

**PENDETEKSIAN POIN-POIN MINUTIAE
PADA CITRA SIDIK JARI DENGAN BERBASIS PADA
PEMROSESAN CITRA *DIGITAL***

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

oleh:
OCMITA TYAS HADHANI
0310960059-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2007**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

**PENDETEKSIAN POIN-POIN *MINUTIAE*
PADA CITRA SIDIK JARI DENGAN BERBASIS PADA
PEMROSESAN CITRA *DIGITAL***

Oleh:
OCMITA TYAS HADHANI
0310960059-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 13 Desember 2007
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

Pembimbing

Mengetahui,
a.n Ketua Jurusan Matematika
Fakultas MIPA Universitas
Brawijaya
Sekretaris,

Wayan Firdaus M., S.Si, MT
NIP. 132 158 724

Dra. Ani Budi Astuti, M.Si
NIP. 131 993 385

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Ocmita Tyas Hadhani
NIM : 0310960059
Jurusan : Matematika
Penulis skripsi berjudul : Pendeteksian Poin-Poin
Minutiae pada Citra Sidik Jari dengan Berbasis pada
Pemrosesan Citra Digital.

Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 13 Desember 2007
Yang menyatakan,

(Ocmita Tyas Hadhani)
NIM. 0310960059

UNIVERSITAS BRAWIJAYA



repository.ub.ac

UNIVERSITAS BRAWIJAYA

**PENDETEKSIAN POIN-POIN *MINUTIAE*
PADA CITRA SIDIK JARI DENGAN BERBASIS PADA
PEMROSESAN CITRA *DIGITA***

ABSTRAK

Identifikasi berbasis sidik jari merupakan salah satu teknik biometrik tersukses yang digunakan untuk identifikasi personal. Sidik jari merupakan pola *ridge* dan *valley* yang terbentuk pada permukaan ujung jari. Keunikan sidik jari digambarkan oleh karakteristik-karakteristik *ridge* lokal serta hubungan-hubungannya. Poin-poin *minutiae* merupakan karakteristik-karakteristik *ridge* lokal yang terdapat pada akhir *ridge* (*ridge ending*) dan pencabangan *ridge* (*ridge bifurcation*). *Ridge ending* didefinisikan sebagai poin dimana *ridge* tersebut berakhir atau berhenti, sedangkan *ridge bifurcation* merupakan poin dimana *ridge* terpilah menjadi dua cabang atau lebih. Pendeteksian poin-poin *minutiae* secara otomatis menjadi tugas berat bagi komputer jika citra sidik jari berkualitas rendah, dimana *noise* dan kurangnya kontras mengakibatkan pixel membentuk gambar yang serupa dengan *minutiae* sebenarnya. Sistem pendeteksi *minutiae* yang baik adalah sistem yang dapat meningkatkan mutu gambar berkualitas rendah serta dapat mengekstrak poin-poin *minutiae* dengan tepat. Pada skripsi ini akan dibangun sebuah sistem pendeteksi poin *minutiae* dengan berbasis pada pengolahan citra digital. Secara umum proses ini terbagi menjadi dua tahapan penting. Tahap pertama disebut dengan *Pre-Processing*. Pada tahapan ini citra sidik jari hasil *scanning* akan ditingkatkan kualitasnya melalui beberapa proses, antara lain proses *segmentation*, *normalization*, *orientation image estimation*, *ridge frequency estimation*, dan *filtering*. Sedangkan tahap yang terakhir disebut *Minutiae Detection* atau deteksi *minutiae*. Pada tahap deteksi *minutiae* citra sidik jari yang telah ditingkatkan kualitasnya akan diproses untuk mendapatkan poin-poin *minutiae*nya. Untuk mempermudah pendeteksian poin *minutiae*, sebelumnya pada citra tersebut dikenakan proses binerisasi (*binarization process*) dan proses penipisan atau yang biasa disebut dengan *thinning process*.

Poin-poin *minutiae* yang merupakan hasil pendeteksian sistem akan dikenakan pengujian secara sederhana. Uji coba diterapkan pada lima belas sidik jari orang yang berbeda dimana tiap orang akan

diambil lima *sample* sidik jari. Poin-poin *minutiae* yang menjadi pusat uji coba ini adalah *ridge ending*, *ridge bifurcation* dan *ridge crossing*. Hasil ujicoba dan evaluasi menunjukkan bahwa dari kelima *sample* sidik jari pada masing-masing orang didapatkan jumlah poin-poin *minutiae* yang tidak konsisten, baik pada *sample* citra sidik jari dengan pola masukan yang berbeda maupun dengan pola masukan yang hampir sama. Hal ini disebabkan karena setiap citra sidik jari masukan memiliki karakteristik frekuensi yang berbeda-beda, sehingga saat melakukan proses *filtering* (menggunakan filter gabor) dibutuhkan nilai standar deviasi σ_x dan σ_y yang sesuai dengan karakteristik tersebut.

Penentuan nilai standar deviasi yang statis terhadap variasi citra masukan sangat mempengaruhi proses peningkatan kualitas citra, dimana proses akan bekerja secara maksimal jika nilai standar deviasi cocok dengan karakteristik frekuensi citra masukan. Begitu pula sebaliknya, proses akan bekerja kurang maksimal jika nilai standar deviasi tidak cocok dengan karakteristik frekuensi citra masukan.

Namun jika seluruh hasil dari uji coba sistem terhadap kelima belas orang yang diambil *sample* sidik jarinya dibandingkan dengan menggunakan metode *Analysis of Variance*, maka didapat kesimpulan yang menunjukkan bahwa paling sedikit ada satu pasang sidik jari orang yang berbeda.



THE DETECTION OF MINUTIAE POINTS UPON FINGERPRINT IMAGE BASED ON DIGITAL IMAGING

ABSTRACT

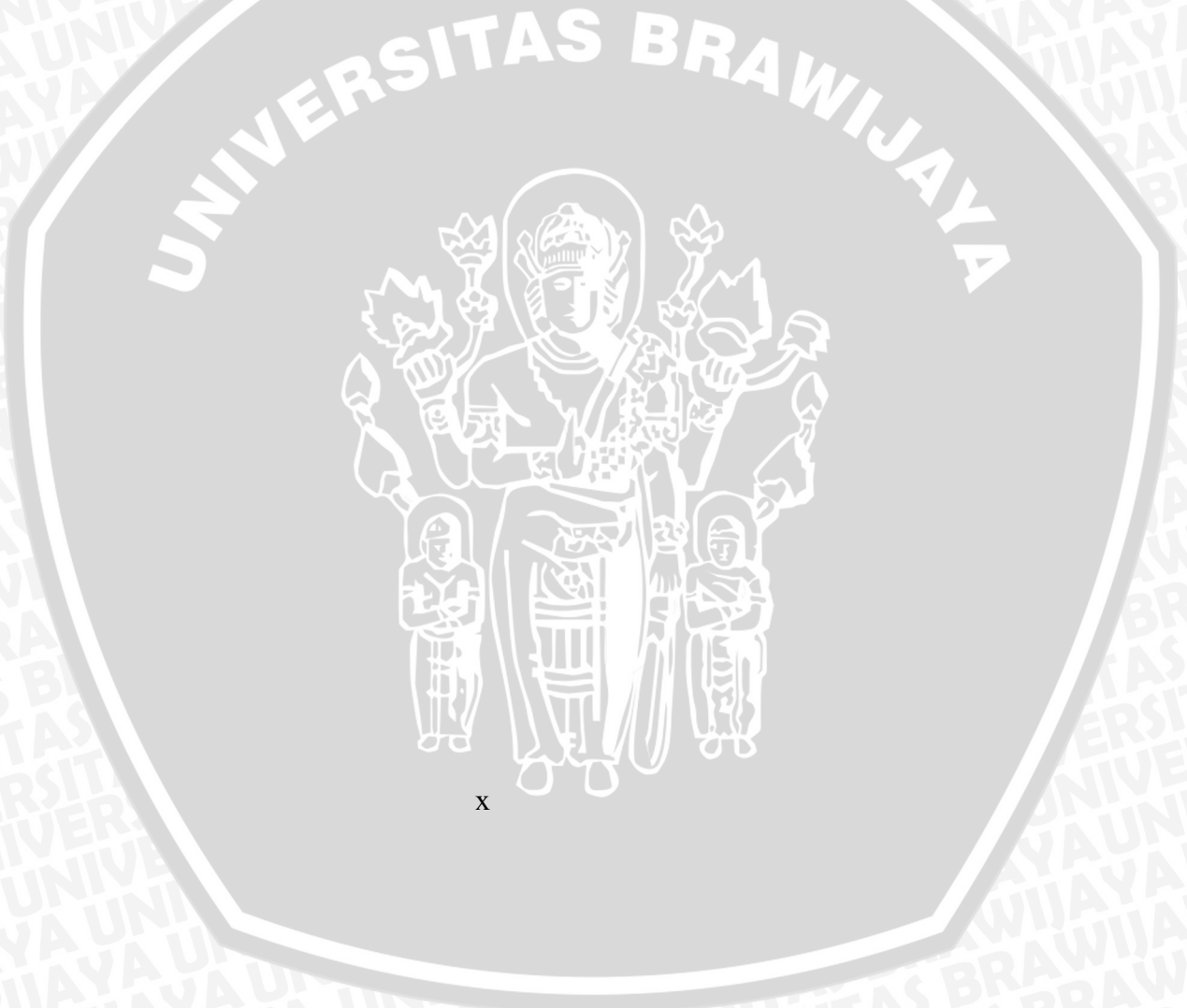
Fingerprint based identification represents as a successful biometric technique in the benefit of personal identification. Fingerprint remains as ridge and valley patterns produced by the tip of finger. The uniqueness of fingerprint seems characterized by local ridge and its attachments. Minutiae points constitute local ridge characteristics shown at ridge ending and ridge bifurcation. One defines ridge ending as a point where ridge ends or stops, while ridge bifurcation submits to a point where ridge divides into two or more branches. Automatic detection upon minutiae points has been a heavy duty for computer when fingerprint image appears in lower quality, particularly that noise and lower contrast cause pixel producing similar picture to the actual minutiae. Favorable detection system for minutiae will be a system to improve the picture quality and to extract minutiae points in precisely manner. In this final assignment, a system to detect minutiae point based on digital imaging may be established. In general, this process has two important stages. First stage consists of Pre-Processing. In this stage, fingerprint image from scanning must be improved for its quality through several processes, such as segmentation, normalization, orientation image estimation, ridge frequency estimation, and filtering. Final stage has been Minutiae Detection. In this stage, the improved fingerprint image quality has to be processed to attain minutiae points. To facilitate minutiae points detection requires binarization process and thinning process subjected to the image.

Minutiae points referred to a result of systemic detection will be subjected to a simple examination. The trial involves fifteen fingerprints of different person in which each of them provides five samplers of fingerprint. Minutiae points as a main consideration of this trial include ridge ending, ridge bifurcation, and ridge crossing. Results of trial and evaluation indicate that of five samples of fingerprint from each person, it obtains inconsistent number of minutiae points, either at image sample of fingerprint in different input pattern or in almost similar input pattern. This seems because inputted fingerprint image has different frequency characteristic such

that when filtering process (using gabor filter), it needs deviation standard σ_x and σ_y , adjusting to the characteristic.

Determining static deviation standard value against input image variation appears affecting the process of image quality improvement, where the process runs maximally if deviation standard value corresponds to the characteristic of input image variation. In contrast, the process will be less maximal when deviation standard value doesn't correspond to the characteristic of input image frequency.

If all outcomes of the trial from fifteen participants who their fingerprints have been taken should be compared with method of *Analysis of Variance*, the conclusion shows that at least a pair of participant's fingerprint differ of each other.



KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahnya, Skripsi yang berjudul “Pendeteksian Poin-Poin *Minutiae* pada Citra Sidik Jari dengan Berbasis pada Pemrosesan Citra *Digital*” ini dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Tidak lupa Shalawat serta salam penulis curahkan kepada Nabi Muhammad SAW yang telah membimbing umat muslim dari jaman kebodohan hingga jaman kejayaan.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Wayan Firdaus Mahmudy, S.Si, MT selaku pembimbing skripsi serta Ketua Program Studi Ilmu Komputer Unibraw Malang. Terima kasih atas semua saran, bantuan, waktu, dan bimbingannya.
2. Drs. Achmad Ridok, M.Kom selaku Penasihat Akademik.
3. Bapak, Ibu dan adik-adik (Tirta dan Umi) serta keluarga besar Alm. Eyang Mangun Witono dan Eyang Harjo, yang senantiasa berdoa dan memberi dukungan serta semangat.
4. Sahabat-sahabat, Mbak Ela, Miko, Yani, Rudi, Rosihan, Iwul, Peno, Aldo, dll. Terima kasih atas semangat dan bantuan yang diberikan.
5. Teman-teman satu kajian, teman-teman UAKI dan PUSKOMDA Malang, terima kasih atas doa dan dukungan yang diberikan.
6. Semua teman-teman Ilmu Komputer angkatan 2003. Terima kasih atas semangat dan doanya.

7. Pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, 13 Desember 2007

Penulis



DAFTAR GAMBAR

Gambar 1.1	Bentuk <i>Ridge ending</i> dan <i>Ridge bifurcation</i> pada sidik jari.....	1
Gambar 2.1	Lima kelas dasar sidik jari bentukan Henry.....	8
Gambar 2.2	Beberapa fitur dasar sidik jari.....	9
Gambar 2.3	Poin <i>core</i> dan <i>delta</i> pada sidik jari.....	10
Gambar 2.4	Fitur pada guratan sidik jari.....	10
Gambar 2.5	Pengelompokan jenis-jenis citra.....	11
Gambar 2.6	Diagram hubungan antara Grafika Komputer, Pengolahan Citra dan Pengenalan Pola.....	15
Gambar 2.7	Hasil dari segmentasi dengan menggunakan <i>variance threshold</i> 100 dan ukuran blok 16 x 16 piksel.....	18
Gambar 2.8	Hasil Normalisasi dengan M_o (Mean yang diharapkan) 100 dan V_o (Varian yang diharapkan) 100.....	19
Gambar 2.9	Orientasi piksel <i>ridge</i> pada sidik jari.....	19
Gambar 2.10	<i>Orientation estimated</i> pada citra sidik jari.....	21
Gambar 2.11	Proyeksi nilai intensitas piksel sepanjang arah orthogonal hingga <i>local ridge orientation</i>	21
Gambar 2.12	<i>Estimated ridge wavelength</i> pada citra sidik jari..	22
Gambar 2.13	Bobot <i>mask Gaussian</i> dengan $\sigma = 1$	23
Gambar 2.14	Bentuk sinusoidal dari alur sidik jari.....	23
Gambar 2.15	Bentuk <i>gabor filter</i> dalam daerah spatial.....	25
Gambar 2.16	Hasil peningkatan citra sidik jari menggunakan <i>Gabor Filter</i>	26
Gambar 2.17	Hasil binerisasi citra sidik jari.....	27
Gambar 2.18	Pemberian nomor pada 8-tetangga piksel p1 untuk proses <i>thinning</i>	28
Gambar 2.19	Hasil penipisan citra sidik jari.....	30
Gambar 2.20	Contoh gambar yang akan <i>dithinning</i> (ditipiskan).....	30
Gambar 2.21	Hasil proses <i>thinning</i> langkah ke-1.....	32
Gambar 2.22	Hasil proses <i>thinning</i> langkah ke-2.....	33
Gambar 2.23	Pemberian nomor pada 8-tetangga piksel P untuk proses <i>minutiae detection</i>	34
Gambar 2.24	Contoh piksel <i>ridge ending</i> dan piksel <i>bifurcation</i>	35

Gambar 2.25	Contoh piksel P yang akan diperiksa.....	35
Gambar 2.26	Pemberian nomor pada 8-tetangga piksel P.....	35
Gambar 3.1	Diagram alir pembuatan perangkat lunak.....	37
Gambar 3.2	Diagram alir proses umum sistem.....	40
Gambar 3.3	Diagram alir proses masukan.....	42
Gambar 3.4	Diagram alir proses segmentasi.....	43
Gambar 3.5	Diagram alir proses normalisasi.....	44
Gambar 3.6	Diagram alir proses <i>gaussian smoothing</i>	44
Gambar 3.7	Diagram alir proses <i>image orientation estimation</i>	46
Gambar 3.8	diagram alir proses <i>ridge frequency estimation</i>	47
Gambar 3.9	Diagram alir proses <i>filtering</i>	48
Gambar 3.10	Diagram alir proses binerisasi.....	49
Gambar 3.11	Pemberian nomor pada 8-tetangga piksel p1 untuk proses <i>thinning</i>	50
Gambar 3.12	Diagram alir proses <i>thinning</i> langkah 1.....	52
Gambar 3.13	Diagram alir proses <i>thinning</i> langkah 2.....	54
Gambar 3.14	Diagram Alir proses deteksi <i>minutiae</i>	55
Gambar 3.15	Rancangan <i>form</i> utama.....	56
Gambar 4.1	Citra Sidik Jari Hasil Proses Segmentasi dengan Nilai $T=100$	62
Gambar 4.2	Citra Sidik Jari hasil proses <i>filtering</i> dengan nilai $\sigma_x = 0,5$ dan $\sigma_y = 3,0$	73
Gambar 4.3	Citra Sidik Jari hasil proses binerisasi dengan <i>threshold</i> $T_1 = 150$ dan $T_2 = 170$	74
Gambar 4.4	Tampilan <i>Form Utama</i>	79
Gambar 4.5	Tampilan hasil peningkatan kualitas citra dan deteksi <i>minutiae</i>	80
Gambar 4.6	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-1.....	81
Gambar 4.7	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-2.....	82
Gambar 4.8	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-3.....	83
Gambar 4.9	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-4.....	84
Gambar 4.10	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-5.....	85

Gambar 4.11	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-6.....	86
Gambar 4.12	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-7.....	87
Gambar 4.13	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-8.....	88
Gambar 4.14	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-9.....	89
Gambar 4.15	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-10.....	90
Gambar 4.16	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-11.....	91
Gambar 4.17	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-12.....	92
Gambar 4.18	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-13.....	93
Gambar 4.19	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-14.....	94
Gambar 4.20	Grafik hasil pendeteksian poin-poin <i>minutiae</i> orang ke-15.....	95
Gambar 4.21	Hasil pengujian terhadap perbandingan <i>ridge ending</i> dari kelima belas orang yang diambil <i>sample</i> sidik jarinya.....	96
Gambar 4.22	Hasil pengujian terhadap perbandingan <i>ridge bifurcation</i> dari kelima belas orang yang diambil <i>sample</i> sidik jarinya.....	98
Gambar 4.23	Hasil pengujian terhadap perbandingan <i>ridge crossing</i> dari kelima belas orang yang diambil <i>sample</i> sidik jarinya.....	100
Gambar 4.24	Hasil proses peningkatan citra sidik jari.....	101
Gambar 4.25	Hasil proses <i>thinning</i> citra sidik jari.....	101
Gambar 4.26	Hasil Proses Pendeteksian Poin-Poin <i>Minutiae</i>	102

UNIVERSITAS BRAWIJAYA



xvi

DAFTAR TABEL

Tabel 2.1	Hasil pemeriksaan pada langkah ke-1 terhadap piksel yang memenuhi syarat.....	31
Tabel 2.2	Hasil pemeriksaan pada langkah ke-2 terhadap piksel yang memenuhi syarat.....	32
Tabel 2.3	Properti <i>Crossing Number</i>	34
Tabel 3.1	Rancangan tabel hasil pendeteksian poin-poin <i>minutiae</i> masing-masing orang.....	57
Tabel 3.2	Rancangan tabel perbandingan hasil pendeteksian <i>ridge ending</i> pada masing-masing orang.....	58
Tabel 3.3	Rancangan tabel perbandingan hasil pendeteksian <i>ridge bifurcation</i> pada masing-masing orang.....	58
Tabel 3.4	Rancangan tabel perbandingan hasil pendeteksian <i>ridge crossing</i> pada masing-masing orang.....	58
Tabel 4.1	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-1.....	81
Tabel 4.2	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-1.....	81
Tabel 4.3	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-2.....	81
Tabel 4.4	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-2.....	82
Tabel 4.5	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-3.....	82
Tabel 4.6	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-3.....	83
Tabel 4.7	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-4.....	83
Tabel 4.8	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-4.....	84
Tabel 4.9	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-5.....	84
Tabel 4.10	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-5.....	85
Tabel 4.11	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-6.....	85
Tabel 4.12	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-6.....	86

Tabel 4.13	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-7.....	86
Tabel 4.14	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-7.....	87
Tabel 4.15	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-8.....	87
Tabel 4.16	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-8.....	88
Tabel 4.17	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-9.....	88
Tabel 4.18	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-9.....	89
Tabel 4.19	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-10.....	89
Tabel 4.20	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-10.....	90
Tabel 4.21	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-11.....	90
Tabel 4.22	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-11.....	91
Tabel 4.23	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-12.....	91
Tabel 4.24	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-12.....	92
Tabel 4.25	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-13.....	92
Tabel 4.26	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-13.....	93
Tabel 4.27	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-14.....	93
Tabel 4.28	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-14.....	94
Tabel 4.29	Hasil pendeteksian poin-poin <i>minutiae</i> orang ke-15.....	94
Tabel 4.30	Hasil perhitungan varian terhadap poin-poin <i>minutiae</i> orang ke-15.....	95
Tabel 4.31	Tabel hasil pendeteksian <i>ridge ending</i> pada masing-masing orang.....	96

Tabel 4.32	Tabel hasil pendeteksian <i>ridge bifurcation</i> pada masing-masing orang.....	98
Tabel 4.33	Tabel hasil pendeteksian <i>ridge crossing</i> pada masing-masing orang.....	99



UNIVERSITAS BRAWIJAYA



xx

DAFTAR SOURCECODE

Sourcecode 4.1	Sourcecode cara membuka file citra.....	60
Sourcecode 4.2	Sourcecode proses pemblokian atau windowing.....	60
Sourcecode 4.3	Sourcecode proses <i>grayscale</i> ing.....	61
Sourcecode 4.4	Sourcecode proses perhitungan <i>mean</i> pada tiap-tiap blok.....	61
Sourcecode 4.5	Sourcecode proses perhitungan varian pada tiap-tiap blok dan proses penentuan <i>background</i> dan objek.....	62
Sourcecode 4.6	Sourcecode proses perhitungan citra ternormalisasi.....	63
Sourcecode 4.7	Sourcecode proses perhitungan <i>gaussian smoothing</i>	64
Sourcecode 4.8	Sourcecode proses perhitungan gradien x dan gradien y.....	65
Sourcecode 4.9	Sourcecode proses perhitungan <i>local</i> <i>orientation</i> $\theta(i,j)$ pada tiap blok.....	66
Sourcecode 4.10	Sourcecode proses pencarian titik x_1, y_1 dan x_2, y_2 untuk membentuk garis orthogonal.....	68
Sourcecode 4.11	Sourcecode proses menggambar garis orthogonal.....	69
Sourcecode 4.12	Sourcecode proses perhitungan <i>ridge</i> <i>frequency</i> $F(i,j)$	71
Sourcecode 4.13	Sourcecode proses <i>filtering</i>	72
Sourcecode 4.14	Sourcecode fungsi <i>Gabor filter</i>	73
Sourcecode 4.15	Sourcecode proses binerisasi.....	74
Sourcecode 4.16	Sourcecode proses <i>thinning</i>	77
Sourcecode 4.17	Sourcecode proses deteksi <i>minutiae</i>	78

UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Lampiran 1	107
Lampiran 2	115
Lampiran 3	119
Lampiran 4	123
Lampiran 5	127
Lampiran 6	135
Lampiran 7	143



UNIVERSITAS BRAWIJAYA



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN.....	v
HALAMAN ABSTRAK.....	vii
KATA PENGANTAR.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xvii
DAFTAR <i>SOURCECODE</i>.....	xxi
DAFTAR LAMPIRAN.....	xxiii
DAFTAR ISI.....	xxv

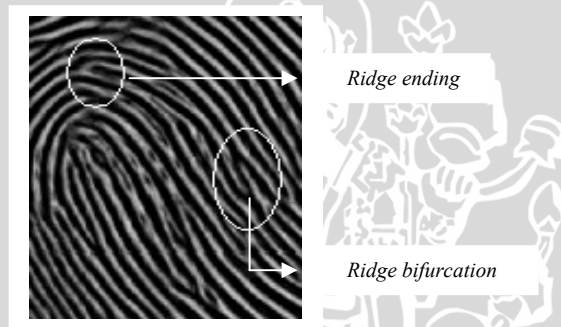
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Pemecahan Masalah.....	4
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Sidik Jari.....	7
2.1.1 Sejarah Sidik Jari.....	7
2.1.2 Klasifikasi Sidik Jari.....	7
2.2 Pemrosesan Citra Digital.....	11
2.2.1 Pengertian Citra.....	11
2.2.2 Pengambilan Citra dan Peralatan Tampilan.....	14
2.2.3 Pengolahan Data Citra.....	15
2.2.4 Konversi Citra <i>True Color</i> ke <i>Grayscale</i>	16
2.2.5 Segmentasi (<i>Segmentation</i>).....	16
2.2.6 Normalisasi (<i>Normalization</i>).....	18
2.2.7 <i>Image Orientation Estimation</i>	19
2.2.8 <i>Ridge Frequency Estimation</i>	21
2.2.9 <i>Gaussian Low Pass Filter</i>	22
2.2.10 <i>Gabor Filter dan Peningkatan Kualitas Citra</i>	23
2.2.11 Binerisasi (<i>Binarization</i>).....	26

2.2.12 Penipisan (<i>Thinning</i>).....	27
2.2.13 <i>Minutiae Detection</i> (Deteksi <i>Minutiae</i>).....	33
BAB III METODE DAN PERANCANGAN.....	37
3.1 Analisis Perangkat Lunak.....	38
3.1.1 Deskripsi Umum Perangkat Lunak.....	38
3.1.2 Batasan Perangkat Lunak.....	38
3.1.3 Analisis Kebutuhan Perangkat Lunak.....	39
3.2 Perancangan Perangkat Lunak.....	41
3.2.1 Perancangan Proses.....	41
3.2.2 Perancangan <i>Interface</i>	56
3.3 Perancangan Uji Coba.....	56
BAB IV HASIL DAN PEMBAHASAN.....	59
4.1 Lingkungan Implementasi.....	59
4.1.1 Lingkungan Perangkat Keras.....	59
4.1.2 Lingkungan perangkat lunak.....	59
4.2 Implementasi Program.....	59
4.2.1 <i>Input</i>	59
4.2.2 Segmentasi (<i>Segmentation</i>).....	60
4.2.3 Normalisasi (<i>Normalization</i>).....	63
4.2.4 <i>Image Orientation Estimation</i>	64
4.2.5 <i>Ridge Frequency Estimation</i>	67
4.2.6 <i>Filtering</i>	71
4.2.7 Binerisasi (<i>Binarization</i>).....	74
4.2.8 <i>Thinning</i> (Penipisan).....	75
4.2.9 <i>Minutiae Detection</i> (Deteksi <i>Minutiae</i>).....	77
4.3 Implementasi <i>Interface</i>	79
4.4 Implementasi Uji Coba.....	80
4.4.1 Evaluasi Hasil 15 Citra Uji.....	80
4.4.2 Analisa Hasil.....	101
BAB V KESIMPULAN DAN SARAN.....	103
5.1 Kesimpulan.....	103
5.2 Saran.....	103
DAFTAR PUSTAKA.....	105
LAMPIRAN	

BAB I PENDAHULUAN

1.1 Latar Belakang

Sidik jari, yang telah digunakan sekitar 100 tahun lamanya, merupakan tanda pengenal biometrik tertua. Studi ilmiah pada sidik jari telah dilakukan pada abad ke-16, namun pondasi indentifikasi sidik jari modern baru dibangun oleh F.Galton dan E.Henry pada studi mereka di akhir abad ke-19(Jain dkk, 1999). Identifikasi berbasis sidik jari merupakan salah satu teknik biometrik tersukses yang digunakan untuk indentifikasi personal(Chama). Sidik jari merupakan pola *ridge* (area gelap) dan *valley* (area terang) pada Gambar 1.1 yang terbentuk pada permukaan ujung jari. Keunikan sidik jari digambarkan oleh karakteristik-karakteristik *ridge* lokal serta hubungan-hubungannya. Poin-poin *minutiae* merupakan karakteristik-karakteristik *ridge* lokal yang terdapat pada akhir *ridge* (*ridge ending*) dan pencabangan *ridge* (*ridge bifurcation*). *Ridge ending* didefinisikan sebagai poin dimana *ridge* tersebut berakhir atau berhenti, sedangkan *ridge bifurcation* merupakan poin dimana *ridge* terpilah menjadi dua cabang atau lebih(Chama).



Gambar 1.1 Bentuk *Ridge ending* dan *Ridge bifurcation* pada sidik jari

Pendeteksian poin-poin *minutiae* secara otomatis menjadi tugas berat bagi komputer jika citra sidik jari berkualitas rendah, dimana *noise* dan kurangnya kontras mengakibatkan pixel membentuk gambar yang serupa dengan *minutiae* sebenarnya. Sistem pendeteksi

minutiae yang baik adalah sistem yang dapat meningkatkan mutu gambar berkualitas rendah serta dapat mengekstrak poin-poin *minutiae* dengan tepat. Sidik jari yang berkualitas baik biasanya terdiri 40-100 *minutiae* (Hong dkk, 1998). Sistem pendeteksi *minutiae* juga sangat menentukan tingkat kesuksesan dalam pengidentifikasian sidik jari (Chama).

Beberapa penelitian dengan berbagai metode dan pendekatan telah dilakukan untuk menghasilkan sistem pendeteksi *minutiae* yang baik. Zhixin Shi dan Venu Govindaraju memanfaatkan Skema *Chaincode* sebagai metode untuk mengekstrak *minutiae* sidik jari. Di lain pihak, Victor Lopez Lorenzo beserta rekan-rekannya membangun sistem pendeteksi *minutiae* melalui tiga tahapan utama, yaitu *Fingerprint enhancement* untuk meningkatkan kualitas citra, *Minutiae extraction* untuk mendeteksi *minutiae*, dan *Post-Processing* untuk menghilangkan poin-poin *minutiae* yang salah. Sharat Chikkerur dan kawan-kawan menggunakan pendekatan sistematis untuk mendeteksi poin *minutiae*. Dalam penelitian tersebut, mereka mengajukan dua algoritma original untuk mengekstrak atau mendeteksi *feature* global dan lokal pada sidik jari.

Mengikuti jejak para peneliti di atas, pada skripsi ini akan dibangun sebuah sistem pendeteksi poin *minutiae* dengan berbasis pada pengolahan citra digital. Secara umum proses ini terbagi menjadi dua tahapan penting. Tahap pertama disebut dengan *Pre-Processing*. Pada tahapan ini citra sidik jari hasil *scanning* akan ditingkatkan kualitasnya melalui beberapa proses, antara lain proses *segmentation*, *normalization*, *orientation image estimation*, *ridge frequency estimation*, dan *filtering*. Sedangkan tahap yang terakhir disebut *Minutiae Detection* atau deteksi *minutiae*. Pada tahap deteksi *minutiae*, citra sidik jari yang telah ditingkatkan kualitasnya akan diproses untuk mendapatkan poin-poin *minutiae*nya. Untuk mempermudah pendeteksian poin *minutiae*, sebelumnya pada citra tersebut dikenakan proses binerisasi (*binarization process*) dan proses penipisan atau yang biasa disebut *thinning process*. Poin-poin *minutiae* yang terdeteksi pada citra sidik jari akan ditandai dengan titik merah.

Dengan penggabungan beberapa metode di atas, diharapkan sistem pendeteksi poin *minutiae* yang dibangun mampu meningkatkan kualitas citra sidik jari, sehingga didapat hasil pendeteksian poin-poin *minutiae* yang stabil untuk sidik jari orang

yang sama. Berdasarkan latar belakang yang telah dijelaskan di atas, maka sangat tepat jika skripsi ini diberi judul “***Pendeteksian Poin-Poin Minutiae pada Citra Sidik Jari dengan Berbasis pada Pemrosesan Citra Digital***”.

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka dalam skripsi ini dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang sebuah perangkat lunak dengan berbasis pada pengolahan citra digital yang mampu meningkatkan kualitas citra sidik jari, sehingga didapat hasil pendeteksian poin-poin *minutiae* yang stabil untuk sidik jari orang yang sama?
2. Bagaimana kestabilan poin-poin *minutiae* hasil pendeteksian sistem untuk sidik jari orang yang sama?

1.3 Batasan Masalah

Dari permasalahan diatas, berikut ini diberikan batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan:

1. Pengambilan objek dilakukan secara *off line*.
2. Objek merupakan citra sidik jari yang disimpan dalam ekstensi BMP dan dengan ukuran 192 x 192 piksel.
3. Objek merupakan objek tunggal, maksudnya dalam satu citra hanya terdapat satu objek sidik jari.
4. Sidik jari yang akan dideteksi poin *minutiae*nya memiliki sudut sembarang.
5. Sudut rotasi hanya pada bidang x dan y (sumbu rotasi pada sumbu z).
6. Poin-poin *minutiae* yang terdeteksi ditandai dengan titik berwarna merah.
7. Jenis poin *minutiae* yang menjadi pusat penelitian adalah *ridge ending*, *ridge bifurcation* dan *ridge crossing*.

1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai adalah:

1. Membangun aplikasi sistem pendeteksi poin *minutiae* dengan berbasis pada pengolahan citra digital yang mampu meningkatkan kualitas citra sidik jari , sehingga didapat hasil

pendeteksian poin-poin *minutiae* sidik jari yang stabil untuk sidik jari orang yang sama.

