



UNIVERSITAS INDONESIA

**STUDI PENYEMPURNAAN IDENTIFIKASI SIDIK JARI
PADA ALGORITMA MINUTEA**

SKRIPSI

**SUROTO
0706199962**

**FAKULTAS TEKNIK
PROGRAM SARJANA EKSTENSI
DEPOK
JUNI 2009**



UNIVERSITAS INDONESIA

**STUDI PENYEMPURNAAN IDENTIFIKASI SIDIK JARI
PADA ALGORITMA MINUTEA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

**SUROTO
0706199962**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2009**

HALAMAN PERNYATAAN ORISINALITAS

**Tugas akhir ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : SUROTO
NPM : 0706199962

Tanda Tangan :
Tanggal : 7 Juli 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : SUROTO
NPM : 0706199962
Program Studi : Teknik Elektro
Judul Skripsi : Studi Penyempurnaan Identifikasi Sidik Jari
pada Algoritma Minutea

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian pernyataan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Abdul Muis ST, M.Eng
NIP. 132233210 (.....)

Penguji : Dr. Abdul Muis ST, M.Eng
NIP. 132233210 (.....)

Penguji : Arief Udhiarto ST,MT
NIP. 0400500032 (.....)

Penguji : Dr. Abdul Halim M.Eng.
NIP. (.....)

Ditetapkan di : Ruang Multimedia B, Lt.2
Universitas Indonesia Depok

Hari / Tanggal : Selasa, 7 Juli 2009

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada **ALLAH SWT**, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tugas akhir ini. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan Tugas akhir ini, sangatlah sulit bagi saya untuk menyelesaikan tugas akhir ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Bpk Dr.Abdul Muis, ST.M.eng, selaku dosen dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan Tugas akhir ini;
- (2) Pihak - pihak lain seperti Laboratorim UI, PT IBM Indonesia serta PT Intikom Berlian Mustika yang telah banyak membantu dalam usaha memperoleh data;
- (3) Orang tua dan keluarga kami yang telah memberikan bantuan dukungan doa, material dan moral.
- (4) Sahabat yang telah banyak membantu kami dalam menyelesaikan Tugas Akhir ini.

Akhir kata, kami berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 16 Juni 2009

Penulis

ABSTRAK

Nama : Suroto
Program Studi : Teknik Elektro
Judul : Studi Penyempurnaan Identifikasi Sidik jari
pada Algoritma Minutae

Sidik jari memiliki suatu orientasi dan struktur periodik berupa komposisi garis-garis gelap dari kulit yang naik (*ridges*) dan garis-garis terang dari kulit yang turun (*furrows*) yang berliku-liku membentuk suatu pola yang berbeda-beda. Walaupun garis-garis alur tangan terbentuk berbeda-beda, tetapi sifat-sifat khusus dari sidik jari yang disebut dengan *minutiae* adalah unik untuk setiap individu. Ciri-ciri ini membentuk pola khusus yang terdiri dari terminasi/ perpisahan atau percabangan dari alur.

Untuk memeriksa apakah dua sidik jari berasal dari jari yang sama atau bukan, para ahli mendeteksi *minutae* tersebut. Sistem Identifikasi Sidik Jari Otomatis (AFIS) akan mengambil dan membandingkan ciri-ciri tersebut untuk menentukan suatu kecocokan.

Metode klasik pengenalan sidik jari menggunakan pattern sekarang ini sudah jarang dipakai dikarenakan memerlukan alokasi penyimpanan data yang besar dan memerlukan waktu identifikasi yang lama. Pembuatan suatu model sidik jari diperlukan untuk menyederhanakan algoritma menjadi lebih efisien.

Dalam Tugas Akhir ini dikembangkan peningkatkan keberhasilan pengenalan sidik jari dengan menambahkan pengolahan gambar berdasarkan *Histogram Equalization* dan *Fast Fourier Transform* (FFT). Dari hasil pengujian diperoleh hasil bahwa dengan menggunakan *Image Enhancement* didapatkan hasil Minutae yang lebih banyak dari pada menggunakan binarisasi tanpa image enhancement.

Kata kunci: Sidik jari , Minutae , Identifikasi, Verifikasi

ABSTRACT

Name : Suroto
Study Program: Electrical Engineering
Title : Study an Fingerprint Identification Enhancement based on
Minutia Algorithm

Human fingerprints have an orientation and periodic structure composed of dark upward skin curves called Ridges, and a lighter colored skin curves called Furrows, which then form many various different pattern. Although that curves make many various shapes and pattern, but special characteristic from that prints which we called Minutia is unique in every individual. These characteristics forms a special pattern which consist of termination, or branches.

To check if two fingerprints come from the same finger or not, experts investigate that Minutia. AFIS or Automated Fingerprint Identification System will take and compare that characteristic to find a match.

Traditional fingerprint identification using pattern recognition is rarely used nowadays, because it needs a very large amount of space for data and require a long time for identification. Fabrication of a fingerprint model is necessary to simplify the algorithm.

In this Final Project is developed to increase success with the introduction of fingerprint image processing add Histogram Equalization, and based on Fast Fourier Transform (FFT). From the results of the test results obtained with that obtained using the Image Enhancement results Minutia more on the use of image enhancement without binarization.

Key words: Fingerprint, Minutia, Identification , Verification

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Suroto
NPM : 0706199962
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik Universitas Indonesia
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

”Studi penyempurnaan Identifikasi Sidik Jari pada Algoritma Minutea”.

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Universitas Indonesia berhak menyimpan, mengalih-media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai Penulis/ Pencipta dan sebagai Pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di :
Pada tanggal :

Yang menyatakan

(Suroto)

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
BAB II DASAR TEORI	4
2.1 Teknologi Sidik Jari	4
2.2 PengolaHan Citra Digital	8
2.2.1 Citra kontinyu dan citra digital	8
2.2.2 Pengenalan pola sidik jari	8
BAB III PERANCANGAN DAN REALISASI	13
3.1 Analisa Kebutuhan	13
3.1.1 Kebutuhan Perangkat	13
3.1.2 Kebutuhan Perangkat Pendukung.....	13
3.2 Perancangan sistem	14
3.3 Proses Identifikasi dan verifikasi sidik jari	36
BAB IV PENGUJIAN DAN ANALISA SISTEM	37
4.1 Mekanisme Pengujian	37
4.2 Spesifikasi Sistem	37
4.3 Pengujian Algoritma dari sistem	38
4.3.1 Pengujian Algoritma sidik jari menggunakan minutea	38
BAB V KESIMPULAN	55
DAFTAR ACUAN	56
DAFTAR PUSTAKA	57

DAFTAR GAMBAR

GAMBAR 2.1	Contoh pola sidik jari ridge berwarna Hitam sedangkan valley berwarna putih	4
GAMBAR 2.2	Algoritma Pattern	5
GAMBAR 2.3	a. berbagai tipe minutea; b. ride ending dan bifurcation	6
GAMBAR 2.4	Penyesuaian minutea (matching) adalah proses dimana dua kumpulan dari minutea dibandingkan untuk memutuskan apakah mereka memiliki citra yang sama	7
GAMBAR 2.5	Feature guratan sidik jari	10
GAMBAR 2.6	Karakteristik sidik jari	11
GAMBAR 3.1	Ekstraksi Minutea	15
GAMBAR 3.2	Histogram asli sidik jari	17
GAMBAR 3.3	Histogram setelah Histogram equalization	17
GAMBAR 3.4	Hasil dari Fourier transformation	20
GAMBAR 3.5	Hasil dari binerisasi	21
GAMBAR 3.6	Beberapa tipe ROI	23
GAMBAR 3.7	Urutan langkah Proses Sidik Jari : a) Original (pattern), b) orientation, c) binarized, d) thinned, e) minutiae, dan minutia graph	37
GAMBAR 3.8	Sistem identifikasi sidik jari	38
GAMBAR 3.9	Hasil tahapan-tahapan algoritma ekstraksi minutiae	39
GAMBAR 3.10	Pencocokan minutea sidik jari	39
GAMBAR 4.1	Hasil citra sidik jari menggunakan DFRU 200	41
GAMBAR 4.2	Sampel sidik jari pada proses pengujian dengan tiga variasi	42
GAMBAR 4.3	Hasil proses Images enhancement	42
GAMBAR 4.4	Load image file ke program sidik jari	43
GAMBAR 4.5	Hasil proses image enhancement, orisinil citra (kiri), Hasil his equalization (kanan)	43
GAMBAR 4.6	Hasil proses image enhancement Hasil FFT (kanan)	44
GAMBAR 4.7	Hasil proses hasil binerisasi, orisinil citra (kiri), Hasil hasil binerisasi	45
GAMBAR 4.8	Hasil proses image segmentasi, orisinil citra (kiri), Hasil ROI (kanan)	46
GAMBAR 4.9	Hasil proses ekstrak minutea, Image asli (kiri), Hasil Thinning (kanan)	46
GAMBAR 4.10	Hasil ekstrak minutea, image asli (kiri) Hasil Ekstrak minutea (kanan)	47
GAMBAR 4.11	Hasil proses post prosesing, image asli (kiri) Hasil menghapus minutea (kanan)	48
GAMBAR 4.12	Hasil proses post prosesing, image asli (kiri) Hasil orientasi (kanan)	48
GAMBAR 4.12	Hasil Validasi (kanan)	49
GAMBAR 4.13	Hasil proses post prosesing, Image asli (kiri)	49

GAMBAR 4.14 Hasil Proses penyimpanan minuteda	50
GAMBAR 4.15 Hasil Proses matching	50
GAMBAR 4.16 Hasil Pengolahan sidik jari (kualiatas bagus) menggunakan image Enhancement	51
GAMBAR 4.17 Hasil Pengolahan sidik jari tanpa (kualiatas bagus) menggunakan image Enhancement	51
GAMBAR 4.16 Hasil Pengolahan sidik jari (kualiatas kurang bagus) menggunakan image Enhancement.....	52
GAMBAR 4.17 Hasil Pengolahan sidik jari tanpa (kualiatas kurangbagus) menggunakan image Enhancement	52

DAFTAR TABEL

TABEL 4.1 Hasil Pengujian sidik jari.....53



BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem teknologi biometrik saat ini mencapai perkembangan yang luar biasa dalam menggantikan sistem konvensional. Pemanfaatan anggota tubuh yang unik ini digunakan untuk membedakan antara satu orang dengan orang lain dan telah dibuktikan memberikan hasil yang lebih akurat dalam pengidentifikasian.[1] Seiring perkembangan sistem security, mulai dari sistem identifikasi konvensional yang menggunakan *password* atau kartu untuk mengenal identitas seseorang ternyata dirasa kurang handal. Karena sistem keamanannya mudah di tembus ketika *password* dan kartu tersebut digunakan oleh pengguna yang tidak berwenang. Maka mulai di kenal teknik security menggunakan biometrik. [10]

Teknik identifikasi biometrik didasarkan pada karakteristik alami manusia, yaitu karakteristik fisiologis dan karakteristik perilaku seperti wajah, sidik jari, suara, telapak tangan, iris, retina mata, DNA, dan tanda tangan.[2] Identifikasi biometrik memiliki keunggulan dibanding dengan metode konvensional dalam hal tidak mudah dicuri atau digunakan oleh pengguna yang tidak berwenang. Sistem pengenalan sidik jari lebih sering digunakan.[1] Hal ini disebabkan sidik jari telah terbukti unik, akurat, aman, mudah, dan nyaman untuk dipakai sebagai identifikasi bila dibanding dengan sistem biometrik lainnya.

Berbagai teknik klasifikasi sidik jari telah dikembangkan. Pengenalan dan klasifikasi sidik jari dengan cara mendeteksi jumlah titik fokal, *whorl*, *core*, dan parameter gradien antara dua titik fokal tersebut. [Ade Chopie Novira] meneliti pengenalan pola sidik jari berbasis jaringan syaraf tiruan (JST). [Elvayandri] melakukan penelitian pengenalan pola sidik jari berbasis jaringan syaraf tiruan yang menggabungkan dua arsitektur JST, yaitu arsitektur Jaringan Widrow - Hoff (JWH) dan Jaringan Propagasi - Balik (JPB). Keduanya mengekstraksi ciri dengan menguraikan ciri minutea cabang (*bifurcation*). [Kusworo Adi] melakukan penelitian sistem verifikasi sidik jari dengan ekstraksi ciri berbasis *filter bank gabor*. [7]

1.2 Perumusan Masalah

Dalam penyusunan Tugas Akhir ini terdapat beberapa permasalahan yaitu bagaimana mendapatkan template citra yang cukup efisien (dengan bantuan image enhancement) sehingga hasil dari citra tersebut bisa diolah menggunakan perangkat lunak Aplikasi Tehnik yang nantinya akan diekstrak ke dalam titik minutea agar dapat digunakan sebagai sistem biometrik.

1.3 Tujuan

Tujuan penyusunan Tugas Akhir ini adalah untuk membahas serta menyempurnakan identifikasi sidik jari menggunakan algoritma minutea dengan meningkatkan kualitas dari sidik jari, sehingga kualitas citra sidik jari cukup baik untuk diolah oleh sistem. Dalam tugas akhir ini dikembangkan *Histogram Equalization* dan *Fast Fourier Transform (FFT)* untuk meningkatkan kualitas citra sidik jari.

1.4 Batasan Masalah

Pembatasan masalah pada penyusunan Tugas Akhir ini adalah pada sistem identifikasi sidik jari menggunakan algoritma minutea yang memanfaatkan pola dari titik pertemuan serta titik perpisahan, dimana titik – titik minutea tersebut dirasa cukup unik untuk tiap individu. Penyempurnaan pada teknik ini dilakukan dengan menambahkan proses pengolahan gambar berdasarkan *Histogram Equalization* dan *Fast Fourier Transform (FFT)* agar bisa menghasilkan minutea yang lebih *valid*.

1.5 Metodologi Penelitian

Metode penulisan yang digunakan adalah sebagai berikut :

1. Studi Pustaka

Membaca dan mempelajari bahan – bahan tentang sistem pengenalan sidik jari baik menggunakan algoritma minutea maupun pattern yang dapat membantu dalam penyusunan Tugas Akhir.

2. Perancangan Sistem

Melakukan proses penggunaan teknik dan prinsip minutea dan penyempurnaanya yang didapat dari studi pustaka untuk mendefinisikan proses ataupun sistem secara detail.

1.6 Sistematika Penulisan

Sistematika pembahasan yang akan diuraikan dalam Tugas Akhir ini terbagi dalam beberapa bab yang akan dibahas sebagai berikut:

Bab I : Pendahuluan

Menjelaskan latar belakang pemilihan Tugas Akhir, batasan masalah, tujuan penulisan, metodologi penulisan dan sistematika penulisan.

Bab II : Dasar teori

Membahas mengenai pengenalan sidik jari menggunakan algoritma minutea yang nantinya akan diproses menggunakan perangkat lunak aplikasi tehnik secara umum, menjelaskan konsep dari sidik jari berupa pengertian, komponen, kegunaan dan lain – lain. Konsep – konsep ini akan digunakan sebagai bahan acuan untuk merancang sistem.

