

LEMBAR PENGESAHAN PEMBIMBING

**RANCANG BANGUN APLIKASI
PEMBUAT POLA STRIMIN DARI SUATU GAMBAR
MENGUNAKAN BORLAND DELPHI 7.0**

TUGAS AKHIR

oleh:

Nama : Totong Sutisno

No. Mahasiswa : 02 523 275

Yogyakarta, 18 September 2007

Pembimbing



Taufiq Hidayat, ST., MCS

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertanda tangan di bawah ini,

Nama : Totong Sutisno
NIM : 02 523 275

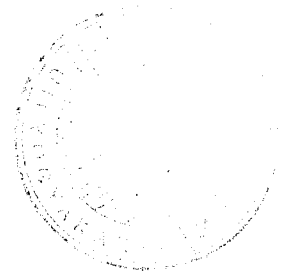
Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri.

Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 Oktober 2007


Totong Sutisno



LEMBAR PENGESAHAN PENGUJI

RANCANG BANGUN APLIKASI PEMBUAT POLA STRIMIN DARI SUATU GAMBAR MENGUNAKAN BORLAND DELPHI 7.0

TUGAS AKHIR

Oleh :

Nama : Totong Sutisno

No. Mahasiswa : 02 523 275

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

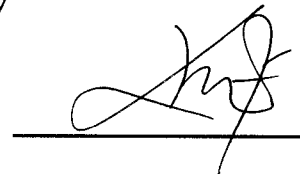
Yogyakarta, 30 Oktober 2007

Tim Penguji

Taufiq Hidayat, ST., MCS
Ketua

Yudi Prayudi, SSi., M.Kom
Anggota

Ami Fauziah, ST., MT
Anggota



Mengetahui,

Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia



Yudi Prayudi, SSi., M.Kom

Persembahan

Dengan Hati Yang Tulus dan Ikhlas Kupersembahkan setitik pencapaian awal dari masa depanku dengan penuh cinta dan kerendahan hati

Teruntuk:

*Allah SWT, rasa syukurku yang teramat hanya untuk Allah semata,
Semoga selalu memberikanku petunjuk pada jalan yang diridhai-Nya...*

*Sepasang Pelita Hidupku, Berkah yang luar biasa dari-Nya, Matahari dan Rembulan
Hidupku, Bapak dan Emakku tercinta yang selalu dan selalu aku susahkan, terima kasih atas
segalanya, doa, didikan serta kasih sayang yang tak terhingga laksana udara yang takkan
pernah habis aku hirup. Restumu untukku ialah cahaya-Nya nan abadi yang kan selalu
menerangi setiap jalan hidupku. Semoga pahala yang berlipat yang mereka dapatkan karena
ketidakmungkinanku untuk membalasnya untuk selamanya...*

*Serta Kakak-kakaku
yang telah banyak membantu dan memberikan doa,
motivasi dan dukungan yang tidak akan pernah terlupakan.....
terima kasih atas segalanya, doa, didikan serta kasih sayang yang tak terhingga.*

*Adikku yang kusayang,
semoga apa yang kamu harapkan dan cita-citakan dapat terwujud dengan doa dan ikhtiarmu
Lebih giat lagi belajar dan persembahkanlah yang terbaik buat Bapak dan Emak,
Buat keponakanku Afida dan ferdy yang selalu menghiasi hari-hariku dengan senyuman dan
selalu membangkitkan semangat di saat –saat rasa jenuh dan sulit untuk berkonsentrasi
mulai menyerang.*

*Almamaterku Universitas Islam Indonesia, terima kasih untuk segala sesuatunya
Buat teman-teman semua terima kasih dorongan semangat dan dukungannya, terutama untuk
seluruh teman-teman informatika VII 'VoIP 02' tak terkecuali, juga buat semua teman-
teman lain yang tak mungkin aku lupakan dan aku sebutkan satu-persatu.
'aku orang yang paling beruntung bisa mengenal kalian'. Untuk pak sabar, mas keliq, aa' n
te"hi burjo, 'terimakasih udah menyediakan makanan hangat disela dingin malam dan
menemani malam-malam sepiku'. Semua sahabat, kawan, sobat, pren, mitra, dan relasiku..
Serta buat semua oarng yang mengenalku yang tidak dapat disebutkan satu per satu, saya
ucapkan banyak terimakasih.*

*Pada suatu saat di dunia,
Dimana seseorang berdoa sepertimu.
Menjaga dan merindukan dirimu...
Sewaktu yang kau lakukan sewaktu sholat...
Akankah kau tahu..*

MOTTO

"Bacalah, dengan nama Tuhanmu Dzat yang menciptakan".

(Q.S. Al 'Alaq: 153)

"...Jadilah sabar dan sholat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar".

(Q.S. Al Baqarah ayat 153)

"Apa yang menimpamu berupa kebaikan, itulah yang datang dari Allah dan apa yang menimpamu berupa kejahatan (keburukan) itu datang dari dirimu sendiri..."

(Q.S. An-Nisa : 79)

"... Bertolong-tolonglah kamu dalam kebaikan dan dalam melaksanakan takwa, jangan kamu tolong-menolong dalam dosa dan permusuhan ..."

(QS. AL-Maaidah : 2)

"... Allah akan meninggikan orang-orang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat ..."

(QS. Al-Mujaadilah : 11)

"Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain".

(Q.S. AL-Insyirah : 6 dan 7)

"Jika kamu menginginkan dunia, raihlah dengan ilmu. Jika kamu menginginkan akhirat, raihlah dengan ilmu. Dan jika kamu menginginkan keduanya, maka raihlah semua itu dengan ilmu"

(Rosulullah Muhammad SAW)

Dimana langit dijunjung, disitu bumi di pijak, katakanlah hitam itu hitam dan putih itu putih. Berusahalah dengan seluruh kemampuanmu untuk menggapai semua impianmu dan imbangi semua itu dengan kesabaran, doa yang khusyu' dan tawadhu', serta dekatkan dirimu dengan Allah SWT karena jika Allah SWT dekat denganmu, maka Dia akan selalu membantumu dan selalu mendampingimu

(Sepasang Pelita Hidupku, Matahari dan Rembulan Hidupku, Bapak dan Emak Tercinta)

"Ada seribu satu cara untuk gagal tapi cuma ada satu cara untuk sukses dan berhasil, cobalah sampai menemukan satu cara tersebut, janganlah bosan".

(Totong Sutisno)

"Dunia hanya berjalan tiga hari, yaitu : Kemarin, yang kita tidak berpengharapan apa-apa lagi darinya. Hari ini, yang harus kita peroleh kebaikan dan kesuksesannya. Dan esok hari, yang tidak kita ketahui apakah kita termasuk yang masih hidup atau yang tergolong sudah meninggal".

(Al Hasan Al Bashri)

"Berkata dengan perbuatan lebih baik daripada berbuat dengan perkataan"

(Aa' Gym)

"Jika seseorang terpanggil menjadi tukang sapu jalan, hendaklah ia menyapu jalan sebagaimana Michael Angelo melukis atau Beethoven menciptakan musik atau Shakespeare menulis puisi. Hendaklah ia menyapu jalan dengan sangat baik sehingga segenap isi surga dan bumi serentak, menghentikan kegiatan mereka dan berkata, disini pernah tinggal seorang penyapu jalan yang agung yang menjalankan tugasnya dengan sangat baik".

(Martin Luther King)

"Kemenangan terbesar adalah melawan kekerasan tanpa kekerasan"

(Mahatma Gandhi)

"Kita Menang Saat Kita Berjuang apa yang kita anggap benar "

(Alfred Batman & Robin)

Kata Pengantar



السلام عليكم ورحمة الله وبركاته

Dengan Mengucapkan Alhamdulillah, segala puji dan syukur kami panjatkan kehadiran Allah SWT, atas berkah, rahmat, hidayah dan ridhoNya. Sesungguhnya segala puji dan ampunan hanya kepada Allah kita panjatkan dan kepada penutup para Nabi shallallahu 'alaihi shalatuwas salam kita bershalawat. Sehingga penulisan laporan tugas akhir yang berjudul : **Rancang Bangun Aplikasi Pembuat Pola Strimin Dari Suatu Gambar Menggunakan Borland Delphi 7.0** dapat penulis selesaikan dengan baik.

Semoga sholawat serta salam dilimpahkan kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, dan pengikut-pengikut beliau sampai hari kiamat nanti (amin).

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia Yogyakarta dan juga untuk dapat dipelajari dan dievaluasi sebagai bahan masukan pada pihak universitas atas apa yang telah diajarkan selama di perkuliahan baik secara teori maupun praktek. Disamping itu laporan ini juga dapat digunakan sebagai dokumentasi hasil penelitian, di samping laporan itu sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Dalam pembuatan tugas akhir ini, penulis mendapat banyak bimbingan, pengarahan serta bantuan baik secara materiil maupun non materiil yang sangat bermanfaat dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Bapak dan Emak di rumah yang telah membesarkan saya dengan ikhlas sehingga bisa menjadi orang yang terpelajar dan atas dukungan, doa dan restu, kepercayaan dan dorongan baik materiil maupun spiritual serta segala hal yang telah diberikan kepada saya selama ini.
2. Kakakku Adi Sujono beserta keluarga dan mbakku Eli Yulianti beserta keluarga terima kasih atas dukungan do'a, motivasi, dan dorongan baik materiil maupun spiritual serta kasih sayangnya. *I love you...*
3. Adikku serta keponakan-keponakanku yang selalu membuat kangen rumah dan tersenyum disela kepenatan hari-hariku
4. Guru-guru SDN Tegalsari III yang membimbingku mengenal huruf dan angka
5. Bapak Prof. Dr. H. Edy Suandi Hamid, M.Ec selaku Rektor Universitas Islam Indonesia.
6. Bapak Fathul Wahid, ST, M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Yudi Prayudi, S.Si., M.Kom. selaku Ketua Jurusan Teknik Informatika Terima kasih atas kemudahan dan dukungan yang telah diberikan.
8. Bapak Taufiq Hidayat, ST., M.Cs selaku dosen pembimbing tugas akhir Terimakasih atas segala bantuan, dukungan, pengetahuannya, kemudahan yang diberikan, kritik, saran serta kesabaran beliau dalam membimbing.
9. Bapak Yudi Prayudi, S.Si., M.Kom dan Ibu Ami Fauziah, ST., MT. selaku dosen penguji pendadaran yang sangat menyenangkan.
10. Mas M. Erwin AH, ST., MT terima kasih banyak untuk semuanya dan semoga sukses di Jakarta dan juga cepat mendapat momongan ya mas
11. Misbahul Munir, Amd. Terima kasih banyak untuk semua informasinya Abah

12. Dosen-dosen Jurusan Teknik Informatika terima kasih atas semua ilmu pengetahuan dan motivasi serta bantuannya.
13. Teman-teman Voip '02 khususnya dan informatika pada umumnya. Apapun yang terjadi tetap SATU untuk semua dan semua untuk SATU !!!!!
14. Teman-teman kos Wisma Tenang (Agus, Heri, Melky, Adit, Firly, Togel dan Wahyu Bandem) dan squad lama yang tidak bisa saya sebutkan satu per satu. Terima kasih atas seluruh persahabatan sejati, pengalaman yang tak terlupakan, kesedihan, kegembiraan dan keceriaan bersama serta support, informasi dan bantuan pinjaman komputernya selama komputerku rusak.
15. Bapak Sutar dan Ibu yang telah mengizinkan saya telat bayar kost.
16. Mas Anca, Sunarmaya, Ridho, Arie Jax, Yopie terima kasih banyak sudah bersedia meminjamkan laptop
17. Mas Handoko terima kasih sudah mengenalkan dunia kerja yang sebenarnya
18. Teman-teman SMANDA dan teman-teman yang ada di Cirebon, Jogja, Jakarta, Riau dan Palembang yang sudah memberikan bantuan dan dukungan.
19. Cak Nun terima kasih wejangannya, Jend. Purn. Wiranto terima kasih senandungnya, Bpk Matt Deris terima kasih sudah membuka mata saya untuk melihat dunia pendidikan di luar negeri
20. Bang Uli, Mas Fauzi, Mas Gesa, Aa' Dani, Radja, Gus One, Bebek terima kasih sudah membuat hari-hari penuh tawa dan gembira
21. Mas Ari terima kasih banyak sudah mau menjawab pertanyaan-pertanyaan saya, Bpk Balza terima kasih telah bersedia menjawab imel dari saya
22. Bidadari kecilku, Seseorang yang selalu menyayangi dan selalu mengerti aku semoga apa yang dicita-citakan dapat terwujud, terima kasih selalu membuat hidup ini terasa lebih indah.
23. Sahabat-sahabatku Tetaplah menjadi pelita yang selalu memberi senyuman, tawa, kebahagiaan serta keceriaan. *Thank's for your friendship...*
24. Ir. Soekarno yang telah membawa udara kemerdekaan di tanah indonesia sehingga saya bisa hidup dan belajar di alam kemerdekaan
25. Untuk Indonesiaku, Aku tak akan pernah melupakan wanginya tanah dan segarnya airmu yang tiap hari aku minum.

