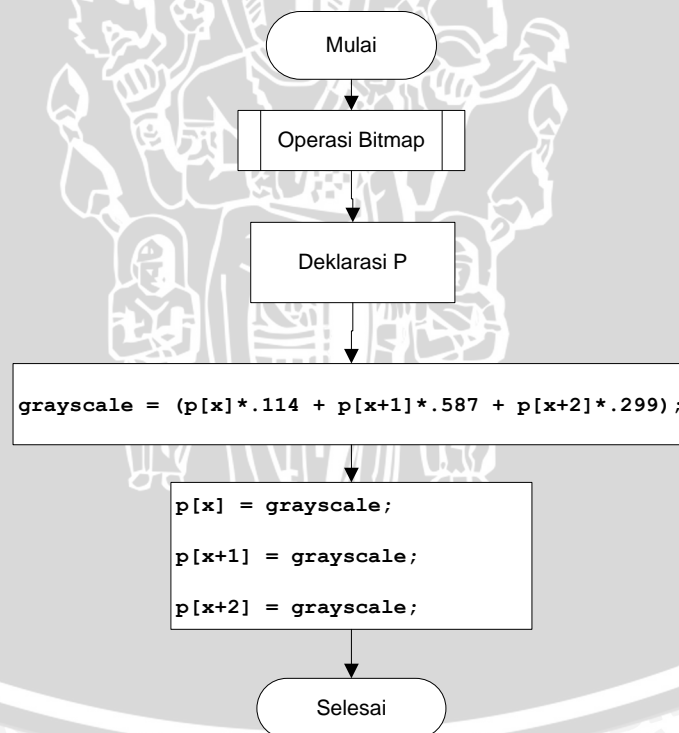
Gambar 4.3 Flowchart *Preprocess*

Sumber : Perancangan

Berikut penjelasan flowchart preprocess :

1. Hasil citra yang akan diproses merupakan citra berwarna oleh karena itu untuk mempermudah proses selanjutnya maka citra berwarna RGB dikonversi menjadi citra grayscale.
2. Setelah proses konversi citra berwarna menjadi citra grayscale selanjutnya dilakukan proses segmentasi dimana proses ini dilakukan untuk merubah citra grayscale menjadi citra biner. Proses binerisasi pada citra ini dilakukan dengan mendapatkan nilai batas ambang(threshold) pada citra grayscale secara otomatis menggunakan yaitu metode otsu.

4.2.1.1 Perancangan Sub Proses Bagian Konversi Grayscale



Gambar 4.4 Flowchart proses bagian konversi grayscale

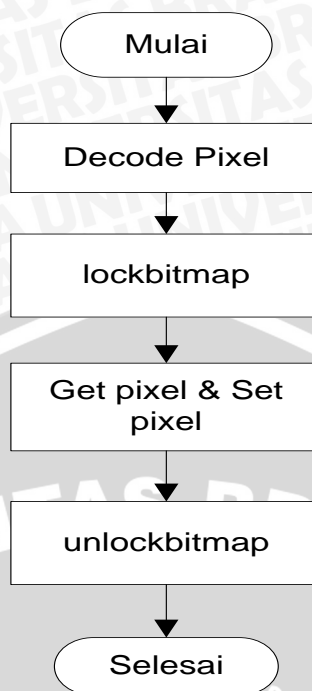
Sumber : Perancangan

Penjelasan flowchart sebagai berikut :

1. Melakukan operasi *bitmap* untuk untuk memanipulasi array *bitmap* piksel secara langsung pada memori.
2. Deklarasi P dimana nilai yang ditunjuk oleh P yaitu alamat dari elemen pertama yaitu nilai pixel B yang di ikuti nilai pixel G dan nilai pixel R
3. Setelah mendapatkan nilai pixel pada kanal RGB kemudian dikalikan dengan koefisien untuk merubah dari nilai RGB menjadi grayscale, dimana nilai $B * 0.114$, $G * 0.587$, $R * 0.299$.
4. Mendapatkan nilai grayscale dari hasil penjumlahan nilai RGB setelah dikalikan dengan koefisiennya.
5. Set nilai grayscale pada masing - masing index untuk mendapatkan citra grayscale.

4.2.1.1.1 Perancangan Sub Pengolahan Citra Bagian Operasi *Bitmap*

Class *bitmap* pada GDI+ menyediakan fitur method *lockBits* dan *unlockBit* yang memungkinkan untuk memanipulasi sebagian dari array *bitmap* piksel secara langsung pada memori, akses langsung ini dapat berupa seleksi piksel atau mengganti bit-bit pada *bitmap* dengan data yang sudah dimodifikasi. Flowchart operasi *bitmap* seperti pada Gambar 4.7.



Gambar 4.5 Flowchart operasi *bitmap*

Sumber : Perancangan

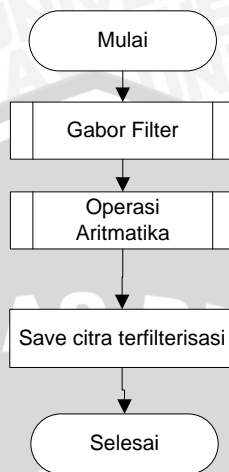
Penjelasan flowchart sebagai berikut:

1. Program melakukan decode piksel, height width beserta nilai pada kanal RGB yang digunakan mengakses piksel secara langsung melalui memori.
2. Setelah proses decode, jalankan fungsi *lockbitmap* untuk mendapatkan lebar dan alamat penyimpanan pada memori guna proses seleksi piksel. Setelah itu lakukan akses piksel dengan menggunakan fungsi set dan get piksel
3. Setelah pengaksesan piksel sampai selesai jalankan fungsi *unlockbitmap*

4.2.2 Perancangan Proses

Setelah citra melewati *preprocess* maka dilanjutkan pemrosesan citra untuk menghasilkan citra terfilterisasi. Proses yang dilakukan pada tahap ini yaitu citra yang telah dibinerisasikan diteruskan dengan menggunakan metode gabor filter untuk mendapatkan 4 citra terfilterisasi dengan sudut orientasi

0°, 45°, 90°, 135°. Setelah mendapatkan 4 citra terfilterisasi dilakukan proses pengurangan dan penjumlahan citra untuk mendapatkan satu citra hasil image processing.