Bab III : Perancangan dan realisasi

Membahas mengenai tahapan - tahapan perancangan yang dilakukan dan proses pengerjaan/ realisasi sistem yang dibuat.

Bab IV : Pengujian dan analisa data

Membahas mengenai teknologi biometrik sidik jari, yaitu menjelaskan mengenai operasi sistem secara keseluruhan dari identifikasi, verifikasi serta pengujiannya.

Bab V : Kesimpulan

Membahas kesimpulan dari seluruh penyusunan Tugas Akhir.

BAB II DASAR TEORI

2.1 Teknologi Sidik Jari

Identifikasi dengan menggunakan pola sidik jari merupakan teknik biometrik yang tertua di dunia. Sejarah kembali ke zaman 6000 tahun sebelum masehi. Penggunaan sidik jari sebagai kode personal, memiliki tradisi yang panjang dan telah digunakan oleh bangsa Assyiria, Babylonia, Jepang dan Cina. Bangsa Cina kuno menggunakan sidik jari sebagai alat identifikasi pembuktian seseorang sebagai pengarang dari suatu dokumen. Sejak tahun 1897, *dactyloscopy* (sinonim dari identifikasi sidik jari tanpa berbasis komputer) telah digunakan untuk identifikasi kejahatan.[7]

Karakteristik sidik jari dari setiap orang adalah unik dan secara normal tidak berubah selama hidupnya. Dari penelitian yang seksama, diperkirakan bahwa peluang dua orang, meskipun kembar, mempunyai pola sidik jari yang sama adalah lebih kecil dari satu di dalam satu milyar.[6]

Suatu pola sidik jari normal dibentuk dari garis-garis (*lines*) dan spasi (*spaces*). Garis-garis ini dinamakan *ridge* sedangkan spasi antara garis-garis ini dinamakan *valley*. Dengan melalui pola dari *ridge* dan *valley* inilah sebuah sidik jari yang unik disesuaikan untuk verifikasi dan otorisasi.[1] Ciri yang unik dari sidik jari ini bisa dicapture kedalam bentuk algoritma, yaitu algoritma pattern (berdasarkan pola) dan minuteda (berdasarkan titik).



Gambar 2.1 Contoh pola sidik jari Ridge berwarna hitam sedangkan valley berwarna putih

Sebagian besar sistem otomatisasi digunakan untuk perbandingan sidik jari didasari oleh matching algoritma baik pattern atau pun minutea. perbedaan dari kedua algoritma diatas bisa kita pembahasan pembahasan dibawah ini:

1. Algoritma Pattern

Merupakan pola pengenalan sidik jari dengan cara meng-capture perblok dari punggung atau lekukan sidik jari yang sudah dibagi kedalam sektor (kotak kecil). Dimana dalam pembagiannya dibedakan berdasarkan *shape of phase, pitch and ridges*, seperti gambar dibawah ini:[8]

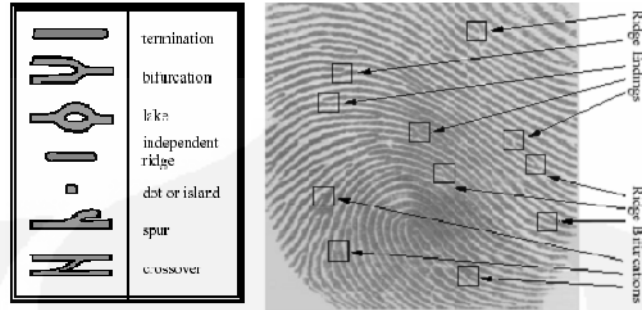


Gambar 2.2 Algoritma Pattern

Pattern recognition (pengenalan pola) merupakan kemampuan mengenali objek-objek berdasarkan ciri – ciri dan pengetahuan yang pernah diamati dari objek – objek tersebut. Tujuannya untuk mengklarifikasi dan mendeskripsikan pola/ subjek kompleks melalui pengukuran sifat – sifat/ciri – ciri objek tersebut. [8]

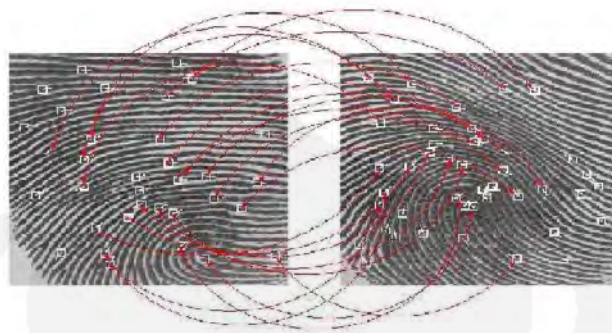
2. Algoritma minutea

Minutea merupakan rumusan pola sidik jari yang paling populer saat ini, karena karakteristik minutea adalah *local discontinue* dalam pola sidik jari yang merepresentasikan *termination* (titik akhir) dan *bifurcation* (pencabangan dua). *Ridge termination* didefinisikan sebagai titik dimana sebuah ridge berakhir, sedangkan *ridge bifurcation* sebagai titik dimana sebuah *ridge* bercabang. Contoh dari minutea ditunjukkan pada gambar dibawah ini:[8]



Gambar 2.3 a. berbagai tipe minuteda;
b. ride ending dan bifurcation

Teknologi sidik jari merupakan teknologi yang paling banyak digunakan dibanding teknologi biometrik yang lain (seperti iris, retina scan, *face scan*, *voice scan*, dan *type signature*). Hal ini dikarenakan sidik jari termasuk yang paling murah (*cost effective*) akan tetapi tetap mempertahankan tingkat keamanan yang tinggi dan kemudahan dalam penggunaannya. Kekuatan sidik jari terletak pada dukungannya (*acceptance*), kemudahan, dan keandalannya. Seseorang membutuhkan sedikit waktu dan usaha ketika menggunakan peralatan identifikasi sidik jari untuk mendapatkan hasil scan sidik jarinya.[10] Dari penyelidikan juga diketahui bahwa menggunakan sidik jari sebagai alat identifikasi adalah paling mudah dibandingkan dengan teknik biometrik lainnya. Verifikasi sidik jari juga sangat cepat dan dapat diandalkan. Umumnya pengguna mengalami sedikit error didalam proses matching ketika menggunakan sidik jari dibandingkan menggunakan metode biometrik lainnya. Selain itu peralatan identifikasi sidik jari biasanya memerlukan ruang yang cukup kecil pada komputer atau dalam sebuah mesin. Beberapa perusahaan bahkan telah memproduksi unit capture (*scanner*) yang lebih kecil dari sebuah kartu. [10]



Gambar 2.4 Penyesuaian minuteda (matching) adalah proses dimana dua kumpulan dari minuteda dibandingkan untuk memutuskan apakah mereka memiliki citra yang sama.

Penggunaan sidik jari sebagai salah satu metode verifikasi telah digunakan secara luas dalam identifikasi kriminal, verifikasi akses otorisasi, konfirmasi transfer keuangan, dan banyak aplikasi lainnya. Aplikasi sidik jari yang umumnya dilihat saat ini adalah akses jaringan komputer dan alat kunci masuk pintu bangunan yang memanfaatkan teknologi sidik jari.[7] Kartu pembaca sidik jari (*Fingerprint reader*) digunakan pada bank - bank untuk meng-otorisasi ATM dan kini sudah banyak tersedia pada toko bahan makanan yang digunakan untuk mengenali secara otomatis pelanggan yang telah terdaftar sehingga dapat membayar lewat kartu kredit atau debit account. Teknologi sidik jari ini juga sedang digunakan sebagai cara baru pembayaran makanan di beberapa kantin sekolah menengah di Pennsylvania yang mendukung program subsidi makanan pemerintah AS.[10] Pelajar yang menerima subsidi makanan tersebut memiliki wewenang untuk makan dengan terlebih dulu memverifikasi sidik jarinya. Pembelian dengan kartu makanan dari pemerintah lewat *checkout* sebagai pengganti uang tunai juga akan digunakan untuk mengidentifikasi pelajar.[10] Dengan demikian, sekolah memberikan pilihan baru bagi pelajar dengan adanya penggunaan peripheral ini untuk membeli makanan. Pada akhir bulan, rekening dikirimkan ke orang tua mereka atau ke pemberi subsidi untuk pembayaran makanan tersebut. Penggunaan scan sidik jari ini akan memberi keyakinan

kepada orangtua atau pemerintah untuk menanggung pembayaran makanan bagi pelajar tersebut.

2.2 Pengolahan Citra Digital

2.2.1 Citra kontinyu dan citra digital

Citra kontinyu yang dimaksud dalam pengolahan citra dapat diartikan sebagai suatu fungsi kontinyu dari intensitas cahaya $f(x,y)$ dalam bidang dua dimensi, dengan (x,y) menyatakan suatu koordinat dan nilai f pada setiap titik (x,y) menyatakan intensitas, tingkat kecerahan atau derajat keabuan. Citra digital merupakan suatu citra kontinyu yang diubah kedalam bentuk diskrit, baik koordinat maupun intensitas cahayanya. Dengan kata lain, citra digital dibuat dengan cara mencuplik suatu citra kontinyu dengan jarak seragam. Suatu titik terkecil pada citra digital sering disebut sebagai *picture element* atau *pixel*. [1] Citra ini dapat berupa citra vektor ataupun citra bitmap. Citra ini mengandung persamaan - persamaan matematis dari bentuk - bentuk dasar yang membentuk citra tersebut.

Operasi pengolahan citra digital umumnya dilakukan dengan tujuan memperbaiki kualitas suatu gambar, sehingga dapat dengan mudah di-interpretasi oleh mata manusia dan untuk mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis. [6]

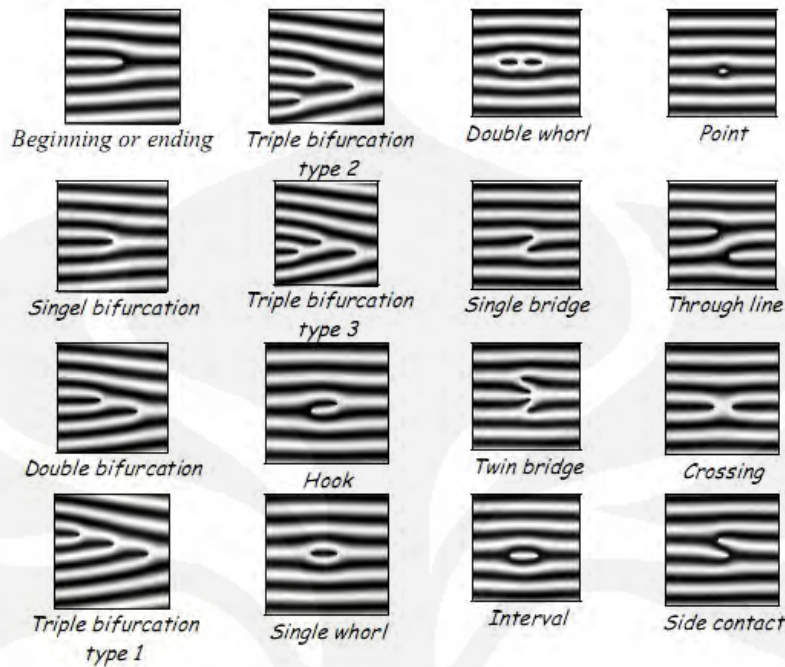
2.2.2 Pengenalan pola sidik jari

Pola atau *pattern* adalah pengaturan khusus dari elemen - elemen struktural misalnya pola - pola awan yang berbeda - beda. Adapun pengenalan pola adalah proses mengidentifikasi suatu objek dalam citra. Secara garis besar, metode - metode pengenalan pola dapat dibagi menjadi tiga kelompok, yaitu metode statistik, metode struktural maupun metode jaringan syaraf tiruan. Pada *metode structural*, dicari ciri *khas/fitur* yang unik dari suatu citra yang dapat dimanfaatkan pada proses pengenalan pola. Pada penelitian ini digunakan metode pengenalan pola dengan *metode struktural*. [9] Sebagaimana prinsip kerjanya, maka pada metode ini harus ada fitur yang diekstrak dan untuk verifikasi sidik jari ini, fitur yang diambil adalah *minutae* sebagai model strukturnya. [1]

Sistem biometrik merupakan sebuah sistem pengenalan pola yang melakukan identifikasi personal dengan menentukan keotentikan dari karakteristik fisiologis dan perilaku tertentu yang dimiliki seseorang. Secara logika sistem ini dibagi menjadi dua modul: modul pendaftaran dan modul identifikasi. Modul pendaftaran berfungsi untuk mengambil data dari individu dan menyimpannya ke dalam sistem. Pada saat pendaftaran, karakteristik biometrik discan terlebih dahulu oleh sebuah pemindai biometrik untuk menghasilkan sebuah representasi digital yang belum diolah. Untuk dapat digunakan dalam proses pencocokan, representasi digital tersebut diproses lebih lanjut untuk mendapatkan representasi hasil yang cukup untuk mewakilinya, yang nantinya disebut sebagai *template*. [1] *Template* ini kemudian disimpan dalam database (*database* ini terdapat pada sistem biometrik). Modul identifikasi berfungsi untuk mengidentifikasi individu pada titik akses. Pada saat tahap pengoperasian, pemindai biometrik menangkap karakteristik yang akan diidentifikasi dan diubah menjadi format digital, kemudian oleh ekstraktor fitur diproses menjadi representasi yang sama dengan *template*-nya dan kemudian dicocokkan untuk mendapatkan suatu identitas. [8]

Karakteristik fisiologis manusia untuk biometrik harus memenuhi beberapa kriteria yaitu: Universal, unik, permanen, dan dapat diukur secara kuantitatif dan nilai dari pengukuran tersebut dapat digunakan untuk kepentingan verifikasi dan identifikasi. [1] Meski demikian, pada prakteknya tidak semua karakteristik biometrik yang memenuhi syarat diatas dapat diwujudkan dalam suatu sistem biometrik yang nyata. Ada beberapa hal yang perlu diperhatikan, yaitu: a. Unjuk kerja (*performance*) b. Tingkat penerimaan (*acceptability*), dan c. Tingkat keamanan (*circumvention*). [10]

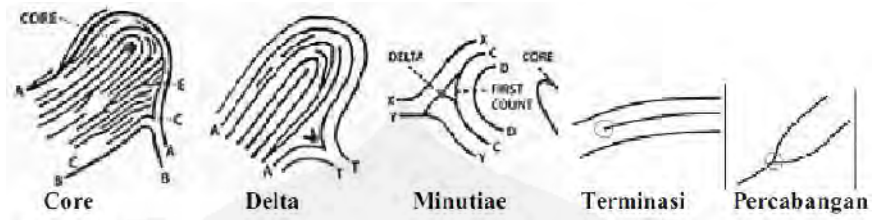
Teknologi biometrik yang menggunakan sidik jari lebih memanfaatkan guratan - guratan yang terdapat di sidik jari masing-masing, berikut ini beberapa macam bentuk dari guratan sidik jari: [2]



Gambar 2.5 Fitur guratan sidik jari

Sedangkan ciri lokal sidik jari ditentukan oleh jumlah dan posisi garis alur dan banyaknya percabangan dari garis - garis alur yang terdiri dari:

1. **Inti** (*core*) didefinisikan sebagai titik yang didekatnya terdapat alur - alur yang membentuk susunan semi-melingkar. Inti ini digunakan sebagai titik pusat lingkaran balik garis alur yang menjadi titik acuan pembacaan dan pengklasifikasian sidik jari.[8]
2. **Delta** didefinisikan sebagai suatu titik yang terdapat pada suatu daerah yang dibatasi oleh tiga sektor yang masing - masing memiliki bentuk hiperbolik. Titik ini merupakan pertemuan curam atau titik *divergensi* dari pertemuan dua garis alur.[8]
3. **Minutae** didefinisikan sebagai titik - titik terminasi (*ending*) dan titik - titik awal percabangan (*bifurcation*) dari garis - garis alur yang memberikan informasi yang unik dari suatu sidik jari.[8]



Gambar 2.6 Karakteristik sidik jari

BAB III

PERANCANGAN DAN REALISASI

Dalam setiap melakukan perancangan sistem diperlukan metode atau cara serta memerlukan perangkat yang nantinya akan mempermudah kerja, seperti kebutuhan perangkat keras maupun perangkat lunak. Dimana nantinya perangkat ini lah yang dapat menentukan persentase keberhasilan rancangan.