26. Teman-teman Informatika semua angkatan dan terutama angkatan 2002.
27. Doni, Phili, Ridho, terima kasih pinjaman buku delphinya. Ririn, agung terima kasih support dan tips pendadarannya, Andikay terima kasih kiriman imelnya.
28. Semua pihak yang telah memberikan bantuan dan dorongan yang tidak bisa penulis sebutkan satu persatu.

Semoga Allah SWT melimpahkan banyak nikmat, pahala dan hidayahNya kepada semua pihak yang membantu dan atas apa yang telah mereka berikan dengan keikhlasan.

Penulis menyadari tidak ada yang sempurna di dunia ini dan penulis sebagai manusia pasti tidak akan terlepas dari kesalahan, karena keterbatasan pengetahuan dan pangalaman, namun penulis berusaha untuk mengerjakan laporan tugas akhir ini dengan sebaik-baiknya. Penulis menyampaikan permohonan maaf sebelumnya serta mengharapkan masukan berupa kritik dan saran yang sifatnya membangun untuk memperbaiki dan meningkatkan serta penyempurnaan di masa mendatang.

Akhir kata semoga laporan tugas akhir ini dapat bermanfaat bagi pembaca dan semua orang. Semoga dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya dan semoga Allah SWT. Selalu menyertai setiap langkah kita, amiin.

والسلام عليكم ورحمة الله وبركاته

Yogyakarta, 30 Oktober 2007

(Totong Sutisno)

SARI

Seiring berjalannya waktu, keinginan dan daya imajinasi manusia yang semakin kompleks, keinginan para pengrajin untuk menuangkan gambar yang diinginkan ke dalam kain strimin semakin beraneka ragam. Sedangkan motif yang tersedia di buku motif hanya terbatas pada gambar bunga, taman, binatang peliharaan, pemandangan alam terbuka, bangunan sampai wajah tokoh idola. Bagi mereka yang menginginkan gambar yang dibuat di atas kain strimin adalah gambar yang tidak ada di dalam buku motif, akan sangat sulit sekali menuangkan gambar yang diinginkan ataupun gambar imajinasi yang ada di dalam pikiran ke atas kain strimin. Kalaupun bisa akan timbul permasalahan penskalaan dan motif yang dibuat tidak proporsional. Baik dari segi pewarnaan, ukuran, dan letak detail suatu motif.

Dengan adanya aplikasi pembuat pola strimin dari suatu gambar menggunakan Borland Delphi 7.0 ini dapat mempermudah dalam membuat dan menuangkan gambar di atas kain strimin dari pola-pola yang diinginkan sehingga kerajinan tangan sulam strimin menjadi sangat mudah, menarik dan menyenangkan. Dan bagi mereka yang ingin serius di bidang ini akan membantu mereka dalam menciptakan lapangan usaha baru yang sekaligus dapat menambah penghasilan keluarga sehingga seni kerajinan tangan sulam strimin bukan merupakan kegiatan yang 'kosong' semata.

Kata kunci : Strimin, motif, *user*.

TAKARIR

<i>Aspect ratio</i>	Aspek rasio
<i>Coding</i>	Kode bahasa pemrograman
<i>Data Flow Diagram (DFD)</i>	Diagram aliran data
<i>Divergensi</i>	Penyebaran
<i>File</i>	Arsip
<i>Flowchart</i>	Diagram alir
<i>Form</i>	Formulir
<i>Full color</i>	Warna sesuai aslinya, dalam komputer didefinisikan sekitar 16 juta warna
<i>Hardware</i>	Perangkat keras
<i>Height</i>	Tinggi
<i>Image</i>	Gambar
<i>Image processing</i>	Pengolahan citra
<i>Input</i>	Masukan
<i>Independent</i>	Merdeka atau berdiri sendiri
<i>Intensity</i>	Intensitas
<i>Interface</i>	Antarmuka
<i>Nearest neighbour</i>	Interpolasi tetangga terdekat
<i>Output</i>	Keluaran
<i>Pixel</i>	Satuan panjang dalam komputer
<i>Software</i>	Perangkat lunak
<i>Strimin</i>	Jenis kain yang berlubang
<i>True color</i>	Warna sesuai aslinya
<i>User friendly</i>	Mudah digunakan pengguna
<i>User</i>	Pengguna
<i>Width</i>	Lebar

DAFTAR ISI

JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	iii
LEMBAR PENGESAHAN PENGUJI	iv
HALAMAN PERSEMBAHAN.....	v
MOTTO.....	vii
KATA PENGANTAR.....	ix
SARI.....	xiii
TAKARIR	xiv
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	4
1.6.1 Metode Pengumpulan Data	4
1.6.2 Metode Pengembangan Perangkat Lunak	4
1.7 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI	8
2.1 Strimin.....	8
2.2 Format Citra	8
2.2.1 Komponen Citra Digital	8
2.2.2 Citra Biner (Monokrom)	9
2.2.3 Citra Skala keabuan (gray scale).....	10
2.2.4 Citra Warna (True color).....	10

2.3	Pengolahan Citra	11
2.4	Operasi Titik.....	12
2.4.1	Peningkatan Kontras	13
2.4.2	Konversi Citra True Color ke Keabuan.....	14
2.5	Operasi Global.....	15
2.5.1	Ekualisasi Histogram.....	15
2.6	Operasi Geometri	16
2.6.1	Interpolasi Tetangga Terdekat.....	16
2.6.2	Operasi Penskalaan (Scaling).....	17
2.7	Operasi Bertetangga	17
2.8	Deteksi Tepi	20
2.8.1	Deteksi Tepi Menggunakan Operator Gradien	20
2.9	Warna	23
2.9.1	Atribut Warna.....	24
2.9.2	Sistem Koordinat Warna	25
2.9.3	Model warna CMY dan CMYK.....	26
2.9.4	Transformasi Warna RGB ke IHS	27
BAB III METODOLOGI		30
3.1	Metode Analisis.....	30
3.2	Hasil Analisis	30
3.1.1	Masukan Sistem	31
3.2.2	Kebutuhan Proses.....	31
3.2.3	Keluaran Sistem	31
3.2.4	Kebutuhan Perangkat Keras	31
3.2.5	Kebutuhan Perangkat Lunak	32
3.2.6	Antarmuka Sistem	32
3.2.7	Kinerja yang Diharapkan	32
3.3	Metode Perancangan	33
3.4	Hasil Perancangan.....	33
BAB IV PERANCANGAN PERANGKAT LUNAK		34
4.1	Metode Perancangan Sistem	34

4.2	Hasil Perancangan	34
4.2.1	DFD (<i>Data Flow Diagram</i>).....	34
4.2.2	Diagram Korteks (<i>Cortext Diagram</i>)	35
4.2.3	Perancangan Diagram Alir Sistem	35
4.2.4	Flowchart konversi format gambar	39
4.2.5	Flowchart peningkatan kontras gambar	41
4.2.6	Flowchart pengaturan ukuran gambar pola strimin.....	45
4.2.7	Flowchart pengaturan ukuran gambar pada interface	49
4.2.8	Flowchart menggambar grid	53
4.2.9	Flowchart prosedur cetak hasil.....	55
4.3	Rancangan Antarmuka	57
4.3.1	Perancangan Antarmuka Splash awal	57
4.3.2	Perancangan Antarmuka Halaman Depan.....	58
4.3.3	Perancangan Antarmuka konfirmasi keluar dari program	59
4.3.4	Perancangan Antarmuka menu utama program	59
4.3.5	Perancangan Antarmuka form utama program	60
4.3.6	Perancangan Antarmuka informasi kesalahan proses	63
4.3.7	Perancangan Antarmuka informasi kesalahan input nilai kontras	64
4.3.8	Perancangan Antarmuka informasi pembatalan.....	64
4.3.9	Perancangan Antarmuka konfirmasi kembali	65
4.3.10	Perancangan Antarmuka Halaman Depan.....	65
4.4	Rancangan Struktur Data	66
BAB V IMPLEMENTASI PERANGKAT LUNAK.....		71
5.1	Batasan Implementasi	71
5.2	Implementasi Perangkat Lunak	72
5.2.1	Pengembangan Antarmuka Perangkat Lunak	72
4.4.1.1	Form Splash Screen Awal	72
4.4.1.2	Form Halaman Depan	73
4.4.1.3	Form Utama.....	74
4.4.1.4	Form Splash Screen Keluar.....	83
5.2.2	Implementasi Prosedural	84

5.2.2.1	Prosedur Buka	84
5.2.2.2	Prosedur Tingkat Kontras.....	85
5.2.2.3	Prosedur Peningkatan Kontras	85
5.2.2.4	Prosedur pengaturan ukuran gambar pada strimin	87
5.2.2.5	Prosedur pengaturan ukuran gambar pada antarmuka	87
5.2.2.6	Prosedur Pembuatan Grid.....	87
5.2.2.7	Prosedur Cetak	88
5.2.2.8	Prosedur Contoh Warna	89
BAB VI ANALISIS KINERJA PERANGKAT LUNAK		90
5.1	Pengujian Sistem	90
5.1.1	Pengujian Data Normal	90
5.1.1.1	Pengujian Input Gambar.....	91
5.1.1.2	Pengujian Dengan Input Gambar .bmp	91
5.1.1.3	Pengujian Dengan Input Gambar .jpg	92
5.1.1.4	Pengujian Dengan Input Gambar .wmf.....	93
5.1.1.5	Pengujian Penyimpanan Gambar	93
5.1.1.6	Pengujian Mencetak Gmbar	94
5.1.2	Pengujian Data Tidak Normal (Proses yang tidak berurutan).....	95
5.1.2.1	Tingkat Kontras.....	95
5.2	Analisis Hasil Pengujian Sistem	95
6.2.1	Pengujian Dengan Data Normal.....	95
6.2.2	Pengujian Dengan Data Tidak Normal	96
BAB VII SIMPULAN DAN SARAN		97
7.1	Kesimpulan.....	97
7.2	Saran.....	97
DAFTAR PUSTAKA		98

DAFTAR GAMBAR

Gambar 2.1 Hubungan Antara Koordinat Pada Citra dan Index Larik.....	9
Gambar 2.2 Citra Biner dan Representasinya Dalam Data Digital.....	10
Gambar 2.3 Citra Skala Keabuan 4 bit dan Representasinya Dalam Data Digital	10
Gambar 2.4 Citra True Color dan Representasinya Dalam Data Digital.....	11
Gambar 2.5 Pengolahan Citra.....	12
Gambar 2.6 Fungsi Transformasi Skala Keabuan (GST).....	13
Gambar 2.7 Fungsi GST untuk Peningkatan Kontras Citra.....	14
Gambar 2.8 Korespondensi Satu-satu Antara Obyek dan Citra Ideal.....	18
Gambar 2.9 Hubungan Antara Titik pada Obyek dan Citra yang Aktua.....	18
Gambar 2.10 Operator Gradien Untuk Deteksi Tepi.....	22
Gambar 2.11 Model IHS.....	25
Gambar 2.12 Segitiga IHS.....	28
Gambar 4.1 Diagram Korteks (DFD Level 0).....	35
Gambar 4.2 Flowchart Aplikasi Pembuat Pola Strimin.....	37
Gambar 4.3 Flowchart Konversi Format Gambar.....	40
Gambar 4.4 Flowchart Peningkatan Kontras Gambar.....	42
Gambar 4.5 Flowchart Pengaturan Ukuran Gambar Pola Strimin.....	46
Gambar 4.6 Flowchart Proses Pengaturan Ukuran Gambar Pada Interface.....	50
Gambar 4.7 Flowchart Menggambar Grid.....	53
Gambar 4.8 Flowchart Mencetak Gambar Pola Strimin.....	56
Gambar 4.9 Antarmuka Splash Screen Program.....	58
Gambar 4.10 Antarmuka Halaman Depan.....	58
Gambar 4.11 Antarmuka Konfirmasi Keluar Dari Program.....	59
Gambar 4.12 Antarmuka Menu Utama Program.....	60
Gambar 4.13 Antarmuka Form Utama Program.....	62
Gambar 4.14 Antarmuka Gambar Pola Strimin.....	63
Gambar 4.15 Antarmuka Informasi Kesalahan Proses.....	63
Gambar 4.16 Antarmuka Informasi Kesalahan Input Nilai Kontras.....	64