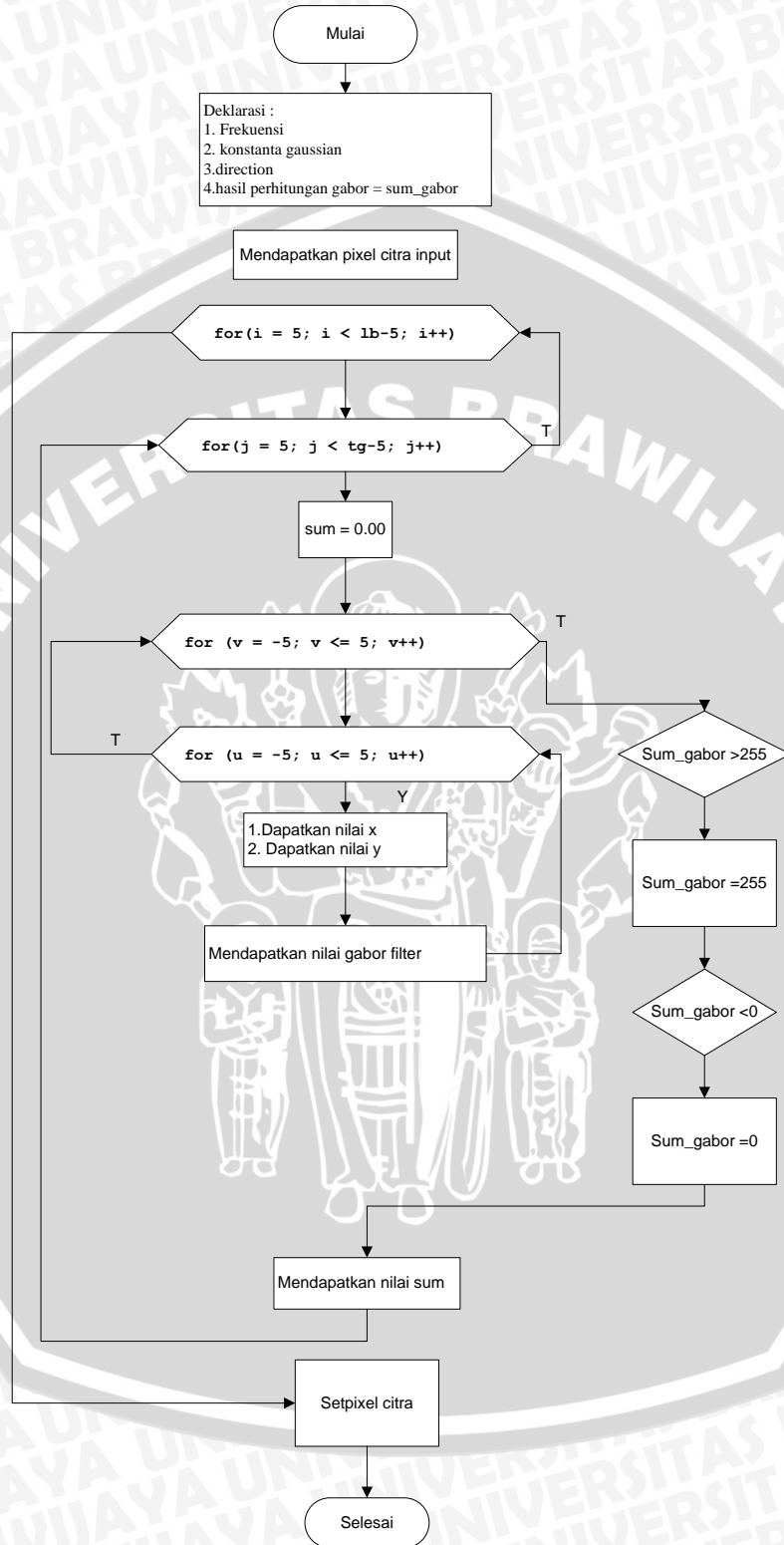
Gambar 4.6 Flowchart *process*

Sumber : Perancangan

Penjelasan flowchart sebagai berikut :

1. Citra hasil segmentasi menggunakan metode otsu digunakan sebagai citra input. Selanjutnya dilakukan proses gabor filter. Proses ini akan dilakukan sebanyak 4 kali dengan sudut orientasi yang berbeda untuk mendapatkan 4 citra hasil filterisasi.
2. Hasil yang diperoleh dari proses gabor filter ini selanjutnya akan dilakukan proses aritmatika yaitu proses pengurangan image dan penjumlahan image untuk mendapatkan satu citra hasil filterisasi.
3. Citra hasil proses pengurangan image dan penjumlahan image merupakan citra akhir yang kemudian disimpan sebagai hasil.