2. Mengevaluasi kestabilan poin-poin *minutiae* sidik jari hasil pendeteksian sistem untuk sidik jari orang yang sama.

1.5 Manfaat Penelitian

Dari penelitian ini, diharapkan dapat diciptakan program aplikasi (*software*) yang mampu meningkatkan kualitas citra sidik jari, sehingga didapat hasil pendeteksian poin-poin *minutiae* sidik jari yang stabil untuk sidik jari orang yang sama.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah:

1. Studi Literatur
Mempelajari teori-teori yang berhubungan dengan karakteristik sidik jari dan konsep pemrosesan citra digital dari berbagai referensi.
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut yaitu membuat aplikasi pendeteksi poin-poin *minutiae* sidik jari.
4. Uji coba dan analisa hasil implementasi
Menguji perangkat lunak, dan menganalisa hasil dari implementasi tersebut apakah telah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN
Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan karakteristik sidik jari dan konsep pemrosesan citra digital.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode yang digunakan dalam peningkatan kualitas citra sidik jari serta pendeteksian poin-poin *minutiae* citra sidik jari tersebut.

4. BAB IV IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem perangkat lunak yang dibangun, yaitu apakah aplikasi tersebut mampu mendeteksi poin-poin *minutiae* citra sidik jari dengan stabil.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.

UNIVERSITAS BRAWIJAYA





BAB II

TINJAUAN PUSTAKA

2.1 Sidik Jari

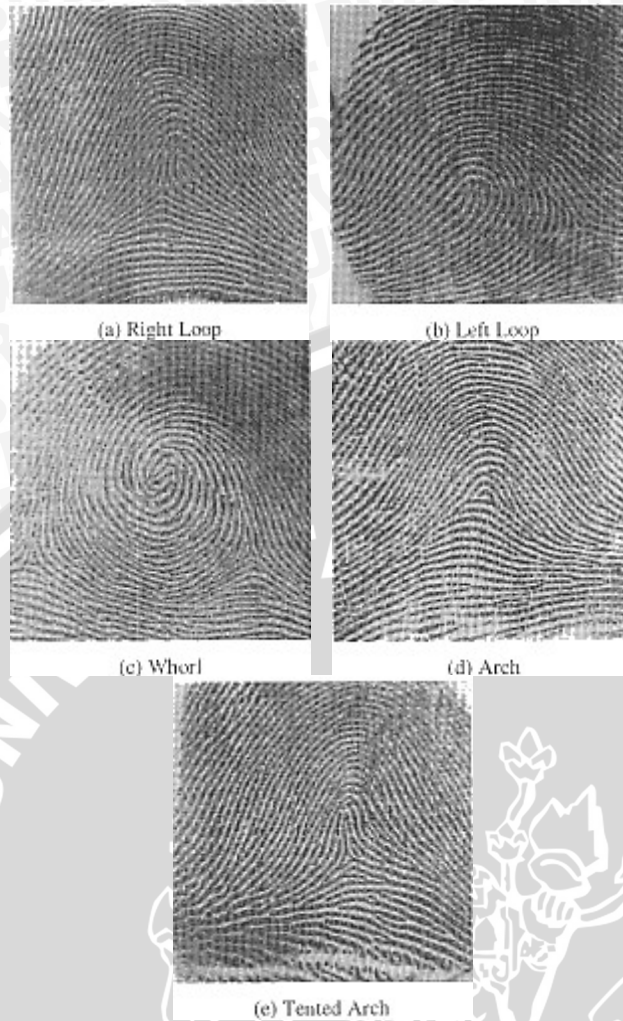
2.1.1 Sejarah Sidik Jari

Seerti yang disebutkan pada bab sebelumnya, penelitian tentang identifikasi sidik jari modern telah dilakukan pada akhir abad ke-19 oleh F.Galton dan E.Henry. Pada studinya, Galton memperkenalkan *minutiae* yang merupakan diskontinuitas lokal pada pola alur *ridge*(Jain dkk, 1999). Studi yang dilakukan oleh Galton tersebut menguji secara eksklusif detil yang ada pada sidik jari. Sidik jari diuji secara morfologi dan eksperimen dilakukan pada beberapa kelompok usia dan ras yang berbeda. Dua kesimpulan penting dihasilkan oleh Galton. Kesimpulan yang pertama adalah sidik jari pada setiap orang bersifat permanen, karakteristiknya tetap terjaga, dan bertahan dari lahir hingga meninggal. Hasil yang kedua adalah setiap individu memiliki sidik jari yang unik. Berdasarkan bukti eksperimental, tidak ada dua orang pun yang memiliki sidik jari yang sama, bahkan kembar identik pun memiliki sidik jari yang berbeda(Jain dkk,1999). Pada tahun 1901, sistem identifikasi kejahatan berdasarkan penelitian Galton mulai mendapat tempat di Inggris. Pasalnya, jejak sidik jari kerap tercecer. Metode pengelompokan Galton pun masih berlaku hingga hari ini dalam ilmu forensik untuk mengidentifikasi tersangka(Arvian, 2005).

Di lain pihak, Henry meneliti tentang struktur global sidik jari dan membuat klasifikasi sidik jari yang terkenal dengan nama “Sistem Henry”, dimana sistem tersebut merupakan metode efektif untuk *indexing* sidik jari dan masih digunakan dalam kebanyakan sistem identifikasi(Jain dkk, 1999).

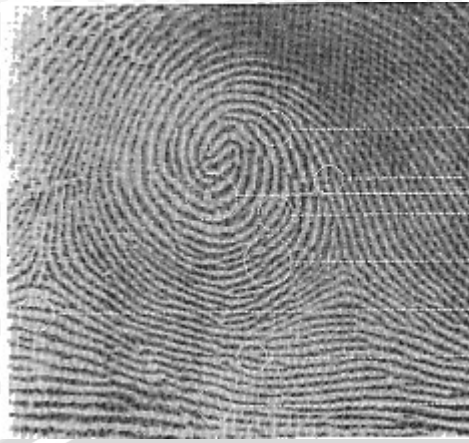
2.1.2 Klasifikasi Sidik Jari

E.R. Henry meneliti struktur global pada sidik jari dan menemukan suatu metode klasifikasi untuk membagi sidik jari ke dalam lima kelas(Jain dkk, 1999), yaitu *Right Loop (R)*, *Left Loop (L)*, *Whorl (W)*, *Arch (A)*, dan *Tented Arch (T)*(Jain dkk, 1999). Contoh dari kelima kelas di atas dapat dilihat pada Gambar 2.1.



Gambar 2.1 Lima kelas dasar sidik jari bentukan Henry

Sedangkan Fitur Galton lebih kepada detil yang terbentuk pada alur *ridge*. Suatu *ridge* dapat didefinisikan sebagai segmen kurva tunggal. Kombinasi dari beberapa *ridge* tersebut membentuk suatu pola sidik jari. Fitur kecil yang terbentuk dari persilangan dan akhir *ridge* pada sidik jari disebut *minutiae*. Pada penelitiannya, Galton mendefinisikan empat karakteristik, yaitu awal dan akhir *ridge*, *forks*, *island*, dan *enclosures* (Jain dkk, 1999).



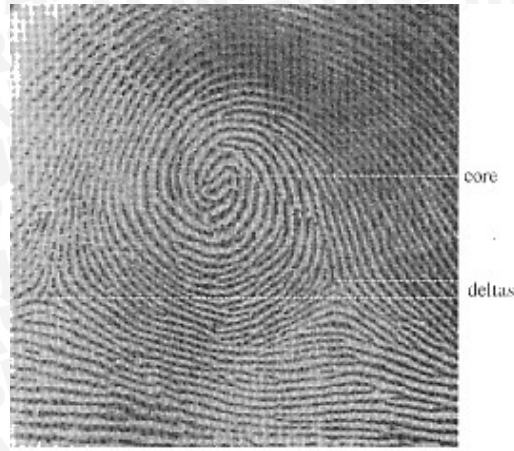
bifurcation
ridge ending
short ridge
bridge
island
dot
crossover
spur

Feature	Dot	Ridge End	Island	Bifurcation	Short Ridge	Crossover	Bridge	Spur
Sample								

Gambar 2.2 Beberapa fitur dasar sidik jari

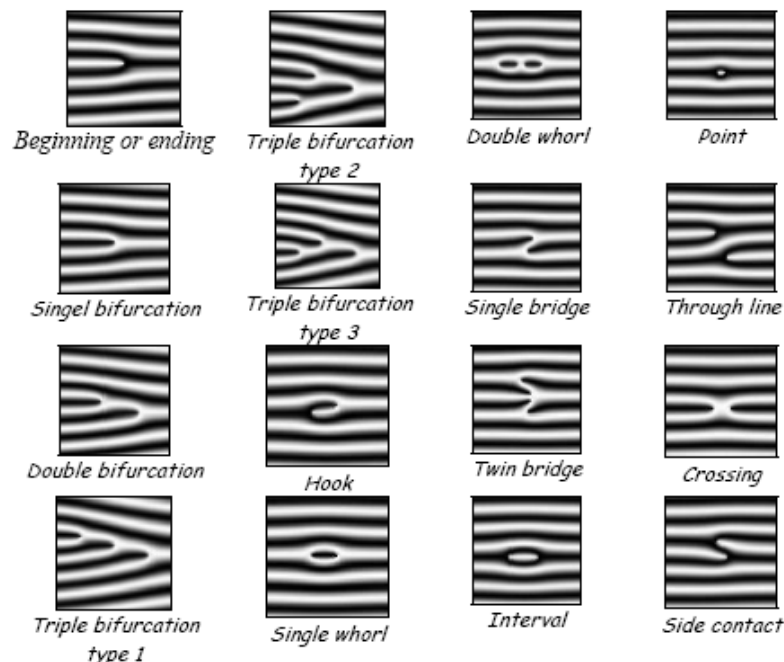
Setelah Henry dan Galton, penelitian terhadap identifikasi sidik jari dan spesifikasinya mulai ditambah dan diperbaiki(Jain dkk, 1999). Penambahan Fitur Galton ditunjukkan pada Gambar 2.2. Seperti yang terlihat pada gambar, ada beberapa penambahan pada Fitur Galton yang asli, tapi kebanyakan tidak digunakan pada sistem identifikasi sidik jari otomatis(Jain dkk, 1999).

Bersamaan dengan itu, sidik jari terdiri dari dua tipe fitur khusus yang disebut dengan poin *core* dan *delta*. Poin-poin tersebut sering ditunjuk sebagai poin istimewa dari suatu sidik jari(Jain dkk, 1999). Contoh dari *core* dan *delta* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Poin *core* dan *delta* pada sidik jari

Berdasarkan jurnal *project* akhir yang dikerjakan oleh Elvayandri, terdapat 16 fitur pada guratan sidik jari seperti yang terlihat pada Gambar 2.4.



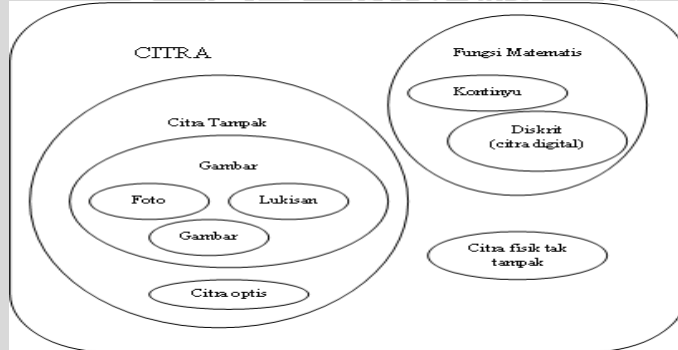
Gambar 2.4 Fitur pada guratan sidik jari

2.2 Pemrosesan Citra Digital

2.2.1 Pengertian Citra

Definisi citra menurut Kamus Webster adalah “suatu representasi, kemiripan atau imitasi dari suatu obyek atau benda”(Achmad dan Firdausy, 2005). Citra dapat dikelompokkan menjadi citra tampak dan citra tidak tampak(Achmad dan Firdausy, 2005). Contoh citra tampak dalam kehidupan sehari-hari adalah foto, lukisan, sesuatu yang nampak pada layar monitor dan televisi, serta hologram (citra optis). Sedangkan citra tak tampak misalnya data gambar dalam file (citra digital), dan citra yang dipresentasikan menjadi fungsi matematis. Di samping itu terdapat juga citra fisik tak tampak, misalnya citra distribusi panas di kulit manusia serta peta densitas dalam suatu material. Untuk dapat terlihat oleh mata manusia, citra tak tampak ini harus diubah menjadi citra tampak, misalnya dengan menampilkannya di monitor, dicetak di atas kertas, dan sebagainya(Achmad dan Firdausy, 2005).

Di antara jenis-jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain, jika hendak diolah dengan komputer, harus diubah dulu menjadi citra digital, misalnya foto di-*scan* dengan *scanner*, persebaran panas tubuh ditangkap dengan kamera infra merah dan diubah menjadi informasi numeris, informasi densitas dan komposisi bagian dalam tubuh manusia ditangkap dengan bantuan pesawat sinar-X dan sistem deteksi radiasi menjadi informasi *digital*. Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*) (Achmad dan Firdausy, 2005). Gambar 2.5 menunjukkan pengelompokan beberapa jenis citra.



Gambar 2.5 Pengelompokan jenis-jenis citra

Setiap citra *digital* memiliki beberapa karakteristik, antara lain ukuran citra, resolusi, dan format nilainya. Umumnya citra *digital* berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau piksel, sehingga ukuran citra selalu bernilai bulat.

Ukuran citra dapat juga dinyatakan secara fisik dalam satuan panjang (misalnya mm atau *inch*). Dalam hal ini tentu saja harus ada hubungan antara ukuran titik penyusun citra dengan satuan panjang. Hal tersebut dinyatakan dengan resolusi yang merupakan ukuran banyaknya titik untuk setiap panjang. Biasanya satuan yang digunakan adalah dpi (*dot per inch*). Makin besar resolusi makin banyak titik yang terkandung dalam citra dengan ukuran fisik yang sama. Hal ini memberikan efek penampakan citra menjadi semakin halus (Achmad dan Firdausy, 2005).

Komputer dapat mengolah isyarat-isyarat elektronik *digital* yang merupakan kumpulan sinyal biner (bernilai dua: 0 dan 1). Untuk itu, citra *digital* harus mempunyai format tertentu yang sesuai sehingga dapat merepresentasikan obyek pencitraan dalam bentuk kombinasi data biner.

Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data *digital* tersebut merepresentasikan warna dari citra yang diolah, dengan demikian format data citra *digital* berhubungan erat dengan warna. Format citra *digital* yang banyak dipakai adalah citra biner, skala keabuan, warna, dan warna berindeks (Achmad dan Firdausy, 2005).

Pada citra biner, setiap titik bernilai 0 atau 1, masing-masing merepresentasikan warna tertentu. Contoh yang paling lazim: warna hitam bernilai 0 dan warna putih bernilai 1. Setiap titik pada citra hanya membutuhkan 1 bit, sehingga setiap *byte* dapat menampung informasi 8 titik (Achmad dan Firdausy, 2005).

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain di antara nilai minimum (biasanya = 0) dan nilai maksimumnya. Banyaknya kemungkinan nilai dan nilai maksimumnya bergantung pada jumlah bit yang digunakan. Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah $2^4 = 16$, dan nilai maksimumnya adalah $2^4 - 1 = 15$; sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$, dan nilai maksimumnya adalah $2^8 - 1 = 255$.

Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimum dan warna putih sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu. Namun pada prakteknya warna yang dipakai tidak terbatas pada warna abu-abu; sebagai contoh dipilih warna minimalnya adalah putih dan warna maksimalnya adalah merah, maka semakin besar nilainya semakin besar pula intensitas warna merahnya. Beberapa buku menyebut format citra ini sebagai citra intensitas (Achmad dan Firdausy, 2005).

Pada citra warna, setiap titik mempunyai warna yang spesifik yang merupakan kombinasi dari 3 warna dasar, yaitu: merah, hijau dan biru. Format citra ini sering disebut sebagai citra RGB (*red-green-blue*). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit), misalnya warna kuning merupakan kombinasi warna merah dan hijau sehingga nilai RGB-nya adalah 255 255 0; sedangkan warna ungu muda nilai RGB-nya adalah 150 0 150. Dengan demikian setiap titik pada citra warna membutuhkan data 3 byte.

Jumlah kombinasi warna yang mungkin untuk format citra ini adalah 2^{24} atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada, inilah sebabnya format ini dinamakan *true color* (Achmad dan Firdausy, 2005).

Jumlah memori yang dibutuhkan untuk format citra warna *true color* adalah 3 kali jumlah titik yang ada dalam citra yang ditinjau. Di lain pihak, pada kebanyakan kasus, jumlah warna yang ada dalam suatu citra terkadang sangat terbatas (jauh di bawah 16 juta kemungkinan warna yang ada), karena banyaknya warna dalam sebuah citra tidak mungkin melebihi banyaknya titik dalam citra itu sendiri. Untuk kasus tersebut, disediakan format citra warna berindeks. Pada format ini, informasi setiap titik merupakan indeks dari suatu tabel yang berisi informasi warna yang tersedia, yang disebut palet warna (beberapa buku menyebutkan sebagai *color map*).

Jumlah bit yang dibutuhkan oleh setiap titik pada citra bergantung pada jumlah warna yang tersedia dalam palet warna. Sebagai contoh, untuk palet berukuran 16 warna, setiap titik membutuhkan 4 bit; dan untuk palet berukuran 256 warna, setiap titik membutuhkan 8 bit atau 1 byte. Palet warna merupakan bagian

dari citra warna berindeks, sehingga pada saat menyimpan citra ini ke dalam file, informasi palet warna juga harus disertakan.

Keuntungan pemakaian palet warna ini adalah kita dapat dengan cepat memanipulasi warna tanpa harus mengubah informasi pada setiap titik dalam citra. Keuntungan lainnya adalah besarnya data yang diperlukan untuk menyimpan citra ini lebih kecil dibandingkan dengan citra warna *true color* (Achmad dan Firdausy, 2005).

2.2.2 Pengambilan Citra dan Peralatan Tampilan

Data citra yang diproses oleh komputer berbentuk data angka. Citra yang berbentuk data analog harus dikonversi ke dalam bentuk data angka sebelum diproses komputer. Proses ini disebut digitasi (*digitization*). Proses digitasi melibatkan dua proses lain, yaitu proses *sampling* dan *quantization*. Proses *sampling* berhubungan dengan pengambilan/pemilihan titik-titik keseluruhan citra yang diobservasi. Karakteristik citra untuk setiap titik diukur dengan suatu besaran bit-bit memori komputer, yang disebut dengan proses *quantization*. Kualitas citra yang diperoleh tergantung dari kepadatan titik sample (*spatial resolution*) dan ketelitian dari ukuran (*grey level resolution*). Sedangkan ketelitian ukuran akan tergantung dari kapasitas bit setiap satuan memori komputer yang digunakan (Sumardika, 1996).

Pengambilan citra suatu obyek memerlukan alat yang dapat melakukan transformasi informasi cahaya ke dalam sinyal listrik. Dari sinyal listrik ini kemudian dilakukan proses *sampling* dan *quantization* dengan menggunakan ADC (*Analog to Digital Converter*). Alat tersebut dapat berupa kamera televisi atau video (Sumardika, 1996).

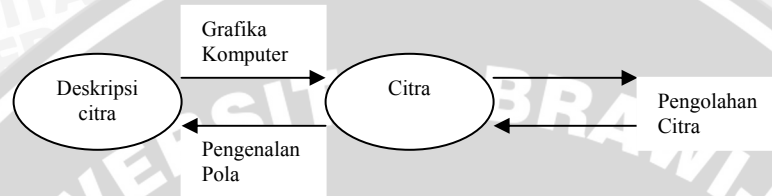
Ada beberapa alat pengambilan citra yang berkecepatan lebih lambat dari kamera televisi atau video dengan hasil yang berkualitas jauh lebih baik. Sebagai contoh misalnya *scanner* (*drum scanner*, *flying spot scanner* atau *handy scanner*) (Sumardika, 1996).

Proses transformasi data citra (data digital) ke dalam bentuk citra pada peralatan tampilan memerlukan waktu yang lebih lama dibandingkan dengan proses pengambilan data citra objek. Hal ini disebabkan oleh bentuk data citra masukan yang harus diolah dan dikonversi ke dalam bentuk data citra yang dapat diterima oleh peralatan tampilan yang digunakan. Sebagai contoh pada peralatan tampilan berupa CRT (*Cathode Ray Tube device*), terjadi proses transformasi data digital ke dalam bentuk data analog (*Digital-to-*

Analog). Demikian juga misalkan pada peralatan pencetak (*hard copy*), data citra masukan harus diurut terlebih dulu, karena pada umumnya peralatan pencetak melakukan proses yang bersifat *one-pass* (satu kali jalan) (Sumardika, 1996).

2.2.3 Pengolahan Data Citra

Pengolahan data citra membutuhkan bentuk dan metode yang berbeda sesuai dengan tujuan aplikasi yang dibuat. Pengolahan data citra ini dapat dibedakan menjadi tiga bidang yaitu Grafika Komputer (*Computer Graphics*), Pengolahan Citra (*Image Processing*), dan Pengenalan Pola (*Pattern Recognition*) (Sumardika, 1996). Gambar 2.6 menunjukkan hubungan antara ketiga bidang diatas.



Gambar 2.6 Diagram hubungan antara Grafika Komputer, Pengolahan Citra dan Pengenalan Pola

Grafika komputer merupakan proses pembuatan citra dari informasi bukan gambar, yang mencakup bermacam-macam aplikasi yang memiliki kompleksitas program ataupun kebutuhan perhitungan untuk menghasilkan suatu tampilan gambar. Contoh dari tampilan gambar yang dimaksud misalnya pada sebuah program simulasi permainan yang memerlukan animasi gambar yang terkadang membutuhkan suatu perhitungan seperti perhitungan jarak pandang perspektif dan penempatan atau pembuatan gambar pada posisi tertentu (Sumardika, 1996).

Pengolahan citra adalah pemrosesan informasi masukan yang berupa data citra dengan langkah dan metode tertentu untuk menghasilkan citra dengan bentuk dan karakteristik yang diharapkan. Contohnya adalah sistem transmisi citra yang berhubungan dengan penghilangan gangguan (*noise*). Citra yang terlalu kabur, gelap atau terlalu terang dapat diperbaiki dengan teknik peningkatan citra (*image enhancement techniques*). Contoh lain misalnya

penyederhanaan citra yang memiliki tingkat warna keabuan (*grey level*) yang relatif besar menjadi citra yang memiliki dua warna, hitam dan putih. Selain itu, dari tingkat warna keabuan dan karakteristik tekstur yang dimiliki oleh suatu citra dapat dibentuk citra berarah (*directional image*) atau bertipe vektor (Sumardika, 1996).

Pengenalan pola berlawanan dengan grafika komputer. Pada pengenalan pola terdapat metode-metode ekstraksi citra untuk menghasilkan deskripsi citra dalam bentuk kumpulan bilangan, untaian simbol atau sebuah *graph*. Untuk pemrosesan lebih lanjut, dari bentuk ini menghasilkan penggolongan citra asal ke dalam suatu kelas berdasarkan keistimewaan yang dimiliki citra tersebut. Dengan deskripsi dan penggolongan ini dapat dilakukan pengenalan kembali citra yang dimaksud (Sumardika, 1996).

2.2.4 Konversi Citra *True Color* ke *Grayscale*

Citra *true color* dapat dikonversi menjadi citra *grayscale* dengan operasi titik. Secara mudahnya, intensitas didefinisikan sebagai nilai rata-rata dari ketiga nilai elemen warna, sehingga nilai keabuan yang mempresentasikan intensitas dapat dihitung dengan persamaan 2.1 (Achmad dan Firdausy, 2005).

$$Grayscale = \frac{R + B + G}{3} \dots\dots\dots 2.1$$

dimana *Grayscale* merupakan nilai keabuan pada suatu piksel. Sedangkan *R* (*red*), *G* (*green*), dan *B* (*blue*) merupakan tiga warna dasar yang dimiliki tiap piksel pada citra *true color* (citra warna).

2.2.5 Segmentasi (*Segmentation*)

Segmentasi sebuah citra sangat penting untuk menganalisa data dari citra tersebut. Segmentasi merupakan proses untuk memisahkan objek pada suatu citra dari daerah *background*-nya (Thai, 2003). Metode segmentasi dapat dibagi menjadi tiga bagian penting, pertama *Global Knowledge* yang berisi tentang histogram dari citra, kedua *Edge-Based* merupakan segmentasi yang dititikberatkan pada pencarian garis dari citra, dan yang ketiga *Region-Based* merupakan segmentasi yang membagi citra kedalam bagian-bagian seperti tingkat kecerahannya (Gonzalez dan Woods, 1992).

Pada segmentasi citra yang menggunakan *Simple Global Thresholding*, pixel-pixel dari citra akan ditandai apakah berupa objek atau *background*. Pengenalan objek dengan *background* didapatkan dari pixel-pixel citra tersebut melalui perbedaan nilai skala keabuan(*grey-level*) (Gonzalez dan Woods, 1992).

Pada proses segmentasi citra sidik jari, daerah objek yang akan dipisahkan dengan *background*nya adalah area yang berisi *ridge* dan *valley*. Sedangkan *background*nya sendiri adalah daerah di luar batas sidik jari yang tidak berisi informasi sidik jari yang valid. Jika algoritma pendeteksi *minutiae* dijalankan pada *background* citra sidik jari, maka akan dihasilkan *minutiae* yang salah. Maka dari itu segmentasi digunakan untuk membuang daerah *background* tersebut untuk memudahkan pendeteksian *minutiae* yang sebenarnya. *Background* tersebut biasanya memperlihatkan nilai varian *grey-scale* yang sangat rendah, tapi sebaliknya daerah objek memiliki nilai varian yang sangat tinggi. Maka dari itu, metode dengan berbasis pada pengembangan varian (*variance thresholding*) dapat dimanfaatkan untuk segmentasi. Pertama, citra dibagi ke dalam beberapa blok dengan ukuran $W \times W$ piksel dan pada tiap blok dihitung varian *grey-scale*nya. Jika nilai varian kurang dari sama dengan *global threshold* ($\leq T$), maka blok yang bersangkutan ditandai sebagai daerah *background* dengan memberinya nilai hitam (0). Sebaliknya, jika nilai varian lebih dari *global threshold* ($> T$), maka blok yang bersangkutan ditandai sebagai daerah objek. Varian *grey-level* pada suatu blok dengan ukuran $W \times W$ dapat dihitung dengan persamaan 2.3.

$$M(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} I(i, j) \dots\dots\dots 2.2$$

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (I(i, j) - M(k))^2 \dots\dots\dots 2.3$$

dimana $V(k)$ adalah varian pada blok k , $I(i, j)$ adalah nilai *grey-level* pada piksel (i, j) , dan $M(k)$ adalah mean dari nilai *grey-level* pada blok k (Thai, 2003). Sedangkan nilai untuk *variance threshold*nya (T) adalah 100 (Thai, 2003).



Gambar 2.7 Hasil dari segmentasi dengan menggunakan *variance threshold* 100 dan ukuran blok 16 x 16 piksel.
(a) Citra asli. (b) Citra Tersegmentasi.

Gambar 2.7 menunjukkan hasil dari proses segmentasi pada citra sidik jari dengan menggunakan *variance threshold* 100 dan ukuran blok 16 x 16 piksel. Area gelap pada gambar tersebut merupakan *background* dari objek sidik jari.

2.2.6 Normalisasi (*Normalization*)

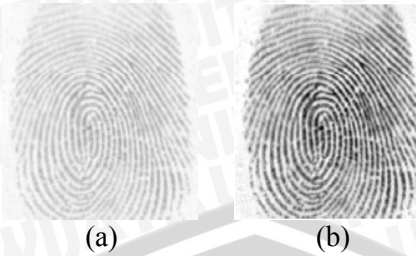
Pada pemrosesan citra, normalisasi adalah suatu proses yang digunakan untuk menstandarisasi nilai intensitas pada suatu citra dengan menyesuaikan cakupan nilai *grey-level* sehingga berada pada cakupan nilai yang diharapkan. Dimisalkan $I(i,j)$ merepresentasikan nilai *grey-level* pada pixel (i,j) , dan $N(i,j)$ merepresentasikan nilai *grey-level* yang dinormalisasikan pada pixel (i,j) . Gambar yang dinormalisasi didefinisikan pada persamaan 2.4.

$$N_{(i,j)} = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i,j) - M)^2}{V}} & \text{if } I(i,j) > M \\ M_0 - \sqrt{\frac{V_0(I(i,j) - M)^2}{V}} & \text{otherwise,} \end{cases} \quad \dots 2.4$$

dimana M dan V adalah mean dan varian dari $I(i,j)$, sedangkan M_0 dan V_0 adalah nilai mean dan varian yang diharapkan (Thai, 2003).

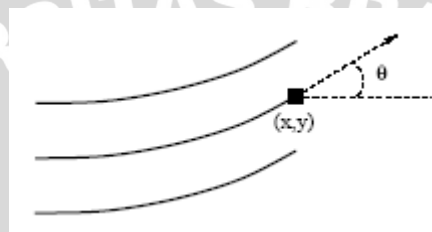
Normalisasi tidak mengubah struktur *ridge* pada sidik jari. Normalisasi digunakan untuk menstandarisasi level dinamik dari varian pada nilai *grey-level*, sehingga akan memudahkan langkah-langkah peningkatan citra berikutnya (Thai, 2003). Normalisasi bermanfaat untuk meningkatkan kontras antara pola *ridge* dan

valley(Thai, 2003). Gambar 2.8 merupakan contoh citra sidik jari hasil normalisasi dengan M_o dan V_o masing-masing 100 dan 100.



Gambar 2.8 Hasil Normalisasi dengan M_o (Mean yang diharapkan) 100 dan V_o (Varian yang diharapkan) 100.
(a) Citra asli (b) Citra ternormalisasi

2.2.7 Image Orientation Estimation



Gambar 2.9 Orientasi piksel *ridge* pada sidik jari

Bidang orientasi dari citra sidik jari mendefinisikan sudut dari *local ridge* (*local ridge orientation*) yang ada pada sidik jari. Penghitungan orientasi (*Orientation estimation*) merupakan langkah mendasar pada proses peningkatan citra sidik jari dimana langkah pemfilteran *Gabor* selanjutnya bergantung pada *local orientation* tersebut sehingga dapat melakukan peningkatan kualitas citra sidik jari secara efektif(Thai, 2003). Berikut merupakan langkah-langkah untuk menghitung orientasi pada piksel (i,j) (Thai, 2003).

1. Pertama, suatu blok berukuran $W \times W$ dipusatkan pada piksel (i,j) pada citra sidik jari ternormalisasi.
2. Pada masing-masing piksel pada blok dihitung gradien $\partial_x(i,j)$ dan $\partial_y(i,j)$, yang merupakan *gradient magnitude* arah x dan y . Operator Sobel horizontal digunakan untuk menghitung gradien $\partial_x(i,j)$:

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \dots\dots\dots 2.5$$

dan Operator Sobel vertikal digunakan untuk menghitung gradien $\partial_y(i,j)$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \dots\dots\dots 2.6$$

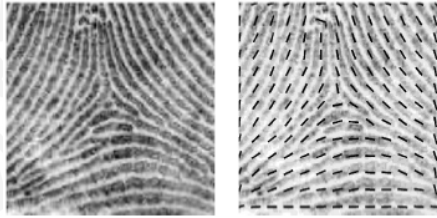
3. *Local Orientation* $\theta(i,j)$ pada piksel (i,j) dapat dihitung menggunakan persamaan 2.9.

$$V_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v), \dots\dots\dots 2.7$$

$$V_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \partial_x^2(u,v)\partial_y^2(u,v), \dots\dots\dots 2.8$$

$$\theta(i,j) = \frac{1}{2} \tan^{-1} \frac{V_y(i,j)}{V_x(i,j)}, \dots\dots\dots 2.9$$

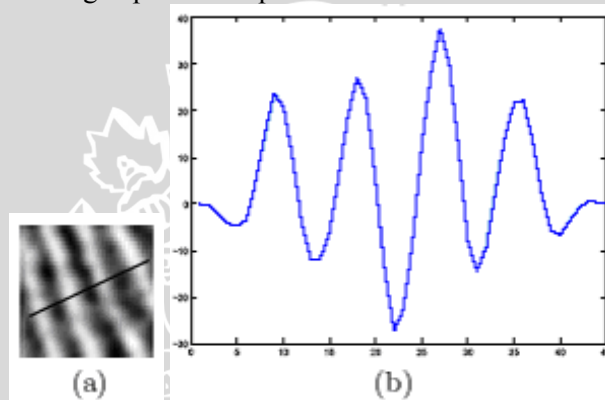
Terkadang perhitungan *local orientation* dari citra asli yang ternormalisasi menunjukkan hasil yang kurang akurat. Oleh karena itu, sebelum menghitung *local orientation*, citra ternormalisasi dihaluskan terlebih dahulu dengan menggunakan *Gaussian Low Pass Filter*. Gambar 2.10 menunjukkan *orientation estimated* pada citra sidik jari.