Gambar 4.17 Antarmuka Konfirmasi Pembatalan	64
Gambar 4.18 Antarmuka Konfirmasi Kembali.....	65
Gambar 4.19 Antarmuka Splash Screen Keluar Dari Program.....	65
Gambar 5.1 Form Splash Screen Awal	72
Gambar 5.2 Form Halaman Depan.....	73
Gambar 5.3 Form Utama.....	74
Gambar 5.4 Menu Buka Berkas.....	75
Gambar 5.5 Form Buka Berkas.....	75
Gambar 5.6 Menu Simpan Berkas.....	76
Gambar 5.7 Form Simpan Berkas.....	76
Gambar 5.8 Pola Strimin Gambar	77
Gambar 5.9 Menu Cetak Hasil.....	77
Gambar 5.10 Form Cetak Hasil.....	78
Gambar 5.11 Menu Keluar Program.....	78
Gambar 5.12 Form Konfirmasi Keluar Program.....	78
Gambar 5.13 Menu About.....	79
Gambar 5.14 Form About.....	80
Gambar 5.15 Menu Help	80
Gambar 5.16 Form Hasil Program.....	81
Gambar 5.17 Form Konfirmasi Pembatalan.....	82
Gambar 5.18 Form Konfirmasi Kembali.....	82
Gambar 5.19 Tombol Keluar.....	83
Gambar 5.20 Form Splash Screen Keluar.....	83
Gambar 6.1 Pembatasan File Gambar yang Dapat Diinputkan.....	91
Gambar 6.2 Pengujian Menggunakan File Gambar Dengan Format .bmp.....	92
Gambar 6.3 Pengujian Menggunakan File Gambar Dengan Format .jpg.....	92
Gambar 6.4 Pengujian Menggunakan File Gambar Dengan Format .wmf.....	93
Gambar 6.5 Penyimpanan Gambar Pola Strimin.....	93
Gambar 6.6 Pencetakan Gambar Pola Strimin.....	94
Gambar 6.7 Kesalahan Tingkat Kontras.....	95

BAB I

PENDAHULUAN

1.1 Latar Belakang

Membuat kerajinan tangan sulam strimin merupakan suatu hobi yang mengasyikkan dan salah satu kegiatan yang dilakukan untuk mengisi waktu luang bagi mayoritas kalangan ibu rumah tangga dan remaja putri. Motif polanya pun bermacam-macam mulai dari gambar bunga, taman, binatang peliharaan, pemandangan alam terbuka, bangunan sampai wajah tokoh idola yang membutuhkan ketelitian dan detail yang tinggi. Cara pengerjaannya pun sangat mudah, hanya dengan mengikuti panduan dari gambar yang sudah disediakan oleh buku panduan dan motif strimin.

Hanya dengan menyulam mengikuti motif dan ketentuan warna yang telah ada dalam gambar di dalam buku motif maka akan menjadi sebuah lukisan suatu gambar sesuai motif dalam buku motif ke atas kain strimin. Adapun bagi mereka yang telah mahir dapat mengubah skala gambar motif yang ada di dalam buku motif sesuai dengan keinginan.

Seiring berjalannya waktu, keinginan dan daya imajinasi manusia yang semakin kompleks, maka timbul masalah di dalam seni kerajinan tangan sulam strimin. Diantaranya keinginan para pengrajin untuk menuangkan gambar yang diinginkan ke dalam kain strimin semakin beraneka ragam. Sedangkan motif yang tersedia di buku motif hanya terbatas pada gambar bunga, taman, binatang

peliharaan, pemandangan alam terbuka, bangunan sampai wajah tokoh idola. Timbulah masalah bagi mereka yang menginginkan gambar yang dibuat di atas kain strimin adalah gambar foto diri, foto keluarga, foto bintang idola tertentu, gambar kaligrafi, gambar pemandangan tertentu, dan lain sebagainya yang semuanya tidak ada di dalam buku motif. Hal ini sangat membatasi daya kreatifitas dan hobi bagi mereka para penggemar kerajinan tangan sulam strimin, sehingga menggambar motif di atas kain strimin menjadi suatu hobi yang sulit. Bagi mereka yang belum mahir akan sangat sulit sekali menuangkan gambar yang diinginkan ataupun gambar imajinasi yang ada di dalam pikiran ke atas kain strimin. Kalaupun bisa akan timbul permasalahan penskalaan dan motif yang dibuat tidak proporsional. Baik dari segi pewarnaan, ukuran, dan letak detail suatu motif.

Dengan adanya aplikasi pembuat pola strimin dari suatu gambar menggunakan Borland Delphi 7.0 ini diharapkan dapat membantu mereka yang senang dan hobi membuat kerajinan sulam kain strimin dan dapat mempermudah dalam membuat dan menuangkan gambar di atas kain strimin dari pola-pola yang diinginkan sehingga kerajinan tangan sulam strimin menjadi sangat mudah, menarik dan menyenangkan. Dan bagi mereka yang ingin serius di bidang ini akan membantu mereka dalam menciptakan lapangan usaha baru yang sekaligus dapat menambah penghasilan keluarga sehingga seni kerajinan tangan sulam strimin bukan merupakan kegiatan yang 'kosong' semata.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka dapat dirumuskan permasalahannya adalah :

1. Bagaimana merancang dan membangun suatu aplikasi yang dapat mengubah suatu gambar menjadi gambar pola strimin
2. Bagaimana menghasilkan pola strimin dari gambar yang diinginkan dalam ukuran yang diinginkan
3. Bagaimana mengubah gambar *full color* menjadi pola strimin dengan warna yang terbatas namun tetap fokus dan proporsional.

1.3 Batasan Masalah

Agar topik yang dibahas tidak terlalu luas maka diperlukan penyederhanaan masalah. Adapun batasan masalah dalam penelitian ini adalah :

1. Aplikasi ini dibangun menggunakan Borland Delphi 7.0
2. Gambar yang diubah mempunyai format JPG, BMP, WMF
3. *Output* dari program adalah gambar dalam bentuk pola strimin.

1.4 Tujuan Penelitian

Diharapkan dengan adanya aplikasi ini dapat membantu mereka yang senang dan hobi membuat kerajinan sulam kain strimin dan dapat mempermudah dalam membuat dan menuangkan gambar di atas kain strimin dari pola-pola yang diinginkan sehingga kerajinan tangan sulam strimin menjadi sangat mudah, menarik dan menyenangkan.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat :

1. Membantu penggemar kerajinan sulam kain strimin untuk membuat motif strimin sesuai dengan yang diinginkan
2. Mengaplikasikan model Pengolahan Citra Digital

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data adalah metode yang digunakan untuk mengumpulkan data yang diperlukan dalam penelitian. Metode ini meliputi studi pustaka yaitu pengumpulan data dengan cara melakukan studi, analisis, dan dokumentasi literatur, serta sumber-sumber catatan lainnya yang berkaitan dengan topik dan permasalahan yang dibahas.

1.6.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak disusun berdasarkan hasil dari data yang sudah diperoleh. Metode ini meliputi :

1. Analisis Kebutuhan

Melakukan analisis terhadap permasalahan yang ada untuk mendapatkan gambaran umum mengenai aplikasi yang dibangun, juga kebutuhan perangkat lunak dan perangkat keras yang dibutuhkan dalam pembangunan dan pengembangan sistem.

2. Perancangan

Tahap ini merupakan tahap perancangan sistem dari aplikasi yang dibangun, yaitu mendefinisikan kebutuhan yang diperlukan, mendeskripsikan gambaran tentang aplikasi yang akan dibangun, selanjutnya adalah persiapan untuk membangun aplikasi yang diinginkan.

3. Implementasi

Implementasi dilakukan untuk menerjemahkan data atau pemecahan masalah yang telah dianalisis dan dirancang sedemikian rupa ke dalam bahasa pemrograman komputer yang telah ditentukan.

4. Pengujian

Pengujian dilakukan setelah aplikasi selesai dibangun. Pengujian ini dapat dilakukan dengan menggunakan kondisi-kondisi yang berbeda untuk menghasilkan suatu aplikasi atau software yang bermanfaat dan sesuai dengan target yang diharapkan.

1.7 Sistematika Penulisan

Berikut merupakan sistematika penulisan tugas akhir yang akan dilakukan:

BAB I PENDAHULUAN

Pada bab ini berisi mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi mengenai tinjauan pustaka dan teori-teori yang berhubungan dengan penelitian diantaranya mengenai teori tentang komponen

citra digital, Timage, operasi titik, operasi bertetangga, operasi geometri, interpolasi, reduksi warna, penskalaan.

BAB III METODOLOGI

Metode Analisis

Berisi mengenai metode-metode yang dipakai pada analisis kebutuhan perangkat lunak.

Hasil Analisis

Berisi mengenai hasil analisis, yaitu pilihan kebutuhan-kebutuhan dalam pembuatan perangkat lunak yang meliputi fungsi-fungsi yang dibutuhkan, kinerja atau tujuan yang dicapai serta pemilihan kebutuhan dalam pembuatan *interface* atau antarmuka.

BAB IV PERANCANGAN PERANGKAT LUNAK

Metode Perancangan

Berisi tentang metode-metode yang dipakai dalam perancangan perangkat lunak.

Hasil Perancangan

Berisi mengenai hasil perancangan perangkat lunak, hasil perancangan merupakan terjemahan kebutuhan perangkat lunak yang meliputi struktur data, arsitektur perangkat lunak, prosedur-prosedur, fungsi-fungsi dan *interface*.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Batasan Implementasi

Berisi mengenai batasan implementasi perangkat lunak yang dapat meliputi asumsi-asumsi yang dipakai, lingkungan pengembangan, bahasa

pemrograman yang digunakan dan batasan lain yang dibuat dan ditemui selama pengembangan perangkat lunak.

Implementasi

Berisi mengenai dokumen implementasi perangkat lunak yang meliputi implementasi struktur data, prosedur-prosedur, fungsi-fungsi dalam bahasa pemrograman yang dipakai serta antarmukanya.

BAB VI ANALISIS KINERJA PERANGKAT LUNAK

Berisi mengenai dokumentasi hasil pengujian terhadap perangkat lunak yang dibandingkan kebenarannya dan kesesuaiannya dengan kebutuhan perangkat lunak dan kemudian dianalisa.

BAB VII SIMPULAN DAN SARAN

Simpulan

Berisi kesimpulan-kesimpulan yang dapat diambil dari proses pengembangan perangkat lunak, baik pada tahap analisis kebutuhan perangkat lunak, perancangan, implementasi dan terutama pada analisis kinerja perangkat lunak.

Saran

Berisi mengenai saran yang perlu diperhatikan berdasar keterbatasan-keterbatasan yang ditemukan dan asumsi yang dibuat selama tugas akhir untuk perbaikan perangkat lunak yang dihasilkan untuk masa yang akan datang. Pada bagian akhir laporan Tugas akhir ini memuat daftar pustaka dan lampiran.

BAB II

LANDASAN TEORI

2.1 Strimin

Strimin adalah salah satu nama jenis kain yang digunakan sebagai media menggambar menggunakan benang sulam berwarna. Tekniknya adalah dengan memindahkan atau menggambar setiap kotak yang berskala 2 mm dengan warna yang sesuai dengan motif.

2.2 Format Citra

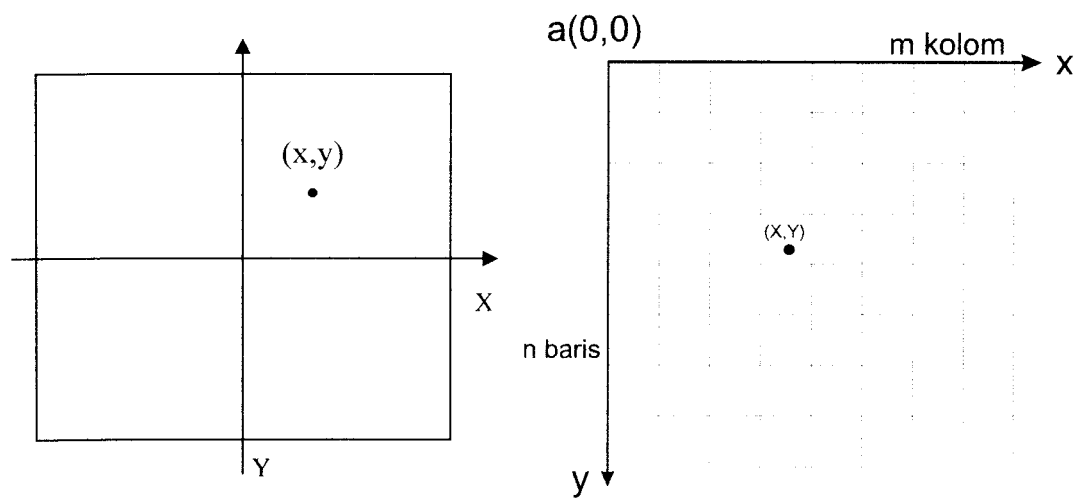
2.2.1 Komponen Citra Digital

Setiap citra digital memiliki beberapa karakteristik, antara lain ukuran citra, resolusi dan format nilainya. Ukuran citra dapat dinyatakan secara fisik dalam satuan panjang (misal mm atau inch). Dalam hal ini harus ada hubungan antara ukuran titik penyusun citra dengan satuan panjang. Hal tersebut dinyatakan dengan resolusi yang merupakan ukuran banyaknya titik untuk setiap satuan 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 [AHM05].