3.1 Analisa Kebutuhan

Dalam merencanakan perancangan aplikasi berupa perangkat lunak komputer atau sejenisnya dibutuhkan beberapa prasarana agar dalam pelaksanaannya tidak mengalami kendala. Berikut beberapa kebutuhan yang dibutuhkan dalam perancangan Tugas akhir ini.

3.1.1 Kebutuhan Perangkat

- Perangkat lunak

Perangkat lunak yang digunakan dalam sistem ini adalah:

1. Operating sistem (OS): Windows Se7en, Vista atau XP
2. Perangkat lunak simulasi: Perangkat lunak aplikasi tehnik

- Perangkat keras

1. Hardware PC (Personal Computer) yang dibutuhkan untuk menjalankan simulasi ini memiliki spesifikasi sebagai berikut:
2. Intel Pentium II atau processor yang kualitasnya menyamai bahkan lebih
3. VGA card yang grafiknya bisa kualitas warna high (16 bit) dan resolusi 1024 x 768
4. RAM minimal 512 MB (direkomendasikan memori 1 GB)
Monitor 32 bit dengan resolusi 800 x 600 pixel.

3.1.2 Kebutuhan Perangkat Pendukung

Pada tugas akhir ini dibuat sebuah aplikasi yang dapat menampung template dari citra sidik jari yang nantinya akan digunakan sebagai database pada sistem. Data di template tersebut didapat dengan melakukan *enroll* (pendaftaran sidik jari) dengan bantuan *sidik jari reader*. [10] Setelah citra sidik jari didapat dari *proses enroll*, maka data tersebut disimpan ke dalam template database pada

aplikasi yang dirancang menggunakan perangkat lunak pengolah data matematis. Pada proses identifikasi sidik jari ada 2 tahapan penting, yaitu: pertama pendaftaran (*enroll*) dan yang kedua identifikasi (pengenalan).[1]

Pada tahap identifikasi ini digunakan algoritma *minutec*, dimana metode ini memanfaatkan hasil citra dari proses *enroll* (pendaftaran) ke dalam bentuk titik *minutec* yang berupa kumpulan titik percabangan (*ridge bifurcation*) dan titik perpisahan (*ridge termination*). Titik - titik inilah yang nantinya digunakan sebagai data, apakah sidik jari dianggap cocok/*matching* atau ditolak/*direject*.

3.2 Perancangan

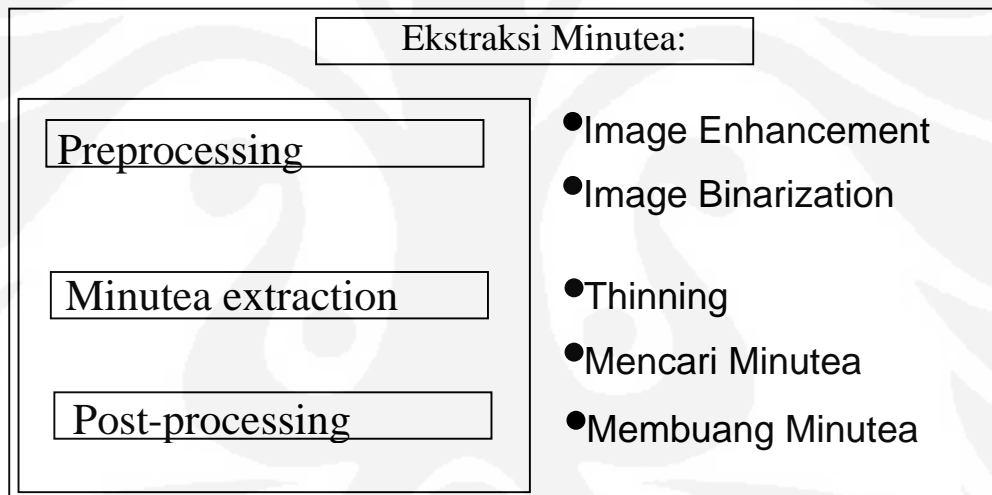
- Batasan Perancangan

Aplikasi yang digunakan untuk *enrollment* serta identifikasi sidik jari ini menggunakan algoritma *minutec* yang terfokus pada peningkatan citra gambar sidik jari menggunakan Image enhancement (Histogram equalization dan Fourier Transform). Algoritma *minutec* terfokus pada titik perpisahan (*termination*) dan titik percabangan (*bifurcation*). Dimana penentuan titik - titik tersebut dimulai dengan menempatkan titik tetap (inti) sebagai acuan posisi x dan y dari *minutec* secara absolut maupun relatif. Titik - titik tersebut nantinya akan digunakan sebagai acuan untuk digunakan kedalam sistem identifikasi, yaitu untuk menentukan apakah sidik jari dianggap sesuai dengan data yang ada di database.[1]

- Perancangan Arsitektur

Perancangan arsitektur dalam tugas akhir ini menggunakan perangkat lunak Aplikasi Teknik. Arsitektur dari program atau aplikasi yang dibuat ini akan menentukan titik - titik *minutec* menggunakan perangkat lunak aplikasi tehnik dimana pada proses menentukan titik *minutec* meliputi beberapa tahapan seperti *Load images - His Equalization - FFT - Binerisasi - Thinning - ROI - Mencari Minutec - Menghapus minutec - Orientasi - Validasi - Save Minutec - Matching*, dimana hasilnya berupa titik - titik koordinat X dan Y . Sedangkan untuk mengidentifikasi dan verifikasi dari citra sidik jari. Pada tahap identifikasi minimal dibutuhkan 2 citra sidik jari yang akan dilihat kemiripannya (*matching*),

setiap citra sidik jari minimal harus memiliki 10 titik minutea sebagai *threshold*, jika jumlah minutea dibawah 10 maka di anggap tidak sesuai *required*, maka perlu diganti dengan citra yang baru. Ini dikarenakan pada sidik jari manusia tidak ada kemiripan struktur sidik jari yang memiliki lebih dari 8 titik minutea yang sama, bahkan untuk orang yang kembar sekalipun, sehingga pada aplikasi ini dibuat *threshold* pada jumlah 10 titik minutea.[9] sedangkan untuk proses identifikasi, sebelumnya kita harus mendaftarkan sidik jari (*enrollment*) ke dalam sistem agar citra sidik jari tersimpan didalam template atau database sistem, sehingga pada proses identifikasi data citra yang kita masukkan ke dalam sistem akan dicocokkan atau dilihat kemiripannya dengan data yang ada didalam database sistem. Dalam proses pencarian titik minutea, terdapat tiga level desain selalu dipakai. Ketiga proses tersebut adalah: [10]



Gambar 3.1 Ekstraksi Minutea

a) **Algoritma penentuan minutea menggunakan Perangkat lunak Aplikasi Teknik**

Penentuan titik minutea yang menggunakan perangkat lunak Aplikasi Teknik sebagai *base program* dengan memanfaatkan *citra processing toolbox* dimulai dengan memasukkan citra sidik jari ke dalam dimensi digital (*citra biner*) dibagi ke dalam tiga level desain yaitu *Preprocessing*, *Minutea extraction* dan *Post Processing* dimana disetiap level desain dibagi lagi kedalam sub level desain

seperti terlihat [digambar 3.1. Mengenai Ekstraksi Minutea].[3] Berikut ini tahapan - tahapan penentuan titik minutea pada algoritma program yang dirancang: *Load images - His Equalization - FFT - Binerisasi - Thinning - ROI - Mencari Minutea - Menghapus minutea - Orientasi - Validasi - Save Minutea - Matching*, Dimana *Histogram equalization* dan *Fourier transform (FFT)* merupakan proses penyempurnaan untuk pengenalan gambar agar diperoleh hasil citra sidik jari lebih baik. Berikut ini penjelasan dari semua tahapan yang ada.

a. *Load Images*

Pada setiap aplikasi pemrosesan citra, tahapan pertama yang harus dilakukan adalah memasukan citra yang akan kita olah kedalam aplikasi atau sistem yang kita rancang, mengenai citra yang bias diproses ini mencakup beberapa tipe citra seperti : **.bmp; *.BMP; *.tif; *.TIF; *.jpg*.[3] Berikut ini algoritma dari load images pada sistem identifikasi sidik jari menggunakan algoritma minutea:[3]

```

%=====
% Fungsi Call Bak Ketika Menu Buka Gambar Dipanggil
%=====
function pb_loadimage_Callback(hObject, eventdata,
handles)
[file_name, path_name]=uigetfile(...
    {'*.bmp;*.jpg;*.tiff','Citra File
 (*.bmp,*.jpg,*.tiff)';
    '*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File jpeg (*.jpg)';...
    '*.tiff','File tiff (*.tiff)';...
    '*.*','All (*.*)'},...
    'Open Citra File');

if file_name ~= 0
cd(path_name);
image1=imread(char(file_name));
colormap(gray);
%image1=double(image1)
%image1=255-double(image1);

guidata(hObject,handles);
axes(handles.imageasli);
imshow(image1);
guidata(hObject,handles);

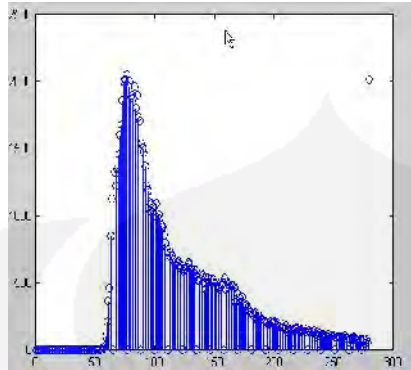
guidata(hObject,handles);
handles.image1 = image1;
guidata(hObject,handles);
%setappdata(handles.NewFingerprint,'image1',image1);
else
return
end;

```

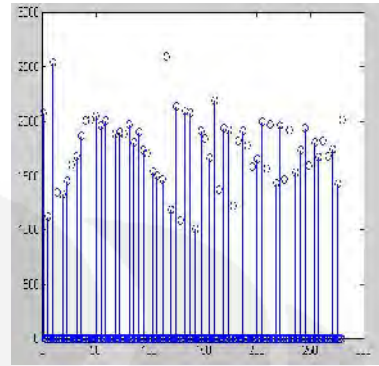
Setelah citra sidik jari sudah di *upload* atau dimasukkan ke dalam sistem, maka citra sidik jari nantinya akan mengalami beberapa tahap proses seperti *preprocessing*, *ekstraksi minuteda* dan *post processing*. Pada tahapan *preprocessing*, terdapat sub level desain, yaitu *image enhancement*, *binerisation* dan *segmentation*. Proses *images enhancement* menggunakan *histogram equalization* dan *fourier tranformation (FFT)*. Penggunaan *image enhancement* yang terfokus pada penggunaan *his equalization* dan *FFT* memiliki dampak yang cukup besar pada kinerja sistem, karena dapat membantu proses tanpa harus memilih masukan data. Dengan adanya *Image enhancement* maka apapun masukan data (baik citra dengan kualitas baik ataupun buruk) nantinya akan dapat diproses dengan meningkatkan kualitas serta informasi dari citra tersebut. Berikut ini penjelasan dari tahapan - tahapan *images enhancement*:

b. *Histogram Equalization*

Proses ini merupakan penambahan proses penyempurnaan pada tugas akhir ini, dimana tanpa proses ini sistem identifikasi sidik jari hanya bisa mengolah citra sidik jari yang mempunyai kualitas bagus. *Histogram equalization* menyatakan frekuensi kemunculan berbagai derajat keabuan dalam citra. Teknik pemodelan histogram mengubah citra hingga memiliki histogram sesuai keinginan. Contohnya adalah ekualisasi histogram yang bertujuan mendapatkan histogram citra dengan distribusi seragam.[3] Merupakan metode yang digunakan untuk meningkatkan atau memperluas nilai pixel dari suatu citra biner serta untuk meningkatkan pemahaman dan informasi mengenai citra tersebut, model orisinal dari citra sidik jari adalah tipe bimodal, seperti ditunjukkan pada gambar dibawah ini: [3]



Gambar 3.2 Histogram asli dari sidik jari



Gambar 3.3 Histogram setelah His Equalization

Berikut ini listing programnya:

```

=====
% Fungsi Call Back ketika button histogram diklik
=====
function pb_histogram_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
image2 = handles.image1;
guidata(hObject,handles);
if ndims(image2)==3
image2= rgb2gray(image2)
end
image2=histeq(uint8(image2));

guidata(hObject,handles);
axes(handles.imageproses);
imagesc(image2);
guidata(hObject,handles);

guidata(hObject,handles);
handles.image2 = image2;
guidata(hObject,handles);
  
```

c. *FFT (Fourier Transform)*

Selain Histogram equalization yang merupakan salah satu cara untuk meningkatkan kualitas sidik jari, satu lagi cara untuk meningkatkan kualitas sidik jari adalah mengubah dari besaran spasial ke besaran frekuensi. Proses tersebut adalah Fourier transform. Fourier transform adalah salah satu teknik perbaikan citra (*image enhancement*). Pada tugas akhir ini FFT merupakan penambahan fitur untuk mendapatkan kualitas citra sidik jari. Pada dasarnya perbaikan citra dapat dibagi

kedalam kedua kelompok, yakni perbaikan citra dalam domain spasial dan perbaikan citra dalam domain frekuensi (Fourier).).[4] Pembagian ini didasarkan pada kawasan dilakukannya proses perbaikan dan seringkali proses perbaikan pada kedua kawasan ini ekuivalen. Jika demikian halnya, pemilihan pengolahan pada kawasan Fourier memberikan keuntungan dengan tersedianya algoritma cepat *FFT* (*Fast Fourier Transform*). FFT dapat digunakan untuk meningkatkan kualitas dari citra. Fourier transform dapat membagi citra menjadi blok kecil - kecil (32 x 32 pixel) dengan rumus tranformasi fourier sebagai berikut: [4]

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left\{-j2\pi X\left(\frac{ux}{M} + \frac{vy}{N}\right)\right\} \quad (3.1)$$

Untuk nilai $u = 0, 1, 2, \dots, 31$ dan $v = 0, 1, 2, \dots, 31$.