Gambar 2.10 *Orientation estimated* pada citra sidik jari

2.2.8 Ridge Frequency Estimation

Parameter penting lainnya yang digunakan dalam pembangunan *Gabor filter* adalah *local ridge frequency* (frekuensi ridge lokal). *Frequency image* merepresentasikan frekuensi lokal dari *ridge* yang ada pada sidik jari. Langkah awal pada tahap *frequency estimation* adalah membagi citra ke dalam beberapa blok dengan ukuran $W \times W$. Langkah selanjutnya adalah memproyeksikan nilai *grey-level* dari piksel-piksel pada masing-masing blok sepanjang arah yang tegak lurus dengan arah *local ridge orientation*. Proyeksi ini membentuk hampir seperti gelombang *sinusoidal* dengan poin minimum lokal sesuai dengan *ridge* pada sidik jari (Thai, 2003). Contoh dari proyeksi bentuk gelombang dapat dilihat pada Gambar 2.11.



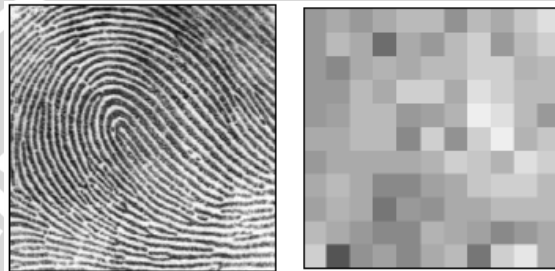
Gambar 2.11 Proyeksi nilai intensitas piksel sepanjang arah orthogonal hingga *local ridge orientation*.

- (a) Blok berukuran 32x32 dari citra sidik jari.
- (b) Proyeksi bentuk gelombang dari blok tersebut

Pengukuran jarak *ridge* $S(i,j)$ dilakukan dengan menghitung nilai median piksel diantara poin-poin minimum yang berurutan pada gelombang proyeksi. Untuk mendapatkan poin-poin minimum yang akurat, maka citra sidik jari dihaluskan dengan menggunakan *Gaussian Low Pass Filter*. Kemudian frekuensi *ridge* $F(i,j)$ untuk blok yang dipusatkan pada piksel (i,j) dapat dihitung dengan persamaan 2.10(Thai, 2003):

$$F(i,j) = \frac{1}{S(i,j)} \dots\dots\dots 2.10$$

Gambar 2.12 menunjukkan *Estimated ridge wavelength* pada citra sidik jari.



Gambar 2.12 *Estimated ridge wavelength* pada citra sidik jari

2.2.9 Gaussian Low Pass Filter

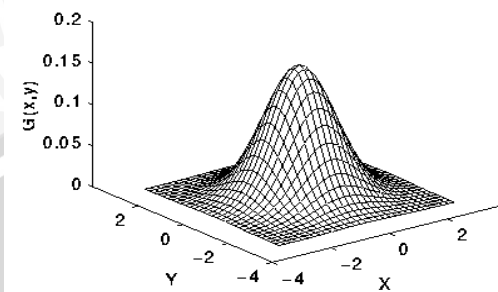
Gaussian low pass filter merupakan suatu tipe filter penghalus citra yang menggunakan distribusi normal untuk menghitung transformasi pada tiap piksel citra. *Gaussian low pass filter* sangat membantu dalam meminimalisir *noise* yang ada pada citra. Persamaan 2.11 menunjukkan perhitungan *Gaussian filter* dua dimensi.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \dots\dots\dots 2.11$$

dimana σ merupakan standar deviasi *Gaussian*.

Gambar 2.13 menunjukkan plot bobot *mask Gaussian* dengan $\sigma = 1$. Terlihat titik pada pusat *mask* memiliki bobot yang paling tinggi. Makin jauh jarak sebuah titik dengan titik pusat tersebut makin kecil

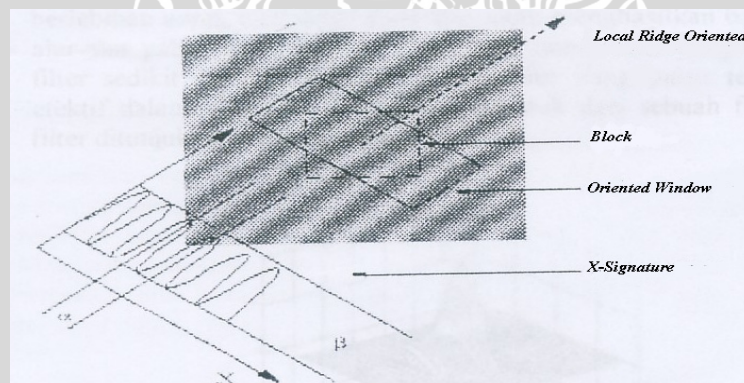
bobotnya, atau dengan kata lain makin kecil pengaruhnya terhadap titik pusat. Untuk nilai σ yang berbeda, bentuk plot masih sama, hanya kelandaannya yang berubah. Nilai σ yang besar menyebabkan plot makin pendek (landai) namun lebar, sehingga lebih banyak titik tetangga yang memberi kontribusi terhadap operasi konvolusi, sementara nilai σ yang kecil menyebabkan plot makin tinggi dan sempit (Achmad dan Firdausy, 2005).



Gambar 2.13 Bobot mask Gaussian dengan $\sigma = 1$

2.2.10 Gabor Filter dan Peningkatan Kualitas Citra

Konfigurasi alur-alur (*ridge* dan *valley*) yang paralel dari sebuah sidik jari mendefinisikan parameter-peremeter frekuensi dan arah dari alur tersebut. Parameter-parameter ini merupakan informasi yang berguna untuk menghilangkan noise yang tidak diinginkan (Meidiyanto, 2002). Bentuk gelombang sinusoidal dari kumpulan *ridge* dan *valley* ditunjukkan pada Gambar 2.14.



Gambar 2.14 Bentuk sinusoidal dari alur sidik jari

Bentuk gelombang sinusoidal dari *ridge* dan *valley* berubah-ubah secara pelan dalam sebuah arah yang tetap. Oleh karena itu, sebuah *bandpass filter* yang mempunyai parameter frekuensi dan arah/orientasi akan dapat mengurangi *noise* yang tidak diinginkan secara efisien dan menjaga bentuk/struktur alur-alur sidik jari sesuai dengan keadaan yang sesungguhnya (Meidiyanto, 2002).

Gabor filter memiliki kedua parameter tersebut, yaitu selektif terhadap frekuensi dan orientasi. Oleh sebab itu *gabor filter* dapat digunakan sebagai sebuah *bandpass filter* untuk menghilangkan *noise* dan dengan tetap menjaga bentuk asli dari alur-alur pada sebuah gambar sidik jari. Rumus umum dari *even-symmetric gabor filter* di dalam *spatial domain* dapat dilihat pada persamaan 2.12 (Meidiyanto, 2002).

$$G(x, y; \theta, f) = \exp \left\{ -\frac{1}{2} \left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2} \right] \right\} \cos(2\pi f x_{\theta}) \dots\dots 2.12$$

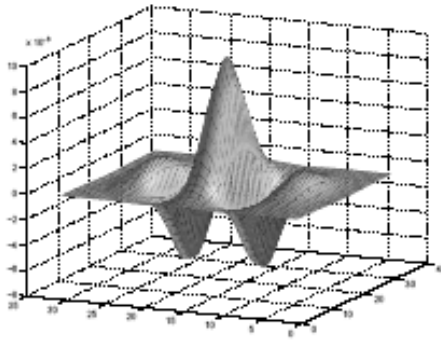
dimana:

$$x_{\theta} = x \cos \theta + y \sin \theta, \dots\dots\dots 2.13$$

$$y_{\theta} = -x \sin \theta + y \cos \theta, \dots\dots\dots 2.14$$

θ adalah orientasi dari *gabor filter*, f adalah frekuensi dari gelombang sinusoidal dari sebuah gambar sidik jari, dan σ_x dan σ_y adalah standard deviasi fungsi Gaussian sepanjang sumbu x dan y . Pemilihan nilai σ_x dan σ_y ditentukan secara eksperimen. Nilai yang terlalu besar terlalu berlebihan untuk memfilter *noise* dan akan menghasilkan banyak sekali alur-alur palsu. Sedangkan di lain pihak, untuk nilai yang terlalu kecil filter sedikit sekali menghasilkan alur-alur yang palsu tetapi kurang efektif dalam menghilangkan *noise* (Meidiyanto, 2002). Bentuk dari sebuah fungsi *gabor filter* ditunjukkan pada Gambar 2.15.





Gambar 2.15 Bentuk *gabor filter* dalam daerah spasial

Filter Gabor diterapkan pada citra sidik jari dengan membalut keseluruhan citra dengan filter. Pembalutan piksel (i,j) pada citra membutuhkan nilai orientasi yang bersesuaian $\theta(i,j)$ dan nilai frekuensi *ridge* $F(i,j)$ dari piksel tersebut. Maka dari itu, aplikasi dari filter Gabor G untuk menghasilkan peningkatan citra E dapat ditunjukkan pada persamaan 2.15.

$$E(i,j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G(u,v; \theta(i,j), F(i,j)) N(i-u, j-v) \dots 2.15$$

dimana θ merupakan citra orientasi (*orientation image*), F adalah citra frekuensi *ridge* (*ridge frequency image*), N citra sidik jari ternormalisasi, dan w_x dan w_y adalah lebar dan tinggi *Gabor filter mask* (Thai, 2003).

Luas bidang filter, yang menentukan cakupan frekuensi, ditentukan oleh parameter standar deviasi σ_x dan σ_y . Pada algoritma asli milik Hong dan kawan-kawan, secara empiris nilai σ_x dan σ_y diset 4,0 dan 4,0 secara berturut-turut (Thai, 2003). Kemudian nilai w_x dan w_y diset 11 untuk keduanya (Thai, 2003).



Gambar 2.16 Hasil peningkatan citra sidik jarimenggunakan *Gabor Filter*. (a) Citra asli
(b) Citra yang ditingkatkan kualitasnya (*enhanced image*)

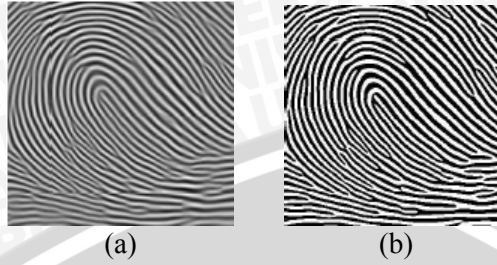
2.2.11 Binerisasi (*Binarization*)

Definisi *binary image* adalah citra yang direpresentasikan sebagai himpunan dari *pixel-pixel foreground* (obyek depan). *Binary image* merupakan citra yang hanya mempunyai dua informasi intensitas warna, yaitu hitam dan putih. konvensi untuk menentukan intensitas warna pada *binary image* bisa berbeda-beda, namun pada umumnya digunakan konvensi yang mendeskripsikan nilai 0 sebagai warna hitam dan nilai 1 atau 255 sebagai warna putih. Warna putih biasanya juga digunakan untuk warna *foreground* (objek), sedangkan warna hitam adalah warna *background*, namun sekali lagi ketentuan ini bukan merupakan keharusan tergantung dari *binary image* yang dihasilkan. *Binary image* seringkali dihasilkan dari proses *thresholding* suatu *grayscale image* (citra yang memiliki informasi tingkat keabuan) dan *color image* (citra yang memiliki informasi intensitas warna yang bervariasi) (Ananta, 2003).

Algoritma deteksi *minutiae* lebih banyak beroperasi pada citra biner dimana hanya ada dua level yang menjadi pusat perhatian, yaitu piksel hitam untuk merepresentasikan *ridge* dan piksel putih untuk merepresentasikan *valley*. Binerisasi digunakan untuk meningkatkan kontras antara *ridge* dan *valley* pada citra sidik jari dan tentu saja memudahkan dalam pendeteksian *minutiae* (Thai, 2003).

Binerisasi citra sidik jari dilakukan dengan menggunakan dua *threshold* (T_1 dan T_2). Proses binerisasi mengukur nilai *grey-level* pada citra yang telah ditingkatkan kualitasnya. Jika nilainya kurang dari sama dengan T_1 maka piksel diberi nilai biner 0, dan jika

nilainya lebih dari sama dengan T_2 maka piksel diberi nilai biner 1. Hasil dari citra biner berisi dua level informasi, yaitu objek *ridge* dan *background valley*. Gambar 2.17 menunjukkan citra hasil proses binerisasi.



Gambar 2.17 Hasil binerisasi citra sidik jari
(a) *Enhanced Image* (b) Citra biner

2.2.12 Penipisan (*Thinning*)

Definisi *image thinning* adalah proses *morphology image* yang merubah bentuk asli *binary image* menjadi *image* yang menampilkan batas-batas obyek/*foreground* hanya setebal satu *pixel*. Algoritma *thinning* secara iteratif ‘menghapus’ *pixel-pixel* pada *binary image*, dimana transisi dari 0 ke 1 (atau dari 1 ke 0 pada konvensi lain) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu *pixel*) terhubung menjadi suatu garis (Nugroho dkk, 2003).

Sepintas, *image thinning* mempunyai kemiripan dengan *edge detection* dalam hal output dari citra yang dihasilkan, kedua proses tersebut sama-sama menampilkan batas obyek pada citra. Namun, tetap saja ada perbedaan antara *Image Thinning* dengan *Edge Detection* dari segi prinsip kerjanya, yaitu :

- *Edge detection* : merubah *graylevel image* atau *color image* menjadi *image* yang menampilkan batas-batas/*boundaries* obyek berdasarkan kekontrasan warna antar *pixel*.
- *Image Thinning* : mereduksi *pixel-pixel* pada obyek *binary image* menjadi *pixel* yang bernilai sama dengan nilai *pixel* pada *background*. Menghasilkan *binary image* dengan informasi berupa batas-batas obyek berdasarkan *pixel* dengan ketebalan satu *pixel* (Nugroho dkk, 2003).

Tujuan *image thinning* adalah untuk menghilangkan *pixel-pixel* yang berada didalam obyek depan (*foreground object*) pada *binary images*(Nugroho dkk, 2003).

Manfaat *image thinning* adalah sebagai berikut :

- Biasanya diterapkan pada proses *skeletonisasi*.
- Berguna untuk merapikan/menyempurnakan hasil output proses *edge detection* dengan cara mengurangi lebar sisi/batas/*edge*(Nugroho dkk, 2003).

Langkah akhir dalam usaha peningkatan kualitas citra sidik jari adalah *thinning*. *Thinning* merupakan proses analisis secara berturut-turut mengikis piksel-piksel hingga menjadi satu piksel saja. Aplikasi algoritma *thinning* citra sidik jari mempertahankan konektifitas struktur *ridge* saat membentuk versi *skeleton* citra biner. Citra *skeleton* ini kemudian digunakan dalam proses pendeteksian *minutiae*(Thai, 2003).

Diasumsikan objek pada citra bernilai 1 dan *background*-nya bernilai 0. Metode ini terdiri dari 2 langkah dasar yang dikenakan terhadap *contour points* dari suatu *region*, dimana *contour points* adalah sembarang piksel dengan nilai 1 dan memiliki paling sedikit satu dari 8-tetangga bernilai 0. Algoritma ini menggunakan tanda untuk memilih piksel mana yang akan dihapus. Aturannya, 8-tetangga terdekat dari setiap piksel p_1 dinomori p_2 (untuk piksel di atas p_1) sampai dengan p_9 sesuai dengan arah jarum jam seperti yang terlihat pada Gambar 2.18(Ananta dkk, 2003).

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Gambar 2.18 Pemberian nomor pada 8-tetangga piksel p_1 untuk proses *thinning*

Langkah 1

Beri tanda sebuah *contour point* p_1 untuk dihapus apabila memenuhi semua kondisi-kondisi berikut ini :

- $2 \leq N(p_1) \leq 6$;
- $S(p_1) = 1$;

$$(c) \quad p_2 \cdot p_4 \cdot p_6 = 0;$$

$$(d) \quad p_4 \cdot p_6 \cdot p_8 = 0;$$

dimana $N(p_1)$ adalah jumlah dari tetangga-tetangga dari p_1 yang bukan nol; sehingga, $N(p_1) = (p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9)$ dan $S(p_1)$ adalah jumlah dari transisi 0 ke 1 dalam urutan $p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$.

Langkah 2

Hampir sama dengan langkah 1, hanya saja pada langkah 2 ini bagian (c) dan (d) berubah menjadi sebagai berikut :

$$(c) \quad p_2 \cdot p_4 \cdot p_8 = 0;$$

$$(d) \quad p_2 \cdot p_6 \cdot p_8 = 0;$$

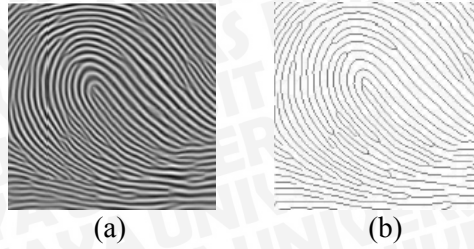
Langkah 1 diterapkan untuk *border pixel* pada *binary region*. Jika satu atau lebih dari kondisi (a) sampai dengan (d) tidak dipenuhi, maka nilai dari *point* yang diperiksa tidak berubah (tidak perlu diberi tanda). *Points* yang diperiksa tidak akan dihapus sampai semua *border points* selesai diproses. Setelah langkah 1 telah selesai dilakukan terhadap semua *border points*, semua piksel yang telah diberi tanda dihapus (diubah ke 0). Lalu, langkah 2 baru dijalankan terhadap hasil data persis sama seperti pada langkah 1.

Jadi satu iterasi dalam algoritma *thinning* ini terdiri dari :

1. Pengerjaan langkah 1 untuk memberi tanda pada *border points* untuk dihapus.
2. Penghapusan *points* yang telah diberi tanda.
3. Pengerjaan langkah 2 untuk memberi tanda *border points* yang tersisa untuk dihapus.
4. Penghapusan *points* yang telah diberi tanda.

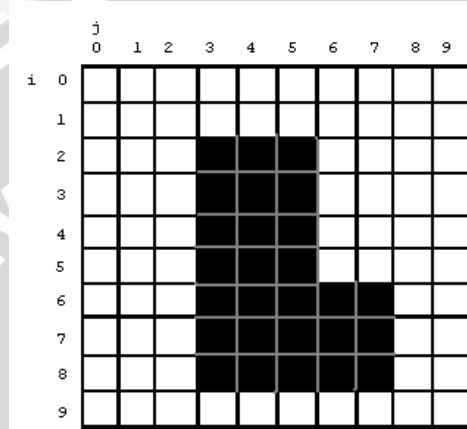
Prosedur dasar ini akan beriterasi hingga tidak ada *points* yang dapat dihapus lagi sehingga hasil yang didapat adalah *skeleton* (kerangka) dari *region* (Ananta dkk, 2003). Gambar 2.19 menunjukkan citra hasil proses *thinning*.





Gambar 2.19 Hasil penipisan citra sidik jari
(a) *Enhanced image* (b) *Thinned image*

Berikut merupakan contoh manual proses *thinning*. Gambar 2.20 merupakan contoh gambar yang akan *dithinning* (ditipiskan).



Gambar 2.20 Contoh gambar yang akan *dithinning* (ditipiskan)

Diasumsikan piksel berwarna hitam bernilai 1 dan piksel berwarna putih bernilai 0. Perlu diingat bahwa piksel yang akan diperiksa adalah piksel yang bernilai 1 (berwarna hitam) dan memiliki paling sedikit 1 dari 8-tetangga yang bernilai 0 (berwarna putih). Tabel 2.1 merupakan hasil pemeriksaan pada langkah ke-1 terhadap piksel yang memenuhi syarat.

Tabel 2.1 Hasil pemeriksaan pada langkah ke-1 terhadap piksel yang memenuhi syarat

Piksel [i,j]	N(P ₁)	S(P ₁)	$2 \leq N(P_1) \leq 6$	$S(P_1) = 1$	$p_2 \cdot p_4 \cdot p_6 = 0$	$p_4 \cdot p_6 \cdot p_8 = 0$	hasil
[2,3]	3	1	T	T	T	T	X
[2,4]	5	1	T	T	T	F	-
[2,5]	3	1	T	T	T	T	X
[3,3]	5	1	T	T	F	T	-
[3,5]	5	1	T	T	T	T	X
[4,3]	5	1	T	T	F	T	-
[4,5]	5	1	T	T	T	T	X
[5,3]	5	1	T	T	F	T	-
[5,5]	6	1	T	T	T	T	X
[6,3]	5	1	T	T	F	T	-
[6,5]	7	1	F	T	F	F	-
[6,6]	6	1	T	T	T	F	-
[6,7]	3	1	T	T	T	T	X
[7,3]	5	1	T	T	F	T	-
[7,7]	5	1	T	T	T	T	X
[8,3]	3	1	T	T	T	T	X
[8,4]	5	1	T	T	T	T	X
[8,5]	5	1	T	T	T	T	X
[8,6]	5	1	T	T	T	T	X
[8,7]	3	1	T	T	T	T	X

Keterangan :

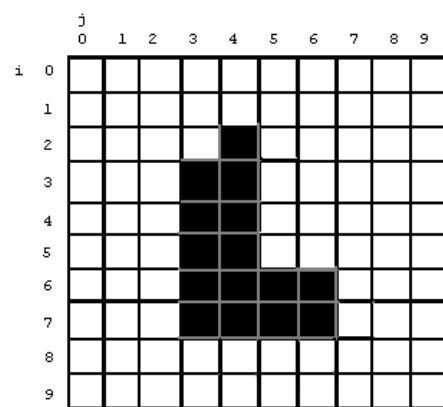
T : Terpenuhi

F : Tidak terpenuhi

X : Piksel [i,j] ditandai

- : Piksel [i,j] tidak ditandai

Tahap selanjutnya adalah menghapus semua piksel yang telah ditandai. Gambar 2.21 merupakan hasil dari proses *thinning* langkah ke-1.



Gambar 2.21 Hasil proses *thinning* langkah ke-1

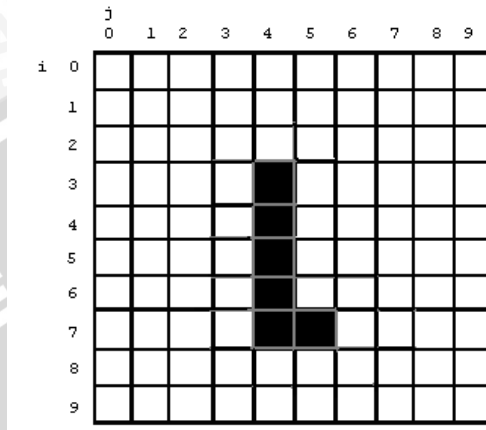
Kemudian proses *thinning* dilanjutkan pada langkah ke-2. Tabel 2.2 merupakan hasil pemeriksaan pada langkah ke-2 terhadap piksel yang memenuhi syarat.

Tabel 2.2 Hasil pemeriksaan pada langkah ke-2 terhadap piksel yang memenuhi syarat

Piksel [i,j]	$N(P_1)$	$S(P_1)$	$2 \leq N(P_1) \leq 6$	$S(P_1) = 1$	$p_2 \cdot p_4 \cdot p_8 = 0$	$p_2 \cdot p_6 \cdot p_8 = 0$	hasil
[2,4]	2	1	T	T	T	T	X
[3,3]	4	1	T	T	T	T	X
[3,4]	4	2	T	F	T	F	-
[4,3]	5	1	T	T	T	T	X
[4,4]	5	1	T	T	T	F	-
[5,3]	5	1	T	T	T	T	X
[5,4]	6	1	T	T	T	F	-
[6,3]	5	1	T	T	T	T	X
[6,4]	7	1	F	T	F	F	-
[6,5]	6	1	T	T	T	T	X
[6,6]	3	1	T	T	T	T	X
[7,3]	3	1	T	T	T	T	X
[7,4]	5	1	T	T	F	T	-
[7,5]	5	1	T	T	F	T	-
[7,6]	3	1	T	T	T	T	X

Keterangan :
 T : Terpenuhi
 F : Tidak terpenuhi
 X : Piksel $[i,j]$ ditandai
 - : Piksel $[i,j]$ tidak ditandai

Semua piksel yang telah ditandai pada proses langkah ke-2 dihapus. Gambar 2.22 merupakan hasil dari langkah ke-2 dan sekaligus hasil akhir dari proses *thinning*.



Gambar 2.22 Hasil proses *thinning* langkah ke-2

2.2.13 Minutiae Detection (Deteksi Minutiae)

Metode paling umum yang digunakan dalam pendeteksian poin *minutiae* adalah *Crossing Number* (CN). Metode ini melibatkan penggunaan citra skeleton dimana pola alir *ridge* merupakan *eight-connected*. Poin *minutiae* dideteksi dengan men-scan tetangga lokal pada masing-masing piksel *ridge* pada citra menggunakan ukuran *window* 3 x 3. Kemudian nilai CN dihitung, yang didefinisikan sebagai separuh penjumlahan dari perbedaan antara pasangan-pasangan piksel yang bersebelahan pada *eight-neighbourhood*. Dengan menggunakan properti CN yang ditunjukkan pada Tabel 2.3, maka piksel *ridge* dapat diklasifikasikan sebagai *ridge ending*, *bifurcation* atau poin non-*minutiae*. Sebagai contoh, piksel *ridge* dengan CN 1 sesuai dengan *ridge ending*, dan CN 3 sesuai dengan *bifurcation* (Thai, 2003).

Tabel 2.3 Properti *Crossing Number*

CN	Property
0	Isolated Point
1	Ridge ending point
2	Continuing ridge point
3	Bifurcation point
4	Crossing point

Nilai CN pada *ridge* piksel P didapat dari persamaan 2.16.

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}| \dots\dots\dots 2.16$$

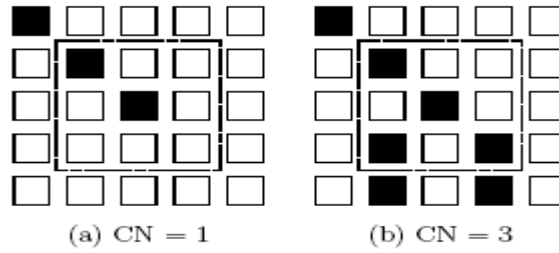
$$P_9 = P_1 \dots\dots\dots 2.17$$

dimana P_i merupakan nilai tetangga dari P . Untuk suatu piksel P , kedelapan piksel tetangganya diperiksa dengan arah berlawanan jarum jam seperti pada Gambar 2.23.

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Gambar 2.23 Pemberian nomor pada 8-tetangga piksel P untuk proses *minutiae detection*.

Setelah CN untuk piksel *ridge* dihitung, maka piksel dapat diklasifikasikan dengan mencocokkannya pada nilai properti CN yang ditunjukkan pada Tabel 2.3. Seperti yang ditunjukkan pada Gambar 2.24, piksel *ridge* dengan CN 1 sesuai dengan *ridge ending*, dan CN 3 sesuai dengan *bifurcation*.

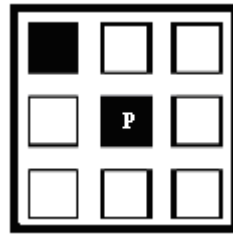


Gambar 2.24 Contoh piksel *ridge ending* dan piksel *bifurcation*.

(a) CN 1 sesuai dengan piksel *ridge ending*

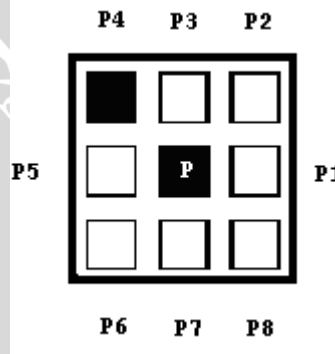
(b) CN 3 sesuai dengan piksel *bifurcation*

Berikut merupakan contoh dari proses deteksi *minutiae* dengan metode *Crossing Number*. Piksel yang akan diperiksa adalah piksel yang berwarna hitam atau bernilai 0. Gambar 2.25 merupakan contoh piksel P yang akan diperiksa.



Gambar 2.25 Contoh piksel P yang akan diperiksa

Kedelapan tetangga terdekat piksel P akan dinomori dengan P_1 hingga P_8 seperti yang terlihat pada Gambar 2.26.

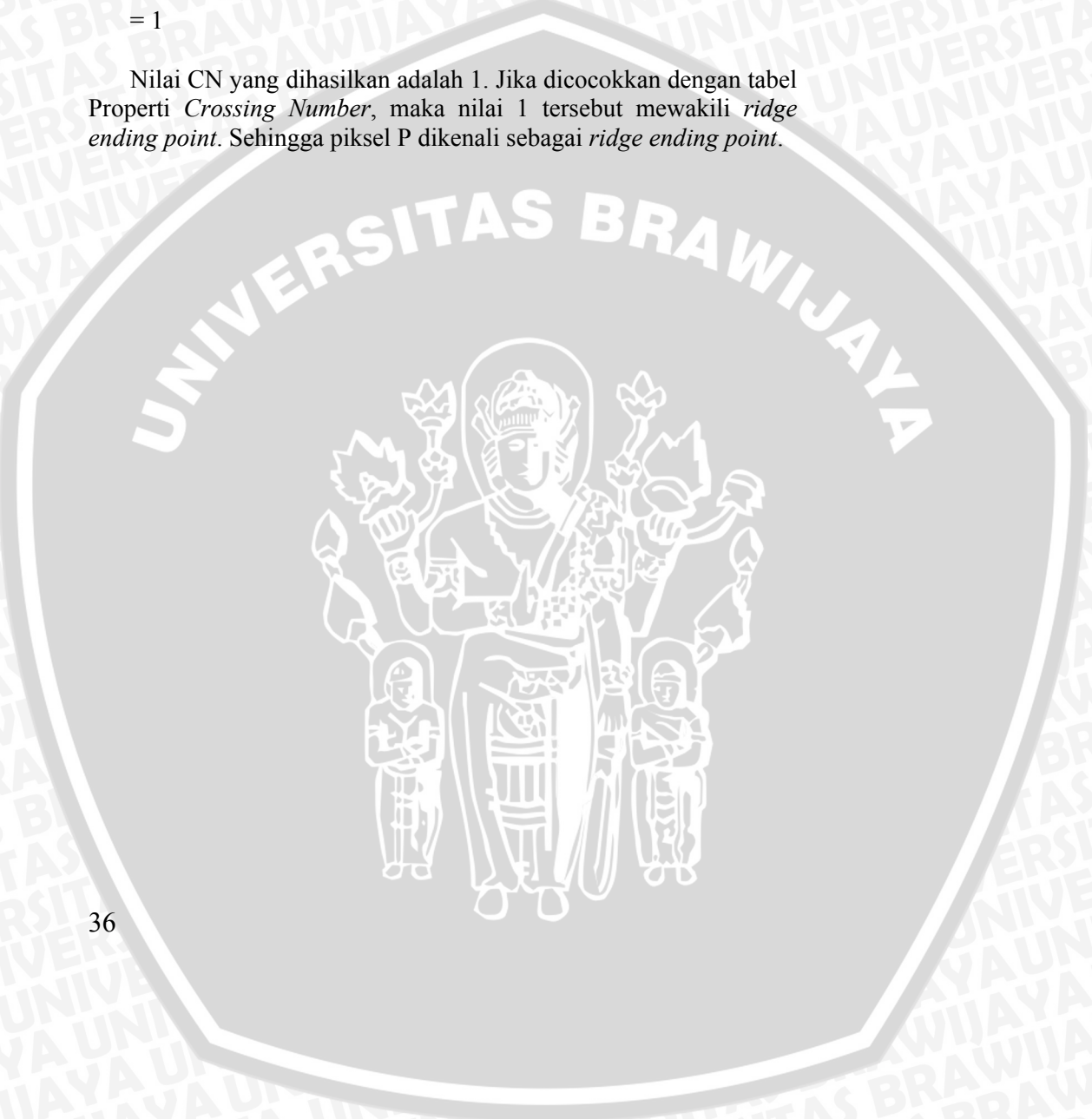


Gambar 2.26 Pemberian nomor pada 8-tetangga piksel P

Setelah kedelapan tetangga terdekat piksel P diberi nomor, maka CN atau *Crossing Number* dapat dihitung dengan persamaan 2.16. Berikut merupakan penyelesaiannya.

$$\begin{aligned}
 \text{CN} &= 0.5 \times (|P_1 - P_2| + |P_2 - P_3| + |P_3 - P_4| + |P_4 - P_5| + |P_5 - P_6| + \\
 &\quad |P_6 - P_7| + |P_7 - P_8| + |P_8 - P_9|) \\
 &= 0.5 \times (|1-1| + |1-1| + |1-0| + |0-1| + |1-1| + |1-1| + |1-1| + |1-1|) \\
 &= 0.5 \times (0 + 0 + 1 + 1 + 0 + 0 + 0 + 0) \\
 &= 0.5 \times 2 \\
 &= 1
 \end{aligned}$$

Nilai CN yang dihasilkan adalah 1. Jika dicocokkan dengan tabel Properti *Crossing Number*, maka nilai 1 tersebut mewakili *ridge ending point*. Sehingga piksel P dikenali sebagai *ridge ending point*.


 The logo of Universitas Brawijaya is a large, light gray shield shape. Inside the shield, the text "UNIVERSITAS BRAWIJAYA" is written in a bold, sans-serif font, arching over a central illustration. The illustration depicts a seated figure, likely a deity or a historical figure, with multiple arms holding various objects. Two smaller figures stand on either side of the central figure. The entire logo is centered on the page.

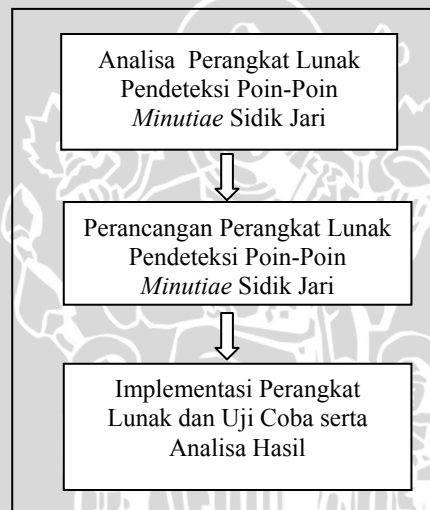
BAB III METODE DAN PERANCANGAN

Pada bab ini akan dibahas langkah-langkah pembuatan perangkat lunak pendeteksi poin-poin *minutiae* sidik jari serta metode-metode yang digunakan di dalamnya.