Citra digital tersusun atas titik-titik yang biasanya berbentuk persegi panjang atau bujur sangkar yang secara beraturan membentuk baris-baris dan kolom-kolom. Setiap titik memiliki koordinat sesuai dengan posisinya dalam

citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili titik tersebut.

Sebuah piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel $(0,0)$ terletak pada sudut kiri atas pada citra. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer [AHM05].

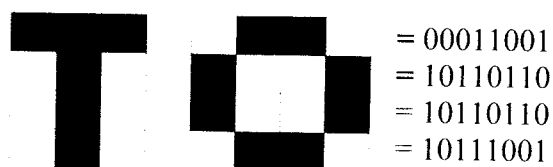


Gambar 2.1 Hubungan antara Koordinat Pada Citra dan Indeks Larik pada Komputer Untuk Menyimpan Data Citra

2.2.2 Citra Biner (Monokrom)

Pada citra biner setiap titik bernilai 0 atau 1, masing-masing merepresentasikan warna tertentu. Contoh yang paling lazim adalah warna hitam

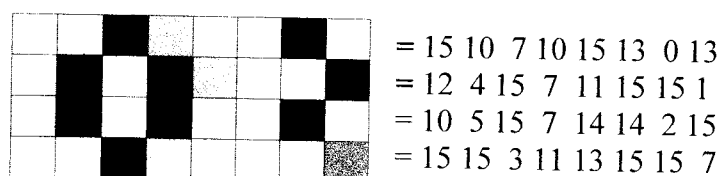
bernilai 0 dan warna putih bernilai 1. Setiap titik pada citra hanya membutuhkan 1 bit, sehingga setiap byte dapat menampung informasi 8 titik.



Gambar 2.2 Citra Biner dan Representasinya Dalam Data Digital

2.2.3 Citra Skala keabuan (gray scale)

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain di antara nilai minimum dan nilai maksimumnya. Banyaknya kemungkinan nilai dan nilai maksimumnya tergantung pada jumlah bit yang digunakan. Contohnya untuk skala keabuan 4 bit, maka jumlah nilai 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 $2^8 - 1 = 255$



Gambar 2.3 Citra Skala Keabuan 4 bit dan Representasinya Dalam Data Digital

2.2.4 Citra Warna (True color)

Pada citra warna setiap titik mempunyai warna yang spesifik yang merupakan kombinasi dari 3 warna dasar yaitu: merah, hijau, 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. Dengan demikian setiap titik pada citra warna membutuhkan data 3 byte.

Jumlah kombinasi yang mungkin untuk format citra ini adalah 2^{24} atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada, oleh sebab itu format ini dinamakan *true color*.

	= 255 255 255 255 0 0 128 128 128 128 128 0
	= 0 255 255 0 128 0 204 255 255 0 0 255
	= 150 150 150 51 51 51 255 255 255 95 95 95
	= 255 204 153 204 255 204 128 0 0 255 0 255

Gambar 2.4 Citra Warna True Color dan Representasinya Dalam Data Digital

2.3 Pengolahan Citra

Definisi citra menurut Kamus Webster adalah “suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda” [ACH05].

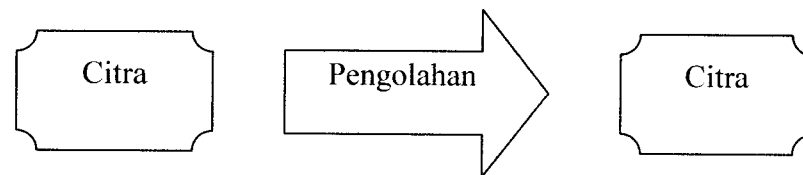
Citra atau *image* merupakan istilah lain untuk gambar. Citra memiliki karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Seperti dalam sebuah pepatah yang mengatakan “sebuah gambar bermakna lebih dari seribu kata”.

Meskipun sebuah citra kaya informasi, namun seringkali citra mengalami penurunan mutu (degradasi) misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur, dan sebagainya. Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin) maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih

baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*).

Pengolahan citra adalah pemrosesan citra , khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik [PRA05].

Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga adalah citra. Namun citra keluaran mempunyai kualitas lebih baik dari pada citra masukan atau kualitas keluaran dapat disesuaikan dengan kebutuhan *user* [MUN04].



Gambar 2.5 Pengolahan Citra

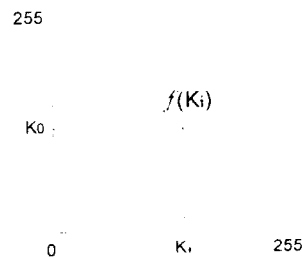
2.4 Operasi Titik

Operasi titik adalah operasi terhadap citra dimana setiap titik diolah secara tak gayut dengan titik-titik yang lain. Setiap titik pada citra mempunyai 2 buah karakteristik, yaitu koordinat yang menunjukkan lokasi dari titik tersebut di dalam citra serta nilai dari titik itu sendiri (tingkat keabuan atau warna) [ACH05].

Operasi titik dilakukan dengan memodifikasi nilai skala keabuan titik yang ditinjau tersebut berdasarkan fungsi tertentu, yang disebut fungsi transformasi skala keabuan atau *Gray Scale Transformation (GST)*. Fungsi ini memetakan tingkat keabuan input (K_i) ke citra keabuan citra output (K_o) [ACH05].

Secara umum fungsi tersebut dapat diformulasikan dengan :

$$K_0 = f(K_i) \quad (2.1)$$



Gambar 2.6 Fungsi Transformasi Skala Keabuan (GST).

Pada citra *true color*, fungsi GST (f) diterapkan untuk ketiga elemen warna yang ada (RGB). Ketiga fungsi GST dapat memiliki bentuk sama ataupun berbeda. Dengan kata lain perlakuan terhadap ketiga elemen warna tersebut tidak harus sama meskipun juga tidak harus berbeda.

$$R_0 = f_R(R_i) \quad (2.2)$$

$$G_0 = f_G(G_i) \quad (2.3)$$

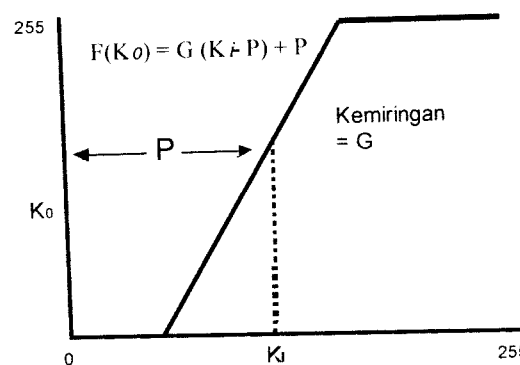
$$B_0 = f_B(B_i) \quad (2.4)$$

2.4.1 Peningkatan Kontras

Jika sebuah citra mempunyai nilai keabuan yang tidak terlalu berbeda untuk semua titik, dengan kurva histogram misalnya, maka citra tersebut akan terlihat kurang kontras. Hal ini disebabkan citra tersebut memiliki kurva histogram yang sempit, dengan tepi kiri dan tepi kanan yang berdekatan, sehingga titik tergelap dalam citra tersebut tidak mencapai hitam pekat dan titik paling terang dalam citra tidak berwarna putih cemerlang [ACH05]. Salah satu fungsi GST yang dapat digunakan untuk melakukan operasi peningkatan kontras adalah:

$$K_0 = G (K_i - P) + P \quad (2.5)$$

Dengan G adalah koefisien penguatan kontras, dan P adalah nilai skala keabuan yang dipakai sebagai pusat pengontrasan. Identy adalah membuat titik yang cenderung gelap menjadi lebih gelap dan titik yang cenderung terang menjadi lebih cemerlang. Plot fungsi GST ini dapat dilihat pada gambar



Gambar 2.7 Fungsi GST untuk Peningkatan Kontras Citra

2.4.2 Konversi Citra True Color ke Keabuan

Citra *true color* dapat dikonversi menjadi citra keabuan dengan operasi titik. Secara mudahnya, intensitas didefinisikan sebagai nilai rerata dari ketiga nilai elemen warna, sehingga nilai keabuan yang merepresentasikan intensitas dapat dihitung dengan rumus

$$K_0 = \frac{R_i + G_i + B_i}{3} \quad (2.6)$$

Atau

$$K_0 = w_R R_i + w_G G_i + w_B B_i \quad (2.7)$$

Dengan w_R , w_G , dan w_B masing-masing adalah bobot untuk elemen warna merah, hijau, dan biru. NTSC (*National Television System Committee*) mendefinisikan bobot untuk konversi citra ke keabuan sebagai berikut: $w_R = 0.299$, $w_G = 0.587$, $w_B = 0.144$.

2.5 Operasi Global

Operasi global adalah operasi yang bergantung pada sifat global citra. Pada operasi global, proses yang dilakukan bergantung pada karakteristik global dari citra yang hendak dimodifikasi. Karakteristik global tersebut biasanya berupa sifat statistik dari citra itu sendiri, yang direpresentasikan dengan histogram tingkat keabuan [ACH05].

2.5.1 Ekualisasi Histogram

Citra yang kurang kontras ditandai dengan sempitnya daerah yang dipakai oleh kurva histogram tingkat keabuan. Dengan operasi peningkatan kontras yang optimal, kurva histogram akan memiliki rentang yang maksimum, dari batas kiri ke batas kanan histogram. Cara lain yang dapat dilakukan untuk mendapatkan kontras yang optimal adalah dengan mendistribusikan kembali nilai-nilai skala keabuan citra untuk memperoleh kurva histogram yang datar atau seragam. Secara matematis proses ekualisasi histogram dapat dilakukan dengan menggunakan persamaan berikut:

$$K_0 = \text{round}(C_i \cdot (2^k - 1) / w.h) \quad (2.8)$$

dimana C_i adalah cacah kumulatif nilai skala keabuan ke- i dari citra asli dan fungsi *round* adalah untuk pembulatan ke bilangan bulat terdekat, contohnya 35.4 menjadi 35 dan 187.5 menjadi 188.

Setelah dilakukan perhitungan dengan persamaan di atas akan diperoleh hubungan antara keabuan input dan keabuan output, yang juga merupakan fungsi GST dari operasi ini. Keunikan dari operasi ini adalah fungsi GST secara eksak tidak akan sama untuk setiap citra. Dengan kata lain untuk citra yang berbeda akan diperoleh fungsi GST yang berbeda. Perbedaan ini terjadi karena sifat global masing-masing citra yang berbeda [ACH05].

2.6 Operasi Geometri

Karakteristik yang dimiliki oleh setiap piksel dalam suatu citra adalah koordinat dan nilai (keabuan atau warna) dari piksel tersebut. Operasi geometri dilakukan terutama dengan memodifikasi koordinat piksel dalam suatu citra dengan kemungkinan mengubah nilai skala keabuan dari titik tersebut dengan pendekatan tertentu. Operasi geometri berhubungan dengan perubahan bentuk geometri citra, baik ukuran maupun orientasinya [ACH05].

2.6.1 Interpolasi Tetangga Terdekat

Pada interpolasi tetangga terdekat (*nearest neighbour*), nilai keabuan titik hasil diambil dari nilai keabuan pada titik asal yang paling dekat dengan koordinat hasil perhitungan dari transformasi spasial. Untuk citra 2 dimensi, tetangga terdekat dipilih di antara 4 titik asal yang paling berbatasan satu sama lain.

Kelebihan dari interpolasi tetangga terdekat adalah kemudahan dan kecepatan eksekusinya. Namun pada banyak kasus, penggunaan interpolasi tetangga terdekat menghasilkan citra yang kurang memuaskan karena timbulnya aliasing pada bagian tepi obyek [ACH05].

2.6.2 Operasi Penskalaan (Scaling)

Operasi penskalaan (scaling) dimaksudkan untuk memperbesar (*zoom-in*) atau memperkecil (*zoom-out*) citra. Hal ini dapat dilakukan dengan mengintroduksi parameter skala, baik ke arah horisontal (S_h) maupun vertikal (S_v). Skala yang bernilai lebih dari 1 akan memperbesar citra asli, sedangkan jika bernilai kurang dari 1 akan memperkecil citra. Apabila *aspect ratio* (perbandingan antara tinggi dan lebar citra) hendak dipertahankan, maka $S_h = S_v$.