4.2.2.1 Perancangan Sub Proses Bagian Gabor Filter

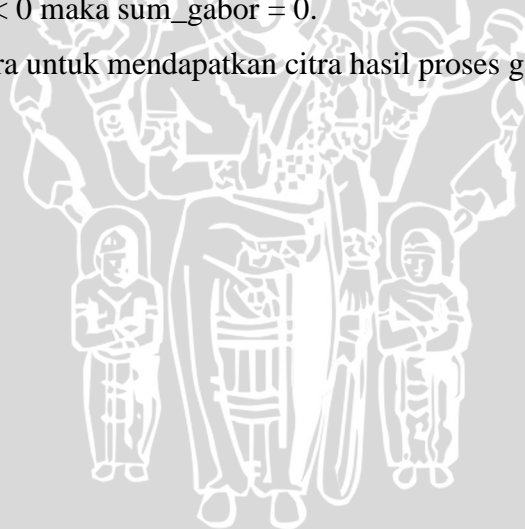


Gambar 4.7 Flowchart proses gabor filter

Sumber : Perancangan

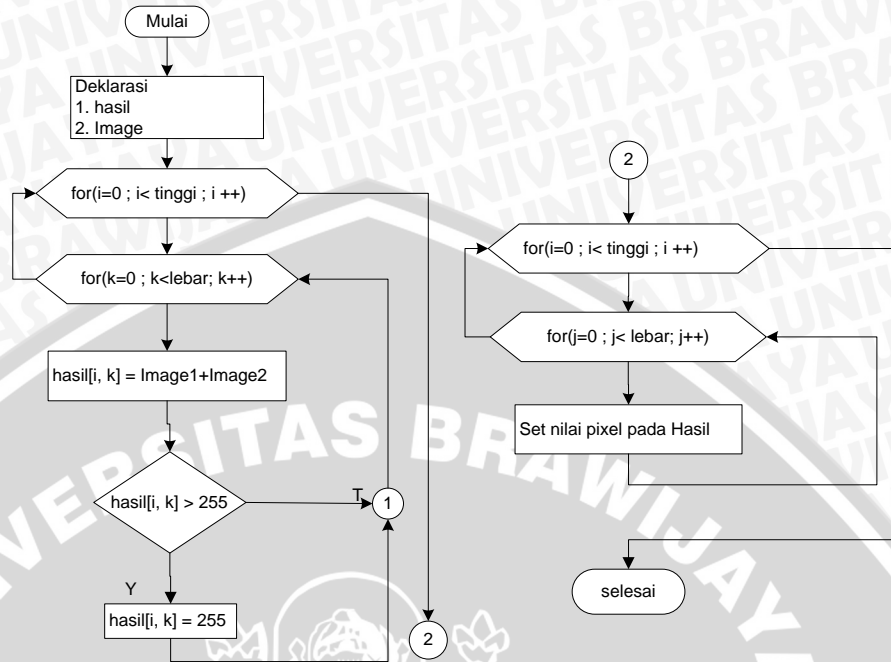
Penjelasan flowchart sebagai berikut :

1. Deklarasi variabel untuk Frekuensi, direction, dan konstanta gaussian yang digunakan sebagai parameter untuk perhitungan gabor filter. Deklarasi hasil perhitungan gabor filter.
2. Lakukan iterasi sepanjang lebar x tinggi citra, dimana lebar dimulai dari 5 hingga lebar - 5 dan tinggi dimulai dari 5 hingga tinggi - 5.
3. Deklarasi $sum_gabor = 0.00$.
4. Lakukan iterasi sepanjang 11 x 11 untuk pemfilteran menggunakan metode gabor filter.
5. Melakukan perhitungan dengan memasukkan frekuensi, direction dan konstanta gaussian untuk mendapatkan hasil penjumlahan gabor filter.
6. Mendapatkan nilai total dari penjumlahan sum_gabor . Apabila nilai total yang didapatkan lebih dari 255 maka akan dilakukan pembatasan nilai, dimana $sum_gabor > 255$ maka $sum_gabor = 255$ sedangkan $sum_gabor < 0$ maka $sum_gabor = 0$.
7. Set pixel citra untuk mendapatkan citra hasil proses gabor filter



4.2.2.2

Perancangan Sub Proses Bagian Penjumlahan Image



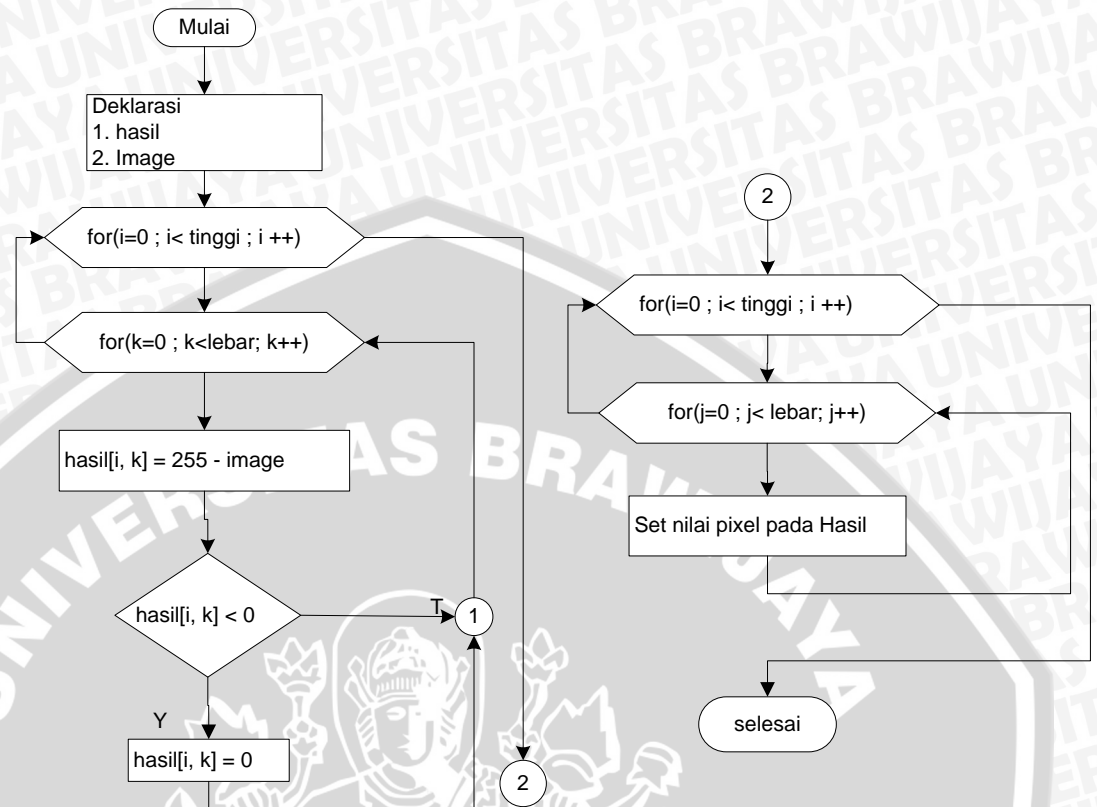
Gambar 4.8 Flowchart proses Penjumlahan Image