Dalam meningkatkan kualitas blok dari citra secara khusus, maka FFT membagi blok - blok tersebut kedalam besaran waktu. Dimana besarnya nilai FFT asli = $\text{abs}(F(u,v)) = |F(u,v)|$. [4]

Dengan meningkatkan kualitas blok mengikuti rumus sbb:

$$g(x,y) = F^{-1}\left\{F(u,v)X|F(u,v)|^k\right\} \quad (3.2)$$

Dimana $F^{-1}(F(u,v))$ akan selesai menggunakan rumus berikut:

$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(u,v) \exp\left\{j2\pi X\left(\frac{ux}{M} + \frac{vy}{N}\right)\right\} \quad (3.3)$$

Untuk nilai $x = 0, 1, 2, \dots, 31$ dan $y = 0, 1, 2, \dots, 31$.

Dengan menggunakan rumus diatas, maka proses peningkatan mutu atau kualitas dari citra sidik jari menggunakan tranformasi fourier dapat dibuat kedalam algoritma pemrograman sebagai berikut: [3]

```

%=====
% Fungsi Slider FFT
%=====
function slider_fft_Callback(hObject, eventdata,
handles)
slider_value1=get(handles.slider_fft, 'Value');
set(handles.ed_fft, 'String', num2str(slider_value1));
handles.fftvalue = slider_value1;

```

```

guidata(hObject,handles);
function slider_fft_CreateFcn(hObject, eventdata,
handles)
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

%=====
% Fungsi Edit
%=====
function ed_fft_Callback(hObject, eventdata, handles)
function ed_fft_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%=====
% Fungsi Call Back ketika button FFT diklik
%=====
function pb_fft_Callback(hObject, eventdata, handles)

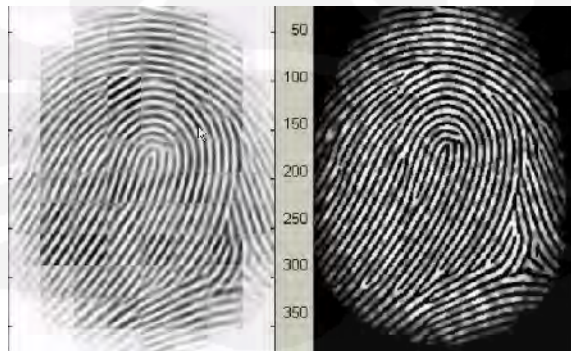
guidata(hObject,handles);
image3 = handles.image2;
nilaiff = handles.fftvalue;
guidata(hObject,handles);
disp(size(image3));

image3=fftenhance(image3,nilaiff);
disp(size(image3));

guidata(hObject,handles);
axes(handles.imageproses);
imagesc(image3);
guidata(hObject,handles);

guidata(hObject,handles);
handles.image3 = image3;
guidata(hObject,handles);

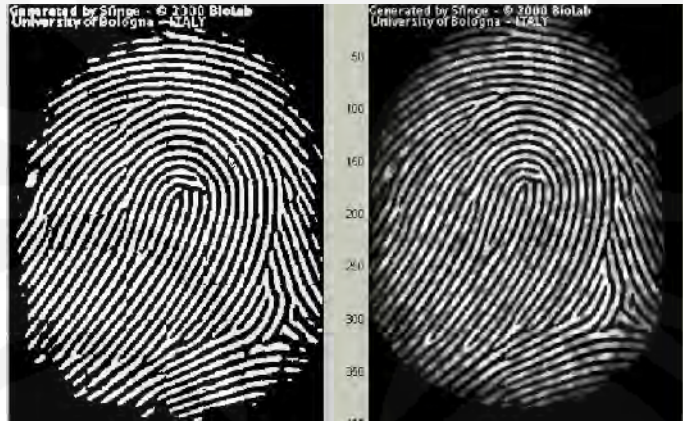
```



Gambar 3.4 Hasil dari Fourier transformation

d. Binerization

Dalam hal ini, citra dari sidik jari akan diubah dari citra warna menjadi citra grayscale lalu menjadi citra biner dengan threshold nilai derajat keabuan bisa diatur dengan default 160. Hasil dari proses ini adalah bentuk citra sidik jari yang berwarna hitam (pixel 0) dan putih (pixel 1). Berikut contoh citra hasil binerisasi: [3]



Gambar 3.5 Hasil dari binerisasi

Binerisasi akan mengubah citra original ke dalam citra biner yang memiliki derajat keabuan sesuai keinginan. Berikut ini listing programnya : [3]

```

=====
% Fungsi Call Back ketika button binerisasi Di Klik
=====
function pb_binerisasi_Callback(hObject, eventdata,
handles)

guidata(hObject,handles);
image4 = handles.image3;
guidata(hObject,handles);

if get(handles.rb_automatibiner, 'value')==1

    BinarizedImage=image4(:,:,1)>160;
else

Threshold=str2num(get(handles.ed_manualbinerisasi, 'string'));
    if Threshold >= 50
        Threshold = 50

```



```

end
BinarizedImage=image4(:,:,1)>Threshold;
end

image5=BinarizedImage*255;
disp(size(image4))
guidata(hObject,handles);
axes(handles.imageproses);
imshow(image5);

guidata(hObject,handles);

guidata(hObject,handles);
handles.image5 = image5;
guidata(hObject,handles);

```

e. *Images segmentation (ROI)*

Setelah citra sidik jari diubah kedalam citra biner, selanjutnya citra tersebut akan melalui proses *image segmentation* atau biasa disebut dengan *Region of interest (ROI)*, dimana images segmentasi bisa dilakukan dengan dua langkah yaitu dengan *blok direction estimation* dan *direction variety check*. berikut ini pembahasan mengenai dua langkah pada proses images segmentasi: [3]

a) Memperkirakan arah dari blok

Proses ini memperkirakan arah setiap blok citra sidik jari kedalam ukuran $W \times W$ dimana setiap W adalah 16 pixel. Pembagian setiap blok citra menggunakan algoritma sebagai berikut:

(a) Menghitung nilai gradient dari G_x dan G_y untuk setiap pixel bloknya, dengan dua sample filter digunakan untuk mengisi tugasnya.[3]

(b) Untuk setiap bloknya, menggunakan formula untuk mendapatkan bentuk formula kubus yang paling sedikit dari setiap blok direction.

$$tg2\beta = 2 \sum \sum (gx * gy) / \sum \sum (gx^2 - gy^2) \quad (3.4)$$

untuk setiap pixel dari blok.

Rumus diatas mudah dipahami dengan menganggap nilai gradient mendekati arah x dan y adalah nilai cosine dan nilai

sine, sehingga nilai tangent dari blok direction seperti mengikuti ilustrasi dari rumus dibawah ini: [3]

$$tg2\theta = 2 \sin \theta \cos \theta / (\cos^2 \theta - \sin^2 \theta) \quad (3.5)$$

setelah selesai memperkirakan arah dari setiap blok, maka blok - blok tanpa informasi yang signifikan akan dibuang dengan menggunakan rumus sebagai berikut :

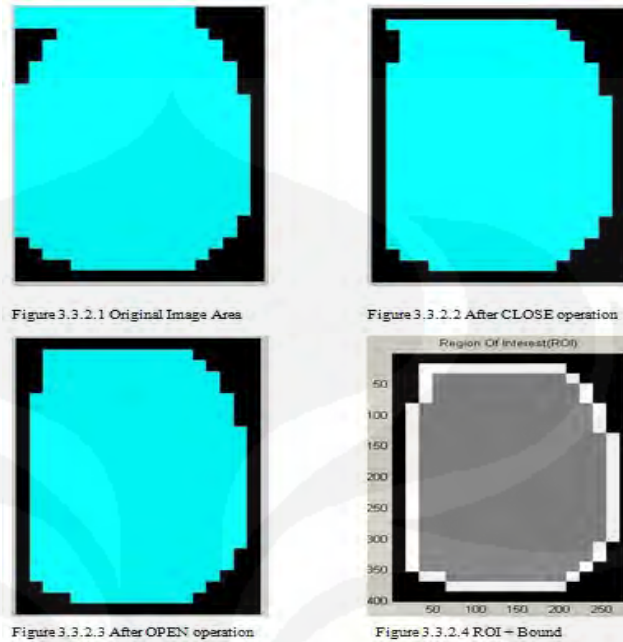
$$E = \{2 \sum \sum (gx * gy) + \sum \sum (gx^2 - gy^2)\} / W * W * \sum \sum (gx^2 - gy^2) \quad (3.6)$$

Gambar diatas menunjukkan arah dari citra tersebut mengikuti diagram dengan asumsi disana hanya ada satu sidik jari dari setiap image-nya. [3]

(c) Ekstrak ROI dengan menggunakan operasi morfologi

Dua operasi morfologi yang digunakan adalah '*OPEN*' dan '*CLOSED*'. Dimana *OPEN* adalah menambahkan citra dan membuang puncak atau tepi dari gangguan di background, sedangkan *CLOSE* adalah operasi yang dapat menyusutkan citra dan membagi kedalam lubang kecil.[9]

Tahapan ini penting karena membantu mempercepat proses *enrollment*, identifikasi serta verifikasi sehingga bisa membantu mempercepat absensi yang digunakan di pabrik pabrik yang memiliki jumlah karyawan yang banyak. Berikut ini algoritma listing program untuk *handle region of interest* (ROI): [3]



Gambar 3.6 Beberapa tipe ROI

```

%=====
% Fungsi Call Back ketika button ROI
%=====
function pb_roi_Callback(hObject, eventdata,
handles)

guidata(hObject,handles);
originalimage = handles.image1;
skeletonimage = handles.image7;
guidata(hObject,handles);

Kopen=imclose(skeletonimage,strel('square',7));
disp(size(skeletonimage));
KopenClean= imfill(Kopen,'holes');
KopenClean=bwareaopen(KopenClean,5);

KopenClean([1 end],:)=0;
KopenClean(:,[1 end])=0;
ROI=imerode(KopenClean,strel('disk',10));
disp(size(KopenClean));

%% Suppress extrema minutea
% Once we defined the ROI, we can suppress minutea
external to this ROI.

guidata(hObject,handles);
CentroidFinX2 = handles.CentroidFinX2;
CentroidFinY2 = handles.CentroidFinY2;
CentroidSepX2 = handles.CentroidSepX2;

```

```

CentroidSepY2 = handles.CentroidSepY2;
guidata(hObject,handles);
[m,n]=size(handles.image3(:,:,1));
disp([CentroidFinX2,CentroidFinY2]);
indFin=sub2ind([n,m],CentroidFinX2,CentroidFinY2);
disp(size(ROI));
disp(size(handles.image3(:,:,1)));
Z=zeros(n,m);
Z(indFin)=1;
ZFin=Z.*ROI';
[CentroidFinX2,CentroidFinY2]=find(ZFin);

indSep=sub2ind([n,m],CentroidSepX2,CentroidSepY2);
Z=zeros(n,m);
Z(indSep)=1;
ZSep=Z.*ROI';
[CentroidSepX2,CentroidSepY2]=find(ZSep);

axes(handles.imageproses);
imshow(originalimage);
hold on
image(255*ROI);
alpha(0.5);
plot(CentroidFinX2,CentroidFinY2,'ro','linewidth',
2);
plot(CentroidSepX2,CentroidSepY2,'go','linewidth',
2);
hold off
guidata(hObject,handles);
handles.CentroidFinX3 = CentroidFinX2;
handles.CentroidFinY3 = CentroidFinY2;
handles.CentroidSepX3 = CentroidSepX2;
handles.CentroidSepY3 = CentroidSepY2;
guidata(hObject,handles);

```

f. *Thinning*

Thinning atau penipisan adalah suatu proses pengurangan objek dalam citra digital hingga ukuran minimal yang diperlukan untuk pengenalan objek tersebut oleh komputer, hasilnya dikenal dengan skeleton (kerangka) dan proses thinning kadang-kadang dikenal juga sebagai skeletonisasi. Algoritma thinning dikelompokkan dalam dua kelompok utama yakni *iterative (pixel based)* dan *non- iterative (non pixel-based)*. Untuk kelompok *iterative* terdiri dari *sequential* dan *parallel*, sedangkan yang *non-iterative* terdiri dari *medial axis transform*, *line following* dan lain-lain.[3] Meskipun secara umum

proses thinning secara berulang akan melakukan proses penghapusan pada titik-titik tepi suatu area, beberapa hal harus diperhatikan :

- (1) tidak menghapus atau menghilangkan titik akhir.
- (2) tidak memutuskan keterhubungan.
- (3) dan tidak terjadi pengikisan yang terlalu berlebihan.

Untuk mendapatkan titik khusus dengan mudah digunakan proses Thinning untuk mendapatkan kerangka citra sidik jari kemudian mengekstraksi titik minuteda dari kerangka tersebut (*Skeleton*).[3]

Berikut ini Listing Program dari proses thinning: [3]

```

%=====
% Fungsi Call Back ketika button thinning di klik
%=====
function pb_thining_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
image6 = handles.image5;
guidata(hObject,handles)

Skeleton=bwmorph(~image6, 'thin', 'inf');
disp(size(image6));
disp(size(Skeleton));
set(handles.rb_skeletonimage, 'enable', 'on');

image7= Skeleton;
image8 = image7*255;

guidata(hObject,handles);
axes(handles.imageproses);
imagesc(image8);
guidata(hObject,handles);

guidata(hObject,handles);
handles.image7 = image7;
guidata(hObject,handles);
disp(size(image7));

```

g. Mencari Minuteda

Pada dasarnya, minuteda merupakan titik - titik yang ada di dalam guratan sidik jari manusia, dimana titik - titik tersebut dipisahkan menjadi dua titik *termination* dan *bifurcation*. [1] Pada tahapan ini, setelah citra sidik jari dari sidik jari di binerisasi dan juga mengalami thinning, maka akan di tentukan titik minutedanya dengan cara membentuk citra ke dalam koordinat x,y dengan nilai 300 x 300 (W x

W). [2] Setelah itu, dengan perhitungan gradien dari G_x dan G_y untuk setiap pixel dalam setiap blok dengan perkiraan orientasi lokal. Dengan W adalah ukuran blok, maka G_x dan G_y adalah nilai gradien dalam arah X dan Y yang akan menentukan letak minutea dari citra sidik jari. Untuk algoritma penentuan minuteanya sebagai berikut: [3]

```

%=====
% Fungsi Call Back ketika button Mencari Minutea di klik
%=====
function pb_mencariminutea_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
image9 = handles.image7;
guidata(hObject,handles);

axes(handles.imageproses);
imshow(255*image9);

fun=@minutie;
L = nfilter(image9,[3 3],fun);
disp(size(L));
axes(handles.imageproses);
%% Termination
LFin=(L==1);
LFinLab=bwlabel(LFin);
propFin=regionprops(LFinLab,'Centroid');
CentroidFin=round(cat(1,propFin(:).Centroid));
CentroidFinX=CentroidFin(:,1);
CentroidFinY=CentroidFin(:,2);
axes(handles.imageproses);
hold on
plot(CentroidFinX,CentroidFinY,'ro')

%% Bifurcation

LSep=(L==3);
LSepLab=bwlabel(LSep);
propSep=regionprops(LSepLab,'Centroid','Image');
CentroidSep=round(cat(1,propSep(:).Centroid));
CentroidSepX=CentroidSep(:,1);
CentroidSepY=CentroidSep(:,2);
plot(CentroidSepX,CentroidSepY,'go');
hold off

guidata(hObject,handles);
handles.CentroidFinX = CentroidFinX;
handles.CentroidFinY = CentroidFinY;
handles.CentroidSepX = CentroidSepX;
handles.CentroidSepY = CentroidSepY;
guidata(hObject,handles);