Transformasi spasial yang dipakai adalah:

$$x' = S_h x \quad (2.9)$$

$$y' = S_v y \quad (2.10)$$

Ukuran citra juga berubah sesuai hubungan berikut:

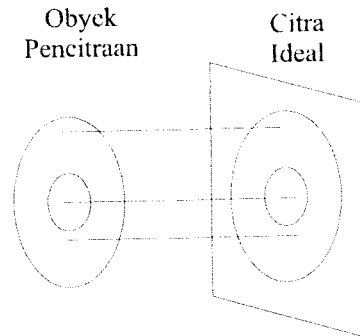
$$w' = S_h w \quad (2.11)$$

$$h' = S_v h \quad (2.12)$$

2.7 Operasi Bertetangga

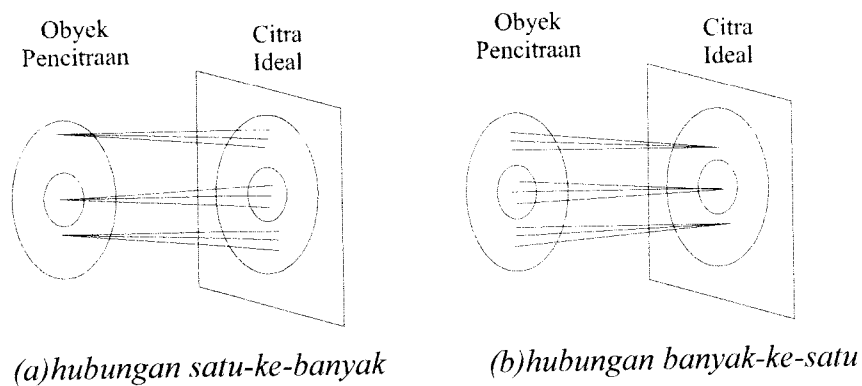
Sebuah citra dikatakan baik apabila benar-benar mencerminkan kondisi sesungguhnya dari obyek yang dicitrakan. Dengan demikian, citra yang ideal mempunyai hubungan korespondensi satu-satu. Artinya, sebuah titik pada obyek

yang dicitrakan dipetakan ke tepat sebuah titik pada citra, demikian pula sebaliknya, setiap titik citra mewakili sebuah titik pada obyek yang dicitrakan.



Gambar 2.8 Korespondensi Satu-satu Antara Obyek dan Citra Ideal.

Meskipun demikian, pada kenyataannya hubungan antara titik dalam obyek dengan titik pada citra adalah hubungan satu-ke-banyak dan banyak-ke-satu. Seperti terlihat pada gambar 2.9. Hubungan satu ke banyak menunjukkan bahwa setiap bagian pada obyek dapat menjadi sumber bagi banyak titik pada citra. Hal ini terjadi apabila sinyal yang berpindah dari obyek pencitraan mengalami penyebaran (divergensi), sehingga ketika sampai di detektor atau sensor, tidak lagi suatu titik namun sudah merupakan luasan (gambar 2.9a). sebaliknya dari sudut pandang detektor, sebuah titik pada detektor mungkin menerima sinyal dari banyak bagian dari obyek (gambar 2.9b).



Gambar 2.9 Hubungan Antar Titik pada Obyek dan Citra yang Aktual.

Melihat kenyataan tersebut, citra yang dihasilkan tidak lagi merupakan representasi 100% dari obyek yang dicitrakan. Oleh karena itu perlu dilakukan pengolahan citra untuk meningkatkan kualitas citra yang berhubungan dengan kondisi di atas. Operasi ini dikenal sebagai operasi bertetangga (*neighborhood operation*). Hal ini dilakukan dengan memodifikasi nilai keabuan sebuah titik berdasarkan nilai-nilai keabuan dari titik-titik yang ada di sekitarnya (bertetangga) yang masing-masing mempunyai bobot tersendiri. Bobot-bobot tersebut nilainya bergantung pada operasi yang hendak dilakukan, sedangkan banyaknya titik tetangga dapat bervariasi, misalkan 2x2, 3x3, 3x4 dan sebagainya [ACH05].

Operasi bertetangga pada dasarnya adalah konvolusi antara citra dengan sebuah filter. Nilai-nilai dalam filter tersebut merupakan bobot-bobot kontribusi titik tetangga terhadap operasi bertetangga yang dilakukan.

Konvolusi satu dimensi dapat diformulasikan dengan :

$$h(x) = f * g = \int_{-\infty}^{\infty} f(u)g(x-u) du \quad (2.13)$$

f adalah filter yang diterapkan, g adalah sinyal input, h adalah hasil konvolusi antara f dan g sedangkan x dan u tak gayut (*independent*) yang memiliki nilai kontinyu.

Operasi konvolusi dua dimensi dirumuskan dengan :

$$h(x,y) = f * g = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v)g(x-u, y-v) dudv \quad (2.14)$$

Untuk pengolahan citra, operasi yang dilakukan adalah diskrit karena nilai koordinat piksel merupakan nilai yang diskrit. Selanjutnya, filter atau *mask* yang digunakan pada pengolahan citra biasanya ukurannya terbatas, dalam artian bobot

atau pengaruh dari titik-titik yang cukup jauh sudah tidak signifikan sehingga dapat diabaikan (dianggap nol). Oleh karena itu batasan integral pada persamaan diatas menjadi terbatas (bukan “ $-\infty$ ” dan “ ∞ ”). Bentuk diskrit dari operasi konvolusi satu dimensi dan dua dimensi pada pengolahan citra masing-masing adalah :

$$h(x) = \sum_{u=-M}^M f(u)g(x-u) \quad (2.15)$$

disini x , y , u dan v adalah variabel tak gayut yang memiliki nilai diskrit, yang berupa posisi titik di dalam citra. M dan N adalah batas titik tetangga yang masih memberikan pengaruh ke titik yang sedang ditinjau untuk arah horizontal dan vertikal. Ukuran *mask* yang digunakan untuk konvolusi adalah $2M+1$ kolom dan $2N+1$ baris. Dengan demikian, banyaknya titik bertetangga yang memiliki pengaruh terhadap titik yang sedang ditinjau adalah $(2M+1) * (2N+1)$. Nilai tersebut termasuk titik yang sedang ditinjau itu sendiri [ACH05].

2.8 Deteksi Tepi

Operasi ini digunakan untuk menemukan lokasi titik-titik yang merupakan tepi obyek. Secara umum, tepi suatu obyek dalam citra dinyatakan sebagai titik yang nilai keabuannya berbeda cukup besar dengan titik yang ada di sebelahnya.

2.8.1 Deteksi Tepi Menggunakan Operator Gradien

Tepi obyek adalah titik yang mempunyai perbedaan keabuan yang cukup besar dengan titik tetangganya, maka pencarian tepi dapat dilakukan dengan menghitung selisih atau diferensi antara 2 buah titik yang bertetangga, sehingga

diperoleh nilai gradien (turunan orde pertama) citra. persamaan dasar untuk menghitung gradien 1 dimensi adalah :

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \quad (2.16)$$

Robert membuat *mask* untuk menerapkan prinsip tersebut seperti terlihat dalam gambar 2.10a dan b. Diferensi diambil secara diagonal untuk mendapatkan *mask* yang berbentuk bujursangkar, dan diperoleh 2 *mask* mengakomodasi kedua arah diagonal. Hasil konvolusi citra asal dengan kedua *mask* tersebut kemudian digabungkan dengan cara tertentu sehingga menghasilkan nilai yang dapat digunakan untuk menentukan apakah sebuah titik merupakan tepi obyek dalam citra. Biasanya tepi naik (hasil positif) dan tepi turun (hasil negatif) tidak dibedakan, sehingga diambil nilai absolutnya [ACH05].

1	0
0	-1

(a) operator Roberts diagonal 1

0	1
-1	0

(b) operator Roberts diagonal 2

-1	0	1
-1	0	1
-1	0	1

(c) Operator Prewitt horizontal

-1	-1	-1
0	0	0
1	1	1

(d) Operator Prewitt vertikal

-1	0	1
-2	0	2
-1	0	1

(e) Operator Sobel horizontal

-1	-2	-1
0	0	0
1	2	1

(f) Operator Sobel vertikal

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

(g) Operator isotropik horizontal

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

(h) Operator isotropik vertikal

Gambar 2.10 Operator Gradien Untuk Deteksi Tepi.

Kombinasi antara kedua hasil konvolusi *mask* tadi biasanya dilakukan dengan mengambil nilai maksimum (persamaan 2.17), rerata (persamaan 2.18) ataupun rerata geometri (persamaan 2.19) dari keduanya.

$$K_0(x,y) = \max (|K_1(x,y)| + |K_2(x,y)|) \quad (2.17)$$

$$K_0(x,y) = \frac{|K_1(x,y)| + |K_2(x,y)|}{2} \quad (2.18)$$

$$K_0(x,y) = \sqrt{K_1(x,y)^2 + K_2(x,y)^2} \quad (2.19)$$

dengan $K_1(x,y)$ dan $K_2(x,y)$ adalah hasil konvolusi menggunakan mask 1 dan mask 2.

Karena ukuran *mask* operator Roberts adalah 2x2, maka posisi titik yang ditinjau tidak berada di pusat *mask*, sehingga tepi yang dihasilkan berada pada salah satu sisi tepi tersebut (tepi atas atau tepi bawah). Prewitt kemudian menggunakan diferensi antara nilai-nilai tetangga titik yang ditinjau untuk membuat citra tepi dengan *mask* berukuran 3x3, sehingga titik yang ditinjau berada tepat di tengah *mask*. Operator Prewitt terdiri atas 2 *mask*, masing-masing untuk mendeteksi tepi dengan arah horizontal dan vertikal (gambar 2.10c dan d). Kombinasinya juga dapat dilakukan dengan persamaan 2.16 sampai dengan 2.18

Sobel memperbaiki operator yang dikembangkan oleh Prewitt dengan memberi nilai bobot yang lebih besar, yaitu dua kalinya, pada baris tengah untuk arah horizontal dan kolom tengah untuk arah vertikal, dengan anggapan baris dan kolom tengah tersebut memberi kontribusi yang lebih besar dibandingkan baris dan kolom lainnya (gambar 2.10e dan f). Kemudian dikembangkan juga operator isotropik yang mirip dengan operator Sobel, namun dengan bobot baris dan kolom tengah adalah $\sqrt{2}$ dari bobot di baris dan kolom lainnya (gambar 2.10g dan h) [ACH05].

2.9 Warna

Warna-warna yang diterima oleh mata manusia merupakan hasil kombinasi cahaya dengan panjang gelombang yang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah *Red (R)*, *Green (G)* dan *Blue (B)*. Ketiga warna tersebut dinamakan warna pokok dan sering disingkat sebagai warna dasar RGB. Warna-warna lain dapat diperoleh dengan

mencampurkan ketiga warna pokok tersebut dengan perbandingan tertentu [MUN04].

2.9.1 Atribut Warna

Selain RGB, warna dapat dimodelkan berdasarkan atribut warna. Setiap warna mempunyai tiga atribut, yaitu *Intensity (I)*, *Hue (H)*, dan *Saturation (S)* [MUN04].

a) *Intensity/Brighness/Luminance*

Atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memedulikan warna. Kisaran nilainya antara gelap (hitam) dan terang (putih).

b) *Hue*

Menyatakan warna sebenarnya, seperti merah, violet, kuning. *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*) dan sebagainya dari cahaya. *Hue* berasosiasi dengan panjang gelombang cahaya.

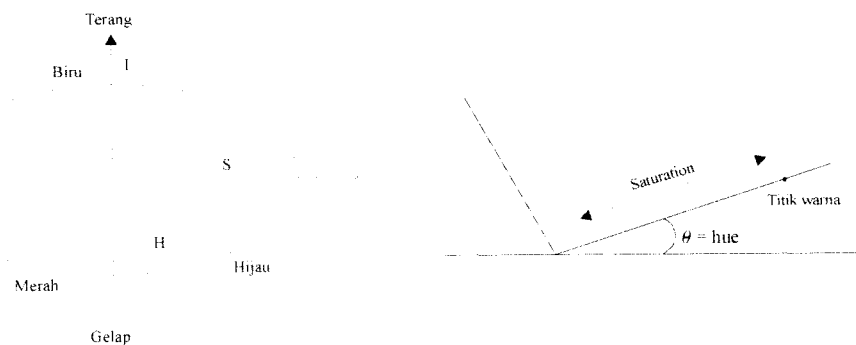
c) *Saturation*

Menyatakan tingkat kemurnian warna cahaya, yaitu mengindikasikan barapa banyak warna putih diberikan pada warna. Sebagai contoh , warna merah adalah 100% warna jenuh (*saturated color*), sedangkan warna pink adalah warna merah dengan tingkat kejenuhan sangat rendah (karena ada warna putih di dalamnya).

Hue dikuantisasi dengan nilai 0 sampai 255. 0 menyatakan merah, kemudian memutar nilai-nilai spektrum tersebut kembali lagi ke 0 untuk menyatakan merah lagi. Ini dapat dipandang sebagai sudut dari 0° sampai 360° .

Jika suatu warna mempunyai $\text{saturation} = 0$, maka warna tersebut tanpa *hue*, yaitu dibuat dari warna putih saja. Jika $\text{saturation} = 255$, maka tidak ada warna putih yang ditambahkan pada warna tersebut [MUN04].