Sumber : Perancangan

Penjelasan flowchart sebagai berikut :

1. Deklarasi variabel untuk hasil dan citra yang akan di proses.
2. Lakukan iterasi berdasarkan panjang x lebar dari citra. Ukuran citra yang akan diproses harus sama.
3. Jumlahkan piksel dari masing – masing citra
4. Apabila hasil penjumlahan lebih besar dari 255 maka akan dibuat menjadi nilai 255 agar tidak menghasilkan nilai diluar range citra.
5. Lakukan iterasi berdasarkan panjang x lebar dari citra. Kemudian lakukan set pixel untuk mendapatkan citra hasil proses penjumlahan image.

4.2.2.3 Perancangan Sub Proses Bagian Invert Image



Gambar 4.9 Flowchart proses invert image

Penjelasan flowchart sebagai berikut :

1. Deklarasi variabel untuk hasil dan citra yang akan di proses.
2. Lakukan iterasi berdasarkan panjang x lebar dari citra. Ukuran citra yang akan diproses harus sama.
3. Hasil inverter merupakan hasil pengurangan dari 255 – (piksel citra)
4. Apabila hasil invert lebih kecil dari 0 maka hasil inverter akan dibuat menjadi nilai 0 agar tidak menghasilkan nilai negatif.
5. Lakukan iterasi berdasarkan panjang x lebar dari citra. Kemudian lakukan set pixel untuk mendapatkan citra hasil proses penjumlahan image.

4.3 Implementasi Sistem

Setelah tahap perancangan, selanjutnya adalah tahap implementasi. Tahap ini merupakan proses transformasi hasil perancangan perangkat lunak yang telah

dibuat ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan.

4.3.1 Lingkungan Implementasi

Aplikasi dibuat dengan menggunakan Microsoft Visual Studio C#. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

1. Perangkat keras (Komputer)

Spesifikasi :

Processor : Intel Core 2 Duo processor T6600 2,20GHz

Memory : 512 MB DDR2

OS : Windows 7

VGA : NVIDIA GeForce G 105M

2. Perangkat Lunak :

- Sistem operasi : Microsoft Windows 7
- Bahasa pemrograman : Microsoft Visual Studio C#.NET 2010

4.3.2 Implementasi Algoritma

4.3.2.1 Proses *Grayscale*

Citra yang akan diproses pada aplikasi ini menggunakan citra grayscale oleh karena itu citra input pada aplikasi ini diubah menjadi citra grayscale terlebih dahulu. Proses *grayscale* ini dilakukan dengan iterasi dimana syarat iterasi menggunakan lebar dan panjang citra berikut listing program dalam syntax C#.

1. Melakukan *lock image* format piksel 24 bpp *bitmap*

```
Rectangle rect = new Rectangle(0, 0, bmp.Width, bmp.Height);
```

```
BitmapData bmData =
```

```
bmp.LockBits(rect, ImageLockMode.ReadWrite, PixelFormat.Format24
```

```
bppRgb);
```

2. Persamaan untuk konversi citra menjadi citra grayscale

```
byte grayscale = (byte) (p[x] * .114 + p[x + 1] * .587 + p[x + 2] * .299);
```

3. Menerapkan hasil proses grayscale

```
p[x] = grayscale;
p[x + 1] = grayscale;
p[x + 2] = grayscale;
```

4.3.2.2 Proses Segementasi

Segmentasi dilakukan untuk proses binerisasi pada image dengan menggunakan nilai threshold. Proses mendapatkan nilai threshold dilakukan secara otomatis dengan metode otsu. Proses metode otsu ini menggunakan bantuan dari aplikasi yang dibuat Tolga Birdal dimana telah menerapkan metode otsu pada bahasa pemrograman C#

1. Melakukan *lock image* format piksel 24 bpp *bitmap*

```
Rectangle rect = new Rectangle(0, 0, bmp.Width,
    bmp.Height);
```

```
BitmapData bmData =
    bmp.LockBits(rect, ImageLockMode.ReadWrite, PixelFormat.Format24bppRgb);
```

2. Memanggil fungsi `getOtsuThreshold` untuk mendapatkan nilai batas ambang dari citra.

```
int otsuThreshold = ot.getOtsuThreshold((Bitmap)temp);
```

3. Proses binerisasi dimana bila nilai grayscale lebih besar dari nilai thershold maka akan dibuat menjadi 255 dan sebaliknya akan dibuat menjadi 0

```
row[x] = (byte)((row[x]>(byte)t) ? 255 : 0);
row[x + 1] = (byte)((row[x + 1]>(byte)t) ? 255 : 0);
row[x + 2] = (byte)((row[x + 2]>(byte)t) ? 255 : 0);
```

- Melakukan proses Unlock *bitmap* bila operasi sudah selesai

```
bmp.UnlockBits (bmData);
```

4.3.2.3 Proses Gabor Filter

Gabor filter dilakukan untuk mendapatkan 4 citra hasil filterisasi menggunakan 4 sudut orientasi berbeda yaitu $0^\circ, 45^\circ, 90^\circ, 135^\circ$. Sehingga untuk proses ini perlu dijalankan sebanyak 4 kali. Berikut listing program dalam syntax C#.