```

Dimana pada penentuan *minutae* dipisahkan menjadi dua, yaitu *termination* dan *bifurcation*, *termination*nya untuk titik akhir sedangkan *bifurcation* adalah titik percabangan.[1] Untuk titik percabangan juga terdapat titik percabangan satu, dua, dan tiga, dimana titik - titik tersebut akan terlihat didalam tanda berwarna, tanda tersebut memiliki letak/posisi dengan koordinat cartesius (X dan Y). [2]

h. Menghapus *minutae* yang salah

Setelah titik *minutae* ditemukan dari adanya titik - titik *termination* dan *bifurcation*, maka semua titik - titik tersebut ada yang ambigu (tidak jelas) dimana titik - titik yang di anggap tidak penting, seperti titik *termination* yang ada di luar atau garis tepi sidik jari, maka *minutae* - *minutae* yang tidak diperlukan harus dibuang agar tidak menyebabkan kesalahan dalam proses identifikasi maupun verifikasi.[2] Berikut ini listing program perangkat lunak Aplikasi Teknik dari tahap *Menghapus Minutae*: [3]

```

%=====
% Fungsi Call Back ketika button removemuthea
%=====
function pb_removemuthea_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
CentroidFinX1 = handles.CentroidFinX;
CentroidFinY1 = handles.CentroidFinY;
CentroidSepX1 = handles.CentroidSepX;
CentroidSepY1 = handles.CentroidSepY;
guidata(hObject,handles);

D=6;
%% Process 1
Distance=DistEuclidian([CentroidSepX1
CentroidSepY1],[CentroidFinX1 CentroidFinY1]);
SpuriousMinutae=Distance<D;
[i,j]=find(SpuriousMinutae);
CentroidSepX1(i)=[];
CentroidSepY1(i)=[];
CentroidFinX1(j)=[];
CentroidFinY1(j)=[];

%% Process 2
Distance=DistEuclidian([CentroidSepX1 CentroidSepY1]);
SpuriousMinutae=Distance<D;
[i,j]=find(SpuriousMinutae);
CentroidSepX1(i)=[];

```

```

CentroidSepY1(i)=[];

%% Process 3
Distance=DistEuclidian([CentroidFinX1 CentroidFinY1]);
SpuriousMinutae=Distance<D;
[i,j]=find(SpuriousMinutae);
CentroidFinX1(i)=[];
CentroidFinY1(i)=[];

guidata(hObject,handles);
image9 = handles. image7;
guidata(hObject,handles);

axes(handles.imageproses);
imshow(255*image9);

hold on
plot(CentroidFinX1,CentroidFinY1,'ro');
plot(CentroidSepX1,CentroidSepY1,'go');
hold off

guidata(hObject,handles);
handles.CentroidFinX2 = CentroidFinX1;
handles.CentroidFinY2 = CentroidFinY1;
handles.CentroidSepX2 = CentroidSepX1;
handles.CentroidSepY2 = CentroidSepY1;
guidata(hObject,handles);
%-----

```

Setelah titik - titik minutea didapatkan, maka minutea tersebut akan di simpan kedalam database yang nantinya akan digunakan untuk proses matching / pencocokan pada saat dilakukan proses verifikasi.[2]

i. Orientasi

Pada tahap orientasi merupakan tahap penentuan titik minutea kedalam derajat cartesius dengan nilai X dan Y serta radian. Penentuan titik - titik minutea tersebut mengikuti listing program berikut: [3]

```

%=====
% Fungsi Call Back ketika button Orientation
%=====
function pb_oreintation_Callback(hObject, eventdata,
handles)
Table=[3*pi/4 2*pi/3 pi/2 pi/3 pi/4
        5*pi/6 0 0 0 pi/6
        pi 0 0 0 0
        -5*pi/6 0 0 0 -pi/6
        -3*pi/4 -2*pi/3 -pi/2 -pi/3 -pi/4];

```



```

guidata(hObject,handles);
CentroidFinX3 = handles.CentroidFinX3;
CentroidFinY3 = handles.CentroidFinY3;
CentroidSepX3 = handles.CentroidSepX3;
CentroidSepY3 = handles.CentroidSepY3;
originalimage = handles.image1;
skeletonimage = handles.image7
guidata(hObject,handles);

%% Termination Orientation
% We have to find the orientation of the termination.
% For finding that, we analyze the position of the pixel
on the boundary of
% a 5 x 5 bounding box of the termination. We compare
this position to the
% Table variable. The Table variable gives the angle in
radian.
for ind=1:length(CentroidFinX3)
    Klocal=skeletonimage(CentroidFinY3(ind)-
2:CentroidFinY3(ind)+2,CentroidFinX3(ind)-
2:CentroidFinX3(ind)+2);
    Klocal(2:end-1,2:end-1)=0;
    [i,j]=find(Klocal);
    if length(i)~=1
        CentroidFinY3(ind)=NaN;
        CentroidFinX3(ind)=NaN;
        OrientationFin3(ind,1)=NaN;
    else
        OrientationFin3(ind,1)=Table(i,j);
    end
end
dxFin=sin(OrientationFin3)*5;
dyFin=cos(OrientationFin3)*5;
axes(handles.imageproses)
imshow(skeletonimage)
hold on
plot(CentroidFinX3,CentroidFinY3,'ro','linewidth',2)
plot([CentroidFinX3 CentroidFinX3+dyFin],...
    [CentroidFinY3 CentroidFinY3-
dxFin'],'r','linewidth',2)

%% Bifurcation Orientation
% For each bifurcation, we have three lines. So we
operate the same
% process than in termination case three times.
for ind=1:length(CentroidSepX3)
    Klocal=skeletonimage(CentroidSepY3(ind)-
2:CentroidSepY3(ind)+2,CentroidSepX3(ind)-
2:CentroidSepX3(ind)+2);
    Klocal(2:end-1,2:end-1)=0;
    [i,j]=find(Klocal);
    if length(i)~=3
        CentroidSepY3(ind)=NaN;
        CentroidSepX3(ind)=NaN;
    end
end

```

```

        OrientationSep3(ind)=NaN;
    else
        for k=1:3
            OrientationSep3(ind,k)=Table(i(k),j(k));
            dxSep(ind,k)=sin(OrientationSep3(ind,k))*5;
            dySep(ind,k)=cos(OrientationSep3(ind,k))*5;
        end
    end
end

plot(CentroidSepX3,CentroidSepY3,'go','linewidth',2)
OrientationLinesX=[CentroidSepX3
CentroidSepX3+dySep(:,1);CentroidSepX3
CentroidSepX3+dySep(:,2);CentroidSepX3
CentroidSepX3+dySep(:,3)'];
OrientationLinesY=[CentroidSepY3 CentroidSepY3-
dxSep(:,1);CentroidSepY3 CentroidSepY3-
dxSep(:,2);CentroidSepY3 CentroidSepY3-dxSep(:,3)'];
plot(OrientationLinesX,OrientationLinesY,'g','linewidth'
,2)
hold off

guidata(hObject,handles);
handles.CentroidFinX4 = CentroidFinX3;
handles.CentroidFinY4 = CentroidFinY3;
handles.CentroidSepX4 = CentroidSepX3;
handles.CentroidSepY4 = CentroidSepY3;
handles.OrientationFin4 = OrientationFin3;
handles.OrientationSep4 = OrientationSep3;
guidata(hObject,handles);

%=====
% Fungsi Call Back ketika button Orientation
%=====
function pb_validation_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
CentroidFinX4 = handles.CentroidFinX4;
CentroidFinY4 = handles.CentroidFinY4;
CentroidSepX4 = handles.CentroidSepX4;
CentroidSepY4 = handles.CentroidSepY4;
OrientationFin4=handles.OrientationFin4;
OrientationSep4=handles.OrientationSep4;
originalimage = handles.image1;
skeletonimage = handles.image7
guidata(hObject,handles);
ValidationGUI(originalimage,CentroidFinX4,CentroidFinY4,
OrientationFin4,CentroidSepX4,CentroidSepY4,OrientationS
ep4);

%=====
% Fungsi Call Back ketika button set minutea di klik
%=====

function pb_setminutea_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);

```

```

CentroidSepX5 = handles.CentroidSepX4;
CentroidSepY5 = handles.CentroidSepY4;
OrientationSep5 = handles.OrientationSep4;
MinutiaSep=[CentroidSepX5 CentroidSepY5
OrientationSep5];
MinutiaSep=MinutiaSep(~isnan(MinutiaSep(:,1)),:);
CentroidFinX5 = handles.CentroidFinX4;
CentroidFinY5 = handles.CentroidFinY4;
OrientationFin5 = handles.OrientationFin4;
MinutiaFin=[CentroidFinX5 CentroidFinY5
OrientationFin5];
MinutiaFin=MinutiaFin(~isnan(MinutiaFin(:,1)),:);

prompt = {'Masukkan File Name:'};
dlg_title = 'Input Untuk Minutia export';
num_lines = 1;
def = {'Username'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
%saveMinutia(answer{1},MinutiaFin,MinutiaSep);
saveMinutia(char(answer{1}),MinutiaFin,MinutiaSep);
guidata(hObject,handles);

%=====
% Fungsi Call Back ketika button Matching di klik
%=====

function pb_matching_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
[templatefile , pathname]= uigetfile('*.dat','Open An
Fingerprint template file');
if templatefile ~= 0
cd(pathname);
finger1=load(char(templatefile));
end;
[templatefile , pathname]= uigetfile('*.dat','Open An
Fingerprint template file');
if templatefile ~= 0
cd(pathname);
finger2=load(char(templatefile));
end;
percent_match=match_end(finger1,finger2,10);

```

j. Validation

Validation merupakan tahapan terakhir sebelum titik munita disimpan ke dalam *.text atau *.dat. Validasi adalah penggambaran letak - letak dari minutea yang didapat dalam proses ekstraksi ke dalam image yang sebenarnya (*original Images*).[1] Pada program aplikasi Ekstraksi sidik jari ini mempunyai 2 Model GUI yang pertama NewFingerprint dan kedua ValidationGUI. Untuk model GUI Pertama sudah dibahas

diatas, sedangkan GUI yang kedua merupakan penggambaran titik - titik minutea ke dalam koordinat X dan Y. berikut ini Listing programnya : [3]

```

%=====
% Fungsi Call Back ketika button Orientation
%=====
function pb_validation_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
CentroidFinX4 = handles.CentroidFinX4;
CentroidFinY4 = handles.CentroidFinY4;
CentroidSepX4 = handles.CentroidSepX4;
CentroidSepY4 = handles.CentroidSepY4;
OrientationFin4=handles.OrientationFin4;
OrientationSep4=handles.OrientationSep4;
originalimage = handles.image1;
skeletonimage = handles.image7
guidata(hObject,handles);
ValidationGUI(originalimage,CentroidFinX4,CentroidFinY4,
OrientationFin4,CentroidSepX4,CentroidSepY4,OrientationS
ep4);

```

Untuk melihat Listing program detail ValidationGUI dapat dilihat dihalaman Lampiran.

k. Simpan Minutea

Setelah letak dari titik minutea didapat beserta derajat radiannya maka titik - titik tersebut disimpan ke dalam file.dat yang nantinya akan digunakan sebagai database sistem, berikut ini listing program : [3]

```

%=====
% Fungsi Call Back ketika button Simpan minutea di klik
%=====
function pb_setminutea_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
CentroidSepX5 = handles.CentroidSepX4;
CentroidSepY5 = handles.CentroidSepY4;
OrientationSep5 = handles.OrientationSep4;
MinutiaSep=[CentroidSepX5 CentroidSepY5
OrientationSep5];
MinutiaSep=MinutiaSep(~isnan(MinutiaSep(:,1)),:);
CentroidFinX5 = handles.CentroidFinX4;
CentroidFinY5 = handles.CentroidFinY4;
OrientationFin5 = handles.OrientationFin4;

```

```

MinutiaFin=[CentroidFinX5 CentroidFinY5
OrientationFin5];
MinutiaFin=MinutiaFin(~isnan(MinutiaFin(:,1)),:);

prompt = {'Masukkan File Name:'};
dlg_title = 'Input Untuk Minutia export';
num_lines = 1;
def = {'Username'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
%saveMinutia(answer{1},MinutiaFin,MinutiaSep);
saveMinutia(char(answer{1}),MinutiaFin,MinutiaSep);
guidata(hObject,handles);

```

1. Matching

Proses pencocokan sidik jari (*Matching*) pada sebuah sistem pemrograman memerlukan sebuah metode, yang pertama metode *Aligment Stage* dengan memberikan dua sidik jari untuk dicocokkan, pilih sala satu minutea dari setiap image. Kemudian menghitung kedekatan dari dua buah ridge yang berdekatan untuk dijadikan sebagai keterangan pencocokan minutea. Yang kedua adalah *Matching Stage*, setelah kedua titik minutea didapatkan melalui tranformasi maka titik minutea tersebut dapat dicocokkan dengan menganggap kedua titik yang berdekatan memiliki letak dan arah yang sama dengan menggunakan metode transformasi elastic.[10] Berikut ini listing programnya : [3]

```

%=====
% Fungsi Call Back ketika button Matching di klik
%=====
function pb_matching_Callback(hObject, eventdata,
handles)
guidata(hObject,handles);
[templatefile , pathname]= uigetfile('*.dat','Open An
Fingerprint template file');
if templatefile ~= 0
cd(pathname);
finger1=load(char(templatefile));
end;
[templatefile , pathname]= uigetfile('*.dat','Open An
Fingerprint template file');
if templatefile ~= 0
cd(pathname);
finger2=load(char(templatefile));
end;
percent_match=match_end(finger1,finger2,10);