Tahapan pembuatannya adalah sebagai berikut:

1. Mempelajari metode yang digunakan dari jurnal yang membahas masalah yang sama.
2. Menganalisa dan merancang perangkat lunak dengan menggunakan penggabungan metode pada penelitian sebelumnya.
3. Membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
4. Uji coba perangkat lunak pada citra sidik jari hasil *scanning* secara *offline*.
5. Evaluasi terhadap hasil uji coba perangkat lunak untuk mengetahui tingkat keberhasilannya.

Langkah-langkah pembuatan perangkat lunak ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir pembuatan perangkat lunak

3.1 Analisis Perangkat Lunak

Pada subbab analisis akan dibahas berbagai hal yang diperlukan dalam proses pendeteksi poin-poin *minutiae* pada citra sidik jari.

3.1.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang akan dibangun merupakan suatu sistem yang dapat membantu *user* dalam melakukan pendeteksian poin-poin *minutiae* secara otomatis pada citra sidik jari. Maksud dari otomatis di sini adalah *user* hanya memasukkan citra sidik jari, kemudian proses selanjutnya dikerjakan sepenuhnya oleh sistem hingga ditemukan poin-poin *minutiae* pada citra sidik jari masukan.

Citra masukan pada sistem ini merupakan citra sidik jari hasil *scanning* secara *offline* dengan ukuran 192 x 192 piksel yang disimpan dalam ekstensi *.bmp*. Metode-metode yang digunakan dalam proses pendeteksian poin-poin *minutiae* ini sepenuhnya berbasis pada pengolahan citra digital.

Saat *user* masuk ke dalam perangkat lunak, maka proses yang terjadi adalah:

- *User* memilih citra sidik jari masukan yang telah tersedia di dalam komputer.
- Oleh perangkat lunak, citra masukan kemudian diubah ke bentuk *grayscale*.
- *Pre-processing*, pada proses ini citra *grayscale* akan ditingkatkan kualitasnya dengan menggunakan beberapa metode, yaitu *segmentation* (segmentasi), *normalization* (normalisasi), *image orientation estimation*, *ridge frequency estimation* dan *filtering* seperti yang telah dijelaskan pada bab 2.
- *Binarization* (binerisasi), citra yang telah ditingkatkan kualitasnya diubah ke bentuk biner.
- *Thinning* (penipisan), objek pada citra biner ditipiskan seperti yang telah dijelaskan pada subbab 2.2.11.
- *Minutiae Detection*, sistem mendeteksi poin-poin *minutiae* pada citra yang telah ditipiskan sebelumnya. Poin yang terdeteksi akan ditandai dengan warna merah dan dicatat jenis poinnya.

3.1.2 Batasan Perangkat Lunak

Batasan dari perangkat lunak yang akan dikembangkan adalah:

1. Citra masukan merupakan citra sidik jari hasil *scanning* secara *offline*.

2. Perangkat lunak hanya dapat memproses citra dengan ekstensi *.bmp*.
3. Ukuran citra 192 x 192 piksel.
4. Poin-poin *minutiae* yang berhasil dideteksi oleh sistem akan ditandai dengan warna merah.
5. Jenis poin *minutiae* yang menjadi pusat penelitian adalah *ridge ending*, *ridge bifurcation*, dan *ridge crossing*.

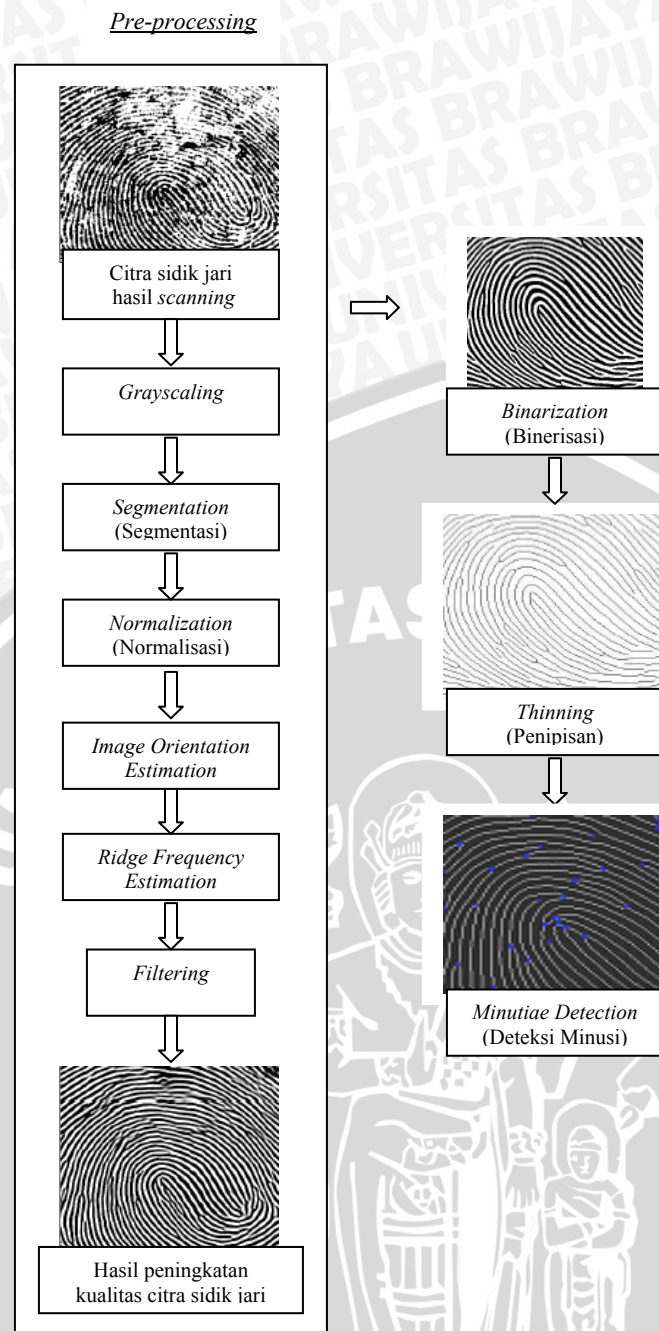
3.1.3 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan ini didasarkan pada pembangunan sistem pendeteksi poin-poin *minutiae* yang baik serta dapat memudahkan *user* dalam mengakses sistem tersebut nantinya. Kebutuhan perangkat lunak meliputi hal-hal penting yang harus dipenuhi, yaitu:

- Sistem harus dapat meningkatkan kualitas citra sekalipun citra dalam kondisi terburuk.
- Sistem harus dapat mendeteksi poin-poin *minutiae* pada citra yang telah ditingkatkan kualitasnya dengan baik sesuai dengan algoritma yang telah ditentukan.
- *Interface* sistem yang dibangun harus bersifat *user friendly* atau dapat dioperasikan dengan mudah oleh *user*.

Diagram alir proses umum sistem dapat dilihat pada Gambar 3.2.





Gambar 3.2 Diagram alir proses umum sistem

3.2 Perancangan Perangkat Lunak

Berdasarkan analisa yang telah dilakukan, maka tahap selanjutnya adalah merancang pembangunan perangkat lunak. Pada subbab ini akan dibahas proses-proses yang terjadi di dalam perangkat lunak, dimana proses-proses tersebut memiliki keterkaitan penting antara satu dengan yang lainnya.

3.2.1 Perancangan Proses

Secara garis besar proses pendeteksian poin-poin *minutiae* pada citra sidik jari dibagi menjadi dua proses penting. Proses yang pertama adalah *Pre-processing*, yang merupakan proses peningkatan kualitas citra sidik jari hasil *scanning* secara *offline*. Sedangkan proses yang kedua adalah *Minutiae Detection* untuk mengekstrak atau mendeteksi poin *minutiae* yang ada pada citra sidik jari. Untuk mempermudah proses pendeteksian *minutiae*, sebelumnya pada citra yang telah ditingkatkan kualitasnya akan dikenakan proses binerisasi dan proses *thinning* atau proses penipisan.

Poin-poin *minutiae* yang terdeteksi oleh sistem akan diperiksa apakah poin tersebut merupakan *ridge ending point*, *ridge bifurcation point*, atau *crossing point*. Poin-poin yang telah terdeteksi akan diberi tanda merah.

3.2.1.1 Perancangan Proses Masukan (*Input*)

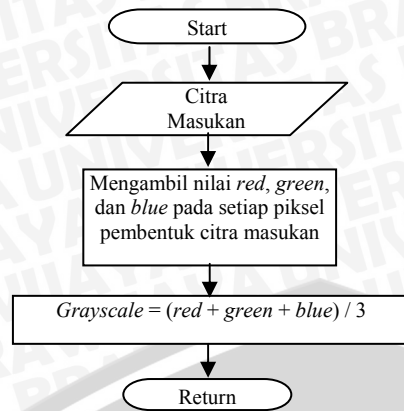
Seperti yang telah disebutkan sebelumnya, citra masukan pada sistem ini harus berupa *file bitmap (.bmp)* dengan ukuran 192 x 192 piksel.

Input

Citra yang menjadi masukan dapat berupa citra *grayscale* atau citra *RGB (true color)*. Namun setiap citra akan dianggap berupa *RGB*, dimana pada awalnya tiap *pixel* mewakili intensitas warna *red, green, blue*.

Grayscale

Citra masukan diubah menjadi citra *grayscale* dengan cara mengambil nilai *red*, *green*, dan *blue* pada setiap piksel pembentuk citra. Ketiga nilai tersebut dijumlahkan dan hasilnya dibagi 3. Hasil dari pembagian tersebutlah yang akan menjadi nilai *greyscale* pada setiap piksel yang dihitung.



Gambar 3.3 Diagram alir proses *grayscale*

3.2.1.2 Perancangan Proses Peningkatan Kualitas Citra (*Pre-processing*)

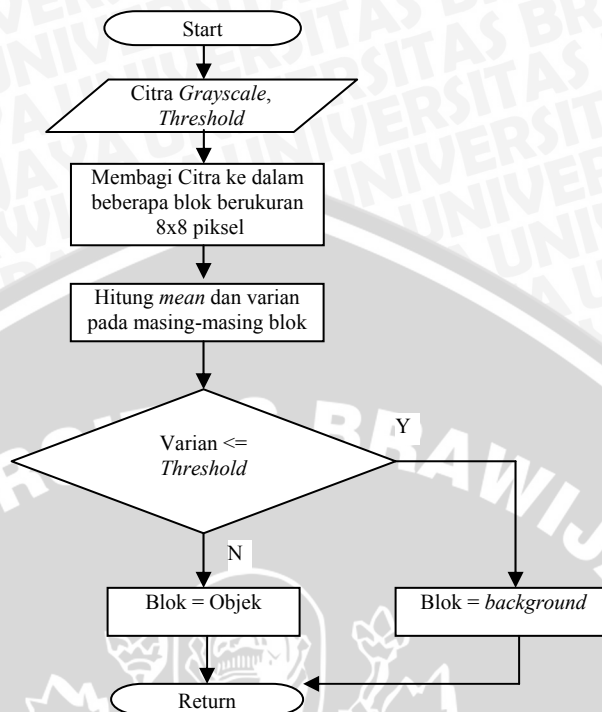
Citra *grayscale* yang didapat dari proses sebelumnya akan ditingkatkan kualitasnya dengan menggunakan gabungan beberapa metode, yaitu *segmentation* (segmentasi), *normalization* (normalisasi), *image orientation estimation*, *ridge frequency estimation* dan *filtering*.

Segmentation (segmentasi)

Untuk mensegmentasi citra digunakan metode yang berbasis pada pengambungan varian (*variance thresholding*) seperti yang telah dijelaskan pada subbab 2.2.5. Hasil dari proses segmentasi ini berupa citra yang telah terdeteksi objek dan *background*-nya. Objek adalah area sidik jari pada citra dan *background* adalah daerah di luar area sidik jari. *Background* yang terdeteksi diberi warna hitam atau bernilai 0. Sedangkan objek yang terdeteksi tetap pada warna semula. Berikut merupakan langkah-langkah segmentasi.

1. Memasukkan citra *grayscale*.
2. Menentukan nilai *threshold*.
3. Membagi citra ke dalam beberapa blok $k = k_1, k_2, k_3, \dots, k_i$ dengan ukuran 8 x 8 piksel.
4. Menghitung *mean* dan varian pada tiap-tiap blok.

- Memeriksa apakah nilai varian pada tiap-tiap blok kurang dari sama dengan *threshold* atau sebaliknya.

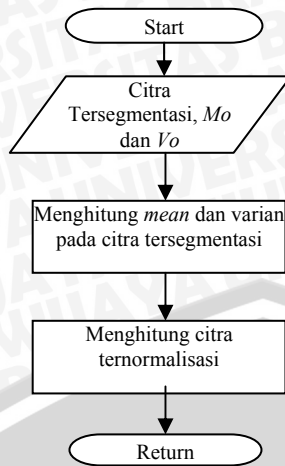


Gambar 3.4 Diagram alir proses segmentasi

Normalization (normalisasi)

Proses selanjutnya untuk meningkatkan kualitas citra adalah normalisasi. Persamaan yang digunakan untuk menormalisasi citra dapat dilihat pada subbab 2.2.6. Citra ternormalisasi $N(i,j)$ akan disimpan dalam *array* dua dimensi dan digunakan dalam proses *image orientation estimation*, *ridge frequency estimation*, dan *filtering*. Proses normalisasi memiliki langkah-langkah sebagai berikut:

- Memasukkan citra hasil proses segmentasi $S(i,j)$.
- Menentukan nilai *mean* yang diharapkan M_o dan nilai varian yang diharapkan V_o .
- Menghitung nilai *mean* dan varian pada citra hasil segmentasi.
- Menghitung citra ternormalisasi $N(i,j)$.



Gambar 3.5 Diagram alir proses normalisasi

Gaussian Smoothing

Gaussian smoothing digunakan untuk menghaluskan citra ternormalisasi. Penghalusan ini berfungsi untuk mendapatkan hasil *local orientation* dan *ridge frequency* yang lebih akurat. Langkah-langkahnya adalah sebagai berikut:

1. Memasukkan standar deviasi *gaussian* $\sigma = 1$ dan *mask gaussian* $W \times W = 9 \times 9$.
2. Menghitung *Gaussian Low Pass Filter* seperti yang telah dijelaskan pada subbab 2.2.9.

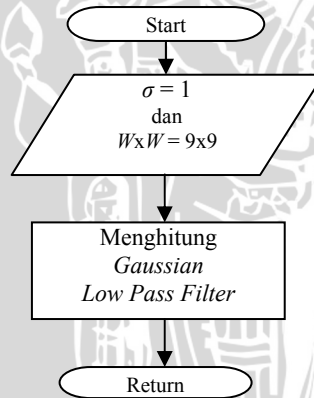
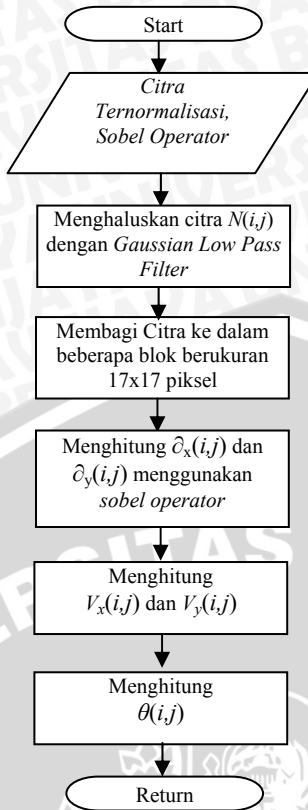
Gambar 3.6 Diagram alir proses *gaussian smoothing*

Image Orientation Estimation

Orientation estimation merupakan langkah penting dalam melakukan peningkatan citra dimana hasil dari *Orientation estimation* akan dijadikan parameter pada tahap pemfilteran selanjutnya. Proses ini menggunakan beberapa perhitungan yang telah dijelaskan pada subbab 2.2.7. Hasil *Orientation estimation* yang berupa derajat kemiringan dari alur *ridge* $\theta(i,j)$ akan disimpan dalam bentuk *array* dua dimensi. Langkah-langkah dari proses ini adalah:

1. Memasukkan citra ternormalisasi $N(i,j)$ dan *sobel operator*.
2. Menghaluskan citra $N(i,j)$ dengan menggunakan *Gaussian Low Pass Filter* seperti yang telah dijelaskan pada subbab 2.2.9.
3. Membagi citra ke dalam beberapa blok $k = k_1, k_2, k_3, \dots, k_i$ dengan ukuran 17x17 piksel.
4. Menghitung gradien $\partial_x(i,j)$ dan $\partial_y(i,j)$ pada tiap piksel di masing-masing blok dengan menggunakan *sobel operator*.
5. Menghitung $V_x(i,j)$ dan $V_y(i,j)$.
6. Menghitung *Local Orientation* $\theta(i,j)$ pada piksel (i,j) .





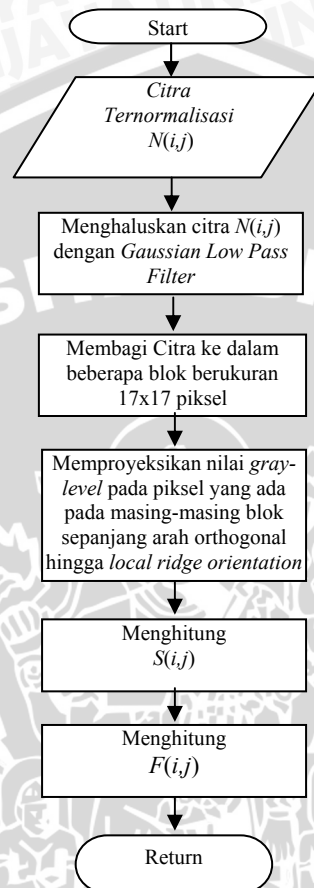
Gambar 3.7 Diagram alir proses *image orientation estimation*

Ridge Frequency Estimation

Frequency image merepresentasikan frekuensi lokal dari *ridge* yang ada pada sidik jari. Proses ini menggunakan beberapa perhitungan yang telah dijelaskan pada subbab 2.2.8. Hasil *Frequency Estimation* $F(i,j)$ akan disimpan dalam bentuk *array* dua dimensi. Langkah-langkah dari proses ini adalah:

1. Memasukkan citra ternormalisasi $N(i,j)$.
2. Menghaluskan citra $N(i,j)$ dengan menggunakan *Gaussian Low Pass Filter* seperti yang telah dijelaskan pada subbab 2.2.9.
3. Membagi citra ke dalam beberapa blok $k = k_1, k_2, k_3, \dots, k_i$ dengan ukuran 17x17 piksel.

4. Memproyeksikan nilai *grey-level* dari piksel-piksel pada masing-masing blok sepanjang arah yang tegak lurus dengan arah *local ridge orientation*.
5. Menghitung jarak $S(i,j)$ dengan menghitung nilai median piksel diantara poin-poin minimum yang berurutan pada gelombang proyeksi.
6. Menghitung *Frequency image* $F(i,j)$.

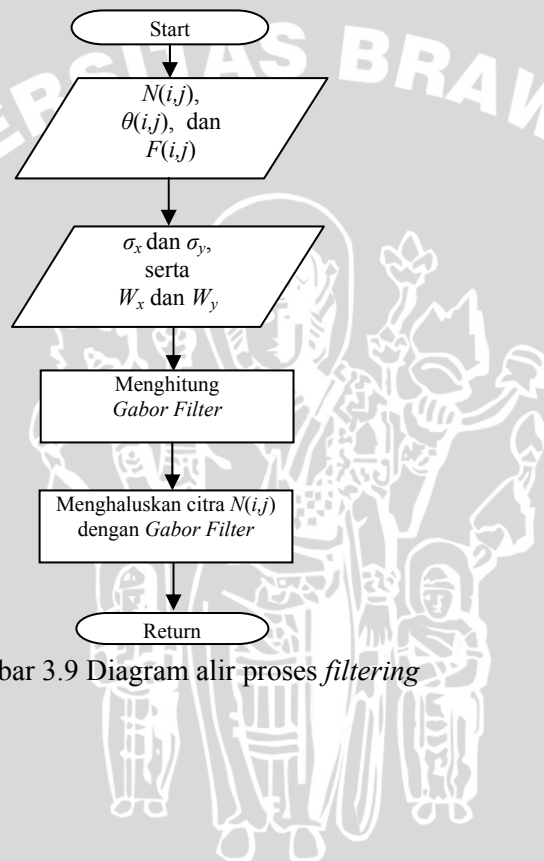


Gambar 3.8 Diagram Alir proses *ridge frequency estimation*

Filtering

Untuk melakukan proses *filtering* digunakan fungsi *Gabor Filter* seperti yang telah dijelaskan pada subbab 2.2.9. Fungsi *Gabor Filter* menggunakan hasil dari *Image Orientation Estimation* dan *Ridge Frequency Estimation* sebagai parameternya. Langkah-langkah dari proses ini adalah:

1. Memasukkan citra ternormalisasi $N(i,j)$, *local orientation* $\theta(i,j)$, *ridge frequency* $F(i,j)$.
2. Menentukan standar deviasi *gaussian* σ_x dan σ_y serta tinggi dan lebar *gabor mask* W_x dan W_y .
3. Menghitung *Gabor Filter* seperti yang ditunjukkan pada persamaan 2.10.
4. Menghaluskan citra ternormalisasi dengan menggunakan *Gabor Filter* seperti yang ditunjukkan pada persamaan 2.11.

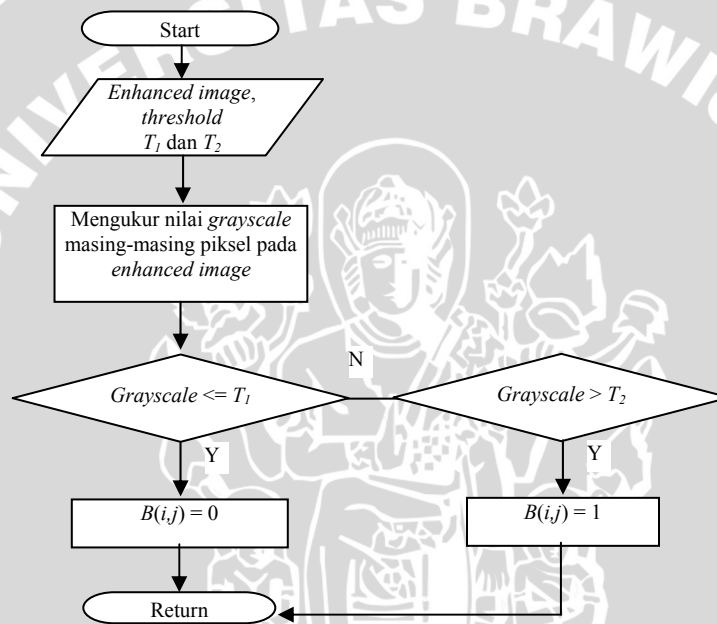


Gambar 3.9 Diagram alir proses *filtering*

3.2.1.3 Perancangan Proses *Binarization* (Binerisasi)

Untuk mempermudah proses pendeteksian poin-poin *minutiae*, maka citra yang telah ditingkatkan kualitasnya dikonversi ke bentuk citra biner $B(i,j)$, dimana di dalamnya hanya terdapat dua informasi nilai atau warna, yaitu 1 (warna putih) dan 0 (warna hitam). Warna hitam untuk merepresentasikan *ridge* dan warna putih untuk merepresentasikan *valley*. Berikut merupakan langkah-langkah dari proses binerisasi.

1. Memasukkan citra yang telah ditingkatkan kualitasnya (*enhanced image*) $E(i,j)$.
2. Menentukan nilai *threshold* T_1 dan T_2 .
3. Memanggil fungsi *grayscale* untuk menghitung nilai *grayscale* masing-masing piksel pada *enhanced image*.
4. Memeriksa apakah nilai *grayscale* masing-masing piksel pada *enhanced image* kurang dari sama dengan *threshold* atau sebaliknya.



Gambar 3.10 Diagram alir proses binerisasi

3.2.1.4 Perancangan Proses *Thinning* (Penipisan)

Thining merupakan proses analisis yang secara berturut-turut mengikis piksel-piksel hingga menjadi satu piksel saja. Aplikasi algoritma *thinning* citra sidik jari mempertahankan konektifitas struktur *ridge* saat membentuk versi *skeleton* citra biner. Citra *skeleton* ini kemudian digunakan dalam proses pendeteksian *minutiae*. Pada proses ini diasumsikan bahwa pola *ridge* memiliki nilai 1 dan pola *valley* dan *background* bernilai 0. Langkah-langkah proses *thinning* ini adalah sebagai berikut:

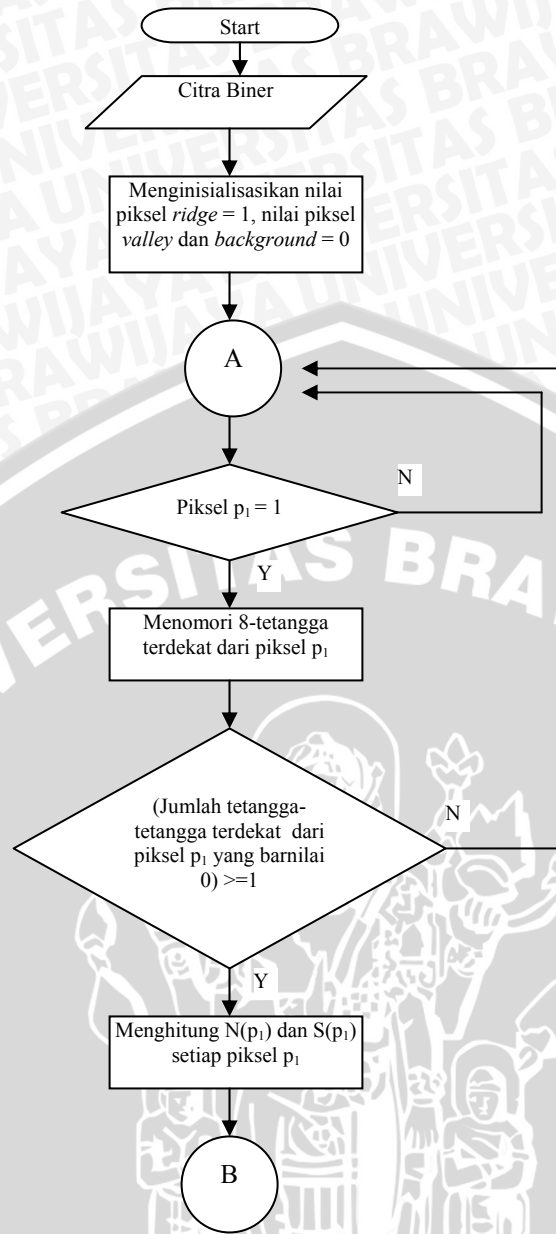
Langkah 1:

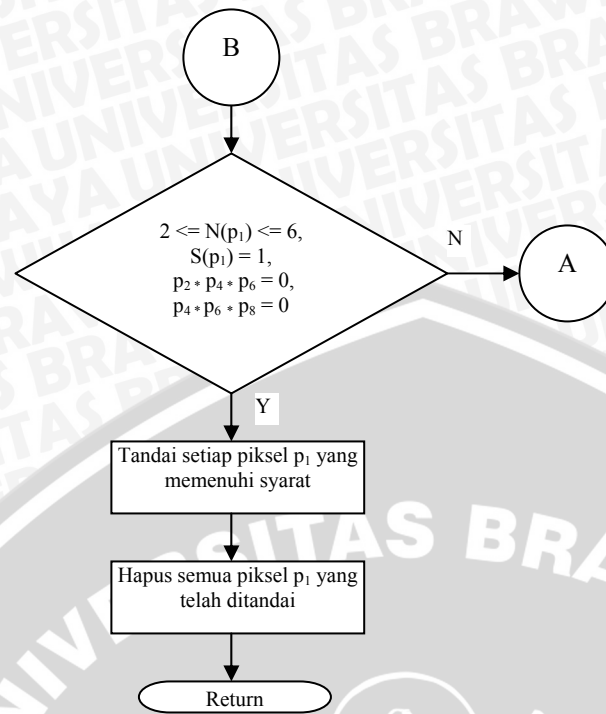
1. Memasukkan citra biner $B(i,j)$.
2. Menginisialisasikan nilai piksel *ridge* = 1, nilai piksel *valley* dan *background* = 0.
3. Memeriksa apakah tiap piksel p_1 bernilai 1.
4. 8-tetangga terdekat dari setiap piksel p_1 dinomori p_2 (untuk piksel di atas p_1) sampai dengan p_9 sesuai dengan arah jarum jam.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Gambar 3.11 Pemberian nomor pada 8-tetangga piksel p_1 untuk proses *thinning*

5. Memeriksa apakah tetangga-tetangga (p_2 - p_9) dari piksel p_1 yang bernilai 0 lebih dari sama dengan 1.
6. Menghitung $N(p_1)$, dimana $N(p_1)$ adalah jumlah tetangga-tetangga dari p_1 yang bukan nol; sehingga, $N(p_1) = (p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9)$.
7. Menghitung $S(p_1)$, dimana $S(p_1)$ adalah jumlah dari transisi 0 ke 1 dalam urutan $p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$.
8. Memeriksa apakah $N(p_1)$ lebih dari sama dengan 2 dan kurang dari sama dengan 6, $S(p_1) = 1$, $p_2 * p_4 * p_6 = 0$, dan $p_4 * p_6 * p_8 = 0$.
9. Memberi tanda setiap piksel p_1 yang memenuhi syarat di atas.
10. Menghapus setiap piksel p_1 yang telah ditandai.

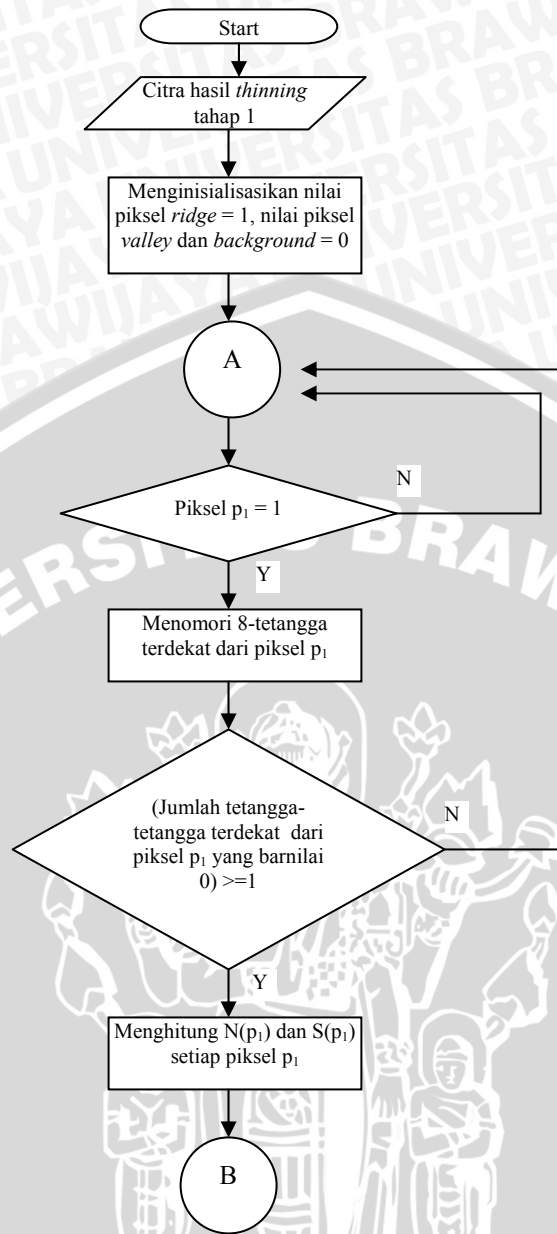


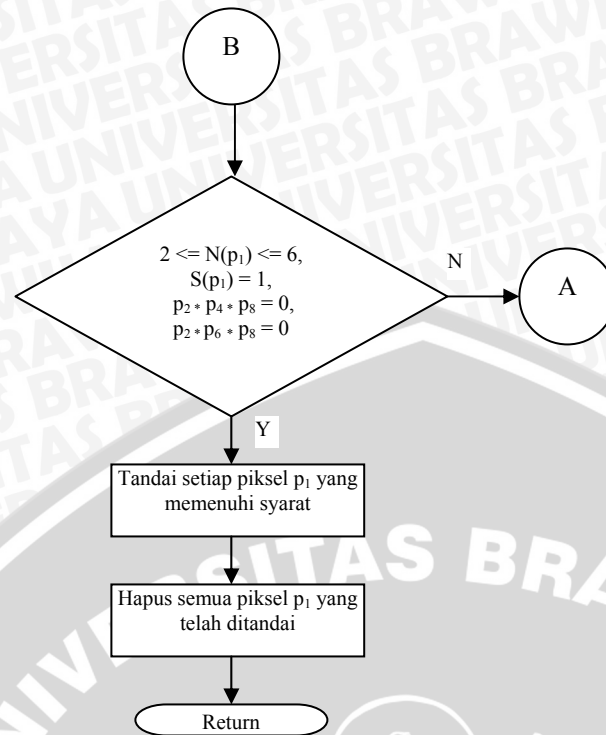


Gambar 3.12 Diagram alir proses *thinning* langkah 1

Langkah 2:

1. Memasukkan citra hasil *thinning* langkah 1.
2. Menginisialisasikan nilai piksel *ridge* = 1, nilai piksel *valley* dan *background* = 0.
3. Memeriksa apakah tiap piksel p_1 bernilai 1.
4. Menomori 8-tetangga terdekat dari setiap piksel p_1 seperti pada langkah 1.
5. Memeriksa apakah tetangga-tetangga (p_2 - p_9) dari piksel p_1 yang bernilai 0 lebih dari sama dengan 1.
6. Menghitung $N(p_1)$ dan $S(p_1)$ seperti pada langkah 1.
7. Memeriksa apakah $N(p_1)$ lebih dari sama dengan 2 dan kurang dari sama dengan 6, $S(p_1) = 1$, $p_2 \cdot p_4 \cdot p_8 = 0$, dan $p_2 \cdot p_6 \cdot p_8 = 0$.
8. Memberi tanda setiap piksel p_1 yang memenuhi syarat di atas.
9. Menghapus setiap piksel p_1 yang telah ditandai.