- Memanggil fungsi Enhance dimana proses ini dilakukan sebanyak 4 kali dengan sudut orientasi berbeda

```
a = gabor.Enhance (mono, 0.00);
b = gabor.Enhance (mono, 45.00);
c = gabor.Enhance (mono, 90.00);
d = gabor.Enhance (mono, 135.00);
```

- Menampilkan satu persatu gambar hasil filterisasi setelah dilakukan filterisasi pada masing – masing sudut(derajat) pada picture box

```
pictureBox2.Image = a;
pictureBox2.Image = b;
pictureBox2.Image = c;
pictureBox2.Image = d;
```

4.3.2.4 Proses Operasi Aritmatika

Proses penjumlahan image dan invert image ini dilakukan untuk mendapatkan satu citra hasil filterisasi berikut listing program dalam syntax C#.

1. Memanggil fungsi ImageEnhancePlus dan Invert untuk melakukan proses penjumlahan dan proses logika NOT citra agar mendapatkan satu citra hasil filterisasi

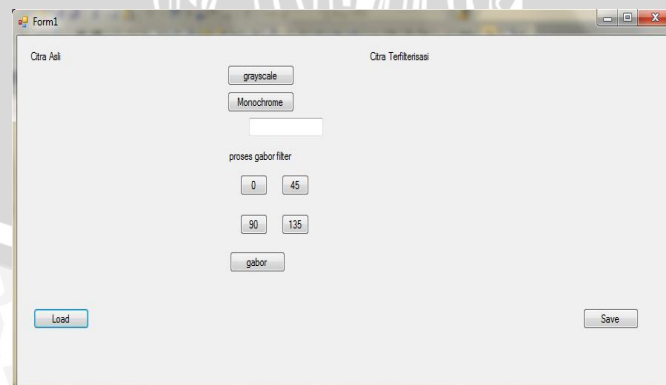
```
a = gabor.invert(a);  
b = gabor.invert(b);  
c = gabor.invert(c);  
d = gabor.invert(d);  
gbor = gabor.ImageEnhancePlus(a, b, c, d);  
gbor = gabor.invert(gbor);
```

2. Menampilkan hasil citra pada aplikasi

```
pictureBox2.Image = gbor;
```

4.4 Implementasi Interface (Antarmuka)

Program *Enhancement* Citra sidik jari ini pengambilan citra sidik jari ini dibuat dengan desain sederhana mungkin dan sesuai dengan langkah-langkah yang berurutan hal ini bertujuan agar pemakai tidak bingung dalam menggunakan program ini. Pada tampilan utama ini terdapat sembilan button yang mempunyai fungsi berbeda – beda. Selain itu pada tampilan utama terdapat dua picture box yang berguna untuk menampilkan citra input dan citra hasil filterisasi.

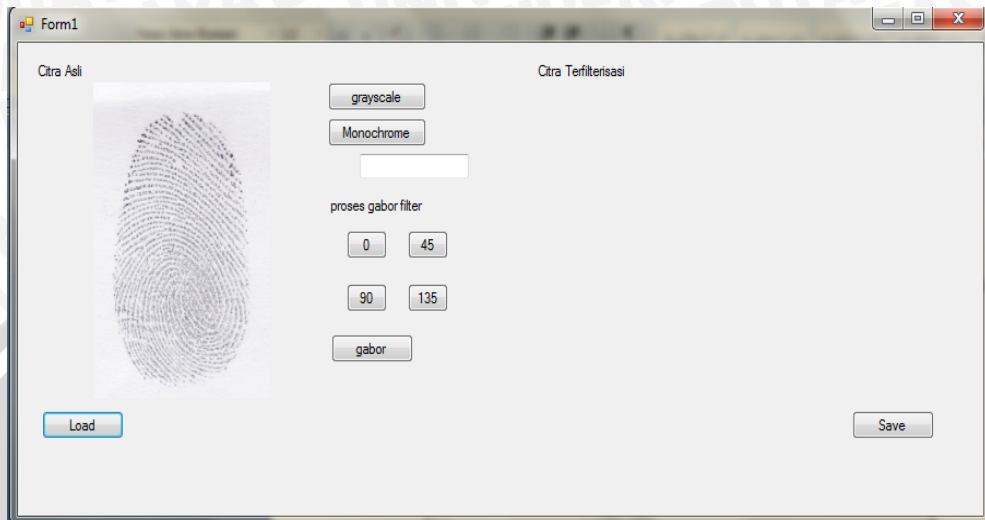


Gambar 4.10 Tampilan utama program

Sumber : Perancangan

4.4.1 Implementasi Antarmuka Untuk Load data

Pada tampilan antarmuka dari *load* data terlihat pada gambar dibawah ini dimana terdapat pictureBox untuk menampilkan data citra yang akan diproses.