Intensitas nilai dari gelap sampai terang (gelap = 0, terang = 255)



Gambar 2.11 Model IHS.

2.9.2 Sistem Koordinat Warna

Standar panjang gelombang warna-warna pokok menurut CIE (*Commission international de l'Eclairage*) adalah

$$R = 700 \text{ nm}$$

$$G = 546.1 \text{ nm}$$

$$B = 435.8 \text{ nm}$$

Warna-warna lain dapat dihasilkan dengan mengkombinasikan ketiga warna pokok tersebut. Model warna yang digunakan sebagai acuan dinamakan model RGB.

Salah satu cara yang mudah untuk menghitung nilai warna dan menafsirkan hasilnya dalam model warna RGB adalah dengan melakukan normalisasi terhadap ketiga komponen warna tersebut [AHM05]

Kromatisitas (*chromaticity of color*) masing-masing warna pokok, menunjukkan persentase relatif suatu warna pokok di antara warna pokok lainnya pada warna yang diberikan, yang didefinisikan sebagai berikut

$$X = \frac{X}{X + Y + Z} \quad (2.20)$$

$$Y = \frac{Y}{X + Y + Z} \quad (2.21)$$

$$Z = \frac{Z}{X + Y + Z} \quad (2.22)$$

Warna putih acuan dinyatakan dengan $X=Y=Z=1$, jumlah seluruh nilai kromatisitas warna adalah satu

$$x+y+z = 1, \text{ atau } z = 1-(x+y) \quad (2.23)$$

2.9.3 Model warna CMY dan CMYK

Warna *cyan* (C), *magenta* (M), dan *yellow* (Y) adalah warna komplementer terhadap *red*, *green*, dan *blue*. Dua buah warna disebut komplementer jika dicampur dengan perbandingan yang tepat menghasilkan warna putih. Misalnya, *magenta* jika dicampur dengan perbandingan yang tepat dengan *green* menghasilkan putih, karena itu *magenta* adalah komplemen dari *green* [MUN04].

$$C = 1 - R \quad (2.24)$$

$$M = 1 - G \quad (2.25)$$

$$Y = 1 - B \quad (2.26)$$

Model CMY dapat digunakan untuk mencetak citra berwarna, tetapi karena ketidaksempurnaan tinta, model CMY tidak dapat menghasilkan warna hitam dengan baik. Karena itu model CMY disempurnakan menjadi model CMYK, yang dalam hal ini K menyatakan warna keempat, dengan perhitungan sebagai berikut :

$$K = \min (C, M, Y) \quad (2.27)$$

$$C = C - K \quad (2.28)$$

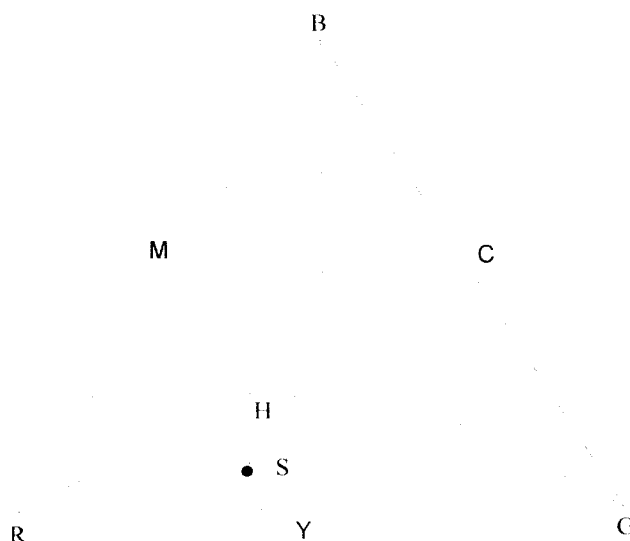
$$M = M - K \quad (2.29)$$

$$Y = Y - K \quad (2.30)$$

2.9.4 Transformasi Warna RGB ke IHS

Meskipun basis RGB bagus untuk menampilkan informasi warna, tetapi tidak cocok untuk beberapa aplikasi pemrosesan citra. Pada aplikasi pengenalan obyek, lebih mudah mengidentifikasi obyek dengan perbedaan *hue*-nya dengan cara memberikan nilai ambang pada rentang nilai-nilai *hue* (panjang gelombang spektrum) yang melingkupi obyek. Aplikasi yang lain misalnya pemampatan citra. Pemampatan citra berwarna lebih relevan bila warna RGB-nya dikonversikan ke *IHS* karena algoritma pemampatan pada citra skala keabuan dilakukan pada komponen I, sedangkan nilai H dan S dikodekan dengan cara yang lain dengan sedikit atau sama sekali tidak ada degradasi [MUN04].

Model warna IHS merepresentasikan warna dalam terminologi *intensity*, *hue*, dan *saturation* [AHM05].



Gambar 2.12 Segitiga HIS.

Komponen RGB dari citra berwarna dapat dikonversikan ke model warna IHS, dengan mengasumsikan komponen RGB telah dinormalisasi ke 1, maka I, H dan S dihitung dengan rumus:

$$I = \frac{1}{3}(R+G+B) \quad (2.31)$$

$$H = \cos^{-1} \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \quad (2.32)$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (2.33)$$

Alternatif lain mengubah model RGB ke IHS adalah sebagai berikut: konversi dari model RGB ke model IHS dilakukan dalam dua tahap. Pertama adalah merotasikan koordinat RGB ke sistem koordinat (I, V_1, V_2) dengan transformasi :

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & \sqrt{3}/3 & \sqrt{3}/3 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.34)$$

Langkah kedua adalah menghitung H dan S dari koordinat (V_1, V_2):

$$H = \tan^{-1}(V_2/V_1) \quad (2.35)$$

$$S = (V_1^2 + V_2^2)^{1/2} \quad (2.36)$$

Nilai H adalah dalam selang $[0, 2\pi]$ atau setara dengan $[0, 360^\circ]$. Transformasi dari model IHS ke model RGB dapat dilakukan dengan prosedur balikan :

$$V_1 = S \cos(H) \quad (2.37)$$

$$V_2 = S \sin(H) \quad (2.38)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & 0 & 2/\sqrt{6} \\ \sqrt{3}/3 & 1/\sqrt{2} & -1/\sqrt{6} \\ \sqrt{3}/3 & -1/\sqrt{2} & -1/\sqrt{6} \end{bmatrix} \begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} \quad (2.39)$$

Dengan transformasi RGB ke IHS, maka algoritma pemrosesan citra yang semula untuk citra skala abu dapat diterapkan pada komponen *intensity*, sedangkan algoritma segmentasi citra dapat dilakukan pada komponen H. Transformasi citra dari basis RGB ke basis IHS dilakukan sebelum pemrosesan citra. Citra yang sudah diproses dapat dikonversikan kembali ke basis RGB untuk tujuan display.

BAB III

METODOLOGI

3.1 Metode Analisis

Metode analisis digunakan untuk mengetahui dan menerjemahkan semua permasalahan serta kebutuhan perangkat lunak dan kebutuhan aplikasi pembuat pola strimin dari suatu gambar yang akan dibangun. Metode analisis kebutuhan perangkat lunak yang dipakai dalam tugas akhir ini menggunakan pendekatan terstruktur. Selain itu bersifat GUI (*Graphical User Interface*) sehingga mudah dalam pengoperasiannya

Oleh karena itu tahap analisis digunakan untuk mendapatkan data-data yang diperlukan untuk kepentingan membangun sistem. Hasil akhir dari analisis diharapkan akan diperoleh suatu sistem yang strukturnya dapat didefinisikan dengan baik dan jelas sehingga dapat diusulkan suatu bentuk perangkat lunak yang diinginkan.

3.2 Hasil Analisis

Berdasarkan analisis yang dilakukan dapat diketahui apa saja yang akan menjadi masukan sistem, fungsi atau metode yang digunakan oleh sistem, keluaran sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antarmuka sistem yang akan dibuat, sehingga sistem yang dibangun sesuai dengan yang diharapkan.

3.2.1 Masukan Sistem

Untuk proses masukan data dari aplikasi pembuat pola strimin dari suatu gambar yang dibuat adalah gambar atau *image* yang mempunyai format BMP, JPG, WMF.

3.2.2 Kebutuhan Proses

Untuk kebutuhan proses dalam sistem yang akan dibangun, proses-proses yang digunakan adalah:

- a. Pemasukan *file* gambar yang mempunyai format BMP, JPG, WMF.
- b. Operasi Titik.
- c. Operasi bertetangga.
- d. Operasi Goemetri.
- e. Reduksi warna.
- f. Penskalaan.

3.2.3 Keluaran Sistem

Data keluaran yang akan ditampilkan oleh Aplikasi pembuat pola strimin dari suatu gambar adalah berupa gambar dalam bentuk motif strimin.

3.2.4 Kebutuhan Perangkat Keras

Untuk membangun Aplikasi pembuat pola strimin dari suatu gambar, diperlukan perangkat keras yang memiliki minimal spesifikasi sebagai berikut :

1. Satu unit komputer dengan spesifikasi minimum prosessor Pentium III.
2. Memori (RAM) 256 MB.
3. Hard Disk 20 GB.
4. Monitor VGA atau SVGA.

5. Mouse.
6. Keyboard.

Namun sangat disarankan menggunakan spesifikasi yang lebih dari pada yang disebut di atas.

3.2.5 Kebutuhan Perangkat Lunak

Untuk membangun Aplikasi pembuat pola strimin dari suatu gambar, diperlukan perangkat lunak antara lain:

1. Sistem Operasi Windows XP.
2. Bahasa pemrograman Borland Delphi 7.0 untuk pembuatan perangkat lunak.
3. Coreldraw Graphics Suite 12 untuk membantu kebutuhan antarmuka

3.2.6 Antarmuka Sistem

Aplikasi pembuat pola strimin dari suatu gambar akan diakses oleh pengguna (*user*) dengan tingkat kemampuan yang berbeda. Oleh karena itu diperlukan antarmuka (*interface*) yang dapat dimengerti dan mudah digunakan oleh pengguna (*userfriendly*) dengan berbagai tingkat kemampuan dalam menggunakan komputer tanpa harus mempelajarinya terlebih dahulu. Kriteria tersebut dapat dipenuhi dengan menggunakan antarmuka (*interface*) berbasis grafis yang lebih mudah dimengerti oleh banyak pengguna.

3.2.7 Kinerja yang Diharapkan

Kinerja yang diharapkan dari hasil analisis di atas adalah perangkat lunak yang dibangun mampu mengubah *file* gambar yang mempunyai format BMP, JPG, WMF menjadi gambar pola strimin yang mempunyai format BMP.

3.3 Metode Perancangan

Metode perancangan yang digunakan untuk mengembangkan sistem aplikasi pembuat pola strimin dari suatu gambar ini adalah metode perancangan terstruktur (*Structured Design Method*), dengan menggunakan alat untuk pengembangan sistem yaitu *Flow Chart* dan *Data Flow Diagram (DFD)*.

3.4 Hasil Perancangan

Berdasarkan analisis yang dilakukan dapat diketahui apa saja yang akan menjadi masukan sistem, fungsi atau metode yang digunakan oleh sistem, keluaran sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antarmuka sistem yang akan dibuat, sehingga sistem yang dibangun sesuai dengan apa yang diharapkan.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan Sistem

Metode perancangan yang digunakan dalam perancangan aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0 adalah dengan menggunakan metode perancangan terstruktur (*Structured Design Method*), dengan menggunakan alat untuk pengembangan sistem yaitu *Flowchart* dan *Data Flow Diagram (DFD)*. *Flowchart* merupakan teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*. *Flowchart* juga digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi.

4.2 Hasil Perancangan

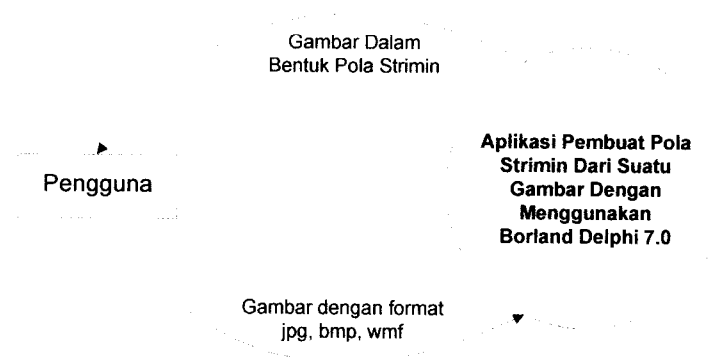
Berikut merupakan hasil perancangan yang dibuat untuk aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0.

4.2.1 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah model perancangan sistem yang menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data . Sebagai perangkat analisis, model ini hanya mampu memodelkan sistem dari sudut pandang fungsi.