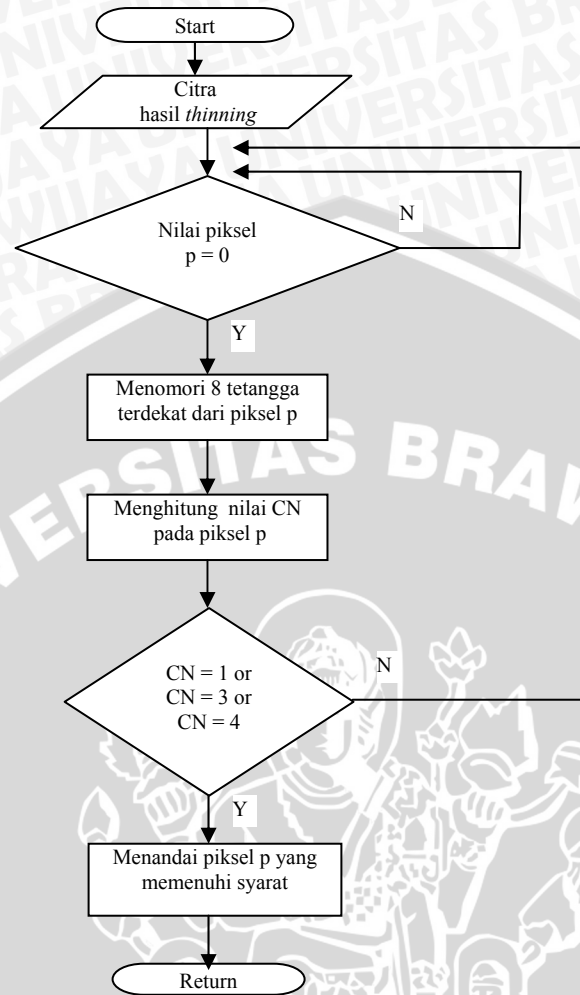


Gambar 3.13 Diagram alir proses *thinning* langkah 2

3.2.1.5 Perancangan Proses *Minutiae Detection*

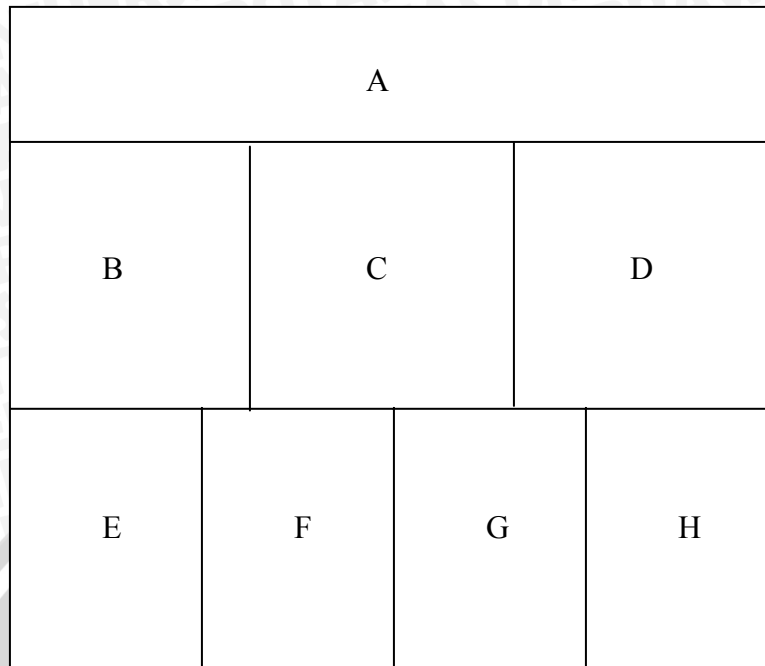
Metode yang digunakan dalam proses pendeteksian poin *minutiae* ini adalah *Crossing Number* (CN) seperti yang telah dijelaskan pada subbab 2.2.12. Metode ini melibatkan penggunaan citra skeleton (citra hasil proses *thinning*) dimana pola alur *ridge* merupakan *eight-connected*. Berikut merupakan langkah-langkah dari proses deteksi minusi.

1. Memasukkan citra hasil *thinning*
2. Memeriksa apakah piksel p bernilai 0
3. Menomori 8-tetangga terdekat dari piksel p
4. Menghitung nilai CN piksel p
5. Memeriksa apakah nilai CN = 1 atau 3 atau 4
6. Menandai piksel p yang memiliki nilai CN = 1 atau 3 atau 4



Gambar 3.14 Diagram Alir proses deteksi *minutiae*

3.2.2 Perancangan *Interface*



Gambar 3.15 Rancangan *form* utama

Rancangan *form* utama dibagi menjadi tiga bagian, yaitu:

- A : Judul atau nama aplikasi.
- B : Menampilkan citra asli yang akan di proses.
- C : Menampilkan citra yang telah ditingkatkan kualitasnya.
- D : Menampilkan citra hasil proses pendeteksian *minutiae*.
- E : Menampilkan citra hasil proses *Orientation Estimatisation*.
- F : Menampilkan citra hasil proses *Frequency Estimatisation*.
- G : Menampilkan citra hasil proses Binerisasi.
- H : Menampilkan citra hasil proses *Thinning*.

3.3 Perancangan Uji Coba

Poin-poin *minutiae* yang merupakan hasil pendeteksian sistem akan dikenakan pengujian secara sederhana. Ada tiga jenis uji coba pada tahap pengujian tersebut, yaitu :

1. Uji coba diterapkan pada sepuluh sidik jari orang yang berbeda. Masing-masing orang tersebut akan diambil lima *sample* sidik jarinya dengan posisi yang berbeda-beda sesuai dengan kriteria yang telah disebutkan pada subbab 1.3. Citra sidik jari yang menjadi *sample* tersebut didownload dari internet dengan alamat www.biometrix.at. Poin-poin *minutiae* yang menjadi pusat uji coba ini adalah *ridge ending*, *ridge bifurcation* dan *ridge crossing*. Jadi, hasil pendeteksian *ridge ending*, *ridge bifurcation* dan *ridge crossing* dari kelima *sample* masing-masing orang akan dibuatkan tabel seperti yang terlihat pada Tabel 3.1. Dari tabel tersebut akan dilihat kestabilan jumlah poin-poin *minutiae* yang berhasil dideteksi. Kemudian poin-poin *minutiae* pada table tersebut akan dihitung jumlah variannya dengan menggunakan *software* SPSS 13.0.

Tabel 3.1 Rancangan tabel hasil pendeteksian poin-poin *minutiae* masing-masing orang

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1			
2			
3			
4			
5			

2. Secara keseluruhan, konsep uji coba kedua hampir sama dengan konsep uji coba pertama. Perbedaannya hanya terletak pada sidik jari yang menjadi objek uji coba. Pada uji coba ini, digunakan lima sidik jari orang yang berbeda sebagai objeknya. Tiap orang tersebut akan diambil lima *sample* sidik jarinya dengan pola pengambilan yang hampir sama.
3. Uji coba ketiga dilakukan dengan membandingkan hasil pendeteksian poin-poin *minutiae* dari kelima belas orang yang diambil *sample* sidik jarinya. Contohnya, hasil pendeteksian *ridge ending* dari kelima belas orang tersebut akan dibuatkan tabel seperti yang terlihat pada Tabel 3.2. Dari table tersebut akan diperiksa apakah jumlah poin-poin *minutiae* hasil pendeteksian sistem pada setiap orang berbeda atau tidak. Metode yang digunakan pada uji coba ini adalah *Analysis of Variance* (Analisa Varian), dimana metode tersebut dijalankan

pada program aplikasi SPSS 13.0. Perlakuan yang sama juga diterapkan pada hasil pendeteksian *ridge bifurcatin* dan *ridge crossing* dari kelima belas orang tersebut.

Tabel 3.2 Rancangan tabel perbandingan hasil pendeteksian *ridge ending* pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1					
2					
3					
⋮					
⋮					
15					

Tabel 3.3 Rancangan tabel perbandingan hasil pendeteksian *ridge bifurcation* pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1					
2					
3					
⋮					
⋮					
15					

Tabel 3.4 Rancangan tabel perbandingan hasil pendeteksian *ridge crossing* pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1					
2					
3					
⋮					
⋮					
15					

Karena yang menjadi pusat perhatian pada pengujian ini adalah orangnya, maka digunakan hipotesa sebagai berikut :

H_0 : Orang ke-1 = Orang ke-2 = Orang ke-3 = ... = Orang ke-15.

H_1 : Paling sedikit ada satu pasang orang yang tidak sama.

BAB IV HASIL DAN PEMBAHASAN

Implementasi merupakan proses transformasi representasi rancangan ke bahasa pemrograman yang dapat dimengerti oleh komputer. Bab ini akan membahas hasil implementasi yang dihasilkan oleh perangkat lunak, untuk selanjutnya dilakukan evaluasi terhadap hasil pendeteksian poin-poin *minutiae* oleh sistem.

4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras serta lingkungan perangkat lunak

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pendeteksi poin-poin *minutiae* ini adalah:

1. Intel(R) Pentium(R) M – Processor 1,73GHz.
2. RAM 758 MB
3. *Harddisk* dengan kapasitas 40 GB
4. Monitor 14 inchi
5. Keyboard
6. Mouse

4.1.2 Lingkungan perangkat lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini adalah :

1. Sistem Operasi *Microsoft Windows XP Professional*
2. *Borland Delphi 7*

4.2 Implementasi Program

Berdasarkan perancangan perangkat lunak pada subbab 3.2 maka pada subbab ini akan dibahas mengenai implementasi dari perancangan tersebut.

4.2.1 Input

Proses *input* yang dilakukan oleh *user* dengan membuka file citra sidik jari yang telah tersimpan sebelumnya. *Sourcecode* 4.1 menunjukkan cara membuka file citra.

```
//prosedur untuk membuka file citra
procedure TForm1.Open1Click(Sender: TObject);
begin
  if (OpenPictureDialog1.Execute) then
  begin
    InputImage.Picture.LoadFromFile
      (OpenPictureDialog1.FileName);
    Image2.Picture.LoadFromFile
      (OpenPictureDialog1.FileName);
  end;
  Button1.Enabled := True;
end;
```

Sourcecode 4.1 Sourcecode cara membuka file citra

4.2.2 Segmentasi (*Segmentation*)

Awalnya citra masukan dibagi kedalam beberapa blok dengan ukuran 8 x 8 piksel.

```
//Membagi citra masukan kedalam blok-blok berukuran 8x8
piksel
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    Status[i+1, j+1] := (j div BlockHeight) *
      (ImageWidth div BlockWidth) +
      ((i div BlockWidth) + 1);
```

Sourcecode 4.2 Sourcecode proses pemblokkan atau *windowing*

Pada tiap-tiap blok dihitung nilai *mean* dan variannya. Karena perhitungan *mean* dan varian membutuhkan nilai *gray-level* dari masing-masing piksel di tiap blok, maka citra masukan harus diubah ke dalam bentuk citra *grayscale* terlebih dulu.

Nilai varian yang dihasilkan kemudian akan dibandingkan dengan nilai *threshold* yang telah ditentukan sebelumnya ($T = 100$) untuk menentukan apakah blok dianggap sebagai *background* atau sebagai objek. Nilai *threshold* tersebut ditentukan berdasarkan hasil percobaan. Hasil dari proses segmentasi dengan beberapa variasi nilai *threshold* dapat dilihat pada lampiran.

Mean dan varian disimpan dalam bentuk array satu dimensi yang berukuran sesuai dengan jumlah blok yang dihasilkan.


```

function TFormUtama.ColorToGrayscale(Color: TColor): TColor;
var
  GrayscaleColor: TColor;
  R, G, B, Avg : Integer;
begin
  R := GetRValue(Color);
  G := GetGValue(Color);
  B := GetBValue(Color);
  Avg := Round((R+G+B)/3);
  GrayscaleColor := RGB(Avg,Avg,Avg);

  Result := GrayscaleColor;
end;

```

Sourcecode 4.3 Sourcecode proses grayscale

```

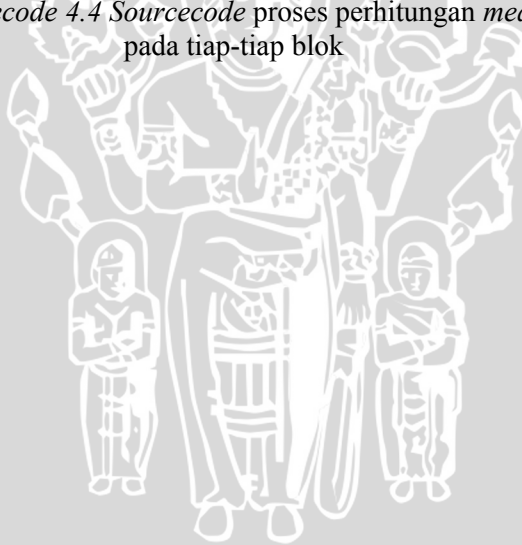
//Menghitung nilai mean pada tiap-tiap blok
for b:=1 to BlockCount do
begin
  jml := 0;

  for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    if Status[i+1, j+1] = b then
      //panggil fungsi ColorToGrayscale
      jml := jml + GetRValue(ColorToGrayscale
        (ImgInput.Canvas.Pixels[i,j]));

  Mean[b] := jml / (BlockWidth * BlockHeight);
end;

```

Sourcecode 4.4 Sourcecode proses perhitungan mean pada tiap-tiap blok



```
//Menghitung nilai varian pada tiap-tiap blok
for b:=1 to BlockCount do
begin
    jml := 0;

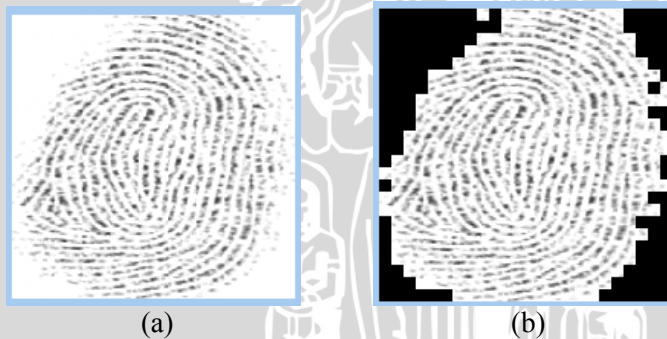
    for i:=0 to ImageHeight-1 do
    for j:=0 to ImageWidth-1 do
        if Status[i+1, j+1] = b then
            begin //panggil fungsi ColorToGrayscale
                tmp := GetRValue(ColorToGrayscale
                    (ImgInput.Canvas.Pixels[i,j]));
                jml := jml + Sqr(tmp - Mean[b]);
            end;

    Variant[b] := jml / (BlockWidth * BlockHeight);

//Menentukan apakah blok merupakan background atau objek
if Variant[b] < ThresholdSeg then
begin
    for i:=0 to ImageHeight-1 do
    for j:=0 to ImageWidth-1 do
        if Status[i+1, j+1] = b then
            ImgInput.Canvas.Pixels[i,j] := clBlack;
    end;
end;
```

Sourcecode 4.5 Sourcecode proses perhitungan varian pada tiap-tiap blok dan proses penentuan background dan objek

Gambar 4.1 menunjukkan citra sidik jari hasil proses segmentasi dengan *threshol* (T) = 100.



(a) (b)
Gambar 4.1 Citra Sidik Jari Hasil Proses Segmentasi dengan Nilai $T=100$.
(a) Citra Asli (b) Citra Tersegmentasi

4.2.3 Normalisasi (*Normalization*)

Citra hasil segmentasi kemudian dinormalisasi. Proses normalisasi diawali dengan menghitung nilai *mean* dan varian pada citra tersegmentasi serta menentukan nilai *mean* yang diharapkan ($M_o = 100$) dan varian yang diharapkan ($V_o = 100$). Nilai tersebut ditentukan berdasarkan jurnal yang ditulis oleh Hong dan kawan-kawan. Setelah keempat parameter tersebut didapat, maka citra ternormalisasi dapat dihitung.

```
Mo := 100;
Vo := 100;

// Menghitung mean
jml := 0;
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    jml := jml + GetRValue(ColorToGrayscale
      (ImgInput.Canvas.Pixels[i,j]));
Mean := jml / (ImageWidth * ImageHeight);

// Menghitung varian
jml := 0;
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    jml := jml + Sqr(GetRValue(ColorToGrayscale
      (ImgInput.Canvas.Pixels[i,j])) - Mean);

Variant := jml / (ImageWidth * ImageHeight);

//Hitung citra ternormalisasi
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    begin
      tmp := GetRValue(ColorToGrayscale
        (ImgInput.Canvas.Pixels[i,j]));

      if tmp > Mean then
        Normalized := Round(Mo +
          Sqrt(Vo * Sqr(tmp - Mean) / Variant))
      else
        Normalized := Round(Mo -
          Sqrt(Vo * Sqr(tmp - Mean) / Variant));

      ImgInput.Canvas.Pixels[i,j] :=
        RGB(Normalized,Normalized,Normalized);
    end;
```

Sourcecode 4.6 Sourcecode proses perhitungan citra ternormalisasi

4.2.4 Image Orientation Estimation

Perhitungan orientasi citra ini dimaksudkan untuk mencari sudut dari pola *ridge* pada citra sidik jari. Awalnya citra ternormalisasi dihaluskan dengan metode *Gaussian smoothing*, kemudian citra tersebut dibagi kedalam blok berukuran 17 x 17 piksel. Masing-masing piksel pada tiap blok dihitung gradien x dan gradien y -nya. Pada tiap blok dihitung *local orientation* $\theta(i,j)$ yang dipusatkan pada piksel (i, j) . Sudut $\theta(i,j)$ yang dihasilkan akan disimpan di dalam *array* dua dimensi.

```
//proses gaussian smoothing
w := 9; thao := 1.0;
for i:=4 to ImageHeight-5 do
begin
  for j:=4 to ImageWidth-5 do
  begin
    jml := 0; p := -4;
    for x:=w-1 downto 0 do
    begin
      q := -4;
      for y:=w-1 downto 0 do
      begin
        imagex := (i - (w div 2) + x + ImageHeight) mod
          ImageHeight;
        imagey := (j - (w div 2) + Y + ImageWidth) mod
          ImageWidth;

        gaussian := (1 / (2 * Pi * sqr(thao))) *
          exp((sqr(p) + sqr(q)) / (2 * sqr(thao)) * -1);
        tmp := GetRValue(ColorToGrayscale
          (InputImage.Canvas.Pixels[imagex, imagey]));
        jml := jml + round(gaussian * tmp);
        inc(q);
      end;
      inc(p);
    end;
    GaussianSmooth[i+1, j+1] := jml;
  end;
end;
```

Sourcecode 4.7 Sourcecode proses perhitungan
gaussian smoothing

```

FilterX: array[1..3,1..3] of Integer =
    ((1, 0, -1), (2, 0, -2), (1, 0, -1));
FilterY: array[1..3,1..3] of Integer =
    ((1, 2, 1), (0, 0, 0), (-1, -2, -1));

// Menghitung gradien x dan y

FilterWidth  := 3;
FilterHeight := 3;

for i:=0 to ImageHeight-1 do
for j:=0 to ImageWidth-1 do
begin
    if (i = 0) or (i = ImageHeight-1) or (j = 0) or
       (j = ImageWidth-1) then
    begin
        DoX[i+1, j+1] := 0;
        DoY[i+1, j+1] := 0;
    end
    else
    begin
        jum1:=0; jum2:=0;
        for x:=0 to FilterWidth-1 do
        for y:=0 to FilterHeight-1 do
        begin
            imagex := (i - (FilterHeight div 2) + x +
                        ImageHeight) mod ImageHeight;
            imagey := (j - (FilterWidth div 2) + Y +
                        ImageWidth) mod ImageWidth;

            jum1 := jum1 + GaussianSmooth[imagex+1, imagey+1]
                        * FilterX[x+1,y+1];
            jum2 := jum2 + GaussianSmooth[imagex+1, imagey+1]
                        * FilterY[x+1,y+1];
        end;
        DoX[i+1, j+1] := jum1; // Gradien x
        DoY[i+1, j+1] := jum2; // Gradien y
    end;
end;
end;

```

Sourcecode 4.8 Sourcecode proses perhitungan gradien x dan gradien y

```

// Menghitung sudut Tetha[i,j]
w := 17; // tinggi dan lebar blok
i := (w div 2); j := (w div 2);
while (i <= 186 - (w div 2)) do
begin
  jum1:=0; jum2:=0; jum3:=0;
  for x:=0 to w-1 do
    for y:=0 to w-1 do
      begin
        imagex := (i - (w div 2) + x + ImageHeight) mod
          ImageHeight;
        imagey := (j - (w div 2) + Y + ImageWidth) mod
          ImageWidth;

        jum1 := jum1 + (2 * DoX[imagex+1,imagey+1] *
          DoY[imagex+1,imagey+1]);
        jum2 := jum2 + (Sqr(DoX[imagex+1,imagey+1]) *
          Sqr(DoY[imagex+1,imagey+1]));
      end;

    if (jum1 = 0) and (jum2 = 0) then
    begin
      jum := degtorad(0);
    end
    else if (jum1 = 0) and (jum2 <> 0) then
    begin
      jum := degtorad(90);
    end
    else if (jum1 <> 0) then
    begin
      jum := ((1/2) * ArcTan(jum2 / jum1));
    end;

  // Simpan sudut tetha ke dalam array dua dimensi
  for x:=0 to w-1 do
    for y:=0 to w-1 do
      begin
        imagex := (i - (w div 2) + x + ImageHeight) mod
          ImageHeight;
        imagey := (j - (w div 2) + Y + ImageWidth) mod
          ImageWidth;

        Tetha[imagex+1, imagey+1] := radtodeg(jum);
      end;

    if (j = 186 - ((w div 2))) then
    begin
      j := (w div 2);
      i := i + w;
    end
    else
    begin
      j := j + w;
    end;
  end;
end;

```

*Sourcecode 4.9 Sourcecode proses perhitungan
local orientation $\theta(i,j)$ pada tiap blok*

4.2.5 Ridge Frequency Estimation

Perhitungan *ridge frequency* ini dimaksudkan untuk mencari frekuensi *ridge* pada citra sidik jari. Awalnya citra ternormalisasi dihaluskan dengan metode *Gaussian smoothing*. Selanjutnya, citra hasil *smoothing* dibagi kedalam blok berukuran 17 x 17 piksel. Pada masing-masing blok digambar garis yang tegak lurus dengan arah *ridge* (kebalikan dari sudut *local orientation*). Setiap piksel yang dilalui garis tersebut akan diambil nilai *gray-level*-nya. Dari nilai *gray-level* yang didapat akan dicari nilai-nilai terendahnya (point minimum). Kemudian jarak antar poin-poin minimum tersebut dijumlahkan. $S(i,j)$ merupakan rata-rata dari jumlah jarak tersebut dan Frekuensi $F(i,j)$ merupakan hasil perhitungan 1 dibagi $S(i,j)$. Kedua hasil perhitungan tersebut masing-masing disimpan di dalam *array* dua dimensi.

```
//prosedur untuk mencari titik x1,y1 dan x2,y2 untuk membuat garis
orthogonal
procedure
  TForm1.CariKoordinatGarisLintas(xc,yc,blockHeight,blockWidth:
  Integer; var x1,y1,x2,y2: Integer; var sudut:Real);
var sudutKuadran: Real;
    batasSudut: Real;
begin
  batasSudut := RadToDeg(arcTan(blockHeight/blockWidth));

  if (sudut < 180) and (sudut >= 180-batasSudut) then
  begin
    sudutKuadran := 180-sudut;
    x1 := xc - Round(blockWidth/2);
    y1 := yc - Round(Round(blockWidth/2)
      * Tan(DegToRad(sudutKuadran)));
    x2 := xc + Round(blockWidth/2);
    y2 := yc + Abs(yc-y1);
  end
  else if (sudut < 180-batasSudut) and (sudut >= 90) then
  begin
    sudutKuadran := sudut-90;
    y1 := yc - Round(blockHeight/2);
    x1 := xc - Round(Round(blockHeight/2)
      * Tan(DegToRad(sudutKuadran)));
    y2 := yc + Round(blockHeight/2);
    x2 := xc + Abs(xc-x1);
  end
  else if (sudut < 90) and (sudut >= batasSudut) then
  begin
    sudutKuadran := 90-sudut;
    y1 := yc - Round(blockHeight/2);
    x1 := xc + Round(Round(blockHeight/2)
      * Tan(DegToRad(sudutKuadran)));
    y2 := yc + Round(blockHeight/2);
    x2 := xc - Abs(xc-x1);
  end
end
```

```
else if (sudut < batasSudut) and (sudut >= 0) then
begin
    sudutKuadran := sudut;
    x1 := xc + Round(blockWidth/2);
    y1 := yc - Round(Round(blockWidth/2)
        * Tan(DegToRad(sudutKuadran)));
    x2 := xc - Round(blockWidth/2);
    y2 := yc + Abs(yc-y1);
end
else if (sudut < 0) and (sudut >= -batasSudut) then
begin
    sudutKuadran := Abs(sudut);
    x1 := xc + Round(blockWidth/2);
    y1 := yc + Round(Round(blockWidth/2)
        * Tan(DegToRad(sudutKuadran)));
    x2 := xc - Round(blockWidth/2);
    y2 := yc - Abs(yc-y1);
end
else if (sudut < -batasSudut) and (sudut >= -90) then
begin
    sudutKuadran := 90-Abs(sudut);
    y1 := yc + Round(blockHeight/2);
    x1 := xc + Round(Round(blockHeight/2)
        * Tan(DegToRad(sudutKuadran)));
    y2 := yc - Round(blockHeight/2);
    x2 := xc - Abs(xc-x1);
end;
end;
```

*Sourcecode 4.10 Sourcecode proses
pencarian titik x1,y1 dan x2,y2
untuk membentuk garis orthogonal*

```

//Prosedur untuk membuat garis orthogonal
procedure TForm1.BresenhamLine(x1,y1,x2,y2: Integer);
var deltax,deltay : Integer;
    i,x,y,den,numadd,numpixels,xinc1,xinc2,yinc1,yinc2: Integer;
    num: Real;
begin
    x := x1;
    y := y1;

    deltax := Abs(x2 - x1);
    deltay := Abs(y2 - y1);

    if (x2 >= x1) then
    begin
        xinc1 := 1;
        xinc2 := 1;
    end
    else
    begin
        xinc1 := -1;
        xinc2 := -1;
    end;

    if (y2 >= y1) then
    begin
        yinc1 := 1;
        yinc2 := 1;
    end
    else
    begin
        yinc1 := -1;
        yinc2 := -1;
    end;

    if (deltax >= deltay) then
    begin
        xinc1 := 0;
        yinc2 := 0;
        den := deltax;
        num := deltax / 2;
        numadd := deltay;
        numpixels := deltax;
    end
    else
    begin
        xinc2 := 0;
        yinc1 := 0;
        den := deltay;
        num := deltay / 2;
        numadd := deltax;
        numpixels := deltay;
    end;
    for i := 0 to numpixels do
    begin
        num := num + numadd;
        if (num >= den) then
        begin
            num := num - den;
            x := x + xinc1;
            y := y + yinc1;
        end;
        x := x + xinc2;
        y := y + yinc2;
        Inc(jumDeret);
        deret[jumDeret] := GaussianSmooth[x+1, y+1];
    end;
end;

```

Sourcecode 4.11 Sourcecode proses menggambar garis yang tegak lurus dengan sudut local orientation


```

//proses ridge frequency estimation
w := 17;
i := (w div 2); j := (w div 2);
while (i <= 186 - (w div 2)) do
begin
  //untuk tiap-tiap blok perhitungkan S(i,j)
  xc := j;
  yc := i;
  sudut := Tetha[i+1,j+1] - 90;
  //cari titik [x1,y1] dan [x2,y2] untuk membuat garis
  orthogonal
  CariKoordinatGarisLintas(xc,yc,w,w,x1,y1,x2,y2,sudut);

  //lakukan looping dari titik x1,y1 sampai x2,y2 untuk
  mendapatkan frekuensi ridge
  xc := x1;
  yc := y1;

  jumDeret := 1;
  deret[jumDeret] := GaussianSmooth[xc+1, yc+1];

  BresenhamLine(x1,y1,x2,y2); //membuat garis orthogonal

  //menghitung jumlah poin minimum jarak antar poin minimum di
  sepanjang garis orthogonal
  jumlahPixelLandai := 0;
  jml := 0;
  d := 1;
  for m := 1 to jumDeret do
  begin
    if (m = 1) then
    begin
      if (deret[m] < deret[m+1]) then
        Inc(jumlahPixelLandai);
    end
    else if (m <> 1) and (m < jumDeret) then
    begin
      if ((deret[m] <= deret[m-1])
        and (deret[m] < deret[m+1])) then
      begin
        Inc(jumlahPixelLandai);
        sum[d] := jml;
        jml := 0;
        inc(d);
      end;
    end
    else if (m = jumDeret) then
    begin
      if ((deret[m] <= deret[m-1])) then
      begin
        Inc(jumlahPixelLandai);
        sum[d] := jml;
        jml := 0;
        inc(d);
      end;
    end;
    inc(jml);
  end;
end;

```

```

jml := 0;
for p := 1 to d-1 do
begin
    jml := jml + sum[p];
end;

//Menghitung S[i,j] dan F[i,j]
if (jumlahPixelLandai-1 <> 0) then
begin
    Minusi := jml / (jumlahPixelLandai-1);
    if (Minusi = 0) then
        freq := 0
    else
        freq := 1 / Minusi;
    end
else if (jumlahPixelLandai-1 = 0) then
begin
    Minusi := 0;
    freq := 0;
end;

for x:=0 to w-1 do
for y:=0 to w-1 do
begin
    imagex := (i - (w div 2) + x + ImageHeight) mod ImageHeight;
    imagey := (j - (w div 2) + Y + ImageWidth) mod ImageWidth;

    F[imagex+1, imagey+1] := freq;
    S[imagex+1, imagey+1] := Minusi;
end;

if (j = 186 - ((w div 2))) then
begin
    j := (w div 2);
    i := i + w;
    InputImage.Repaint;
end
else
begin
    j := j + w;
end;
end;
end;

```

Sourcecode 4.12 Sourcecode proses perhitungan
ridge frequency $F(i,j)$

4.2.6 Filtering

Proses *filterng* merupakan proses terpenting dari serangkaian proses peningkatan citra sidik jari. Pada proses ini, hasil perhitungan *local orientation* dan *ridge frequency* dijadikan sebagai parameter dalam fungsi *Gabor Filter*. Nilai standar deviasi σ_x dan σ_y yang digunakan pada *Gabor Filter* adalah 0,5 dan 3,0 secara berturut-turut. Nilai standar deviasi tersebut didapat dari hasil eksperimen. Perhitungan *Gabor Filter* menghasilkan *mask* yang akan dikalikan

dengan citra ternormalisasi dimana hasilnya disebut dengan *enhanced image* (citra yang telah ditingkatkan kualitasnya). Hasil proses peningkatan kualitas citra dengan beberapa kombinasi nilai σ_x dan σ_y dapat dilihat pada lampiran.

```
//proses filtering
for i:=9 to 186 - 9 do
begin
  for j:=9 to 186 - 9 do
  begin

    thaoX := 0.5;
    thaoY := 3.0;
    wX := 11;
    wY := 11;

    jml := 0;
    p := round(-1 * (wX/2));
    for x:=wX-1 downto 0 do
    begin
      q := round(-1 * (wY/2));
      for y:=wY-1 downto 0 do
      begin
        imagex := (i - (wX div 2) + x + ImageHeight) mod
                  ImageHeight;
        imagey := (j - (wY div 2) + Y + ImageWidth) mod
                  ImageWidth;

        hasil := GaborFilter
                  (p,q,Tetha[i+1, j+1],F[i+1, j+1],thaoX,thaoY);
        tmp := GetRValue(ColorToGrayscale
                          (InputImage.Canvas.Pixels[imagex, imagey]));

        jml := jml + round(hasil * tmp);
        inc(q);
      end;
      inc(p);
    end;
    Enhanced[i+1, j+1] := round(jml);
    EnhancedImage.Canvas.Pixels[i, j] := RGB(Enhanced[i+1,
      j+1],Enhanced[i+1, j+1],Enhanced[i+1, j+1]);
  end;
end;
```

Sourcecode 4.13 Sourcecode proses filtering


```
//fungsi gabor filter
function TForm1.GaborFilter(u,v: Integer;
    orientasi,frequency,Tx,Ty: real): Real;
var
    jum, Xtetha, Ytetha, hasil : real;
begin
    Xtetha := (u * cos(degtorad(orientasi))) +
        (v * sin(degtorad(orientasi)));
    Ytetha := (-1 * u * sin(degtorad(orientasi))) +
        (v * cos(degtorad(orientasi)));

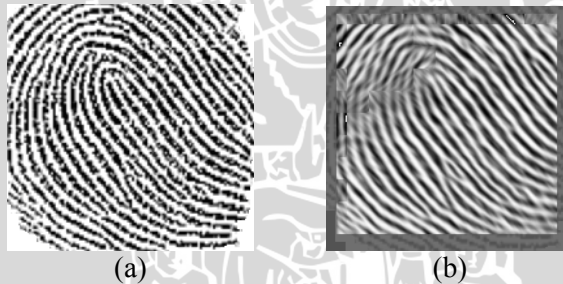
    if (Tx = 0) or (Ty = 0) or (frequency = 0) then
    begin
        jum := 0;
    end
    else
    begin
        jum := (sqr(Xtetha) / sqr(Tx)) +
            (sqr(Ytetha) / sqr(Ty));
    end;

    hasil := Exp(-1 * 0.5 * jum) *
        cos(degtorad(2 * Pi * frequency * Xtetha));

    Result := hasil;
end;
```

Sourcecode 4.14 Sourcecode fungsi Gabor filter

Gambar 4.2 Menunjukkan contoh citra sidik hasil proses *filtering* dengan nilai standar deviasi $\sigma_x = 0,5$ dan $\sigma_y = 3,0$.