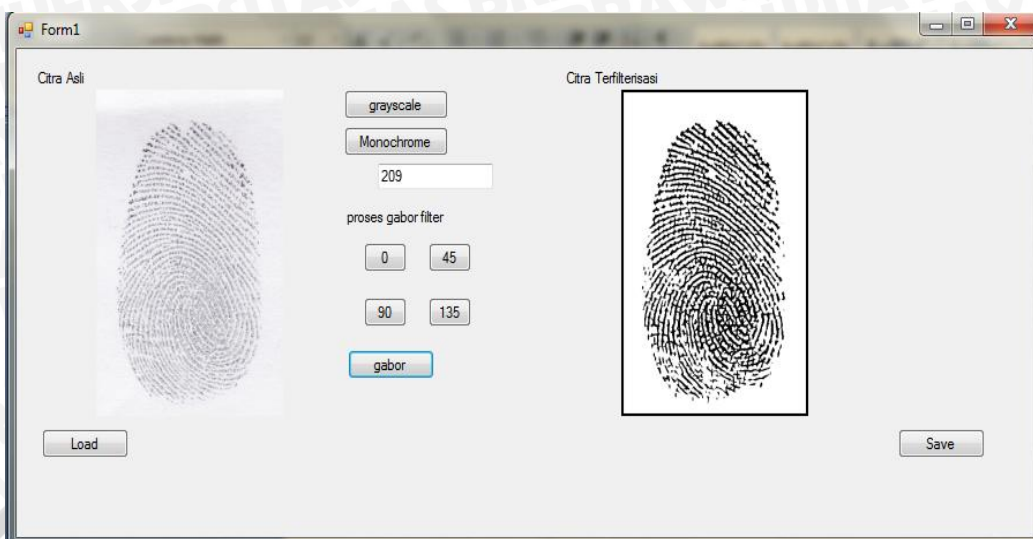


Gambar 4.11 Tampilan Load citra

Sumber : Pengujian

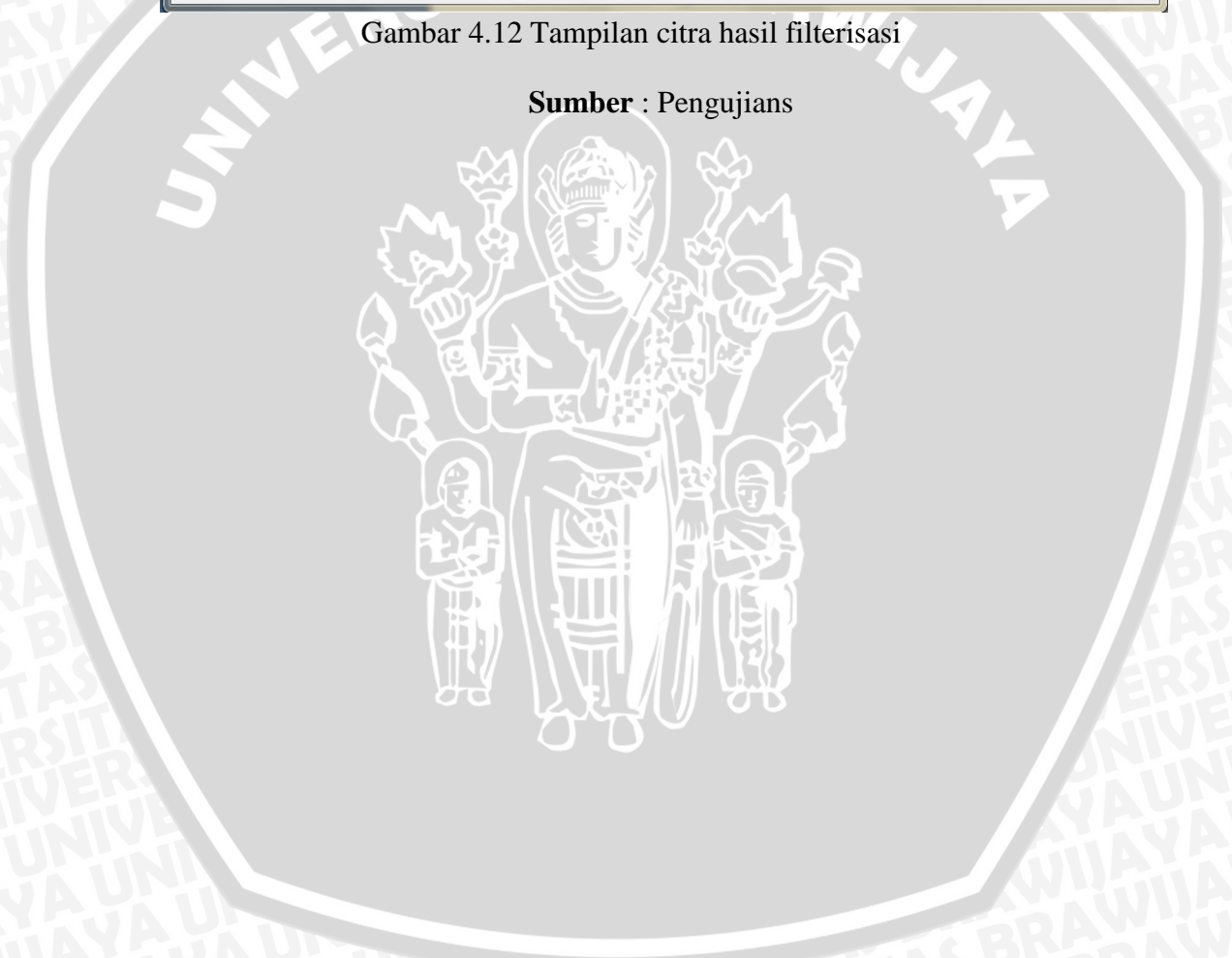
4.4.2 Implementasi Antarmuka Proses Pengolahan Citra

Untuk melakukan proses filterisasi citra harus di ubah menjadi citra grayscale terlebih dahulu. Selanjutnya citra grayscale disegmentasi dengan merubah menjadi citra biner menggunakan metode otsu threshold dengan menekan button Monochrome, di bawah buttoon Monochrome akan muncul nilai dari batas ambang citra input. Setelah itu dilakukan proses filtersisasi dengan menekan buuton 0,45,90, dan 135 dimana button tersebut meerangkann sudut orientasi yang akan dimasukan sebagai parameter. Button gabor digunakan untuk mendapatkan satu citra hasil penjumlahan dan penguranga dari 4 citra hasil filtersisasi.