4.2.2 Diagram Korteks (*Context Diagram*)

Diagram korteks atau DFD *level 0*, yang disebut juga dengan model sistem dasar yang merepresentasikan seluruh elemen sistem sebagai *entitas* dengan data *input* dan *output* yang ditunjukkan oleh anak panah untuk menunjukkan aliran data yang masuk atau keluar dari suatu *entitas* ke proses ataupun proses ke entitas. Aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0 dimulai dari bentuk yang paling global yaitu diagram korteks, kemudian diagram korteks ini diturunkan sampai bentuk yang paling detail. Dari analisis yang telah dilakukan, diperoleh diagram korteks yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram korteks (DFD level 0)

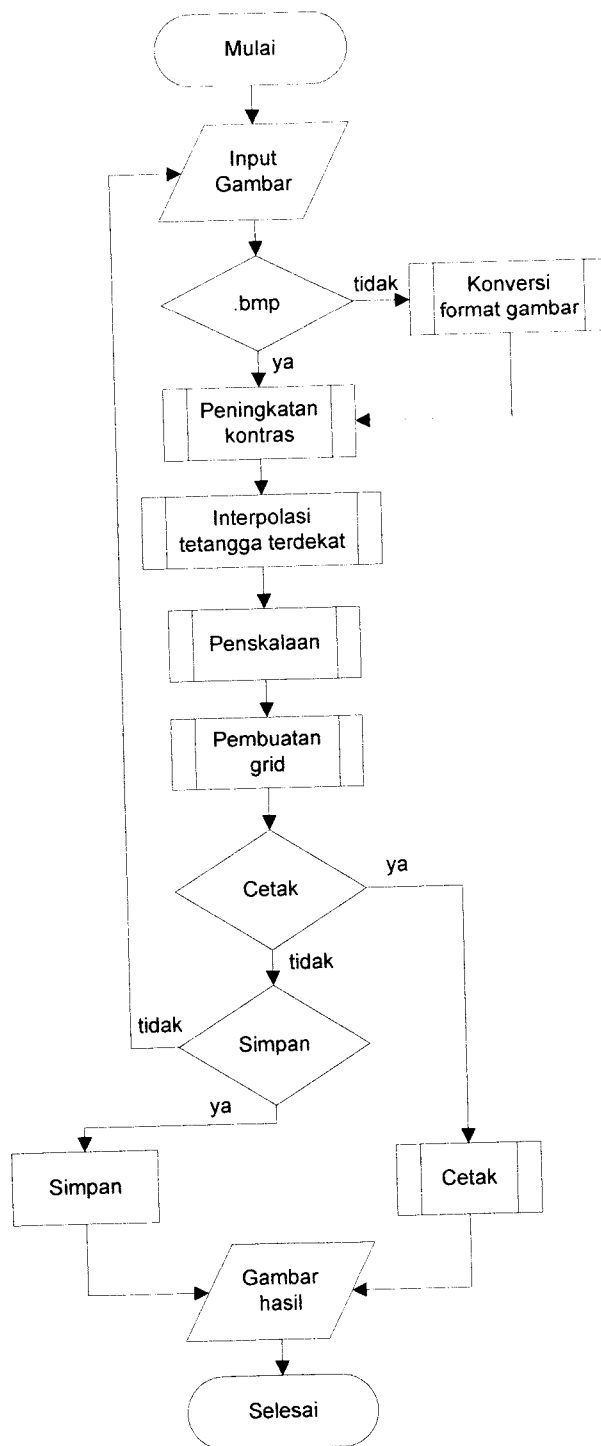
Dalam sistem tersebut terdapat satu entitas saja, yaitu pengguna dan data *input* maupun *output*nya mengarah ke pengguna. Setelah menggambarkan diagram korteks atau DFD level 0, langkah selanjutnya adalah memecah DFD level 0 menjadi bagian-bagian yang lebih terinci sesuai dengan kebutuhan.

4.2.3 Perancangan Diagram Alir Sistem

Untuk mempermudah penggambaran suatu proses menjadi sistem yang mudah untuk dikembangkan secara logika maka digunakan bagan alir atau

flowchart. Diagram Alir (*Flowchart*) yang merupakan bahasa informal yang menyediakan urutan perintah secara logika bagaimana analisis sistem memecahkan suatu masalah tertentu yang digambarkan secara grafis sehingga lebih jelas dan mudah dipahami sehingga akan memudahkan di dalam implementasinya ke *coding* suatu bahasa pemrograman tertentu dalam hal ini Borland Delphi 7.0

Gambar 4.2 menunjukkan *flowchart* Aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0.



Gambar 4.2 Flowchart aplikasi pembuat pola strimin

Pada Gambar 4.2, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flowchart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **Input Gambar**, untuk membuka *file* gambar yang akan diproses.
- 3) **bmp**, untuk mengecek *file* gambar apakah mempunyai format .bmp atau tidak.
 ⇒ Jika “Ya”, maka *file* gambar bisa dikenakan proses selanjutnya.
 ⇒ Jika “Tidak”, artinya *file* gambar tersebut mempunyai format gambar .jpg atau .wmf, maka *file* gambar tersebut akan dikenakan proses konversi format gambar menjadi format .bmp.
- 4) **Peningkatan Kontras**, untuk mengubah tingkat kontras suatu gambar dengan tujuan mengurangi jumlah warna dari gambar tersebut. Adapun skala tingkat kontras yang diizinkan antara 1 sampai dengan 20.
- 5) **Interpolasi tetangga terdekat**, untuk mendekatkan nilai pixel diantara titik-titik yang berdekatan.
- 6) **Penskalaan**, untuk membuat gambar yang akan di proses mempunyai ukuran lebar 800 pixel (100 kotak strimin), dan tinggi menyesuaikan berdasarkan perbandingan dengan gambar aslinya.
- 7) **Pembuatan Grid**, untuk membuat grid atau kotak strimin pada gambar.
- 8) **Gambar Hasil**, merupakan gambar yang sudah berbentuk pola strimin
- 9) **Cetak**, untuk memberikan pilihan kepada pengguna proses selanjutnya yang akan dilakukan.

⇒ Jika “Ya”, maka *file* gambar bisa dikenakan proses cetak.

⇒ Jika “Tidak”, maka proses mulai lagi dari langkah b (*Input* gambar) atau langsung selesai.

10) **Simpan**, untuk memberikan pilihan kepada pengguna proses selanjutnya yang akan dilakukan.

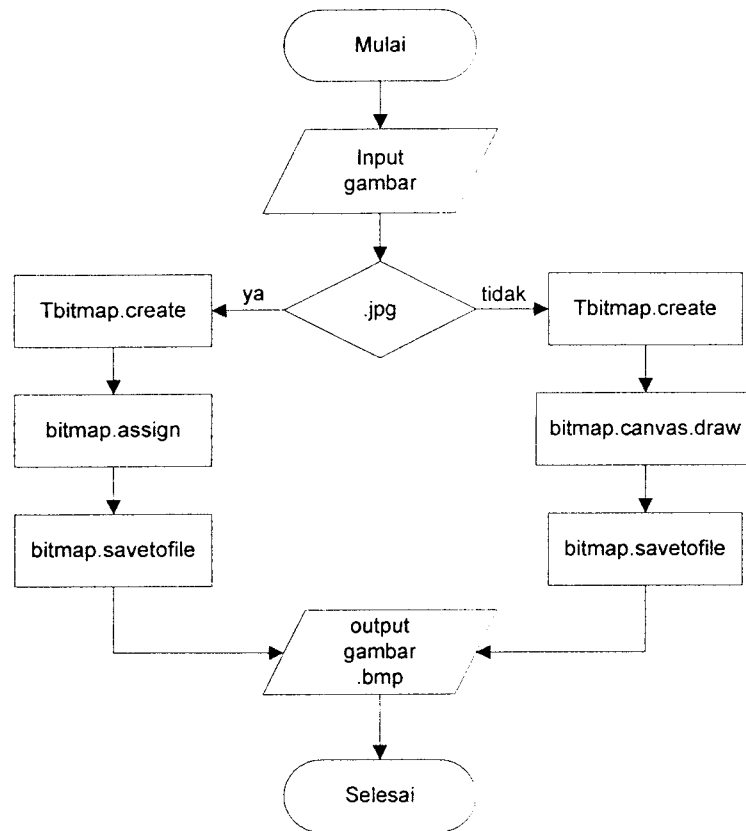
⇒ Jika “Ya”, maka *file* gambar bisa dikenakan proses simpan.

⇒ Jika “Tidak”, maka proses mulai lagi dari langkah b (*Input* gambar) atau langsung selesai.

11) **Selesai**, penanda akhir dari semua proses.

4.2.4 Flowchart konversi format gambar

Flowchart konversi format gambar menggambarkan proses yang dilakukan untuk melakukan konversi format gambar dari format .jpg dan .wmf menjadi gambar dengan format .bmp. Proses dimulai dengan memeriksa *file* masukan apakah berformat .bmp jika tidak maka dilanjutkan ke proses konversi. Proses ini dapat dilihat pada gambar 4.3



Gambar 4.3 Flowchart konversi format gambar

Pada Gambar 4.3, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flowchart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **Input Gambar**, untuk membuka *file* gambar yang akan diproses.
- 3) **.jpg**, untuk mengecek *file* gambar apakah mempunyai format .jpg atau tidak.

⇒ Jika “Ya”, maka *file* gambar akan dibuatkan Tbitmap.create.

a) **bitmap.assign**, untuk menyalin *file* .jpg

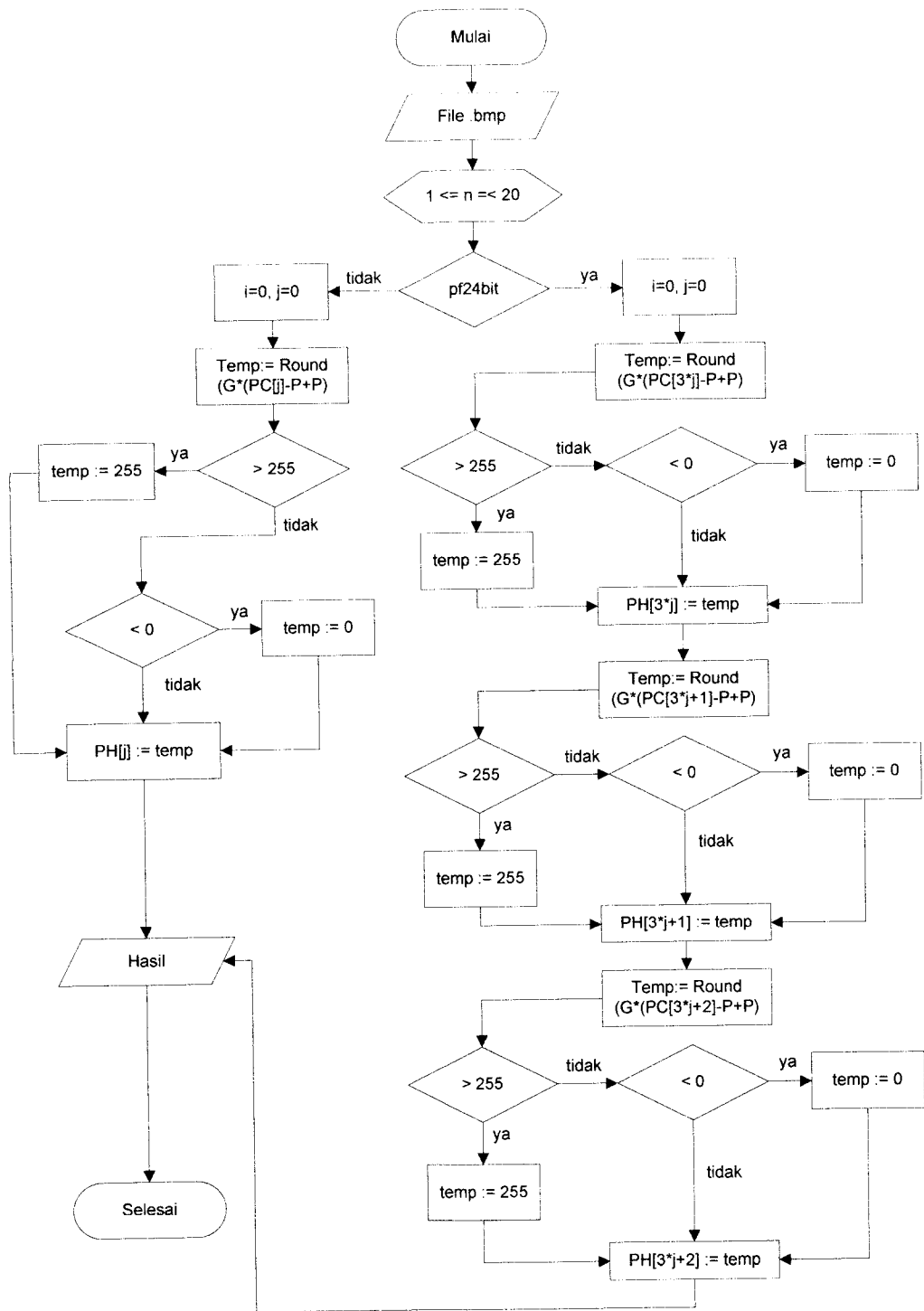
b) **bitmap.SaveToFile**, untuk menyimpan sementara sebelum dikenakan proses selanjutnya

⇒ Jika “Tidak”, artinya *file* gambar tersebut mempunyai format gambar .wmf, maka *file* gambar tersebut akan

- a) **bitmap.canvas.draw**, untuk menggambar *file* .wmf
 - b) **bitmap.SaveToFile**, untuk menyimpan sementara sebelum dikenakan proses selanjutnya.
- 4) **output gambar .bmp**, setelah *file* jpg dan .wmf selesai dikonversi.
 - 5) **Selesai**, penanda akhir dari semua proses.