(a) (b)
Gambar 4.2 Citra Sidik Jari hasil proses *filtering*
dengan nilai $\sigma_x = 0,5$ dan $\sigma_y = 3,0$
(a) Citra Asli (b) *Enhanced Image*

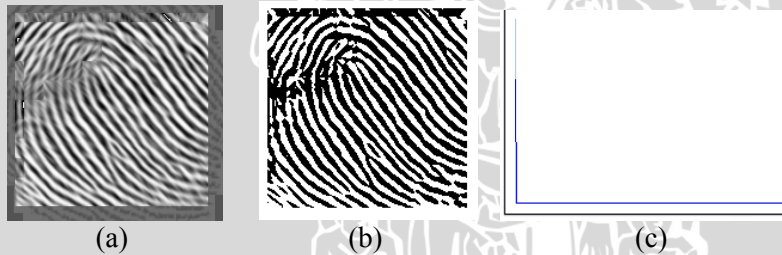
4.2.7 Binerisasi (*Binarization*)

Proses binerisasi pada citra *grayscale* sidik jari diawali dengan menentukan dua *threshold*. $T_1 = 150$ dan $T_2 = 170$. Kedua nilai *threshold* tersebut didapat dari hasil eksperimen. Kemudian nilai *gray-level* dari masing-masing piksel pada *enhanced image* dibandingkan dengan nilai *threshold* yang telah ditentukan untuk mendapatkan nilai biner (0 atau 1).

```
//proses binerisasi
for i:=10 to 176 do
  for j:=10 to 176 do
    begin
      value := GetRValue(ColorToGrayscale
        (EnhancedImage.Canvas.Pixels[i, j]));
      if (value <= 150) then // threshold pertama
        MinutiaeImage.Canvas.Pixels[i,j] := RGB(0,0,0);
        BinaryImage.Canvas.Pixels[i,j] := RGB(0,0,0);
      else if (value >= 170) then // threshold kedua
        MinutiaeImage.Canvas.Pixels[i,j] :=
          RGB(255,255,255);
        BinaryImage.Canvas.Pixels[i,j] :=
          RGB(255,255,255);
    end;
```

Sourcecode 4.15 Sourcecode proses binerisasi

Gambar 4.3 menunjukkan contoh citra sidik jari hasil proses binerisasi dengan *threshold* $T_1 = 150$ dan $T_2 = 170$ serta histogramnya.



(a) (b) (c)
Gambar 4.3 Citra Sidik Jari hasil proses binerisasi
dengan *threshold* $T_1 = 150$ dan $T_2 = 170$
(a) *Enhanced Image* (b) Citra Biner (c) Histogram

Contoh hasil proses binerisasi dengan beberapa kombinasi T_1 dan T_2 dapat dilihat pada lampiran.

4.2.8 Thinning (Penipisan)

Proses *thinning* merupakan proses terakhir sebelum poin-poin minutiae pada citra sidik jari dideteksi. Proses ini akan mengurangi ketebalan pola *ridge* hingga menjadi satu piksel.

```
//proses thinning
//asumsi objek bernilai 1 dan latar bernilai 0
for b := 1 to 30 do
begin
  for i:=0 to ImageHeight-1 do
  begin
    for j:=0 to ImageWidth-1 do
    begin
      color := GetRValue(MinutiaeImage.Canvas.Pixels[i,j]);
      if color = 225 then
        Temp[i+1,j+1] := 0;
      if color = 0 then
        Temp[i+1,j+1] := 1;
      end;
    end;
  end;

  //inisialisasi kondisi
  for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    Kondisi[i+1,j+1] := 1;

  //Tandai langkah 1
  for i:=1 to ImageHeight-2 do
  begin
    for j:=1 to ImageWidth-2 do
    begin
      if Temp[i+1,j+1] = 1 then
      begin
        p[1] := Temp[i+1,j+1];
        p[2] := Temp[i,j+1];
        p[3] := Temp[i,j+2];
        p[4] := Temp[i+1,j+2];
        p[5] := Temp[i+2,j+2];
        p[6] := Temp[i+2,j+1];
        p[7] := Temp[i+2,j];
        p[8] := Temp[i+1,j];
        p[9] := Temp[i,j];
        p[10] := p[2];

        sum := 0;
        for x := 2 to 9 do
          if p[x] = 0 then
            inc(sum);

        if sum >= 1 then
        begin
          N := 0;
          for x := 2 to 9 do
            N := N + p[x];

          Sx := 0;
          for x := 2 to 9 do
            if (p[x] = 0) and (p[x+1] = 1) then
              inc(Sx);
```



```

if (N >= 2) and (N <= 6) and (Sx = 1) and (p[2]*p[4]*p[6] = 0) and
(p[4]*p[6]*p[8] = 0) then
    Kondisi[i+1,j+1] := 0;

    end;
end;
end;
end;

//Tipiskan langkah 1
for i:=0 to ImageHeight-1 do
    for j:=0 to ImageWidth-1 do
        if Kondisi[i+1,j+1] = 0 then
            Temp[i+1,j+1] := 0;

//reinisialisasi kondisi
for i:=0 to ImageHeight-1 do
    for j:=0 to ImageWidth-1 do
        Kondisi[i+1,j+1] := 1;

//Tandai langkah 2
for i:=1 to ImageHeight-2 do
begin
    for j:=1 to ImageWidth-2 do
begin
    if Temp[i+1,j+1] = 1 then
begin
    p[1] := Temp[i+1,j+1];
    p[2] := Temp[i,j+1];
    p[3] := Temp[i,j+2];
    p[4] := Temp[i+1,j+2];
    p[5] := Temp[i+2,j+2];
    p[6] := Temp[i+2,j+1];
    p[7] := Temp[i+2,j];
    p[8] := Temp[i+1,j];
    p[9] := Temp[i,j];
    p[10] := p[2];

    sum := 0;
    for x := 2 to 9 do
        if p[x] = 0 then
            inc(Sum);

    if sum >= 1 then
begin
    N := 0;
    for x := 2 to 9 do
        N := N + p[x];

    Sx := 0;
    for x := 2 to 9 do
        if (p[x] = 0) and (p[x+1] = 1) then
            inc(Sx);

    if (N >= 2) and (N <= 6) and (Sx = 1) and (p[2]*p[4]*p[8] =
0) and (p[2]*p[6]*p[8] = 0) then
        Kondisi[i+1,j+1] := 0;

    end;
end;
end;
end;
end;

```

```
//Tipiskan langkah 2
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    if Kondisi[i+1,j+1] = 0 then
      Temp[i+1,j+1] := 0;

// finishing
for i:=1 to ImageHeight-2 do
  for j:=1 to ImageWidth-2 do
    begin
      if Temp[i+1,j+1] = 0 then
        MinutiaeImage.Canvas.Pixels[i,j] := RGB(255,255,255);
      if Temp[i+1,j+1] = 1 then
        MinutiaeImage.Canvas.Pixels[i,j] := RGB(0,0,0);
    end;
  end;
end;
```

Sourcecode 4.16 Sourcecode proses thinning

4.2.9 Minutiae Detection (Deteksi Minutiae)

Pendeteksian poin-poin *minutiae* ini dipusatkan pada 3 jenis poin *minutiae*, yaitu *ridge ending*, *ridge bifurcation* dan *ridge crossing*. Ketiga poin ini akan dicatat jumlah kemunculannya pada citra sidik jari yang dideteksi. Poin-poin yang terdeteksi tersebut akan ditandai dengan warna merah.

```
//proses deteksi minutiae
//pindahkan nilai tiap pixel ke temp
for i:=1 to ImageHeight-1 do
  for j:=1 to ImageWidth-1 do
    begin
      color := GetRValue(MinutiaeImage.Canvas.Pixels[i,j]);
      if color = 255 then
        Tmp[i+1,j+1] := 1;
      if color = 0 then
        Tmp[i+1,j+1] := 0;

    end;

// inisialisasi status
for i:=0 to ImageHeight-1 do
  for j:=0 to ImageWidth-1 do
    status[i+1,j+1] := 0;

//deteksi poin-poin minusi
CN1 := 0;
CN3 := 0;
CN4 := 0;
for i:=14 to 172 do
  for j:=14 to 172 do
    begin
      if tmp[i+1,j+1] = 0 then
        begin
          pixel[1] := Tmp[i+1,j+2];
          pixel[2] := Tmp[i,j+2];
```

```

pixel[3] := Tmp[i,j+1];
pixel[4] := Tmp[i,j];
pixel[5] := Tmp[i+1,j];
pixel[6] := Tmp[i+2,j];
pixel[7] := Tmp[i+2,j+1];
pixel[8] := Tmp[i+2,j+2];
pixel[9] := pixel[1];

jum := 0;
for N := 1 to 8 do
begin
    jum := jum + abs(pixel[N] - pixel[N+1]);
end;
hsl := round(jum / 2);

if (hsl = 1) or (hsl = 3) or (hsl = 4) then
begin
    status[i+1,j+1] := 1;
    if (hsl = 1) then
        inc(CN1);
    if (hsl = 3) then
        inc(CN3);
    if (hsl = 4) then
        inc(CN4);
    end;
end;

memol.Lines.Add('HASIL DETEKSI DITEMUKAN');
memol.Lines.Add('Ridge Ending' + ' = ' + inttostr(CN1));
memol.Lines.Add('Ridge Bifurcation' + ' = ' + inttostr(CN3));
memol.Lines.Add('Ridge Crossing' + ' = ' + inttostr(CN4));

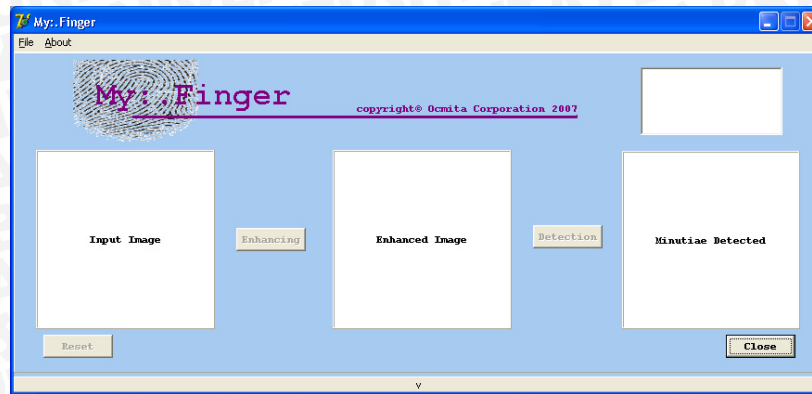
//tandai poin-poin minusi yang terdeteksi
for i:=0 to ImageHeight-1 do
    for j:=0 to ImageWidth-1 do
        if status[i+1,j+1] = 1 then
            begin
                MinutiaeImage.Canvas.Pixels[i,j] := clred;
                MinutiaeImage.Canvas.Pixels[i,j+1] := clred;
                MinutiaeImage.Canvas.Pixels[i-1,j+1] := clred;
                MinutiaeImage.Canvas.Pixels[i-1,j] := clred;
                MinutiaeImage.Canvas.Pixels[i-1,j-1] := clred;
                MinutiaeImage.Canvas.Pixels[i,j-1] := clred;
                MinutiaeImage.Canvas.Pixels[i+1,j-1] := clred;
                MinutiaeImage.Canvas.Pixels[i+1,j] := clred;
                MinutiaeImage.Canvas.Pixels[i+1,j+1] := clred;
            end;
        end;
    end;
end;

```

Sourcecode 4.17 Sourcecode proses deteksi minutiae

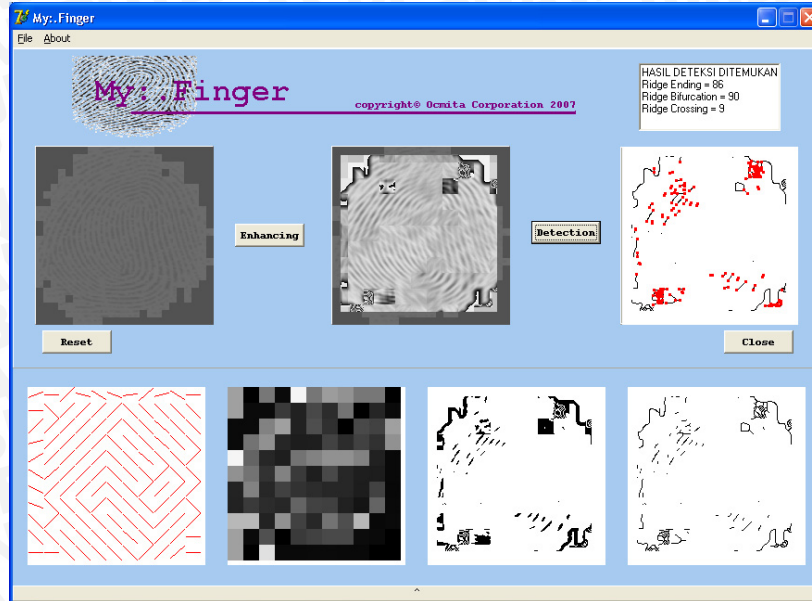
4.3 Implementasi *Interface*

Pembangunan *interface* didasarkan pada perancangan *interface* pada subbab 3.2.2 dengan diberi sedikit tambahan informasi.



Gambar 4.4 Tampilan Form Utama

Menu *File* pada *form* utama berisi submenu *open* untuk membuka *file* citra masukan (*input image*), *save* untuk menyimpan *minutiae detected image* dan *exit* untuk keluar dari sistem. Sedangkan menu *About* berisi submenu *About My:Finger* sebagai informasi mengenai sistem yang dibangun serta *About Programmer* yang berisi informasi tentang pembuat sistem. Pada tampilan *form* utama terdapat tiga buah *field* untuk menampilkan *image*, yaitu *input image*, *enhanced image*, dan *minutiae detected image*. Pada *form* tersebut juga terdapat empat buah *button*. *Button close* untuk menutup aplikasi, *button reset* untuk mengembalikan tampilan *form* utama seperti keadaan awal, *button enhancing* untuk melakukan proses peningkatan kualitas citra, *button detection* untuk mengolah *enhanced image* menjadi *binary image* dan *thinned image* hingga pendeteksian poin-poin *minutiae*. Di pojok kanan atas *form* utama terdapat memo yang digunakan untuk menampilkan informasi jumlah *ridge ending*, *ridge bifurcation*, dan *ridge crossing* pada citra sidik jari yang berhasil dideteksi oleh sistem.



Gambar 4.5 Tampilan hasil peningkatan kualitas citra dan deteksi *minutiae*

Pada Gambar 4.5 terdapat empat *output image* tambahan yang terletak di bawah tiga *field* citra yang telah dijelaskan sebelumnya. Keempat citra tersebut adalah citra hasil proses *image orientation estimation*, citra hasil proses *ridge frequency estimation*, *binary image* dan *thinned image* secara berturut-turut dari sebelah kiri gambar.

4.4 Implementasi Uji Coba

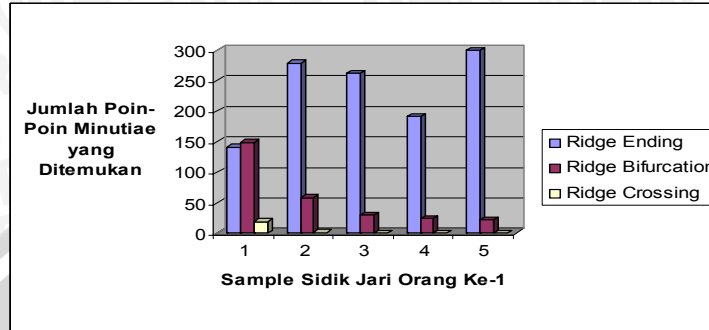
Pada subbab ini akan dilakukan pembahasan mengenai pengujian yang dilakukan pada sistem dan hasil evaluasi dari hasil yang dikeluarkan sistem.

4.4.1 Evaluasi Hasil 15 Citra Uji

Seperti yang telah dijelaskan pada subbab 3.3, uji coba pertama diterapkan pada sepuluh sidik jari orang yang berbeda. Masing-masing orang tersebut akan diambil lima *sample* sidik jarinya dengan posisi yang berbeda-beda. Berikut merupakan hasil uji coba sistem.

Tabel 4.1 Hasil pendeteksian poin-poin *minutiae* orang ke-1

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	141	149	19
2	279	58	1
3	263	29	0
4	192	24	0
5	300	22	0



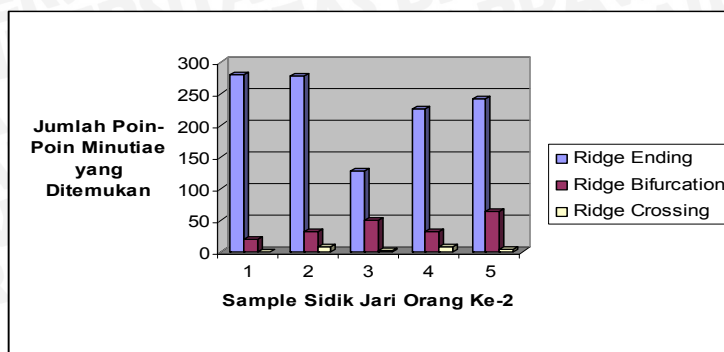
Gambar 4.6 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-1

Tabel 4.2 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-1

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	141	300	235,00	66,389	4407,500
Ridge_Bifurcation	5	22	149	56,40	53,762	2890,300
Ridge_Crossing	5	0	19	4,00	8,396	70,500
Valid N (listwise)	5					

Tabel 4.3 Hasil pendeteksian poin-poin *minutiae* orang ke-2

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	279	21	0
2	278	33	9
3	128	51	2
4	225	33	9
5	242	64	4



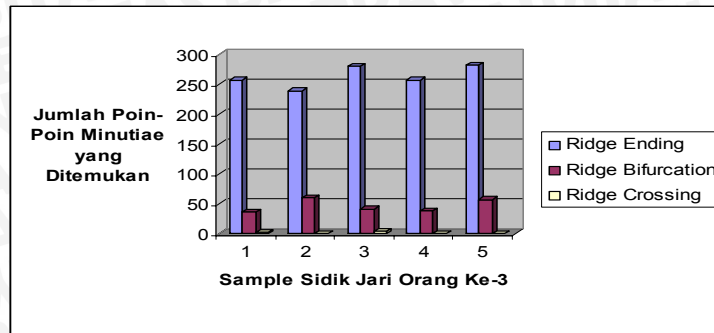
Gambar 4.7 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-2

Tabel 4.4 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-2

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	128	279	230,40	61,800	3819,300
Ridge_Bifurcation	5	21	64	40,40	16,994	288,800
Ridge_Crossing	5	0	9	4,80	4,087	16,700
Valid N (listwise)	5					

Tabel 4.5 Hasil pendeteksian poin-poin *minutiae* orang ke-3

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	257	36	2
2	239	60	1
3	279	41	4
4	257	38	0
5	281	57	1



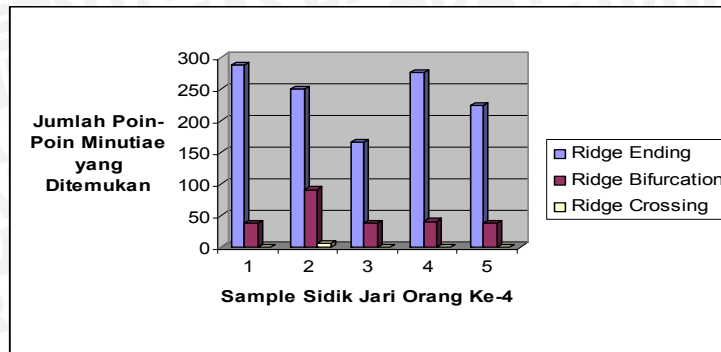
Gambar 4.8 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-3

Tabel 4.6 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-3

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	239	281	262,60	17,516	306,800
Ridge_Bifurcation	5	36	60	46,40	11,238	126,300
Ridge_Crossing	5	0	4	1,60	1,517	2,300
Valid N (listwise)	5					

Tabel 4.7 Hasil pendeteksian poin-poin *minutiae* orang ke-4

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	287	39	0
2	250	91	6
3	166	38	0
4	276	41	0
5	224	39	0



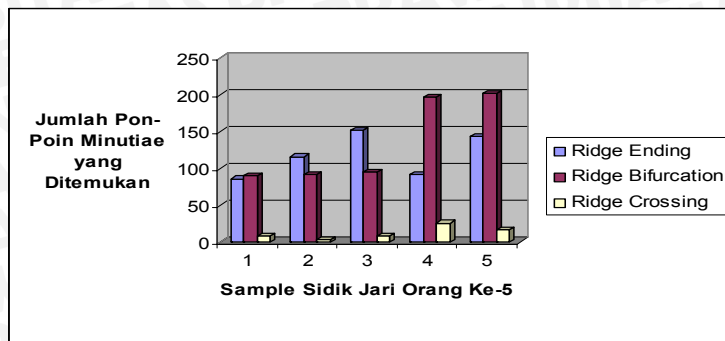
Gambar 4.9 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-4

Tabel 4.8 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-4

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	166	287	240,60	48,309	2333,800
Ridge_Bifurcation	5	38	91	49,60	23,169	536,800
Ridge_Crossing	5	0	6	1,20	2,683	7,200
Valid N (listwise)	5					

Tabel 4.9 Hasil pendeteksian poin-poin *minutiae* orang ke-5

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	86	90	9
2	116	92	4
3	152	95	8
4	92	196	26
5	143	202	17



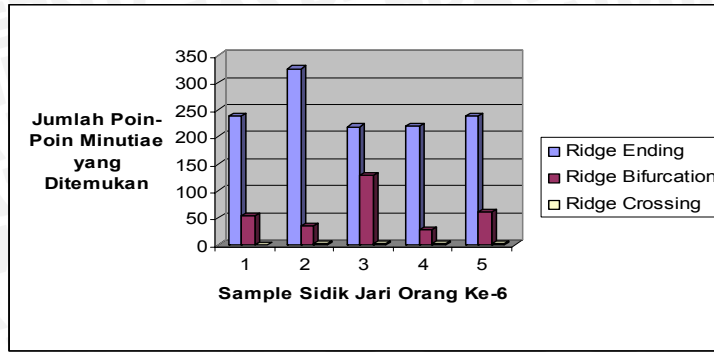
Gambar 4.10 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-5

Tabel 4.10 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-5

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	86	152	117,80	29,516	871,200
Ridge_Bifurcation	5	90	202	135,00	58,489	3421,000
Ridge_Crossing	5	4	26	12,80	8,758	76,700
Valid N (listwise)	5					

Tabel 4.11 Hasil pendeteksian poin-poin *minutiae* orang ke-6

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	237	53	0
2	325	36	2
3	218	129	3
4	220	29	3
5	237	60	3



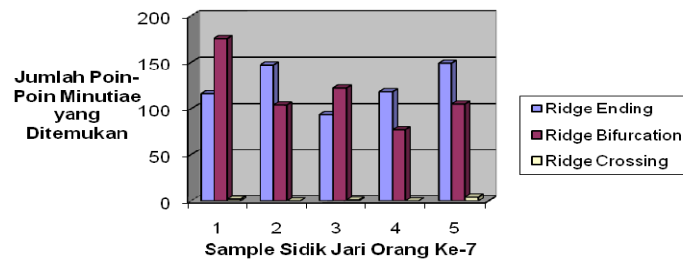
Gambar 4.11 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-6

Tabel 4.12 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-6

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	218	325	247,40	44,309	1963,300
Ridge_Bifurcation	5	29	129	61,40	39,803	1584,300
Ridge_Crossing	5	0	3	2,20	1,304	1,700
Valid N (listwise)	5					

Tabel 4.13 Hasil pendeteksian poin-poin *minutiae* orang ke-7

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	116	175	2
2	147	104	0
3	93	122	1
4	118	77	0
5	149	105	4



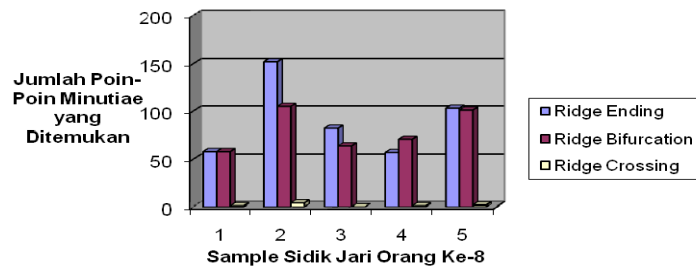
Gambar 4.12 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-7

Tabel 4.14 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-7

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	93	149	124,60	23,522	553,300
Ridge_Bifurcation	5	77	175	116,60	36,405	1325,300
Ridge_Crossing	5	0	4	1,40	1,673	2,800
Valid N (listwise)	5					

Tabel 4.15 Hasil pendeteksian poin-poin *minutiae* orang ke-8

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	58	58	1
2	152	105	4
3	82	64	0
4	57	71	1
5	103	101	2



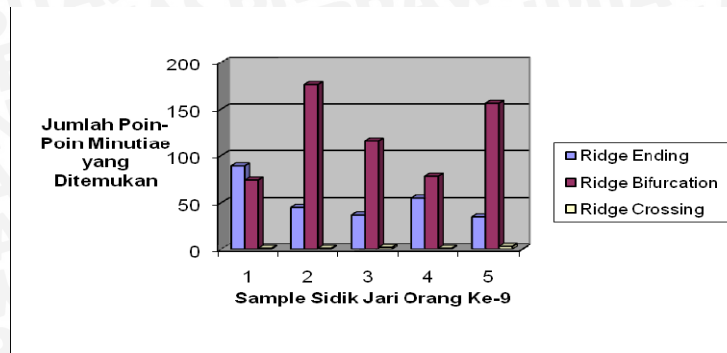
Gambar 4.13 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-8

Tabel 4.16 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-8

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	57	152	90,40	39,336	1547,300
Ridge_Bifurcation	5	58	105	79,80	21,719	471,700
Ridge_Crossing	5	0	4	1,60	1,517	2,300
Valid N (listwise)	5					

Tabel 4.17 Hasil pendeteksian poin-poin *minutiae* orang ke-9

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	89	74	1
2	45	176	1
3	37	116	2
4	55	78	1
5	35	156	3



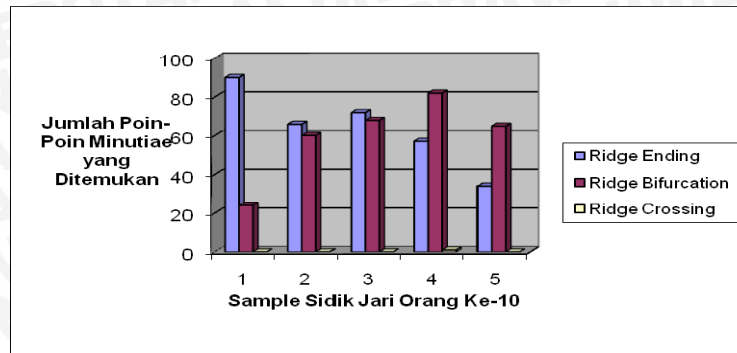
Gambar 4.14 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-9

Tabel 4.18 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-9

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	35	89	52,20	22,027	485,200
Ridge_Bifurcation	5	74	176	120,00	45,629	2082,000
Ridge_Crossing	5	1	3	1,60	,894	,800
Valid N (listwise)	5					

Tabel 4.19 hasil pendeteksian poin-poin *minutiae* orang ke-10

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	90	24	0
2	66	60	0
3	72	68	0
4	57	82	1
5	34	65	0



Gambar 4.15 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-10

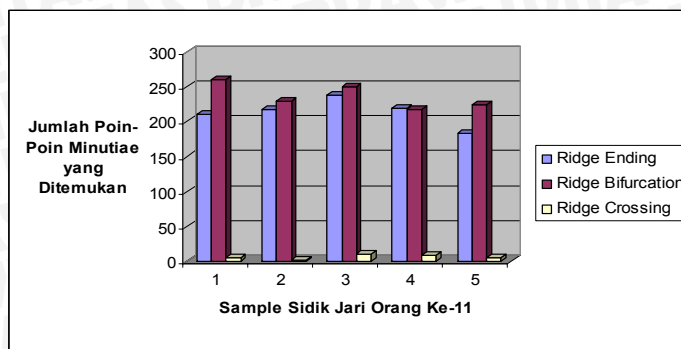
Tabel 4.20 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-10

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	34	90	63,80	20,572	423,200
Ridge_Bifurcation	5	24	82	59,80	21,615	467,200
Ridge_Crossing	5	0	1	,20	,447	,200
Valid N (listwise)	5					

Berikut merupakan hasil dari uji coba kedua, dimana objek yang digunakan adalah lima sidik jari orang yang berbeda. Tiap orang tersebut akan diambil lima *sample* sidik jarinya dengan pola pengambilan yang hampir sama.

Tabel 4.21 Hasil pendeteksian poin-poin *minutiae* orang ke-11

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	211	261	5
2	217	230	2
3	238	250	10
4	219	217	9
5	184	224	5



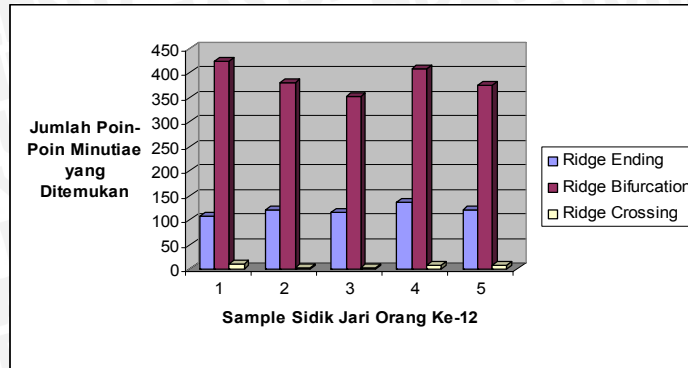
Gambar 4.16 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-11

Tabel 4.22 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-11

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	184	238	213,80	19,486	379,700
Ridge_Bifurcation	5	217	261	236,40	18,447	340,300
Ridge_Crossing	5	2	10	6,20	3,271	10,700
Valid N (listwise)	5					

Tabel 4.23 Hasil pendeteksian poin-poin *minutiae* orang ke-12

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	108	423	10
2	122	380	4
3	115	353	3
4	136	408	8
5	120	375	8



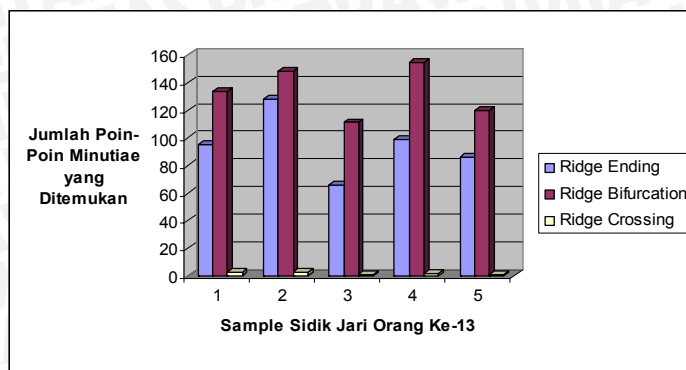
Gambar 4.17 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-12

Tabel 4.24 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-12

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	108	136	120,20	10,354	107,200
Ridge_Bifurcation	5	353	423	387,80	27,761	770,700
Ridge_Crossing	5	3	10	6,60	2,966	8,800
Valid N (listwise)	5					

Tabel 4.25 Hasil pendeteksian poin-poin *minutiae* orang ke-13

Sidik Jari	Jenis Poin <i>Minutiae</i>		
	<i>Ridge Ending</i>	<i>Ridge Bifurcation</i>	<i>Ridge Crossing</i>
1	95	133	3
2	128	148	3
3	66	111	1
4	99	154	2
5	86	120	1



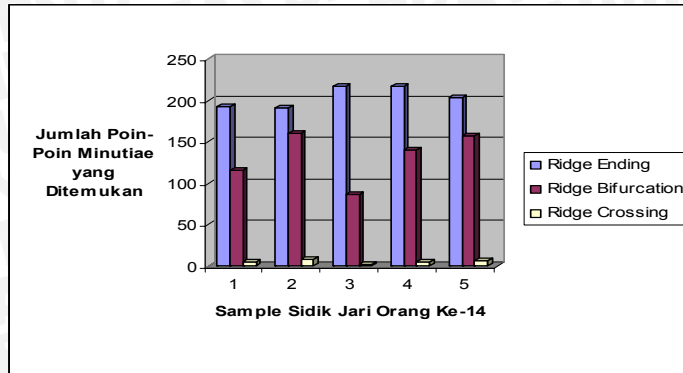
Gambar 4.18 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-13

Tabel 4.26 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-13

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	66	128	94,80	22,510	506,700
Ridge_Bifurcation	5	111	154	133,20	18,158	329,700
Ridge_Crossing	5	1	3	2,00	1,000	1,000
Valid N (listwise)	5					

Tabel 4.27 Hasil pendeteksian poin-poin *minutiae* orang ke-14

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	193	115	4
2	192	160	8
3	218	86	1
4	218	141	5
5	203	158	7



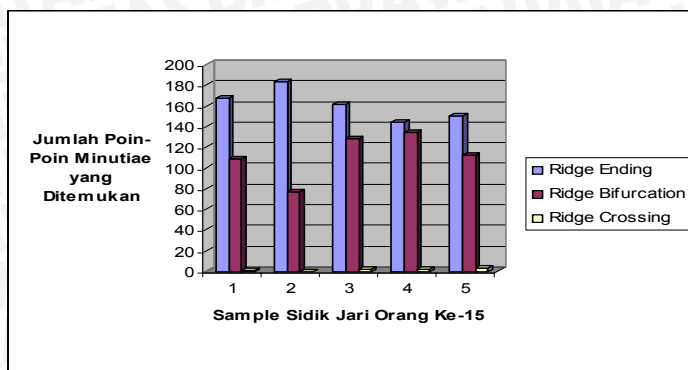
Gambar 4.19 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-14

Tabel 4.28 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-14

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	192	218	204,80	12,795	163,700
Ridge_Bifurcation	5	86	160	132,00	31,409	986,500
Ridge_Crossing	5	1	8	5,00	2,739	7,500
Valid N (listwise)	5					

Tabel 4.29 Hasil pendeteksian poin-poin *minutiae* orang ke-15

Sidik Jari	Jenis Poin Minutiae		
	Ridge Ending	Ridge Bifurcation	Ridge Crossing
1	167	108	1
2	183	77	0
3	161	128	2
4	144	134	2
5	150	112	3



Gambar 4.20 Grafik hasil pendeteksian poin-poin *minutiae* orang ke-15

Tabel 4.30 Hasil perhitungan varian terhadap poin-poin *minutiae* orang ke-15

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Ridge_Ending	5	144	183	161,00	15,248	232,500
Ridge_Bifurcation	5	77	134	111,80	22,253	495,200
Ridge_Crossing	5	0	3	1,60	1,140	1,300
Valid N (listwise)	5					

Seperti yang telah disebutkan pada subbab 3.3, uji coba juga dilakukan dengan membandingkan hasil pendeteksian poin-poin *minutiae* dari kelima belas orang yang diambil *sample* sidik jarinya. Tujuannya adalah untuk memeriksa apakah jumlah poin-poin *minutiae* hasil pendeteksian sistem pada setiap orang berbeda atau tidak. Tabel 4.31, Tabel 4.32 dan Tabel 4.33 menunjukkan hasil pendeteksian *ridge ending*, *ridge bifurcation* dan *ridge crossing* dari kelima belas orang tersebut secara berturut-turut.