```

```

function
[percent_match]=match_end(template1,template2,edgeWidth,
noShow)
% MATCH_END Fingerprint Minutia Matcher Based on Ridge
Alignment
% Decompose the template file into minutea and ridge
matrixes seperately
if or(edgeWidth == 0,isempty(edgeWidth))
    edgeWidth=10;
end;

if or(isempty(template1), isempty(template2))
    percent_match = -1;
else
    length1 = size(template1,1);
    minu1 = template1(length1,3);
    real_end1 = template1(1:minu1,:);
    ridgeMap1= template1(minu1+1:length1,:);

    length2 = size(template2,1);
    minu2 = template2(length2,3);
    real_end2 = template2(1:minu2,:);
    ridgeMap2= template2(minu2+1:length2,:);

    ridgeNum1 = minu1;
    minuNum1 = minu1;
    ridgeNum2 = minu2;
    minuNum2 = minu2;

    max_percent=zeros(1,3);

    for k1 = 1:minuNum1
        %minuNum2

        %=====
        %calculate the similarities between ridgeMap1(k1) and
        ridgeMap(k2)
        %choose the current two minutia as origins
        %=====

        newXY1 =
        MinuOriginTransRidge(real_end1,k1,ridgeMap1);
        for k2 = 1:minuNum2

            newXY2 =
            MinuOriginTransRidge(real_end2,k2,ridgeMap2);

            %choose the minimum ridge length
            compareL = min(size(newXY1,2),size(newXY2,2));
            %compare the similarity certainty of two ridge
            eachPairP =
            newXY1(1,1:compareL).*newXY2(1,1:compareL);
            pairPSquare = eachPairP.*eachPairP;
            temp = sum(pairPSquare);

            ridgeSimCoef = 0;

```

```

    if temp > 0
        ridgeSimCoef = sum(eachPairP)/( temp^.5 );
    end;

    if ridgeSimCoef > 0.8
        %transfer all the minutia in two fingerprint
        based on
        %the reference pair of minutia
        fullXY1=MinuOrigin_TransAll(real_end1,k1);
        fullXY2=MinuOrigin_TransAll(real_end2,k2);

        minuN1 = size(fullXY1,2);
        minuN2 = size(fullXY2,2);
        xyrange=edgeWidth;
        num_match = 0;

        %if two minutia are within a box with width 20
        and height 20,
        %they have small direction variation pi/3
        %then regard them as matched pair

        for i=1:minuN1
            for j=1:minuN2
                if (abs(fullXY1(1,i)-fullXY2(1,j))<xyrange &
                    abs(fullXY1(2,i)-fullXY2(2,j))<xyrange)
                    angle = abs(fullXY1(3,i) - fullXY2(3,j) );
                    if or (angle < pi/3, abs(angle-pi)<pi/6)
                        num_match=num_match+1;
                        break;
                    end;
                end;
            end;
        end;

        % get the largest matching score
        current_match_percent=num_match;
        if current_match_percent > max_percent(1,1);
            max_percent(1,1) = current_match_percent;
            max_percent(1,2) = k1;
            max_percent(1,3) = k2;
        end;
        num_match = 0;

    end;
end;
end;

percent_match = max_percent(1,1)*100/minuNum1;
end;

%if function is called in GUI mode, popup out the
message box
%for final result
if nargin == 3

```

```

text=strcat('Maksimal dari persentase pencocokan
adalah',num2str(percent_match),'%');
msgbox(text);
end;

```

3.3. Sistem Identifikasi dan Verifikasi Sidik Jari

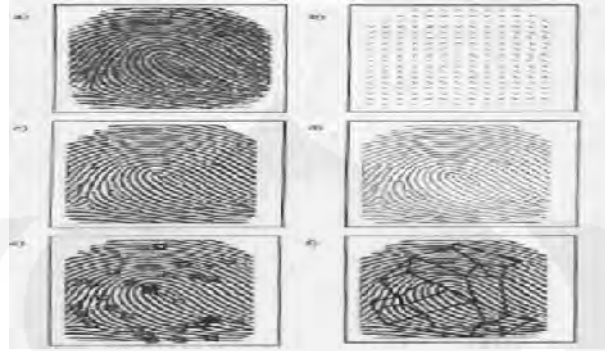
Sistem Identifikasi sidik jari merupakan pola pencocokan sidik jari yang sudah kita daftarkan ke dalam sistem database, dengan sidik jari yang mau di cocokan. Dimana nantinya akan kita kenalin dengan membandingkan sidik jari yang baru dengan data sidik jari yang ada di sistem.[1] Untuk cara mencocokkannya ada dua cara, pertama membandingkan 2 sidik jari langsung (*one to one*) dan yang kedua membandingkan sidik jari dengan data yang ada di database (*one to many*). Jika persentase sidik jari menunjukkan angka hamper mendekati 100% maka kedua sidik jari tersebut dianggap cocok.[2] Berikut struktur sistem identifikasi dan verifikasi sidik jari :

a. Sistem identifikasi sidik jari

Proses peng-*identifikasian* pola sidik jari pada dasarnya adalah sama, walaupun terdapat banyak algoritma peng-*identifikasian* pola sidik jari. Kesamaan proses pengidentifikasian tersebut adalah sebagai berikut:

- a) Suatu citra dari sidik jari *dicapture*.
- b) Citra sidik jari tersebut diproses untuk memecahnya ke dalam *ridge* dan *valley*
- c) Pola *ridge* diperkirakan untuk menemukan adanya ketidak-teraturan dalam pola tersebut, yang dikenal sebagai *minutea* atau kedalam pola yang dikenal dengan *pattern*. [9] Beberapa contoh umum dari *minutea* dikenal sebagai *crossover*, *cores*, *bifurcation*, *ridge ending*, *island*, dan *delta*.
- d) Lokasi relatif dari *minutea* diukur, menghasilkan sebuah pola yang dikenal sebagai “*graph minutea*” yang membentuk dasar dari template sidik jari. [7]

Contoh visual suatu algoritma pemrosesan sidik jari ditunjukkan dalam gambar berikut:



Gambar 3.7 Urutan langkah Proses Sidik Jari : a) Original (pattern), b) Orientation, c) Binerisasi, d) Thining, e) Minutea, dan Minutea graph

Dari sekian banyak algoritma yang ada, seperti di jelaskan pada gambar diatas, hanya ada dua yang sering dipakai/populer dipakai yaitu algoritma menggunakan sidik jari original dengan *metode pattern* (pola) dan *metode minutea* (titik).[2] Langkah - langkah pengenalan dalam suatu sistem identifikasi sidik jari otomatis adalah meliputi:

1. Akuisisi data untuk pembacaan sidik jari
2. Ekstraksi *pattern* / *minutea* untuk identifikasi
3. Pencocokan dan pengambilan keputusan

Jadi sistem identifikasi sidik jari otomatis dapat direduksi menjadi masalah pencocokan posisi titik - titik minutea antara dua pola sidik jari sebesar mungkin. Jika tidak ada deformasi seperti pergeseran, perputaran atau masalah non-linier, maka setiap titik minutea yang terlokalisasi secara tepat akan dibandingkan antara kedua pola untuk menghasilkan suatu identifikasi. Tetapi kenyataannya terjadi deformasi, adanya sejumlah minutea palsu dan sejumlah minutea yang hilang karena tidak terdeteksi. [6] Oleh karena beberapa model simulasi sidik jari digunakan untuk membantu restrukturisasi garis alur pola sidik jari yang mengalami deformasi. Sidik jari dibaca menggunakan *scanner* sidik jari secara *live-scan* (langsung) atau *off-line* (tinta). Citra sidik jari yang dihasilkan kemudian dilakukan prapengolahan, yaitu diperiksa dengan suatu orientasi medan untuk estimasi posisi.[9] Setelah dilakukan prapengolahan, kemudian dilakukan ekstraksi garis-garis alur pola sidik jari untuk menentukan posisi - posisi minutea yang menggunakan metode

minutea. [8] Sedangkan yang menggunakan metode pattern, sistem identifikasinya dengan cara membagi per blok dari citra hasil scanner sidik jari, baik secara langsung (menggunakan livescan) atau secara tidak langsung (menggunakan tinta, yang nantinya pembagian dibedakan ke dalam *shape of phase, pitch and ridges*) [6]



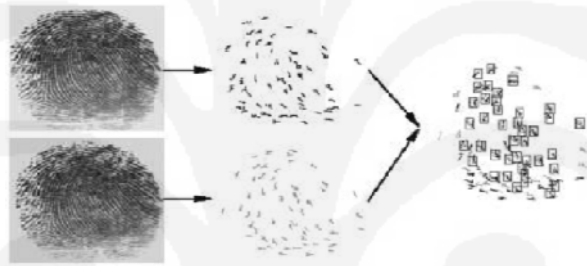
Gambar 3.8 Sistem identifikasi sidik jari

Berikut ini cara identifikasi menggunakan metode minutea, citra sidik jari yang telah mengalami estimasi posisi akan diproses melalui ekstraksi garis-garis alur yang dilakukan melalui tahap binarisasi dan penipisan homotopik.[1] Binarisasi dari citra tingkat keabuan menjadi citra hitam putih direalisasikan dengan suatu batas ambang. Penipisan homotopik diterapkan ke citra biner dengan suatu *filter morfologi* keluarga L, yaitu penipisan obyek secara simetris yang hasil akhir adalah diperolehnya garis tipis yang menggambarkan bentuk dasar pola garis alur sidik jari. Operasi penipisan dengan transformasi homotopi, yaitu agar tetap menjaga konektivitas atau keterhubungan antara komponen-komponen pixel yang ditipiskan, sehingga diperoleh rangka dasar yang mencerminkan garis alur pola sidik jari pada citra. Berikut ini proses tahapan algoritma minutea:



Gambar 3.9 Hasil tahapan-tahapan Algoritma Ekstraksi Minutea

Ekstraksi minuteda diperoleh dengan menggunakan pendekatan matriks dan filter bentuk dengan topologinya dirancang untuk mengeksploitasi dua sifat minuteda, yaitu untuk mendeteksi akhir garis alur atau titik - titik percabangan. Citra awal berhubungan dengan arah garis - garis alur pola sidik jari yang kemudian disekat - sekat kedalam daerah - daerah yang terhubung homogen. Dari terminasi dan percabangan garis alur pola sidik jari, maka diperoleh titik - titik minuteda.[10] Peta dari titik - titik minuteda disimpan sebagai informasi unik berhubungan dengan sidik jari individual, maka teknologi algoritma pengidentifikasian sidik jari ini disebut dengan pengolahan berbasis minuteda.[1]



Gambar 3.10 Pencocokan minuteda sidik jari

Toleransi terhadap pergeseran dan perputaran sidik jari dikoreksi dengan *transformasi Hough*, tetapi menyebabkan pemeriksaan menjadi lambat. Dalam satu sidik jari biasanya diperoleh 30 sampai 40 minuteda. Dari pengamatan diperoleh kenyataan bahwa dari dua individu paling banyak hanya ditemukan kurang dari delapan titik-titik minuteda yang mempunyai posisi yang sama, yaitu hanya 5 sampai 7 minuteda yang sama. Umumnya sidik jari dari tangan yang sama memberikan 20 sampai 40 minuteda yang sama. Kecepatan pencocokan sidik jari sekitar 0,2 sampai 0,5 detik, dimana algoritma dapat mencocokkan 5000 sidik jari per detiknya.[6]

BAB IV

PENGUJIAN DAN ANALISA SISTEM

Untuk mengetahui keberhasilan dari algoritma dan sistem yang dirancang, maka perlu dilakukan pengujian terhadap sistem yang dimiliki. Dalam bab ini akan dibahas mengenai proses pengujian yang dilakukan serta analisa dari hasil yang diperoleh.

4.1 Mekanisme Pengujian

Tujuan dari pengujian adalah untuk mengetahui fungsionalitas algoritma pemrograman serta sistem yang telah dirancang, waktu komputasi yang dibutuhkan, dan juga tingkat keberhasilan dan efektifitas metode. Tingkat keberhasilan menunjukkan seberapa besar kemampuan metode yang dikembangkan dalam memberikan bentuk serta pola minuteda dari sidik jari, dimana untuk kemudian digunakan dalam analisis sistem biometrik.

4.2 Spesifikasi Sistem

Implementasi perangkat keras dan perangkat lunak sangat menentukan keberhasilan dari pengujian algoritma yang telah dirancang. Berikut ini adalah uraian piranti yang digunakan dalam proses pengujian ini.

- **Perangkat Lunak**

Dalam Tugas Akhir ini sebagian proses menggunakan Aplikasi Teknik untuk mengolah citra biner sidik jari. Fungsi - fungsi yang digunakan terdapat dalam *Image Processing Toolbox* yang dijalankan di bawah sistem operasi Windows XP.

- **Perangkat Keras**

Pengambilan citra sidik jari dilakukan dengan menggunakan Fingerprint Reader Identix DFRU 200 dengan kualitas yang memadai. DFRU 200 merupakan salah satu fingerprint yang mempunyai kemampuan untuk melakukan livenesscaning

sidik jari. Sehingga format yang dihasilkan berupa format BMP. Berikut ini hasil scanning fingerscan menggunakan *Fingerprint Identix DFRU 200*.



Gambar 4.1 Hasil citra sidik jari menggunakan DFRU 200

4.3 Pengujian Algoritma dari sistem

Pengujian algoritma dari sistem identifikasi sidik jari menggunakan metode minuteda dilakukan untuk tiap - tiap citra dengan proses yang sama, kecuali pada bagian pengolahan awal citra. Tiap citra mengalami proses yang sedikit berbeda tergantung pada kebutuhan masing-masing. Semakin buruk kualitas citra untuk dianalisis, maka semakin banyak pemrosesan awal yang dilakukan pada citra. Berikut adalah hasil pengujian pemrosesan citra pada data citra sidik jari.