Gambar 4.12 Tampilan citra hasil filterisasi

Sumber : Pengujians



BAB V PENGUJIAN

Untuk mengetahui apakah sistem bekerja dengan baik dan sesuai dengan perancangan, maka diperlukan serangkaian pengujian. Pengujian yang dilakukan dalam bab ini adalah sebagai berikut:

1. Pengujian gabor filter dengan 4 orientasi berbeda.
2. Pengujian operasi aritmatika untuk mendapatkan 1 citra terfilterisasi.
3. Analisis faktor kegagalan.
4. Kesimpulan hasil pengujian.

5.1 Pengujian Gabor Filter Dengan 4 Orientasi Sudut Berbeda










Pengujian ini bertujuan untuk mengetahui apakah program berhasil melakukan filterisasi pada citra dengan 4 sudut orientasi yang berbeda. Sudut orientasi yang digunakan pada aplikasi ini yaitu 0° , 45° , 90° , 135° . Berikut hasil pengujian pada tabel 5.1








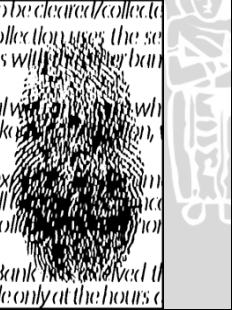
Nama Citra Uji	Sudut Orientasi (derajat)			
	0	45	90	135
Uji_1	Berhasil	Berhasil	Berhasil	Berhasil
Uji_2	Berhasil	Berhasil	Berhasil	Berhasil
Uji_3	Tidak Berhasil	Tidak Berhasil	Tidak Berhasil	Tidak Berhasil
Uji_4	Berhasil	Berhasil	Berhasil	Berhasil
Uji_5	Berhasil	Berhasil	Berhasil	Berhasil
Uji_6	Tidak Berhasil	Tidak Berhasil	Tidak Berhasil	Tidak Berhasil
Uji_7	Berhasil	Berhasil	Berhasil	Berhasil
Uji_8	Berhasil	Berhasil	Berhasil	Berhasil
Uji_9	Berhasil	Berhasil	Berhasil	Berhasil
Uji_10	Berhasil	Berhasil	Berhasil	Berhasil

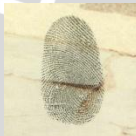




Tabel 5.1 Data pengujian operasi gabor filter

Sumber : Pengujian

Berikut tabel 3 gambar pengujian deteksi objek dari 10 sampel yang digunakan :

Citra Sampel	Filterisasi	Hasil Filterisasi
	0°  	Berhasil
	45°  	Berhasil
	90°  	Berhasil
	135°  	Berhasil

Citra Sampel	Filterisasi	Hasil Filterisasi
	<p>0° →</p> 	<p>Tidak Berhasil</p>
	<p>45° →</p> 	<p>Tidak Berhasil</p>
	<p>90° →</p> 	<p>Tidak Berhasil</p>
	<p>135° →</p> 	<p>Tidak Berhasil</p>

Citra Sampel	Filterisasi		Hasil Filterisasi
	0° →		Berhasil
	45° →		Berhasil
	90° →		Berhasil
	135° →		Berhasil

Tabel 5.2 Tabel Citra Pengujian Operasi Gabor Filter

Sumber : Pengujian

5.2 Pengujian Operasi Aritmatika

Pengujian ini bertujuan untuk mengetahui apakah operasi aritmatika pada 4 citra terfilterisasi berhasil dilakukan untuk mendapatkan 1 citra akhir hasil filterisasi. Berikut hasil pengujian pada tabel 5.1

Nama Citra Uji	Operasi Aritmatika
Uji_1	Berhasil
Uji_2	Berhasil
Uji_3	Tidak Berhasil
Uji_4	Berhasil
Uji_5	Berhasil
Uji_6	Tidak Berhasil
Uji_7	Berhasil
Uji_8	Berhasil
Uji_9	Berhasil
Uji_10	Berhasil

Tabel 5.3 Data Pengujian Operasi Aritmatika


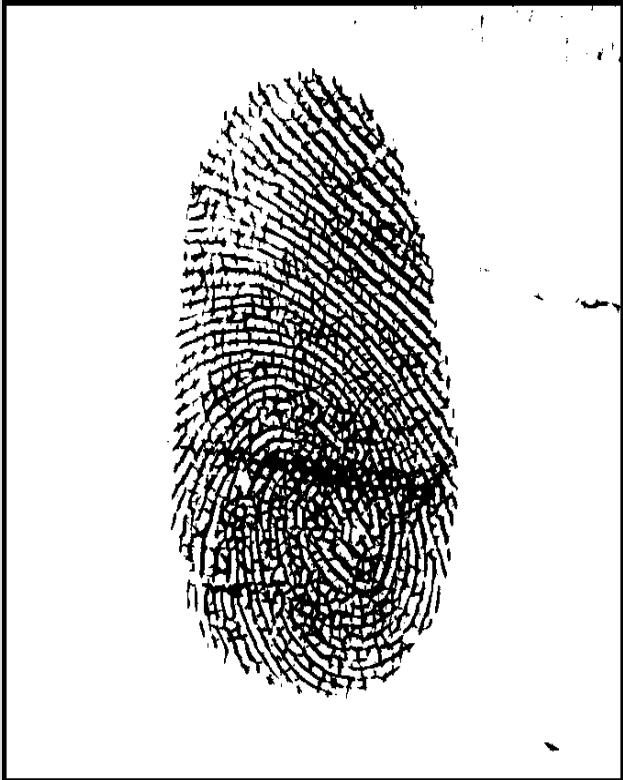

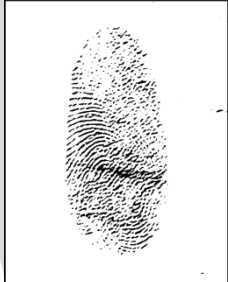
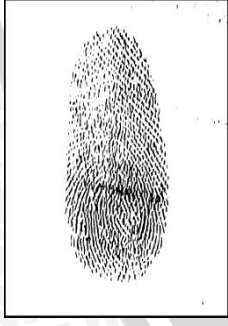
Sumber : Pengujian