Tabel 4.31 Tabel hasil pendeteksian
ridge ending pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1	141	279	263	192	300
2	279	278	128	225	242
3	257	239	279	257	281
4	287	250	166	276	224
5	86	116	152	92	143
6	237	325	218	220	237
7	116	147	93	118	149
8	58	152	82	57	103
9	89	45	37	55	35
10	90	66	72	57	34
11	211	217	238	219	184
12	108	122	115	136	120
13	95	128	66	99	86
14	193	192	218	218	203
15	167	183	161	144	150

Data dari Table 4.31 dijadikan masukan untuk pengujian terhadap perbandingan *ridge ending* dari kelima belas orang yang diambil *sample* sidik jarinya.

Tests of Between-Subjects Effects					
Dependent Variable: Ridge_Ending					
Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	372494,080 ^a	14	26606,720	22,049	,000
Intercept	2016216,120	1	2016216,120	1670,833	,000
Orang	372494,080	14	26606,720	22,049	,000
Error	72402,800	60	1206,713		
Total	2461113,000	75			
Corrected Total	444896,880	74			

a. R Squared = ,837 (Adjusted R Squared = ,799)

Gambar 4.21 Hasil pengujian terhadap perbandingan *ridge ending* dari kelima belas orang yang diambil *sample* sidik jarinya

Dengan hipotesa pada analisa adalah :

H_0 : Orang ke-1 = Orang ke-2 = Orang ke-3 = ... = Orang ke-15.

H_1 : Paling sedikit ada satu pasang orang yang tidak sama.

Analisa dilakukan dengan menggunakan level toleransi α sebesar 0,05. Pada Gambar 4.21, tepatnya pada baris orang memperlihatkan nilai *Significantly* (Sig.) sebesar 0,000. Nilai ini kemudian

dibandingkan dengan nilai α . Jika nilai *Significantly* lebih besar sama dengan ($\geq$) nilai α , maka terima H_0 . Tetapi jika nilai *Significantly* lebih kecil dari ($<$) nilai α , maka terima H_1 . Nilai 0,000 jelas lebih kecil dari nilai α (0,05). Jadi keputusan untuk Table 4.31 adalah terima H_1 , yaitu paling sedikit ada satu pasang orang yang tidak sama. Untuk memperlihatkan perbedaannya secara detil, maka dilakukan uji lanjutan yang disebut dengan *Post Hoc Test*. Hasil *Post Hoc Test* untuk *ridge ending* dapat dilihat pada Lampiran 5.

Tabel *post hoc test* pada Lampiran 5 menunjukkan perbandingan secara detil antara orang yang satu dengan yang lainnya. Sebagai contoh, perbandingan orang ke-1 dengan keempatbelas orang lainnya. Karena perbandingan orang ke-1 dengan orang ke-2, ke-3, ke-4, ke-6, ke-11 dan ke-14 menunjukkan nilai *significantly* (Sig.) yang lebih besar dari nilai α (0,05), yaitu 0,835; 0,214; 0,800; 0,575; 0,338 dan 0,174, maka hasil pendeteksian poin *ridge ending* antara orang ke-1 dengan keenam orang tersebut dianggap sama. Lain halnya dengan perbandingan antara orang ke-1 dengan orang ke-5, orang ke-7 sampai orang ke-13. Perbandingan tersebut menunjukkan nilai *significantly* yang lebih rendah dari nilai α , yaitu 0,000. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa hasil pendeteksian poin *ridge ending* antara orang ke-1 dengan orang ke-5, orang ke-7 sampai orang ke-13 dianggap berbeda. Hal serupa juga terlihat pada perbandingan antara orang ke-1 dengan orang ke-15. Karena nilai *significantly*nya lebih kecil dari nilai α , yaitu 0,001, maka orang ke-1 dengan orang ke-15 dapat dianggap berbeda.



Tabel 4.32 Tabel hasil pendeteksian
ridge bifurcation pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1	149	58	29	24	22
2	21	33	51	33	64
3	36	60	41	38	57
4	39	91	38	41	39
5	90	92	95	196	202
6	53	36	129	29	60
7	175	104	122	77	105
8	58	105	64	71	101
9	74	176	116	78	156
10	24	60	68	82	65
11	261	230	250	217	224
12	423	380	353	408	375
13	133	148	111	154	120
14	115	160	86	141	158
15	108	77	128	134	112

Tests of Between-Subjects Effects

Dependent Variable: Ridge_Bifurcation

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	576208,747 ^a	14	41157,768	38,307	,000
Intercept	1040291,853	1	1040291,853	968,248	,000
Orang	576208,747	14	41157,768	38,307	,000
Error	64464,400	60	1074,407		
Total	1680965,000	75			
Corrected Total	640673,147	74			

a. R Squared = ,899 (Adjusted R Squared = ,876)

Gambar 4.22 Hasil pengujian
terhadap perbandingan *ridge bifurcation*
dari kelima belas orang yang diambil *sample* sidik jarinya

Nilai *significantly* (Sig.) dari baris orang pada Gambar 4.22 menunjukkan angka 0,000. Jika dibandingkan dengan nilai α (0,05), tentu nilai *significantly* lah yang lebih kecil. Jadi keputusan untuk uji perbandingan *ridge bifurcation* ini adalah tolak H_0 , yang berarti paling sedikit ada satu pasang orang yang tidak sama. Detil dari perbandingan tersebut dapat dilihat pada Lampiran 6.

Berdasarkan tabel *post hoc test*, dapat dilihat bahwa hasil pendeteksian poin *ridge bifurcation* antara orang ke-1 dengan orang ke-2, ke-3, ke-4, ke-6, ke-8 dan ke-10 dianggap sama. Hal ini disebabkan oleh nilai *significantly* yang lebih besar bila dibandingkan dengan nilai α , yaitu 0,443; 0,631; 0,744; 0,810; 0,263 dan 0,870. Sedangkan hasil pendeteksian *ridge bifurcation* antara orang ke-1 dengan orang ke-5, orang ke-7, orang ke-9, orang ke-11 sampai ke-15 dianggap berbeda karena nilai *significantly*nya lebih kecil dari nilai α , yaitu 0,000; 0,005; 0,003; 0,000; 0,000; 0,000; 0,001 dan 0,010. Begitu seterusnya untuk melihat perbandingan orang yang lain.

Tabel 4.33 Tabel hasil pendeteksian *ridge crossing* pada masing-masing orang

ORANG	SAMPLE				
	1	2	3	4	5
1	19	1	0	0	0
2	0	9	2	9	4
3	2	1	4	0	1
4	0	6	0	0	0
5	9	4	8	26	17
6	0	2	3	3	3
7	2	0	1	0	4
8	1	4	0	1	2
9	1	1	2	1	3
10	0	0	0	1	0
11	5	2	10	9	5
12	10	4	3	8	8
13	3	3	1	2	1
14	4	8	1	5	7
15	1	0	2	2	3

Tests of Between-Subjects Effects

Dependent Variable: Ridge_Crossing

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	732,720 ^a	14	52,337	3,729	,000
Intercept	929,280	1	929,280	66,219	,000
Orang	732,720	14	52,337	3,729	,000
Error	842,000	60	14,033		
Total	2504,000	75			
Corrected Total	1574,720	74			

^a. R Squared = ,465 (Adjusted R Squared = ,341)

Gambar 4.23 Hasil pengujian terhadap perbandingan *ridge crossing* dari kelima belas orang yang diambil *sample* sidik jarinya

Nilai *significantly* (Sig.) dari baris orang pada Gambar 4.23 menunjukkan angka 0,000. Jika dibandingkan dengan nilai α (0,05), tentu nilai *significantly* lah yang lebih kecil. Jadi keputusan untuk uji perbandingan *ridge bifurcation* ini adalah tolak H_0 , yang berarti paling sedikit ada satu pasang orang yang tidak sama. Gambar 4.19 menunjukkan detil dari perbandingan tersebut.

Hasil dari *post hoc test* untuk poin *ridge crossing* dapat dilihat pada Lampiran 7. Sama halnya dengan *post hoc test* untuk poin *ridge ending* dan *ridge bifurcation* sebelumnya, untuk melihat perbedaan orang yang satu dengan yang lainnya adalah dengan membandingkan nilai *significantly* dengan nilai α . Berdasarkan tabel *post hoc test* pada Lampiran 7, dapat dilihat bahwa hasil pendeteksian poin *ridge crossing* antara orang ke-1 dengan orang ke-5 dianggap berbeda. Hal ini disebabkan oleh nilai *significantly* yang lebih kecil bila dibandingkan dengan nilai α , yaitu 0,000. Sedangkan hasil pendeteksian *ridge crossing* antara orang ke-1 dengan ketigabelas orang lainnya dianggap sama karena nilai *significantly* nya lebih besar dari nilai α .



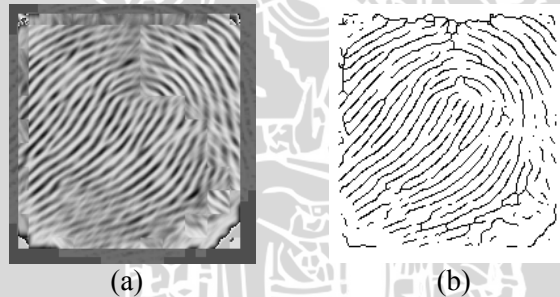
4.4.2 Analisa Hasil

Pengujian sistem dilakukan dengan memberi nilai standar deviasi σ_x dan σ_y secara statis untuk semua citra masukan, yaitu sebesar 0,5 dan 3,0 secara berturut-turut. Pemberian nilai tersebut didasarkan pada beberapa percobaan yang telah dilakukan sebelumnya. Dari pemberian nilai tersebut didapatkan hasil peningkatan kualitas citra sidik jari seperti yang terlihat pada Gambar 4.24.



Gambar 4.24 Hasil proses peningkatan citra sidik jari
(a) Citra Asli (b) *Enhanced Image*

Kemudian jika *enhanced image* yang dihasilkan dikenakan proses *thinning*, maka akan dihasilkan citra seperti yang ditunjukkan pada Gambar 4.25.



Gambar 4.25 Hasil proses *thinning* citra sidik jari
(a) *Enhanced Image* (b) *Thinned Image*

Thinned Image yang dihasilkan akan dideteksi poin-poin *minutiae*nya. Gambar 4.26 menunjukkan citra hasil pendeteksian poin-poin *minutiae*.



Gambar 4.26 Hasil Proses Pendeteksian Poin-Poin *Minutiae*

(a) *Thinned Image*

(b) Citra Hasil Pendeteksian Poin-Poin *Minutiae*

Berdasarkan evaluasi hasil pada subbab 4.4.1, dapat dilihat bahwa dari kelima sample sidik jari pada masing-masing orang didapatkan jumlah poin-poin *minutiae* yang tidak konsisten, baik pada *sample* citra sidik jari dengan pola masukan yang berbeda maupun dengan pola masukan yang hampir sama. Hal ini disebabkan karena setiap citra sidik jari masukan memiliki karakteristik frekuensi yang berbeda-beda, sehingga saat melakukan proses *filtering* (menggunakan filter gabor) dibutuhkan nilai standar deviasi σ_x dan σ_y yang sesuai dengan karakteristik tersebut.

Penentuan nilai standar deviasi yang statis terhadap variasi citra masukan sangat mempengaruhi proses peningkatan kualitas citra, dimana proses akan bekerja secara maksimal jika nilai standar deviasi cocok dengan karakteristik frekuensi citra masukan. Begitu pula sebaliknya, proses akan bekerja kurang maksimal jika nilai standar deviasi tidak cocok dengan karakteristik frekuensi citra masukan.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat ditarik dari skripsi ini adalah dari kelima *sample* sidik jari pada masing-masing orang didapatkan jumlah poin-poin *minutiae* yang tidak konsisten baik pada *sample* citra sidik jari dengan pola masukan yang berbeda maupun dengan pola masukan yang hampir sama. Hal ini disebabkan karena setiap citra sidik jari masukan memiliki karakteristik frekuensi yang berbeda-beda, sehingga saat melakukan proses *filtering* (menggunakan filter gabor) dibutuhkan nilai standar deviasi σ_x dan σ_y yang sesuai dengan karakteristik tersebut.

Penentuan nilai standar deviasi yang statis terhadap variasi citra masukan sangat mempengaruhi proses peningkatan kualitas citra, dimana proses akan bekerja secara maksimal jika nilai standar deviasi cocok dengan karakteristik frekuensi citra masukan. Begitu pula sebaliknya, proses akan bekerja kurang maksimal jika nilai standar deviasi tidak cocok dengan karakteristik frekuensi citra masukan.

Namun jika seluruh hasil dari uji coba sistem terhadap kelima belas orang yang diambil *sample* sidik jarinya dibandingkan dengan menggunakan metode *Analysis of Variance*, maka didapat kesimpulan yang menunjukkan bahwa paling sedikit ada satu pasang sidik jari orang yang berbeda.

5.2 Saran

Beberapa saran pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah :

1. Melakukan percobaan untuk mendapatkan nilai standard deviasi σ_x dan σ_y yang dinamis dimana nilai tersebut dapat berubah-ubah sesuai dengan karakteristik frekuensi citra masukan, sehingga proses peningkatan kualitas citra menjadi lebih maksimal.
2. Melakukan pengembangan terhadap sistem pendeteksi poin-poin *minutiae* ini dengan melanjutkannya ke tahap pengenalan sidik jari.



DAFTAR PUSTAKA

Achmad, Blaza dan Firdausy, Kartika. *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Yogyakarta: Ardi Publishing. 2005.

Ananta, Adityo; Hayurani, Herika; Wulansari, Indah; Azhari, Rakhmad; Kemala, Ratih. *Penggunaan Metode Thinning pada Pemrosesan Gambar*. Jakarta: Fak. Ilmu Komputer, UI. 2003.

Arvian, Yandhrie. *Sidik Jari*. Jakarta: Koran Tempo. 2005.

Chama, Nimitha. *Fingerprint Image Enhancement and Minutiae Extraction*. Dept. Of Electrical and Computer Engineering, Clemson University.

Chikkerur, Sharat; Wu, Chaohang and Govindaraju, Venu. *A Systematic Approach for Feature Extraction in Fingerprint Image*. USA: State University of New York.

Elvayandri. *Sistem Keamanan Akses Menggunakan Pola Sidik Jari Berbasis Jaringan Saraf Tiruan*. Bandung: ITB. 2002.

Gonzalez, Rafael and Woods, Richard. *Digital Image Processing*. USA: Addison-Wesley Publishing Company, Inc, 1992.

Hong, Lin; Wan, Yifei; Jain, Anil. *Fingerprint Image Enhancement: Algorithm and Performance Evaluation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, August 1998.

<http://www.biometrix.at>, tanggal akses : 5 Oktober 2007.

Jain, L.C; Halici, U; Hayashi, I; Lee, S.B and Tsustui.S. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. USA: CRC Press.1999.

Lorenzo, V. Lopez; Pellitero, P. Huerta; Torre, M. I. Jose; Villar, C. Javier. *Fingerprint Minutiae Extraction Based on FPGA and Matlab*. Univesidad Rey Juan Carlos. 2003.

Meidiyanto, Thomas. *Aplikasi FPGA sebagai Pendeteksi Sidik Jari dengan Menggunakan Algoritma Gabor Filter*. Surabaya: Jurusan Teknik Elektro, Fak. Teknologi Industri, ITS. 2002.

Nugroho, Aldiantoro; Cininta; Fitriani, Dhini; Liputo, Rifka dan Lestyaningrum, Yoga. *Image Thinning*. Jakarta: Fak. Ilmu Komputer, UI. 2003.

Shi, Zhixin and Govindaraju, Venu. *A Chaincode Based Scheme for Fingerprint Feature Extraction*. USA: State University of New York. 2005.

Sumardika, I. Wayan. *Perancangan dan Pembuatan Perangkat Lunak Identifikasi Sidik Jari dengan Metode Penelusuran Garis-Garis Papiliar*. Surabaya: Jurusan Teknik Informatika, Fak. Teknologi Industri, ITS. 1996.

Thai, Raymond. *Fingerprint Image Enhancement and Minutiae Extraction*. Australia: The University of Western Australia. 2003.

Lampiran 1

Sample Citra Sidik Jari

Orang ke-1 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-2 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-3 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-4 :



sample ke-1



sample ke-2



sample ke-3

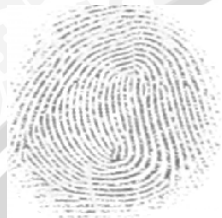


sample ke-4



sample ke-5

Orang ke-5 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-6 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-7 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-8 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-9 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-10 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-11 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-12 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-13 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-14 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Orang ke-15 :



sample ke-1



sample ke-2



sample ke-3



sample ke-4



sample ke-5

Lampiran 2

Hasil Proses Segmetasi
dengan Sepuluh Variasi Nilai *Tresholding* (T)1. Nilai $T = 10$ 

citra asli



citra tersegmentasi

2. Nilai $T = 25$ 

citra asli



citra tersegmentasi

3. Nilai $T = 35$ 

citra asli



citra tersegmentasi

4. Nilai $T = 50$



citra asli



citra tersegmentasi

5. Nilai $T = 75$



citra asli



citra tersegmentasi

6. Nilai $T = 90$



citra asli



citra tersegmentasi

7. Nilai $T = 150$



citra asli



citra tersegmentasi

8. Nilai $T = 175$



citra asli



citra tersegmentasi

9. Nilai $T = 190$



citra asli



citra tersegmentasi

10. Nilai $T = 250$



citra asli



citra tersegmentasi

UNIVERSITAS BRAWIJAYA



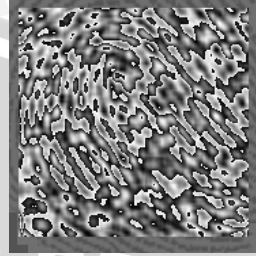
Lampiran 3

Hasil Proses Peningkatan Kualitas Citra dengan Sepuluh Kombinasi σ_1 dan σ_2

1. Kombinasi $\sigma_1 = 4,0$ dan $\sigma_2 = 4,0$



citra asli

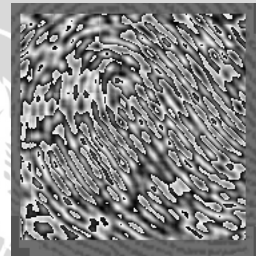


enhanced image

2. Kombinasi $\sigma_1 = 3,0$ dan $\sigma_2 = 4,0$



citra asli

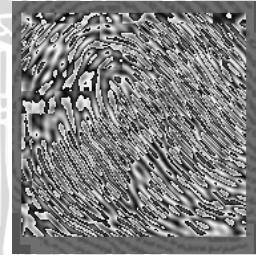


enhanced image

3. Kombinasi $\sigma_1 = 2,0$ dan $\sigma_2 = 4,0$



citra asli

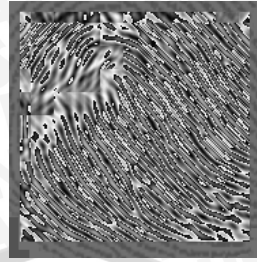


enhanced image

4. Kombinasi $\sigma_1 = 1,0$ dan $\sigma_2 = 4,0$



citra asli

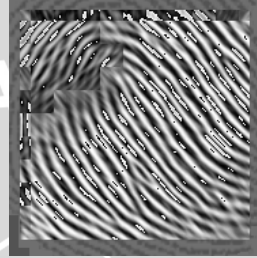


enhanced image

5. Kombinasi $\sigma_1 = 0,5$ dan $\sigma_2 = 4,0$



citra asli

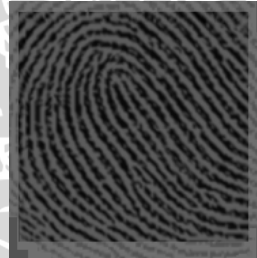


enhanced image

6. Kombinasi $\sigma_1 = 0,5$ dan $\sigma_2 = 1,0$



citra asli

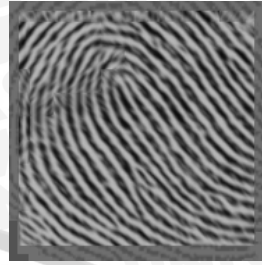


enhanced image

7. Kombinasi $\sigma_1 = 0,5$ dan $\sigma_2 = 2,0$



citra asli



enhanced image

8. Kombinasi $\sigma_1 = 0,5$ dan $\sigma_2 = 5,0$



citra asli

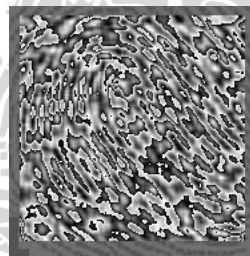


enhanced image

9. Kombinasi $\sigma_1 = 4,0$ dan $\sigma_2 = 5,0$



citra asli

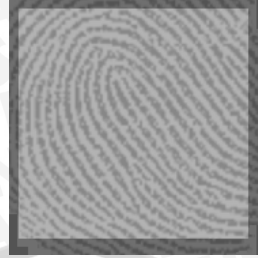


enhanced image

10. Kombinasi $\sigma_1 = 0,5$ dan $\sigma_2 = 0,5$



citra asli



enhanced image

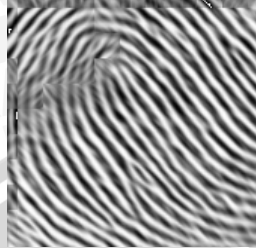
UNIVERSITAS BRAWIJAYA



Lampiran 4

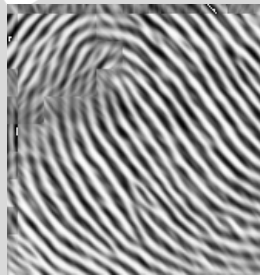
Hasil Proses Binerisasi dengan
Sepuluh Kombinasi T_1 dan T_2

1. Kombinasi $T_1 = 100$ dan $T_2 = 150$

*enhanced image*

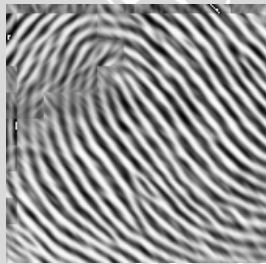
citra biner

2. Kombinasi $T_1 = 125$ dan $T_2 = 150$

*enhanced image*

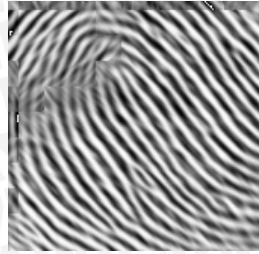
citra biner

3. Kombinasi $T_1 = 90$ dan $T_2 = 125$

*enhanced image*

citra biner

4. Kombinasi $T_1 = 130$ dan $T_2 = 150$

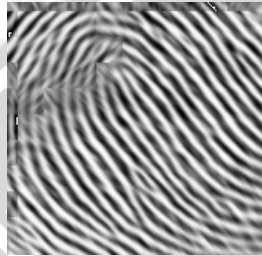


enhanced image



citra biner

5. Kombinasi $T_1 = 150$ dan $T_2 = 160$

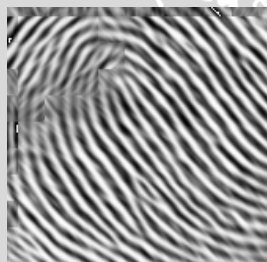


enhanced image



citra biner

6. Kombinasi $T_1 = 130$ dan $T_2 = 160$

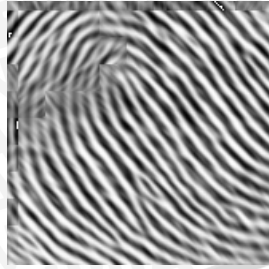


enhanced image



citra biner

7. Kombinasi $T_1 = 90$ dan $T_2 = 160$

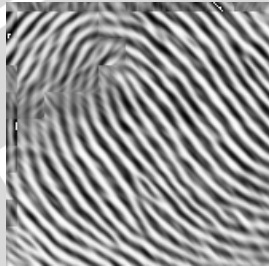


enhanced image



citra biner

8. Kombinasi $T_1 = 90$ dan $T_2 = 170$

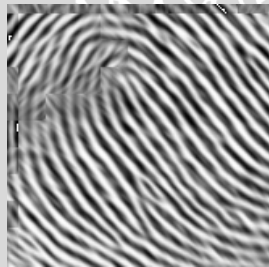


enhanced image



citra biner

9. Kombinasi $T_1 = 130$ dan $T_2 = 170$

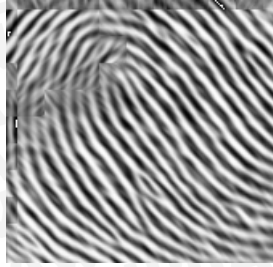


enhanced image



citra biner

10. Kombinasi $T_1 = 127$ dan $T_2 = 170$



enhanced image



citra biner

UNIVERSITAS BRAWIJAYA



Lampiran 5

Hasil Uji *Post Hoc Test* terhadap Perbandingan *Ridge Ending* dari Kelima belas Orang yang Diambil *Sample* Sidik Jarinya

Multiple Comparisons Dependent Variable: Ridge_Ending LSD						
(I) Orang	(J) Orang	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	4,60	21,970	,835	-39,35	48,55
	3	-27,60	21,970	,214	-71,55	16,35
	4	-5,60	21,970	,800	-49,55	38,35
	5	117,20(*)	21,970	,000	73,25	161,15
	6	-12,40	21,970	,575	-56,35	31,55
	7	110,40(*)	21,970	,000	66,45	154,35
	8	144,60(*)	21,970	,000	100,65	188,55
	9	182,80(*)	21,970	,000	138,85	226,75
	10	171,20(*)	21,970	,000	127,25	215,15
	11	21,20	21,970	,338	-22,75	65,15
	12	114,80(*)	21,970	,000	70,85	158,75
	13	140,20(*)	21,970	,000	96,25	184,15
	14	30,20	21,970	,174	-13,75	74,15
	15	74,00(*)	21,970	,001	30,05	117,95
	1	-4,60	21,970	,835	-48,55	39,35
2	3	-32,20	21,970	,148	-76,15	11,75
	4	-10,20	21,970	,644	-54,15	33,75
	5	112,60(*)	21,970	,000	68,65	156,55
	6	-17,00	21,970	,442	-60,95	26,95
	1	-4,60	21,970	,835	-48,55	39,35

3	7	105,80(*)	21,970	,000	61,85	149,75
	8	140,00(*)	21,970	,000	96,05	183,95
	9	178,20(*)	21,970	,000	134,25	222,15
	10	166,60(*)	21,970	,000	122,65	210,55
	11	16,60	21,970	,453	-27,35	60,55
	12	110,20(*)	21,970	,000	66,25	154,15
	13	135,60(*)	21,970	,000	91,65	179,55
	14	25,60	21,970	,249	-18,35	69,55
	15	69,40(*)	21,970	,002	25,45	113,35
	1	27,60	21,970	,214	-16,35	71,55
	2	32,20	21,970	,148	-11,75	76,15
	4	22,00	21,970	,321	-21,95	65,95
	5	144,80(*)	21,970	,000	100,85	188,75
	6	15,20	21,970	,492	-28,75	59,15
	7	138,00(*)	21,970	,000	94,05	181,95
4	8	172,20(*)	21,970	,000	128,25	216,15
	9	210,40(*)	21,970	,000	166,45	254,35
	10	198,80(*)	21,970	,000	154,85	242,75
	11	48,80(*)	21,970	,030	4,85	92,75
	12	142,40(*)	21,970	,000	98,45	186,35
	13	167,80(*)	21,970	,000	123,85	211,75
	14	57,80(*)	21,970	,011	13,85	101,75
4	15	101,60(*)	21,970	,000	57,65	145,55
	1	5,60	21,970	,800	-38,35	49,55
	2	10,20	21,970	,644	-33,75	54,15
	3	-22,00	21,970	,321	-65,95	21,95
	5	122,80(*)	21,970	,000	78,85	166,75
	6	-6,80	21,970	,758	-50,75	37,15
	7	116,00(*)	21,970	,000	72,05	159,95

5	8	150,20(*)	21,970	,000	106,25	194,15
	9	188,40(*)	21,970	,000	144,45	232,35
	10	176,80(*)	21,970	,000	132,85	220,75
	11	26,80	21,970	,227	-17,15	70,75
	12	120,40(*)	21,970	,000	76,45	164,35
	13	145,80(*)	21,970	,000	101,85	189,75
	14	35,80	21,970	,108	-8,15	79,75
	15	79,60(*)	21,970	,001	35,65	123,55
	1	-117,20(*)	21,970	,000	-161,15	-73,25
	2	-112,60(*)	21,970	,000	-156,55	-68,65
	3	-144,80(*)	21,970	,000	-188,75	-100,85
	4	-122,80(*)	21,970	,000	-166,75	-78,85
	6	-129,60(*)	21,970	,000	-173,55	-85,65
	7	-6,80	21,970	,758	-50,75	37,15
	8	27,40	21,970	,217	-16,55	71,35
6	9	65,60(*)	21,970	,004	21,65	109,55
	10	54,00(*)	21,970	,017	10,05	97,95
	11	-96,00(*)	21,970	,000	-139,95	-52,05
	12	-2,40	21,970	,913	-46,35	41,55
	13	23,00	21,970	,299	-20,95	66,95
	14	-87,00(*)	21,970	,000	-130,95	-43,05
	15	-43,20	21,970	,054	-87,15	,75
	1	12,40	21,970	,575	-31,55	56,35
6	2	17,00	21,970	,442	-26,95	60,95
	3	-15,20	21,970	,492	-59,15	28,75
	4	6,80	21,970	,758	-37,15	50,75
	5	129,60(*)	21,970	,000	85,65	173,55
	7	122,80(*)	21,970	,000	78,85	166,75
	8	157,00(*)	21,970	,000	113,05	200,95