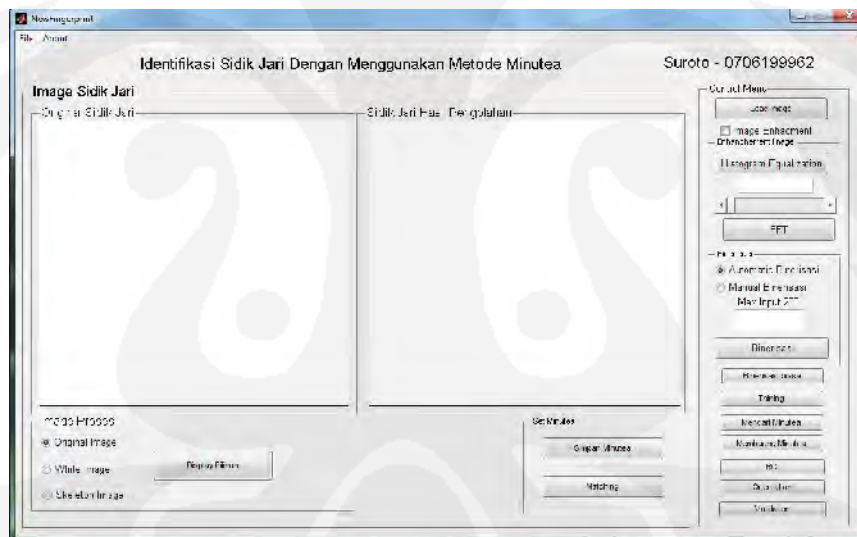
4.3.1 Pengujian Algoritma sidik jari menggunakan minuteda

Dalam proses identifikasi sidik jari menggunakan algoritma minuteda, dalam proses ini diperlukan beberapa sample sidik jari yang diperoleh dengan menggunakan fingerprint reader DFRU 200 merk Identik, dimana hasil yang diperoleh adalah livescanning dari citra sidik jari, sehingga harus dicapture ke dalam bentuk citra biner dengan dimensi $W \times W$. Pada dunia nyata, sesungguhnya terdapat tiga macam pola sidik jari yang berbeda untuk setiap individu, yaitu: (a) *Tipe Whorl*, (b) *Tipe Arch*, (c) *Tipe Loop (Right loop dan left loop)*. Sidik jari dengan tipe loop merupakan sidik jari yang terdistribusi paling banyak atau sebanyak 65 % dari keseluruhan populasi manusia.[10] Berikut ini contoh sampel sidik jari yang akan digunakan untuk penelitian ini dapat dilihat pada gambar berikut



Gambar 4.2 Sampel sidik jari pada proses pengujian dengan tiga variasi

Proses penentuan sidik jari dengan menggunakan metode Minutiae terbagi ke dalam beberapa tahapan, pertama citra dari sidik jari harus diubah ke dalam citra biner yang memiliki derajat keabuan dengan cara *Load images - His Equalization - FFT - Binerisasi - Thinning - ROI - Mencari Minutiae - Menghapus minutiae - Orientasi - Validasi - Save Minutiae - Matching*. berikut ini penjelasan dari beberapa tahapan dalam penentuan minutiae pada sidik jari. Berikut ini tampilan program NewFingerprint GUI:



Gambar 4.3 Main Program Newfingerprint

1. Load images

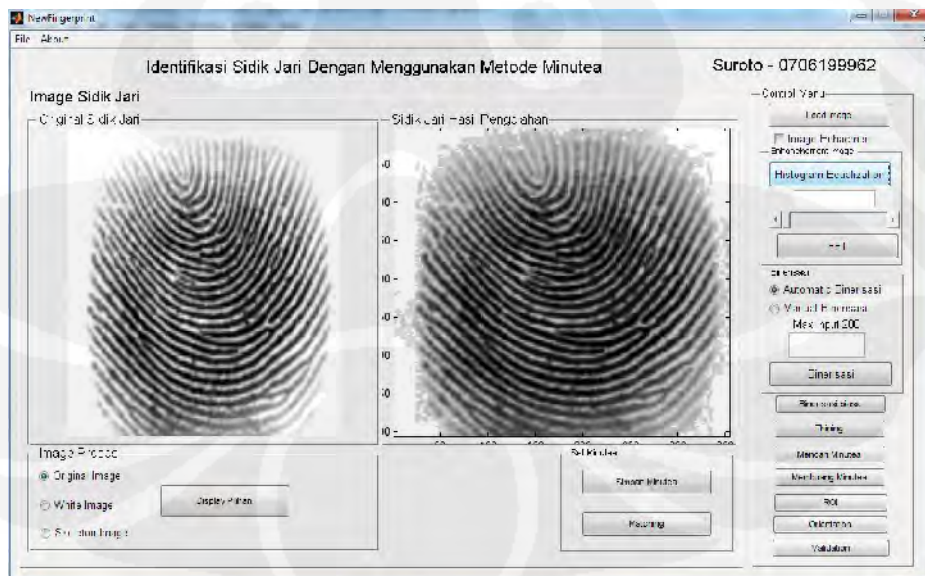
Pada setiap aplikasi pemrosesan citra, tahapan pertama yang harus dilakukan adalah memasukkan citra yang akan kita olah ke dalam aplikasi atau sistem yang kita rancang.



Gambar 4.4 Load image file ke program identifikasi sidik jari

2. Histogram Equalization

Histogram equalization merupakan metode yang digunakan untuk meningkatkan atau memperluas nilai pixel dari suatu citra biner serta untuk meningkatkan pemahaman dan informasi mengenai citra tersebut, model orisinal dari citra sidik jari adalah tipe bimodal. Berikut ini hasil dari prosesnya:[4]



Gambar 4.5 Hasil proses Images enhancement, Orisinal citra (kiri), Hasil his equalization (kanan)

3. FFT (Fourier Transformation)

Pada fingerprint Enhancement, selain menggunakan histogram equalization, juga menggunakan metode fourier transform [membagi citra menjadi blok kecil - kecil (32 x 32 pixel), dimana nantinya blok - blok kecil ini yang akan dimanfaatkan untuk penentuan titik minutea. berikut ini images hasil proses FFT:[5]



Gambar 4.6 Hasil proses Images enhancement, Image asli (kiri), Hasil FFT (kanan)

4. Binerisasi

Binerisasi merupakan tahapan awal dari proses penentuan titik minutea dengan mengubah derajat keabuan dari citra sidik jari, proses binerisasi dimulai dengan mengubah citra orisinil ke dalam citra biner/ hitam putih menggunakan threshold.[5] Berikut ini citra hasil binerisasi:

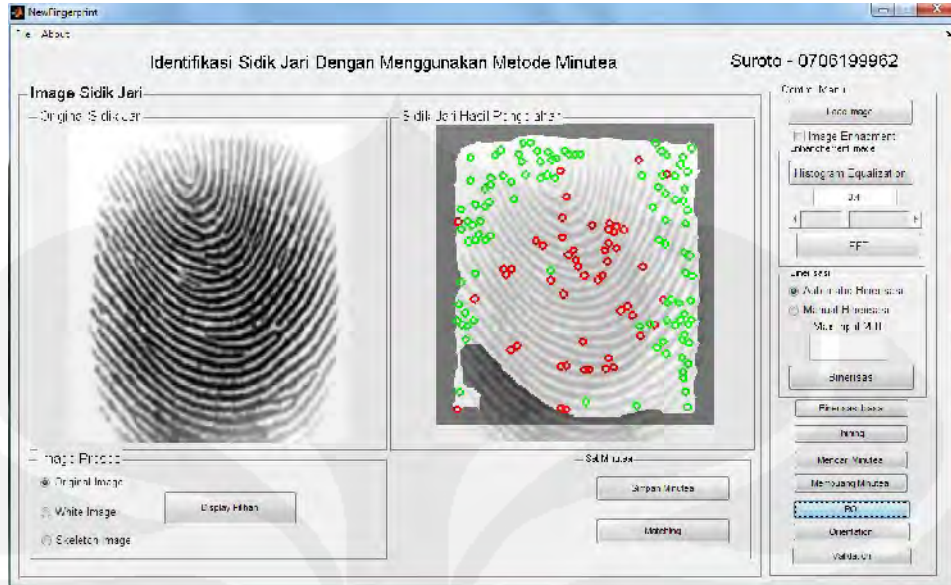


Gambar 4.7 Hasil proses binerisasi, image asli (kiri), hasil binerisasi (kanan)

Citra yang dihasilkan pada tahap binerisasi berupa citra biner/ citra dengan derajat keabuan, sehingga dihasilkan citra berupa hitam - putih.

5. Images Segmentasi (ROI)

Setelah citra sidik jari diubah kedalam citra biner, maka selanjutnya citra tersebut akan melalui proses *image segmentation* atau biasa disebut dengan *Region of interest* (ROI). Sedangkan untuk menentukan region of interest (ROI) juga terdapat dalam proses segmentasi dengan menggunakan operasi morfologi, berikut ini citra hasil dari ROI:



Gambar 4.8 Hasil proses segmentasi, Image asli (kiri),
Hasil ROI (kanan)

6. Thinning

Hasil citra biner yang diperoleh dalam proses binerisasi akan diubah kedalam bentuk garis (*rangka/skeleton*) agar tidak terjadi kesalahan kedalam penentuan titik minutae, sehingga proses thinning merupakan tahapan penting dalam penentuan titik minutae pada algoritma, berikut ini hasil dari proses thinning:



Gambar 4.9 Hasil proses ekstrak minutae, Image asli (kiri),
Hasil Thinning (kanan)

Dengan mengubah citra sidik jari kedalam garis akan membantu penentuan titik minutea, proses thinning digunakan untuk menghindari adanya kesalahan dalam penentuan titik minutea.

7. Mencari Minutea (Penentuan titik minutea)

Pada algoritma minutea yang didasari pada penentuan titik - titik termination dan bifurcation ini, ditentukan dengan pendekatan titik koordinat cartesius dengan nilai x dan y sebagai letak dari titik - titik minutea. pada proses thinning dimana citra biner diubah kedalam garis (*rangka/skeleton*) agar tidak terjadi kesalahan dalam penentuan titik - titik termination dan bifurcation, sehingga memudahkan dalam penentuan minuteanya, berikut ini hasil proses penentuan titik minutea:



Gambar 4.10 Hasil proses ekstrak minutea, Image asli (kiri), Hasil Ekstrak Minutea (kanan)

Titik warna hijau merupakan titik - titik percabangan (bifurcation), sedangkan titik - titik warna merah merupakan titik akhir (termination). Titik - titik minutea diatas akan diubah kedalam titik koordinat x dan y. Dari sekian banyak titik - titik minutea, ternyata ada juga titik - titik minutea yang dianggap tidak berguna karena titik - titik tersebut merupakan titik - titik yang berada diluar area (diluar region of interest / ROI). Berikut ini hasil seleksi titik minutea yang dianggap salah/dieliminasi. [10]



Gambar 4.11 Hasil proses post proseding, Image asli (kiri), Hasil menghapus minutae (kanan)

8. Orientation

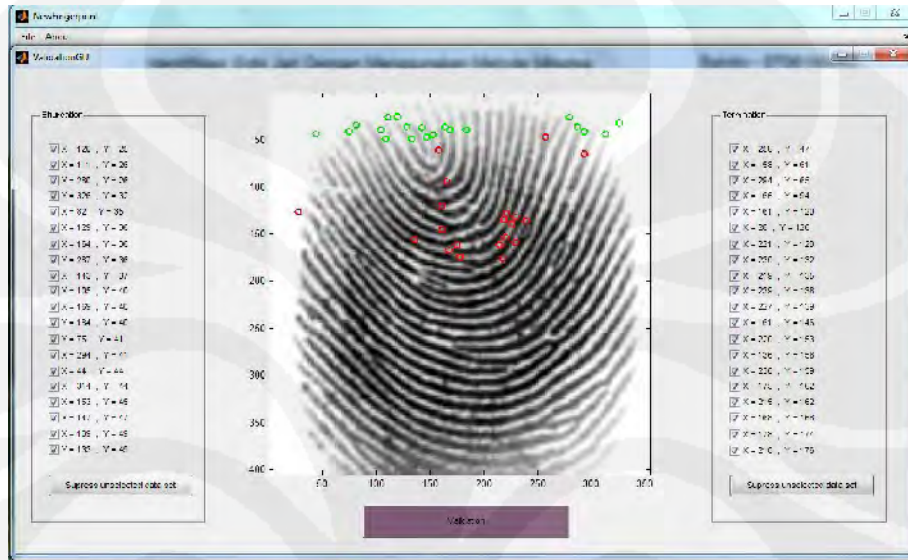
Pada tahap orientasi merupakan tahap penentuan titik minutae kedalam derajat cartesius dengan nilai X dan Y serta radian.[1] Penentuan titik - titik minutae tersebut dapat dilihat pada gambar dibawah:



Gambar 4.12 Hasil proses post proseding, Image asli (kiri), Hasil Orientasi (kanan)

9. Validation

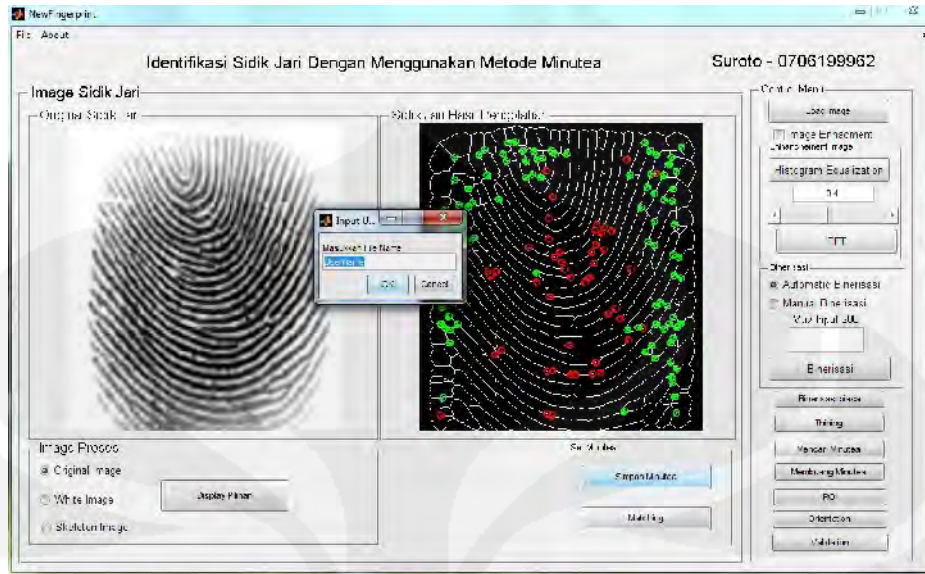
Validation merupakan tahapan terakhir sebelum titik minutea disimpan ke dalam *.text atau *.dat. Validasi adalah penggambaran letak - letak dari minutea yang didapat dalam proses ekstraksi ke dalam image yang sebenarnya (*original Images*). Hasil dari proses validasi dapat dilihat dibawah ini :



Gambar 4.13 Hasil proses post prosing, Image asli (kiri), Hasil Validasi (kanan)

10. Menyimpan Minutea

Setelah letak dari titik minutea didapat beserta derajat radiannya maka titik - titik tersebut disimpan ke dalam file.dat yang nantinya akan digunakan sebagai database sistem, berikut gambar hasil proses save minutea :



Gambar 4.14 Hasil proses menyimpan minutear

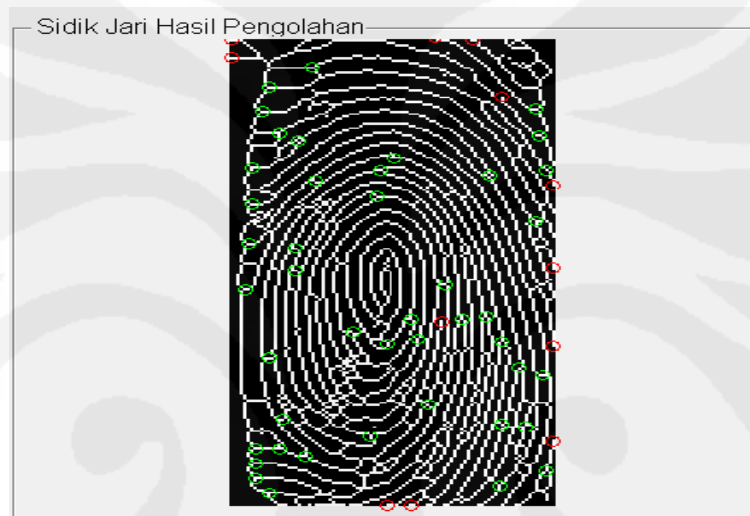
11. Matching

Proses pencocokan sidik jari (*Matching*) pada sebuah sistem pemrograman memerlukan dua buah sidik jari yang akan di lihat kemiripannya, berikut ini gambar hasil percobaan mencocokkan dua buah sidik jari menggunakan algoritma minutear:

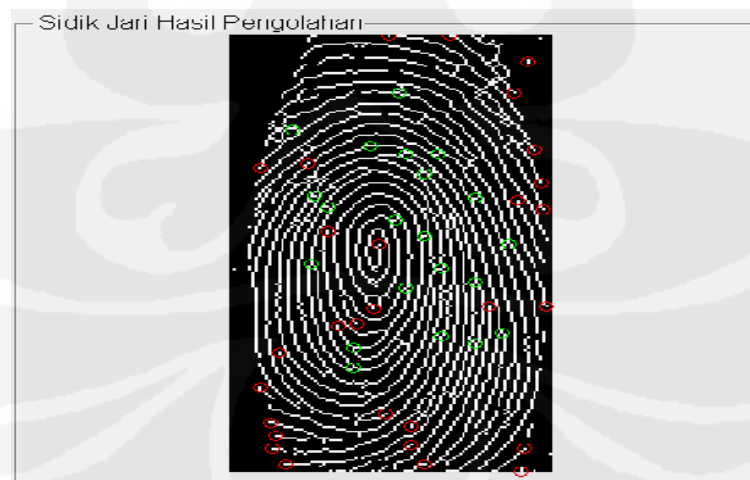


Gambar 4.15 Hasil proses *matching*

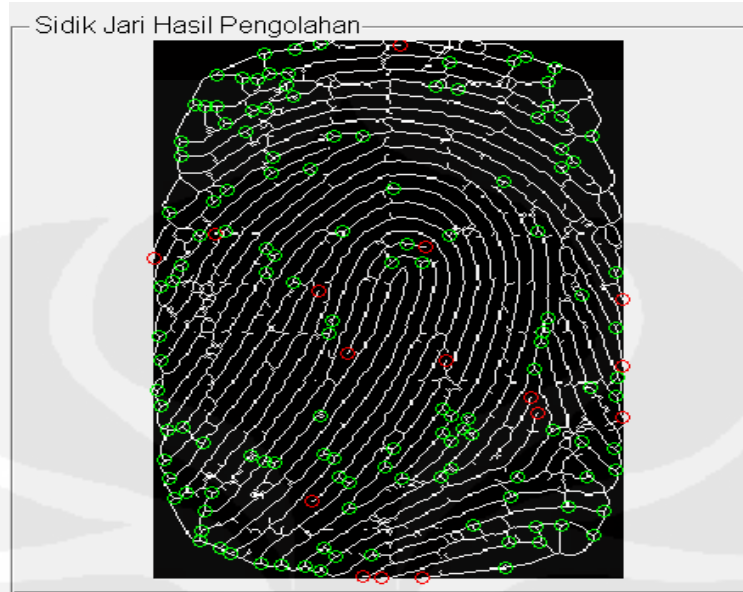
Pada program identifikasi sidik jari menggunakan algoritma minutia ini dibuat dua sistem binerisasi yang pertama binerisasi yang memanfaatkan hasil *outputan* (keluaran) dari *Image enhancement* dan yang kedua adalah binerisasi dari Image asli tanpa ada *image enhancement* untuk yang tanpa image enhancement sering mengalami *error*, ini dikarenakan informasi sidik jari yang dibutuhkan untuk proses pengenalan sidik jari dirasa kurang. Berikut ini hasil pengolahan sidik jari dengan menggunakan Image enhancement dan tanpa image enhancement:



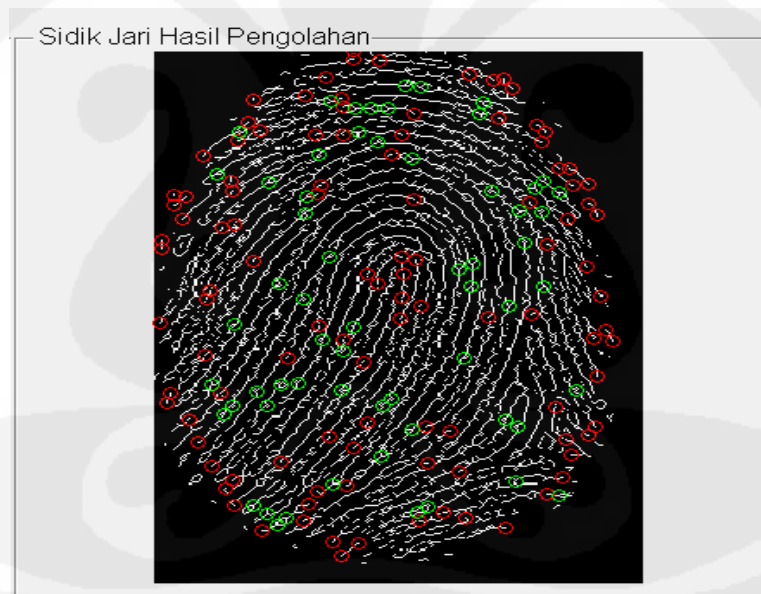
Gambar 4.16 Hasil pengolahan sidik jari (dengan kualitas yang bagus) menggunakan Image enhancement



Gambar 4.17 Hasil pengolahan sidik jari (dengan kualitas yang bagus) Tanpa menggunakan Image enhancement



Gambar 4.18 Hasil pengolahan sidik jari (dengan kualitas yang kurang) menggunakan Image enhancement



Gambar 4.19 Hasil pengolahan sidik jari (dengan kualitas yang kurang) Tanpa menggunakan Image enhancement

Dari gambar diatas bisa diketahui dengan menggunakan Image enhancement dapat membantu untuk memperjelas informasi mengenai sidik jari,

berikut ini data hasil pengujian menggunakan sistem Identifikasi Sidik jari menggunakan Algoritma Minutec :

No	Sample	Template	Matching	Dengan Image enhancement (IE)		Tanpa Image Enhancement (IE)	
				Template dengan IE	Hasil Matching (%)	Template tanpa IE	Hasil Matching (%)
1	Telunjuk User 101	User 101_1	Gbr101_1	Gbr101_1	100	Gbr101_17	100
			Gbr101_2		30.7592		11.1111
			Gbr101_3		34.5154		-
			Gbr101_4		15.3846		-
			Gbr101_5		26.9231		-
			Gbr101_6		34.5154		9.5238
			Gbr101_7		23.0769		-
2	Telunjuk User 102	User 102_1	Gbr102_1	Gbr102_1	100	Gbr102_17	100
			Gbr102_2		41.5557		-
			Gbr102_3		30		4.7619
			Gbr102_4		31.5557		5.5566
			Gbr102_5		20		8.3334
			Gbr102_6		33.3333		-
3	Telunjuk User 103	User 103_1	Gbr103_1	Gbr103_1	100	Gbr103_17	100
			Gbr103_2		34.7826		4.7619
			Gbr103_4		30.4348		1.4493
4	Telunjuk User 104	User 104_2	Gbr104_2	Gbr104_2	100	Gbr104_27	100
			Gbr104_3		42.5532		11.6667
			Gbr104_4		21.2766		-
			Gbr104_5		19.1489		2.0833
			Gbr104_7		42.5532		-

Tabel 4.1 Hasil Pengujian sidik jari

Dari hasil pengujian diatas didapatkan hasil bahwa untuk meningkatkan kualitas sidik jari menggunakan *image enhancement* harus memperhatikan beberapa faktor penting selain dari kualitas dari sidik jari, kedua faktor tersebut adalah:

1. Mengenai nilai faktor yang di masukan untuk *Fast Fourier Transform*.
2. Derajat keabuan (*Binerisasi*).

Kedua faktor tersebut dapat di atur sehingga didapatkan hasil sidik jari yang mempunyai kualitas dan informasi yang *valid*. Dimana untuk nilai *FFT* sendiri semakin besar nilai faktor yang dimasukan maka blok *pixel* yang di buat akan semakin besar. Jika ingin mendapatkan detail informasi sidik jari yang lebih banyak dan akurat maka nilai faktor *FFT* yang dimasukan harus semakin kecil, untuk nilai faktornya antara 0 - 1. Dari analisa pengujian didapatkan rata - rata nilai faktor antara 0.1 sampai 0.2, nilai ini digunakan untuk mendapatkan informasi yang lebih akurat dan juga blok pixel yang semakin banyak. Sedangkan

untuk nilai binerisasi, semakin besar nilai yang dimasukkan maka guratan citra sidik jari semakin menipis. Dimana untuk nilai binerisasi dirancang dengan range antara 0 - 200.

Setelah dilakukan pengujian dengan cara membandingkan hasil ekstraksi sidik jari yang menggunakan image enhancement dan tanpa menggunakan image enhancement didapat kan hasil seperti terlihat diatas (*Tabel 4.1*). Setiap hasil ekstraksi sidik jari dicocokkan dengan sidik jari yang sama (baik menggunakan maupun tidak menggunakan image enhancement) maka hasil presentasinya adalah 100 % cocok, sedangkan jika dibandingkan silang antara menggunakan image enhancement dan tanpa enhancement dengan sidik jari yang sama didapatkan hasil yang bervariasi. Untuk beberapa sidik jari harus mengatur nilai FFT maupun nilai yang dimasukkan pada proses binerisasi, ini berguna untuk mendapatkan kualitas sidik jari yang mempunyai nilai informasi yang *valid*. Dengan mendapatkan informasi yang *valid* dapat membantu atau mempercepat proses ekstraksi minuteda. Nilai minuteda yang dimanfaatkan untuk proses pencocokan sidik jari diatas adalah titik terminasi (titik perpisahan dari guratan sidik jari). Semakin banyak jumlah minuteda yang didapat maka sidik jari tersebut semakin akurat ini disebabkan karena tak ada satu pun manusia yang mempunyai jumlah minuteda yang sama lebih dari 8, bahkan untuk kembar sekalipun. Sehingga dengan banyaknya jumlah minuteda yang didapatkan maka tingkat keamanan sidik jari tersebut bisa dihandalkan.

Dari hasil pengujian sistem identifikasi sidik jari dengan melakukan proses pencocokan pada sidik jari baik dengan menggunakan *Image Enhancement* maupun tidak, didapatkan hasil yang kurang baik. Dengan nilai persentase kurang dari 50 %, ini bisa disebabkan karena sistem identifikasi yang kurang sempurna ataupun karena blm ada sistem penyesuaian diri (*adaptive threshold*) dari aplikasi untuk memilih besarnya nilai - nilai atau parameter yang bisa menentukan keberhasilan pencocokan.

BAB V

KESIMPULAN

1. Untuk meningkatkan kualitas gambar dari sidik jari digunakan dua proses *enhancement*, yaitu *Histogram Equalization* dan *FFT*. Dimana dengan adanya proses *enhancement*, kualitas dari citra sidik jari dapat di tingkatkan walaupun citra yang dimasukan kurang baik, dimana citra yang sebelumnya terlalu gelap ataupun kurang jelas menjadi lebih jelas.
2. Pada proses *images enhancement* ada 2 faktor penting yang harus diperhatikan agar didapatkan hasil citra sidik jari yang berkualitas, yaitu dengan mengatur nilai faktor *FFT*. Untuk nilai faktor *FFT* semakin besar maka citra blok yang dibuat semakin besar. Dari hasil pengujian didapatkan nilai faktor *FFT* yang cukup baik adalah 0.1 - 0.2.
3. Proses binerisasi juga menentukan keberhasilan sistem, semakin besar nilai *threshold* yang dimasukan maka semakin tipis guratan sidik jari pada citra. Dimana letak dan jumlah *minutea* yang dihasilkan akan tergantung pada proses ini.
4. Metode *Minutrea* merupakan salah satu metode yang sangat efektif dalam proses identifikasi sidik jari, karena dengan menggunakan metode ini, selain bisa mempercepat proses identifikasi karena data yang diproses berupa titik - titik *x* dan *y*, serta dengan menggunakan metode ini data *template database* cukup kecil, karena hanya menyimpan titik - titik *minutea* yang berhasil ditemukan. Sehingga *library* dari sistem ini bisa memuat banyak *template/citra enrollment* dari sidik jari.
5. Setelah dilakukan pengujian ternyata hasil dari pencocokan (*Matching*) menggunakan *Image Enhancement* dirasa kurang berhasil itu terbukti dengan kecilnya nilai dari persentase pencocokan sidik jari (kurang dari 50%). Kegagalan ini disebabkan karena sistem dari identifikasi yang kurang sempurna atau stabil dan juga karena belum adanya sistem *adaptive threshold*.

DAFTAR ACUAN

- [1] *Penjelasan Fingerprint Identifikasi*. Diakses tanggal 17 Mei 2009 dari <http://biometrics.cse.msu.edu>
<http://biometrics.cse.msu.edu/fingerprint.html>
- [2] *Penjelasan Matching Fingerprint*. Diakses tanggal 17 Mei 2009 dari <http://biometrics.cse.msu.edu>
<http://biometrics.cse.msu.edu/fingerprint.html>
- [3] *Dasar Image prosesing pada citra biner*. Diakses tanggal 25 Mei 2009 dari <http://www.docstoc.com>
<http://www.docstoc.com/docs/5066383/BASIC-IMAGE-PROCESSING>
http://repository.petra.ac.id/81/1/II-06-naskah-SIIT_05_053.pdf.
- [4] *Perbaikan citra menggunakan FFT*. Diakses tanggal 25 Mei 2009 dari <http://en.wikipedia.org>
http://en.wikipedia.org/wiki/Fourier_transform
- [5] *Perbaikan citra menggunakan HistogramEqualization*. Diakses tanggal 25 Mei 2009 dari <http://fourier.eng.hmc.edu>
http://fourier.eng.hmc.edu/e161/lectures/contrast_transform/node3.html
- [6] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, “*Handbook of Fingerprint Recognition*”, Springer, New York, 2003.
- [7] Anil. K. Jain, “*Handbook of Fingerprint Recognition*”, Springer, New York, 2003.
- [8] Hari Zainuddin Rasyid, “*Pengenalan Sidik Jari Manusia menggunakan Algoritma Ekstraksi Minutea*”. Penelitian, Departemen Teknik Elektro, 2001
- [9] “*Fingerprint Identification*”, Pattern Recognition and Image Processing Laboratory, Department of Computer Science and Engineering, Michigan State University
<http://biometrics.cse.msu.edu/fingerprint.html>
- [10] *Pembahasan Sidik jari serta algoritma Minutea* Diakses tanggal 10 Juni 2009 dari <http://biometrics.cse.msu.edu>
<http://biometrics.cse.msu.edu>

DAFTAR PUSTAKA

- [1] Fahmi, “Image Processing for Iridiology”, International Seminar ICTS 2007, Institut Teknologi Surabaya, Surabaya, 2007
- [2] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, “ *Handbook of Fingerprint Recognition* ”, Springer, New York, 2003.
- [3] Adi, Kusworo, “ Ekstraksi Ciri Berbasis Filter Gabor Sebagai Sistem Verifikasi Sidikjari’ , *Tesis Magister*, ITB, 2002.
- [4] Jain, Anil K. “*Fundamental of Digital Image Processing*”, Prentice Hall, New Jersey, 1989.
- [5] Mengko, Tati Latifah R. “*Diktat Kuliah Pengolahan Citra Digital*”, ITB, tanpa tahun.