Berikut tabel 3 gambar pengujian deteksi objek dari 10 sampel yang digunakan :

Citra Sampel	Filterisasi	Hasil Filterisasi
		<p>Berhasil</p>
		
		
		

Citra Sampel	Filterisasi	Hasil Filterisasi
<p>to be cleared/collected collection uses the services with the other banks</p> 		
		<p>Tidak Berhasil</p>
		
		



Citra Sampel	Filterisasi	Hasil Filterisasi
		<p>Berhasil</p>
		
		
		

Tabel 5.4 Tabel Citra Pengujian Operasi Aritmatika

Sumber : Pengujian

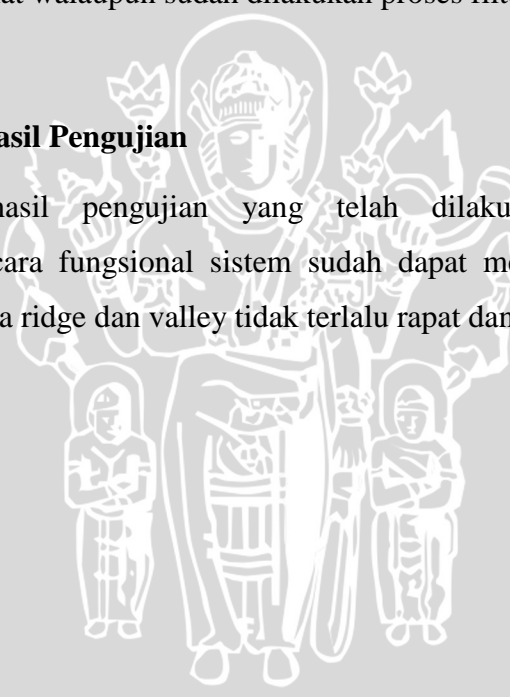
5.4 Analisis Faktor kegagalan

Pada kenyataannya, aplikasi *enhancement* citra sidik jari menggunakan gabor filter tidak selalu berjalan optimal. Hal ini dipengaruhi beberapa faktor antara lain :

1. Kondisi *ridge* dan *valley* pada citra sidik jari input mempengaruhi keberhasilan pada proses filterisasi. Apabila *ridge* terlalu rapat maka setelah dilakukan proses filterisasi pada citra, *ridge* akan semakin tampak lebih tebal sehingga tidak dapat dibedakan antara *ridge* dan *valley* nya
2. Kondisi background pada citra mempengaruhi keberhasilan proses filterisasi. Apabila background citra berwarna hitam, maka background akan tetap terlihat walaupun sudah dilakukan proses filtering pada citra.

5.4 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, memberikan kesimpulan bahwa secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan apabila *ridge* dan *valley* tidak terlalu rapat dan terlihat jelas.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. Citra input berupa citra bitmap. Kemudian dilakukan proses *grayscale* dan segmentasi pada citra. Selanjutnya dilakukan filterisasi citra dengan 4 sudut orientasi berbeda. Setelah proses filterisasi dilakukan proses aritmatika untuk mendapatkan 1 citra terfilterisasi yang selanjutnya dapat diekstraksi titik cirinya.
2. Untuk melakukan pemfilteran citra sidik jari menggunakan metode gabor filter, parameter berupa frekuensi, sudut orientasi, δ_x dan δ_y harus tersedia.

6.2 Saran

Dalam perancangan dan pembuatan aplikasi *enhancement* citra sidik jari menggunakan metode gabor filter ini masih terdapat kekurangan dan kelemahan, oleh karena itu masih diperlukan adanya penyempurnaan dalam rangka pengembangan ke depan. Adapun hal yang dapat disempurnakan yaitu menggunakan metode ridge extracting dan ridge voting untuk menggabungkan 4 citra hasil filterisasi menjadi 1 citra terfilterisasi.

DAFTAR PUSTAKA

- [1] Adi, Kusworo.2003. Pembentukan Vektor Ciri menggunakan Metode *Average Absolute Deviation* (AAD)
- [2] Ahmad, Usman, "Pengolahan Citra Digital dan Teknik Pemrogramannya", Graha Ilmu, Yogyakarta, 2005
- [3] Anonim. *BMP File Format*. http://en.wikipedia.org/wiki/BMP_file_format (diakses tanggal 1 Agustus 2012).
- [4] Burger, Wilhelm., Burge, Burge, Mark James. 2008. *Digital Image Processing*. New York: Spinger.
- [5] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, 2003, *Handbook of Fingerprint Recognition*, Springer, New York.
- [6] Gregory, Peter.2008 *Biometrics for dummies*
- [7] Kirilov, Andrew, "Image Processing and Computer Vision", "<http://aforgenet.com/forum>", Akses tanggal 19 Mei 2012.
- [8] L. Hong, Y. Wan and A.K. Jain, "Fingerprint Image Enhancement: Algorithms and Performance Evaluation", *IEEE Transactions on PAMI*, Vol. 20, No. 8, pp.777-789, August 1998.
- [9] McAndrew, Alasdair "An Introduction to Digital Image Processing With MatLab", School of Computer Science and Mathematics, Victoria University Of Technology, 2004.
- [10] Nasir, Muhammad.2010. *Enhancement citra sidik jari kotor menggunakan teknik hybrid morphology dan gabor filter*.
- [11] Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: Andi
- [12] Thai Raymond. 2003. *Fingerprint Image Enhancement and Minutiae Extraction*. The University of Western Australia.
- [13] Wikipedia, 2012, "Gabor filter", "http://en.wikipedia.org/wiki/Gabor_filter"n Akses tanggal 6 Februari 2012.