7	9	195,20(*)	21,970	,000	151,25	239,15
	10	183,60(*)	21,970	,000	139,65	227,55
	11	33,60	21,970	,131	-10,35	77,55
	12	127,20(*)	21,970	,000	83,25	171,15
	13	152,60(*)	21,970	,000	108,65	196,55
	14	42,60	21,970	,057	-1,35	86,55
	15	86,40(*)	21,970	,000	42,45	130,35
	1	-110,40(*)	21,970	,000	-154,35	-66,45
	2	-105,80(*)	21,970	,000	-149,75	-61,85
	3	-138,00(*)	21,970	,000	-181,95	-94,05
	4	-116,00(*)	21,970	,000	-159,95	-72,05
	5	6,80	21,970	,758	-37,15	50,75
	6	-122,80(*)	21,970	,000	-166,75	-78,85
	8	34,20	21,970	,125	-9,75	78,15
	9	72,40(*)	21,970	,002	28,45	116,35
8	10	60,80(*)	21,970	,008	16,85	104,75
	11	-89,20(*)	21,970	,000	-133,15	-45,25
	12	4,40	21,970	,842	-39,55	48,35
	13	29,80	21,970	,180	-14,15	73,75
	14	-80,20(*)	21,970	,001	-124,15	-36,25
	15	-36,40	21,970	,103	-80,35	7,55
	1	-144,60(*)	21,970	,000	-188,55	-100,65
	2	-140,00(*)	21,970	,000	-183,95	-96,05
	3	-172,20(*)	21,970	,000	-216,15	-128,25
	4	-150,20(*)	21,970	,000	-194,15	-106,25
	5	-27,40	21,970	,217	-71,35	16,55
	6	-157,00(*)	21,970	,000	-200,95	-113,05
	7	-34,20	21,970	,125	-78,15	9,75
	9	38,20	21,970	,087	-5,75	82,15

	10	26,60	21,970	,231	-17,35	70,55
	11	-123,40(*)	21,970	,000	-167,35	-79,45
	12	-29,80	21,970	,180	-73,75	14,15
	13	-4,40	21,970	,842	-48,35	39,55
	14	-114,40(*)	21,970	,000	-158,35	-70,45
	15	-70,60(*)	21,970	,002	-114,55	-26,65
9	1	-182,80(*)	21,970	,000	-226,75	-138,85
	2	-178,20(*)	21,970	,000	-222,15	-134,25
	3	-210,40(*)	21,970	,000	-254,35	-166,45
	4	-188,40(*)	21,970	,000	-232,35	-144,45
	5	-65,60(*)	21,970	,004	-109,55	-21,65
	6	-195,20(*)	21,970	,000	-239,15	-151,25
	7	-72,40(*)	21,970	,002	-116,35	-28,45
	8	-38,20	21,970	,087	-82,15	5,75
	10	-11,60	21,970	,599	-55,55	32,35
	11	-161,60(*)	21,970	,000	-205,55	-117,65
	12	-68,00(*)	21,970	,003	-111,95	-24,05
	13	-42,60	21,970	,057	-86,55	1,35
	14	-152,60(*)	21,970	,000	-196,55	-108,65
	15	-108,80(*)	21,970	,000	-152,75	-64,85
10	1	-171,20(*)	21,970	,000	-215,15	-127,25
	2	-166,60(*)	21,970	,000	-210,55	-122,65
	3	-198,80(*)	21,970	,000	-242,75	-154,85
	4	-176,80(*)	21,970	,000	-220,75	-132,85
	5	-54,00(*)	21,970	,017	-97,95	-10,05
	6	-183,60(*)	21,970	,000	-227,55	-139,65
	7	-60,80(*)	21,970	,008	-104,75	-16,85
	8	-26,60	21,970	,231	-70,55	17,35
	9	11,60	21,970	,599	-32,35	55,55

	11	-150,00(*)	21,970	,000	-193,95	-106,05
	12	-56,40(*)	21,970	,013	-100,35	-12,45
	13	-31,00	21,970	,163	-74,95	12,95
	14	-141,00(*)	21,970	,000	-184,95	-97,05
	15	-97,20(*)	21,970	,000	-141,15	-53,25
11	1	-21,20	21,970	,338	-65,15	22,75
	2	-16,60	21,970	,453	-60,55	27,35
	3	-48,80(*)	21,970	,030	-92,75	-4,85
	4	-26,80	21,970	,227	-70,75	17,15
	5	96,00(*)	21,970	,000	52,05	139,95
	6	-33,60	21,970	,131	-77,55	10,35
	7	89,20(*)	21,970	,000	45,25	133,15
	8	123,40(*)	21,970	,000	79,45	167,35
	9	161,60(*)	21,970	,000	117,65	205,55
	10	150,00(*)	21,970	,000	106,05	193,95
	12	93,60(*)	21,970	,000	49,65	137,55
	13	119,00(*)	21,970	,000	75,05	162,95
	14	9,00	21,970	,684	-34,95	52,95
	15	52,80(*)	21,970	,019	8,85	96,75
12	1	-114,80(*)	21,970	,000	-158,75	-70,85
	2	-110,20(*)	21,970	,000	-154,15	-66,25
	3	-142,40(*)	21,970	,000	-186,35	-98,45
	4	-120,40(*)	21,970	,000	-164,35	-76,45
	5	2,40	21,970	,913	-41,55	46,35
	6	-127,20(*)	21,970	,000	-171,15	-83,25
	7	-4,40	21,970	,842	-48,35	39,55
	8	29,80	21,970	,180	-14,15	73,75
	9	68,00(*)	21,970	,003	24,05	111,95
	10	56,40(*)	21,970	,013	12,45	100,35

	11	-93,60(*)	21,970	,000	-137,55	-49,65
	13	25,40	21,970	,252	-18,55	69,35
	14	-84,60(*)	21,970	,000	-128,55	-40,65
	15	-40,80	21,970	,068	-84,75	3,15
13	1	-140,20(*)	21,970	,000	-184,15	-96,25
	2	-135,60(*)	21,970	,000	-179,55	-91,65
	3	-167,80(*)	21,970	,000	-211,75	-123,85
	4	-145,80(*)	21,970	,000	-189,75	-101,85
	5	-23,00	21,970	,299	-66,95	20,95
	6	-152,60(*)	21,970	,000	-196,55	-108,65
	7	-29,80	21,970	,180	-73,75	14,15
	8	4,40	21,970	,842	-39,55	48,35
	9	42,60	21,970	,057	-1,35	86,55
	10	31,00	21,970	,163	-12,95	74,95
	11	-119,00(*)	21,970	,000	-162,95	-75,05
	12	-25,40	21,970	,252	-69,35	18,55
	14	-110,00(*)	21,970	,000	-153,95	-66,05
	15	-66,20(*)	21,970	,004	-110,15	-22,25
14	1	-30,20	21,970	,174	-74,15	13,75
	2	-25,60	21,970	,249	-69,55	18,35
	3	-57,80(*)	21,970	,011	-101,75	-13,85
	4	-35,80	21,970	,108	-79,75	8,15
	5	87,00(*)	21,970	,000	43,05	130,95
	6	-42,60	21,970	,057	-86,55	1,35
	7	80,20(*)	21,970	,001	36,25	124,15
	8	114,40(*)	21,970	,000	70,45	158,35
	9	152,60(*)	21,970	,000	108,65	196,55
	10	141,00(*)	21,970	,000	97,05	184,95
	11	-9,00	21,970	,684	-52,95	34,95

15	12	84,60(*)	21,970	,000	40,65	128,55
	13	110,00(*)	21,970	,000	66,05	153,95
	15	43,80	21,970	,051	-,15	87,75
	1	-74,00(*)	21,970	,001	-117,95	-30,05
	2	-69,40(*)	21,970	,002	-113,35	-25,45
	3	-101,60(*)	21,970	,000	-145,55	-57,65
	4	-79,60(*)	21,970	,001	-123,55	-35,65
	5	43,20	21,970	,054	-,75	87,15
	6	-86,40(*)	21,970	,000	-130,35	-42,45
	7	36,40	21,970	,103	-7,55	80,35
	8	70,60(*)	21,970	,002	26,65	114,55
	9	108,80(*)	21,970	,000	64,85	152,75
	10	97,20(*)	21,970	,000	53,25	141,15
	11	-52,80(*)	21,970	,019	-96,75	-8,85
	12	40,80	21,970	,068	-3,15	84,75
	13	66,20(*)	21,970	,004	22,25	110,15
	14	-43,80	21,970	,051	-87,75	,15

Based on observed means.

* The mean difference is significant at the ,05 level.

Lampiran 6

Hasil Uji *Post Hoc* Test
terhadap Perbandingan *Ridge Bifurcation* dari
Kelima belas Orang yang Diambil *Sample* Sidik Jarinya

Multiple Comparisons Dependent Variable: Ridge_Bifurcation LSD						
(I) Orang	(J) Orang	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	16,00	20,731	,443	-25,47	57,47
	3	10,00	20,731	,631	-31,47	51,47
	4	6,80	20,731	,744	-34,67	48,27
	5	-78,60(*)	20,731	,000	-120,07	-37,13
	6	-5,00	20,731	,810	-46,47	36,47
	7	-60,20(*)	20,731	,005	-101,67	-18,73
	8	-23,40	20,731	,263	-64,87	18,07
	9	-63,60(*)	20,731	,003	-105,07	-22,13
	10	-3,40	20,731	,870	-44,87	38,07
	11	-180,00(*)	20,731	,000	-221,47	-138,53
	12	-331,40(*)	20,731	,000	-372,87	-289,93
	13	-76,80(*)	20,731	,000	-118,27	-35,33
	14	-75,60(*)	20,731	,001	-117,07	-34,13
	15	-55,40(*)	20,731	,010	-96,87	-13,93
	1	-16,00	20,731	,443	-57,47	25,47
2	3	-6,00	20,731	,773	-47,47	35,47
	4	-9,20	20,731	,659	-50,67	32,27
	5	-94,60(*)	20,731	,000	-136,07	-53,13
	6	-21,00	20,731	,315	-62,47	20,47
	1	-16,00	20,731	,443	-57,47	25,47

3	7	-76,20(*)	20,731	,001	-117,67	-34,73
	8	-39,40	20,731	,062	-80,87	2,07
	9	-79,60(*)	20,731	,000	-121,07	-38,13
	10	-19,40	20,731	,353	-60,87	22,07
	11	-196,00(*)	20,731	,000	-237,47	-154,53
	12	-347,40(*)	20,731	,000	-388,87	-305,93
	13	-92,80(*)	20,731	,000	-134,27	-51,33
	14	-91,60(*)	20,731	,000	-133,07	-50,13
	15	-71,40(*)	20,731	,001	-112,87	-29,93
	1	-10,00	20,731	,631	-51,47	31,47
	2	6,00	20,731	,773	-35,47	47,47
	4	-3,20	20,731	,878	-44,67	38,27
	5	-88,60(*)	20,731	,000	-130,07	-47,13
	6	-15,00	20,731	,472	-56,47	26,47
	7	-70,20(*)	20,731	,001	-111,67	-28,73
4	8	-33,40	20,731	,112	-74,87	8,07
	9	-73,60(*)	20,731	,001	-115,07	-32,13
	10	-13,40	20,731	,520	-54,87	28,07
	11	-190,00(*)	20,731	,000	-231,47	-148,53
	12	-341,40(*)	20,731	,000	-382,87	-299,93
	13	-86,80(*)	20,731	,000	-128,27	-45,33
	14	-85,60(*)	20,731	,000	-127,07	-44,13
	15	-65,40(*)	20,731	,003	-106,87	-23,93
	1	-6,80	20,731	,744	-48,27	34,67
	2	9,20	20,731	,659	-32,27	50,67
	3	3,20	20,731	,878	-38,27	44,67
	5	-85,40(*)	20,731	,000	-126,87	-43,93
	6	-11,80	20,731	,571	-53,27	29,67
	7	-67,00(*)	20,731	,002	-108,47	-25,53

	8	-30,20	20,731	,150	-71,67	11,27
	9	-70,40(*)	20,731	,001	-111,87	-28,93
	10	-10,20	20,731	,624	-51,67	31,27
	11	-186,80(*)	20,731	,000	-228,27	-145,33
	12	-338,20(*)	20,731	,000	-379,67	-296,73
	13	-83,60(*)	20,731	,000	-125,07	-42,13
	14	-82,40(*)	20,731	,000	-123,87	-40,93
	15	-62,20(*)	20,731	,004	-103,67	-20,73
5	1	78,60(*)	20,731	,000	37,13	120,07
	2	94,60(*)	20,731	,000	53,13	136,07
	3	88,60(*)	20,731	,000	47,13	130,07
	4	85,40(*)	20,731	,000	43,93	126,87
	6	73,60(*)	20,731	,001	32,13	115,07
	7	18,40	20,731	,378	-23,07	59,87
	8	55,20(*)	20,731	,010	13,73	96,67
	9	15,00	20,731	,472	-26,47	56,47
	10	75,20(*)	20,731	,001	33,73	116,67
	11	-101,40(*)	20,731	,000	-142,87	-59,93
	12	-252,80(*)	20,731	,000	-294,27	-211,33
	13	1,80	20,731	,931	-39,67	43,27
	14	3,00	20,731	,885	-38,47	44,47
	15	23,20	20,731	,268	-18,27	64,67
6	1	5,00	20,731	,810	-36,47	46,47
	2	21,00	20,731	,315	-20,47	62,47
	3	15,00	20,731	,472	-26,47	56,47
	4	11,80	20,731	,571	-29,67	53,27
	5	-73,60(*)	20,731	,001	-115,07	-32,13
	7	-55,20(*)	20,731	,010	-96,67	-13,73
	8	-18,40	20,731	,378	-59,87	23,07

7	9	-58,60(*)	20,731	,006	-100,07	-17,13
	10	1,60	20,731	,939	-39,87	43,07
	11	-175,00(*)	20,731	,000	-216,47	-133,53
	12	-326,40(*)	20,731	,000	-367,87	-284,93
	13	-71,80(*)	20,731	,001	-113,27	-30,33
	14	-70,60(*)	20,731	,001	-112,07	-29,13
	15	-50,40(*)	20,731	,018	-91,87	-8,93
	1	60,20(*)	20,731	,005	18,73	101,67
	2	76,20(*)	20,731	,001	34,73	117,67
	3	70,20(*)	20,731	,001	28,73	111,67
	4	67,00(*)	20,731	,002	25,53	108,47
	5	-18,40	20,731	,378	-59,87	23,07
	6	55,20(*)	20,731	,010	13,73	96,67
	8	36,80	20,731	,081	-4,67	78,27
	9	-3,40	20,731	,870	-44,87	38,07
8	10	56,80(*)	20,731	,008	15,33	98,27
	11	-119,80(*)	20,731	,000	-161,27	-78,33
	12	-271,20(*)	20,731	,000	-312,67	-229,73
	13	-16,60	20,731	,426	-58,07	24,87
	14	-15,40	20,731	,460	-56,87	26,07
	15	4,80	20,731	,818	-36,67	46,27
	1	23,40	20,731	,263	-18,07	64,87
	2	39,40	20,731	,062	-2,07	80,87
	3	33,40	20,731	,112	-8,07	74,87
	4	30,20	20,731	,150	-11,27	71,67
	5	-55,20(*)	20,731	,010	-96,67	-13,73
	6	18,40	20,731	,378	-23,07	59,87
	7	-36,80	20,731	,081	-78,27	4,67
	9	-40,20	20,731	,057	-81,67	1,27

	10	20,00	20,731	,339	-21,47	61,47
	11	-156,60(*)	20,731	,000	-198,07	-115,13
	12	-308,00(*)	20,731	,000	-349,47	-266,53
	13	-53,40(*)	20,731	,012	-94,87	-11,93
	14	-52,20(*)	20,731	,014	-93,67	-10,73
	15	-32,00	20,731	,128	-73,47	9,47
9	1	63,60(*)	20,731	,003	22,13	105,07
	2	79,60(*)	20,731	,000	38,13	121,07
	3	73,60(*)	20,731	,001	32,13	115,07
	4	70,40(*)	20,731	,001	28,93	111,87
	5	-15,00	20,731	,472	-56,47	26,47
	6	58,60(*)	20,731	,006	17,13	100,07
	7	3,40	20,731	,870	-38,07	44,87
	8	40,20	20,731	,057	-1,27	81,67
	10	60,20(*)	20,731	,005	18,73	101,67
	11	-116,40(*)	20,731	,000	-157,87	-74,93
	12	-267,80(*)	20,731	,000	-309,27	-226,33
	13	-13,20	20,731	,527	-54,67	28,27
	14	-12,00	20,731	,565	-53,47	29,47
	15	8,20	20,731	,694	-33,27	49,67
10	1	3,40	20,731	,870	-38,07	44,87
	2	19,40	20,731	,353	-22,07	60,87
	3	13,40	20,731	,520	-28,07	54,87
	4	10,20	20,731	,624	-31,27	51,67
	5	-75,20(*)	20,731	,001	-116,67	-33,73
	6	-1,60	20,731	,939	-43,07	39,87
	7	-56,80(*)	20,731	,008	-98,27	-15,33
	8	-20,00	20,731	,339	-61,47	21,47
	9	-60,20(*)	20,731	,005	-101,67	-18,73

	11	-176,60(*)	20,731	,000	-218,07	-135,13
	12	-328,00(*)	20,731	,000	-369,47	-286,53
	13	-73,40(*)	20,731	,001	-114,87	-31,93
	14	-72,20(*)	20,731	,001	-113,67	-30,73
	15	-52,00(*)	20,731	,015	-93,47	-10,53
11	1	180,00(*)	20,731	,000	138,53	221,47
	2	196,00(*)	20,731	,000	154,53	237,47
	3	190,00(*)	20,731	,000	148,53	231,47
	4	186,80(*)	20,731	,000	145,33	228,27
	5	101,40(*)	20,731	,000	59,93	142,87
	6	175,00(*)	20,731	,000	133,53	216,47
	7	119,80(*)	20,731	,000	78,33	161,27
	8	156,60(*)	20,731	,000	115,13	198,07
	9	116,40(*)	20,731	,000	74,93	157,87
	10	176,60(*)	20,731	,000	135,13	218,07
	12	-151,40(*)	20,731	,000	-192,87	-109,93
	13	103,20(*)	20,731	,000	61,73	144,67
	14	104,40(*)	20,731	,000	62,93	145,87
	15	124,60(*)	20,731	,000	83,13	166,07
12	1	331,40(*)	20,731	,000	289,93	372,87
	2	347,40(*)	20,731	,000	305,93	388,87
	3	341,40(*)	20,731	,000	299,93	382,87
	4	338,20(*)	20,731	,000	296,73	379,67
	5	252,80(*)	20,731	,000	211,33	294,27
	6	326,40(*)	20,731	,000	284,93	367,87
	7	271,20(*)	20,731	,000	229,73	312,67
	8	308,00(*)	20,731	,000	266,53	349,47
	9	267,80(*)	20,731	,000	226,33	309,27
	10	328,00(*)	20,731	,000	286,53	369,47

	11	151,40(*)	20,731	,000	109,93	192,87
	13	254,60(*)	20,731	,000	213,13	296,07
	14	255,80(*)	20,731	,000	214,33	297,27
	15	276,00(*)	20,731	,000	234,53	317,47
13	1	76,80(*)	20,731	,000	35,33	118,27
	2	92,80(*)	20,731	,000	51,33	134,27
	3	86,80(*)	20,731	,000	45,33	128,27
	4	83,60(*)	20,731	,000	42,13	125,07
	5	-1,80	20,731	,931	-43,27	39,67
	6	71,80(*)	20,731	,001	30,33	113,27
	7	16,60	20,731	,426	-24,87	58,07
	8	53,40(*)	20,731	,012	11,93	94,87
	9	13,20	20,731	,527	-28,27	54,67
	10	73,40(*)	20,731	,001	31,93	114,87
	11	-103,20(*)	20,731	,000	-144,67	-61,73
	12	-254,60(*)	20,731	,000	-296,07	-213,13
	14	1,20	20,731	,954	-40,27	42,67
	15	21,40	20,731	,306	-20,07	62,87
14	1	75,60(*)	20,731	,001	34,13	117,07
	2	91,60(*)	20,731	,000	50,13	133,07
	3	85,60(*)	20,731	,000	44,13	127,07
	4	82,40(*)	20,731	,000	40,93	123,87
	5	-3,00	20,731	,885	-44,47	38,47
	6	70,60(*)	20,731	,001	29,13	112,07
	7	15,40	20,731	,460	-26,07	56,87
	8	52,20(*)	20,731	,014	10,73	93,67
	9	12,00	20,731	,565	-29,47	53,47
	10	72,20(*)	20,731	,001	30,73	113,67
	11	-104,40(*)	20,731	,000	-145,87	-62,93

15	12	-255,80(*)	20,731	,000	-297,27	-214,33
	13	-1,20	20,731	,954	-42,67	40,27
	15	20,20	20,731	,334	-21,27	61,67
	1	55,40(*)	20,731	,010	13,93	96,87
	2	71,40(*)	20,731	,001	29,93	112,87
	3	65,40(*)	20,731	,003	23,93	106,87
	4	62,20(*)	20,731	,004	20,73	103,67
	5	-23,20	20,731	,268	-64,67	18,27
	6	50,40(*)	20,731	,018	8,93	91,87
	7	-4,80	20,731	,818	-46,27	36,67
	8	32,00	20,731	,128	-9,47	73,47
	9	-8,20	20,731	,694	-49,67	33,27
	10	52,00(*)	20,731	,015	10,53	93,47
	11	-124,60(*)	20,731	,000	-166,07	-83,13
	12	-276,00(*)	20,731	,000	-317,47	-234,53
	13	-21,40	20,731	,306	-62,87	20,07
	14	-20,20	20,731	,334	-61,67	21,27

Based on observed means.

* The mean difference is significant at the ,05 level.

Lampiran 7

Hasil Uji *Post Hoc Test* terhadap Perbandingan *Ridge Crossing* dari Kelima belas Orang yang Diambil *Sample* Sidik Jarinya

Multiple Comparisons Dependent Variable: Ridge_Crossing LSD						
(I) Orang	(J) Orang	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-,80	2,369	,737	-5,54	3,94
	3	2,40	2,369	,315	-2,34	7,14
	4	2,80	2,369	,242	-1,94	7,54
	5	-8,80(*)	2,369	,000	-13,54	-4,06
	6	1,80	2,369	,450	-2,94	6,54
	7	2,60	2,369	,277	-2,14	7,34
	8	2,40	2,369	,315	-2,34	7,14
	9	2,40	2,369	,315	-2,34	7,14
	10	3,80	2,369	,114	-,94	8,54
	11	-2,20	2,369	,357	-6,94	2,54
	12	-2,60	2,369	,277	-7,34	2,14
	13	2,00	2,369	,402	-2,74	6,74
	14	-1,00	2,369	,674	-5,74	3,74
	15	2,40	2,369	,315	-2,34	7,14
2	1	,80	2,369	,737	-3,94	5,54
	3	3,20	2,369	,182	-1,54	7,94
	4	3,60	2,369	,134	-1,14	8,34
	5	-8,00(*)	2,369	,001	-12,74	-3,26
	6	2,60	2,369	,277	-2,14	7,34

3	7	3,40	2,369	,156	-1,34	8,14
	8	3,20	2,369	,182	-1,54	7,94
	9	3,20	2,369	,182	-1,54	7,94
	10	4,60	2,369	,057	-,14	9,34
	11	-1,40	2,369	,557	-6,14	3,34
	12	-1,80	2,369	,450	-6,54	2,94
	13	2,80	2,369	,242	-1,94	7,54
	14	-,20	2,369	,933	-4,94	4,54
	15	3,20	2,369	,182	-1,54	7,94
	1	-2,40	2,369	,315	-7,14	2,34
	2	-3,20	2,369	,182	-7,94	1,54
	4	,40	2,369	,866	-4,34	5,14
	5	-11,20(*)	2,369	,000	-15,94	-6,46
	6	-,60	2,369	,801	-5,34	4,14
	7	,20	2,369	,933	-4,54	4,94
4	8	,00	2,369	1,000	-4,74	4,74
	9	,00	2,369	1,000	-4,74	4,74
	10	1,40	2,369	,557	-3,34	6,14
	11	-4,60	2,369	,057	-9,34	,14
	12	-5,00(*)	2,369	,039	-9,74	-,26
	13	-,40	2,369	,866	-5,14	4,34
	14	-3,40	2,369	,156	-8,14	1,34
4	15	,00	2,369	1,000	-4,74	4,74
	1	-2,80	2,369	,242	-7,54	1,94
	2	-3,60	2,369	,134	-8,34	1,14
	3	-,40	2,369	,866	-5,14	4,34
	5	-11,60(*)	2,369	,000	-16,34	-6,86
	6	-1,00	2,369	,674	-5,74	3,74
	7	-,20	2,369	,933	-4,94	4,54

5	8	- ,40	2,369	,866	-5,14	4,34
	9	- ,40	2,369	,866	-5,14	4,34
	10	1,00	2,369	,674	-3,74	5,74
	11	-5,00(*)	2,369	,039	-9,74	-,26
	12	-5,40(*)	2,369	,026	-10,14	-,66
	13	- ,80	2,369	,737	-5,54	3,94
	14	-3,80	2,369	,114	-8,54	,94
	15	- ,40	2,369	,866	-5,14	4,34
	1	8,80(*)	2,369	,000	4,06	13,54
	2	8,00(*)	2,369	,001	3,26	12,74
	3	11,20(*)	2,369	,000	6,46	15,94
	4	11,60(*)	2,369	,000	6,86	16,34
	6	10,60(*)	2,369	,000	5,86	15,34
	7	11,40(*)	2,369	,000	6,66	16,14
	8	11,20(*)	2,369	,000	6,46	15,94
6	9	11,20(*)	2,369	,000	6,46	15,94
	10	12,60(*)	2,369	,000	7,86	17,34
	11	6,60(*)	2,369	,007	1,86	11,34
	12	6,20(*)	2,369	,011	1,46	10,94
	13	10,80(*)	2,369	,000	6,06	15,54
	14	7,80(*)	2,369	,002	3,06	12,54
	15	11,20(*)	2,369	,000	6,46	15,94
	1	-1,80	2,369	,450	-6,54	2,94
6	2	-2,60	2,369	,277	-7,34	2,14
	3	,60	2,369	,801	-4,14	5,34
	4	1,00	2,369	,674	-3,74	5,74
	5	-10,60(*)	2,369	,000	-15,34	-5,86
	7	,80	2,369	,737	-3,94	5,54
	8	,60	2,369	,801	-4,14	5,34

7	9	,60	2,369	,801	-4,14	5,34
	10	2,00	2,369	,402	-2,74	6,74
	11	-4,00	2,369	,097	-8,74	,74
	12	-4,40	2,369	,068	-9,14	,34
	13	,20	2,369	,933	-4,54	4,94
	14	-2,80	2,369	,242	-7,54	1,94
	15	,60	2,369	,801	-4,14	5,34
	1	-2,60	2,369	,277	-7,34	2,14
	2	-3,40	2,369	,156	-8,14	1,34
	3	-,20	2,369	,933	-4,94	4,54
	4	,20	2,369	,933	-4,54	4,94
	5	-11,40(*)	2,369	,000	-16,14	-6,66
	6	-,80	2,369	,737	-5,54	3,94
	8	-,20	2,369	,933	-4,94	4,54
	9	-,20	2,369	,933	-4,94	4,54
8	10	1,20	2,369	,614	-3,54	5,94
	11	-4,80(*)	2,369	,047	-9,54	-,06
	12	-5,20(*)	2,369	,032	-9,94	-,46
	13	-,60	2,369	,801	-5,34	4,14
	14	-3,60	2,369	,134	-8,34	1,14
	15	-,20	2,369	,933	-4,94	4,54
	1	-2,40	2,369	,315	-7,14	2,34
	2	-3,20	2,369	,182	-7,94	1,54
	3	,00	2,369	1,000	-4,74	4,74
	4	,40	2,369	,866	-4,34	5,14
	5	-11,20(*)	2,369	,000	-15,94	-6,46
	6	-,60	2,369	,801	-5,34	4,14
	7	,20	2,369	,933	-4,54	4,94
	9	,00	2,369	1,000	-4,74	4,74

9	10	1,40	2,369	,557	-3,34	6,14
	11	-4,60	2,369	,057	-9,34	,14
	12	-5,00(*)	2,369	,039	-9,74	-,26
	13	-,40	2,369	,866	-5,14	4,34
	14	-3,40	2,369	,156	-8,14	1,34
	15	,00	2,369	1,000	-4,74	4,74
	1	-2,40	2,369	,315	-7,14	2,34
	2	-3,20	2,369	,182	-7,94	1,54
	3	,00	2,369	1,000	-4,74	4,74
	4	,40	2,369	,866	-4,34	5,14
	5	-11,20(*)	2,369	,000	-15,94	-6,46
	6	-,60	2,369	,801	-5,34	4,14
	7	,20	2,369	,933	-4,54	4,94
	8	,00	2,369	1,000	-4,74	4,74
	10	1,40	2,369	,557	-3,34	6,14
10	11	-4,60	2,369	,057	-9,34	,14
	12	-5,00(*)	2,369	,039	-9,74	-,26
	13	-,40	2,369	,866	-5,14	4,34
	14	-3,40	2,369	,156	-8,14	1,34
	15	,00	2,369	1,000	-4,74	4,74
	1	-3,80	2,369	,114	-8,54	,94
	2	-4,60	2,369	,057	-9,34	,14
	3	-1,40	2,369	,557	-6,14	3,34
	4	-1,00	2,369	,674	-5,74	3,74
	5	-12,60(*)	2,369	,000	-17,34	-7,86
	6	-2,00	2,369	,402	-6,74	2,74
	7	-1,20	2,369	,614	-5,94	3,54
	8	-1,40	2,369	,557	-6,14	3,34
	9	-1,40	2,369	,557	-6,14	3,34

	11	-6,00(*)	2,369	,014	-10,74	-1,26
	12	-6,40(*)	2,369	,009	-11,14	-1,66
	13	-1,80	2,369	,450	-6,54	2,94
	14	-4,80(*)	2,369	,047	-9,54	-,06
	15	-1,40	2,369	,557	-6,14	3,34
11	1	2,20	2,369	,357	-2,54	6,94
	2	1,40	2,369	,557	-3,34	6,14
	3	4,60	2,369	,057	-,14	9,34
	4	5,00(*)	2,369	,039	,26	9,74
	5	-6,60(*)	2,369	,007	-11,34	-1,86
	6	4,00	2,369	,097	-,74	8,74
	7	4,80(*)	2,369	,047	,06	9,54
	8	4,60	2,369	,057	-,14	9,34
	9	4,60	2,369	,057	-,14	9,34
	10	6,00(*)	2,369	,014	1,26	10,74
	12	-,40	2,369	,866	-5,14	4,34
	13	4,20	2,369	,081	-,54	8,94
	14	1,20	2,369	,614	-3,54	5,94
	15	4,60	2,369	,057	-,14	9,34
12	1	2,60	2,369	,277	-2,14	7,34
	2	1,80	2,369	,450	-2,94	6,54
	3	5,00(*)	2,369	,039	,26	9,74
	4	5,40(*)	2,369	,026	,66	10,14
	5	-6,20(*)	2,369	,011	-10,94	-1,46
	6	4,40	2,369	,068	-,34	9,14
	7	5,20(*)	2,369	,032	,46	9,94
	8	5,00(*)	2,369	,039	,26	9,74
	9	5,00(*)	2,369	,039	,26	9,74
	10	6,40(*)	2,369	,009	1,66	11,14

13	11	,40	2,369	,866	-4,34	5,14
	13	4,60	2,369	,057	-,14	9,34
	14	1,60	2,369	,502	-3,14	6,34
	15	5,00(*)	2,369	,039	,26	9,74
	1	-2,00	2,369	,402	-6,74	2,74
	2	-2,80	2,369	,242	-7,54	1,94
	3	,40	2,369	,866	-4,34	5,14
	4	,80	2,369	,737	-3,94	5,54
	5	-10,80(*)	2,369	,000	-15,54	-6,06
	6	-,20	2,369	,933	-4,94	4,54
	7	,60	2,369	,801	-4,14	5,34
	8	,40	2,369	,866	-4,34	5,14
	9	,40	2,369	,866	-4,34	5,14
	10	1,80	2,369	,450	-2,94	6,54
	11	-4,20	2,369	,081	-8,94	,54
14	12	-4,60	2,369	,057	-9,34	,14
	14	-3,00	2,369	,210	-7,74	1,74
	15	,40	2,369	,866	-4,34	5,14
	1	1,00	2,369	,674	-3,74	5,74
	2	,20	2,369	,933	-4,54	4,94
	3	3,40	2,369	,156	-1,34	8,14
	4	3,80	2,369	,114	-,94	8,54
	5	-7,80(*)	2,369	,002	-12,54	-3,06
	6	2,80	2,369	,242	-1,94	7,54
	7	3,60	2,369	,134	-1,14	8,34
	8	3,40	2,369	,156	-1,34	8,14
	9	3,40	2,369	,156	-1,34	8,14
	10	4,80(*)	2,369	,047	,06	9,54
	11	-1,20	2,369	,614	-5,94	3,54

15	12	-1,60	2,369	,502	-6,34	3,14
	13	3,00	2,369	,210	-1,74	7,74
	15	3,40	2,369	,156	-1,34	8,14
	1	-2,40	2,369	,315	-7,14	2,34
	2	-3,20	2,369	,182	-7,94	1,54
	3	,00	2,369	1,000	-4,74	4,74
	4	,40	2,369	,866	-4,34	5,14
	5	-11,20(*)	2,369	,000	-15,94	-6,46
	6	-,60	2,369	,801	-5,34	4,14
	7	,20	2,369	,933	-4,54	4,94
	8	,00	2,369	1,000	-4,74	4,74
	9	,00	2,369	1,000	-4,74	4,74
	10	-1,40	2,369	,557	-3,34	6,14
	11	-4,60	2,369	,057	-9,34	,14
	12	-5,00(*)	2,369	,039	-9,74	-,26
	13	-,40	2,369	,866	-5,14	4,34
	14	-3,40	2,369	,156	-8,14	1,34

Based on observed means.

* The mean difference is significant at the ,05 level.