4.2.5 Flowchart peningkatan kontras gambar

Tingkat kontras satu gambar berbeda dengan gambar yang lain. Pada bagian ini kontras gambar diberi nilai antara 1 sampai 20, dimana nilai 1 merupakan batas minimum dan nilai 20 merupakan batas maksimum peningkatan kontras. Idanya adalah membuat gambar yang agak terang menjadi lebih terang dan gambar yang agak gelap menjadi lebih gelap. *Flowchart* peningkatan tingkat kontras gambar dapat dilihat pada gambar 4.4



Gambar 4.4 Flowchart peningkatan kontras gambar

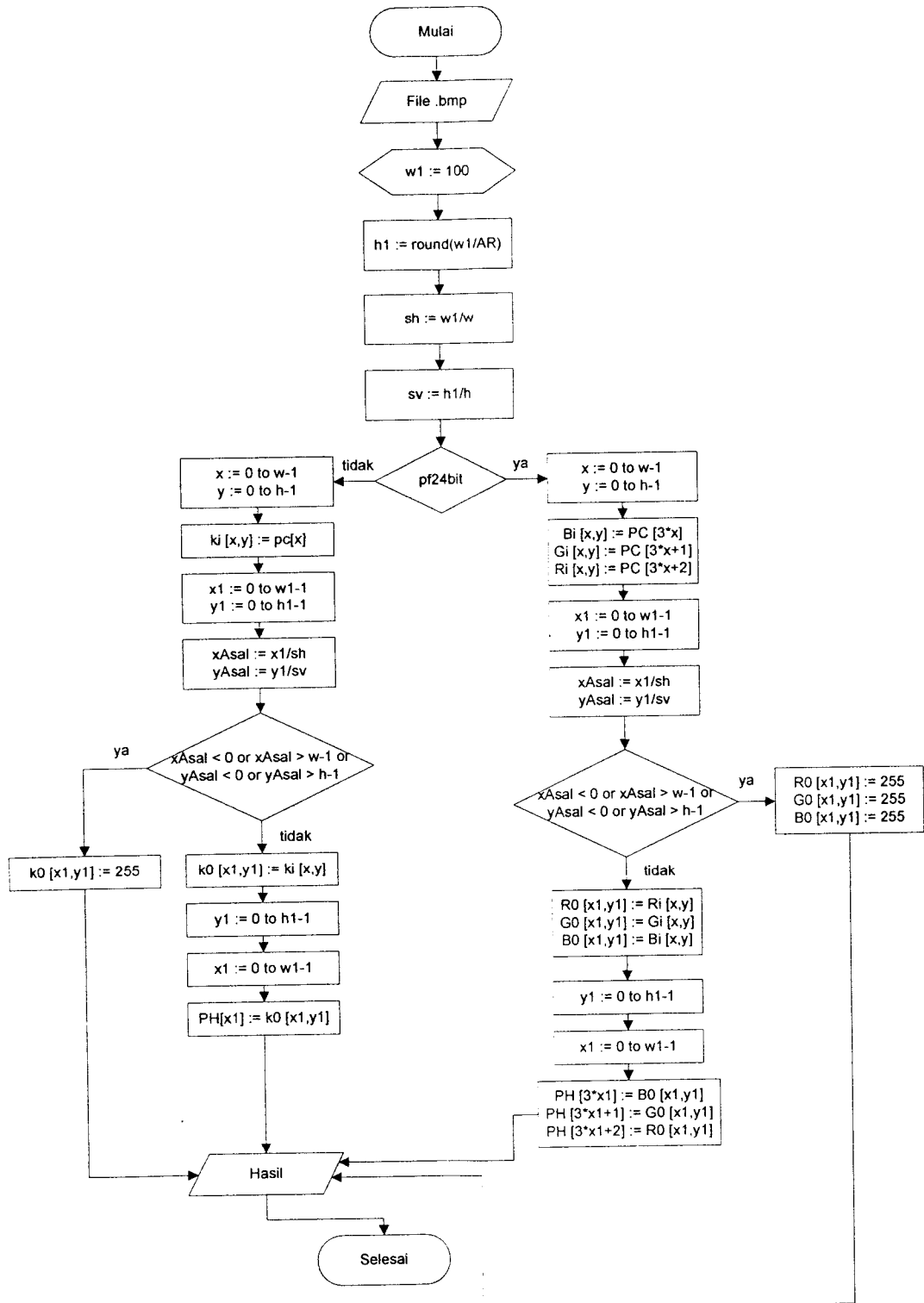
Pada Gambar 4.4, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flowchart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **File .bmp**, membuka *file* gambar yang akan diproses.
- 3) $1 \leq n \leq 20$, menetapkan nilai penguatan kontras
- 4) **Pf24bit**, jika ya jalankan langkah 5 sampai 23, jika tidak jalankan langkah 24 sampai 30.
- 5) **i=0, j=0**, isi nilai i dan j dengan 0
- 6) **Temp:=Round(G*(PC[3*j]P+P))**, isi nilai temp dengan hasil perhitungan.
- 7) Jika nilai temp > 255, isi temp dengan 255, lakukan proses selanjutnya
- 8) Jika nilai temp < 255 maka lakukan langkah 9
- 9) **PH[3*j] := Temp**, isi picture hasil = temp
- 10) Jika nilai temp < 0, isi temp dengan 0, lakukan proses selanjutnya
- 11) Jika nilai temp > 0 maka lakukan langkah 9
- 12) **Temp:=Round(G*(PC[3*j+1]P+P))**, isi nilai temp dengan hasil perhitungan.
- 13) Jika nilai temp > 255, isi temp dengan 255, lakukan proses selanjutnya
- 14) Jika nilai temp < 255 maka lakukan langkah 15
- 15) **PH[3*j+1] := Temp**, isi picture hasil = temp
- 16) Jika nilai temp < 0, isi temp dengan 0. lakukan proses selanjutnya
- 17) Jika nilai temp > 0 maka lakukan langkah 15

- 18) **Temp:=Round(G*(PC[3*j+2]P+P))**, isi nilai temp dengan hasil perhitungan.
- 19) Jika nilai temp > 255, isi temp dengan 255, lakukan proses selanjutnya
- 20) Jika nilai temp < 255 maka lakukan langkah 21
- 21) **PH[3*j+2] := Temp**, isi picture hasil = temp
- 22) Jika nilai temp < 0, isi temp dengan 0, lakukan proses selanjutnya
- 23) Jika nilai temp > 0 maka lakukan langkah 21
- 24) **i=0, j=0**, isi nilai i dan j dengan 0
- 25) **Temp:=Round(G*(PC[j]P+P))**, isi nilai temp dengan hasil perhitungan.
- 26) Jika nilai temp > 255, isi temp dengan 255, lakukan proses selanjutnya
- 27) Jika nilai temp < 255 maka lakukan langkah 28
- 28) Jika nilai temp < 0, isi temp dengan 0, lakukan proses selanjutnya
- 29) Jika nilai temp > 0 maka lakukan langkah 30
- 30) **PH[3*j+2] := Temp**, isi picture hasil = temp
- 31) **Hasil**, nilai picture hasil.
- 32) **Selesai**, penanda akhir dari semua proses.

4.2.6 Flowchart pengaturan ukuran gambar pola strimin

Beragamnya ukuran gambar yang diinputkan ke dalam program menyulitkan pembuatan ukuran pola strimin yang dibentuk. Maka ukuran pola strimin yang dibuat diatur menjadi 100 pixel untuk lebar gambar yang diinputkan dengan tinggi gambar yang menyesuaikan dengan perbandingan ukuran gambar aslinya. Ketentuan ini berlaku untuk gambar dengan format pixel 8 bit dan 24 bit. Setelah masing-masing gambar dipisahkan antara yang 8 bit dan 24 bit maka pembentukan pola strimin dibuat. *Flowchart* pengaturan ukuran gambar pola strimin dapat dilihat pada gambar 4.5



Gambar 4.5 Flowchart pengaturan ukuran gambar pola strimin

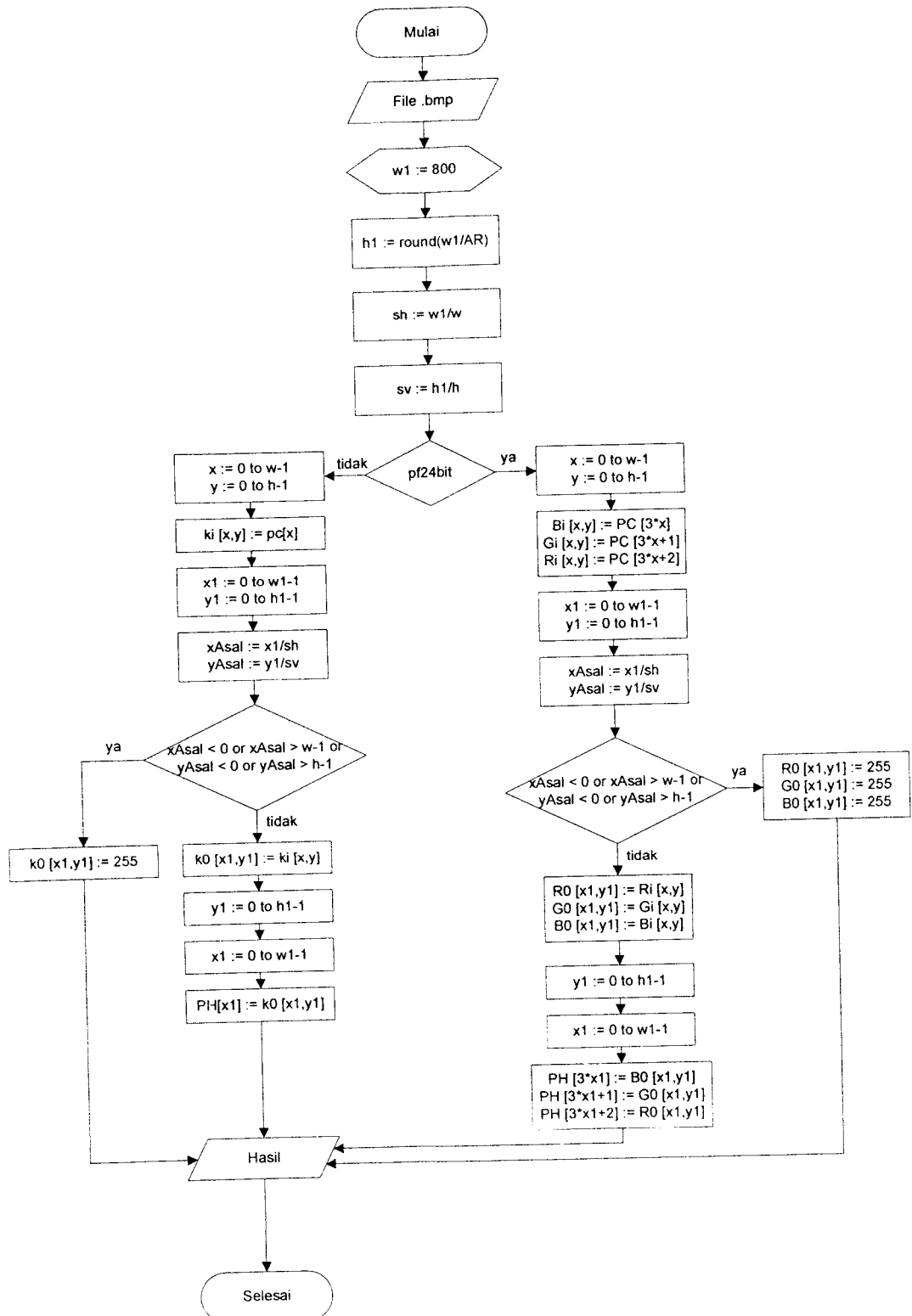
Pada Gambar 4.5, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flowchart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **File .bmp**, membuka *file* gambar yang akan diproses.
- 3) **w1:=100**, menetapkan nilai *Width (w1)* atau lebar gambar hasil
- 4) **h1 := Round(w1/AR)**, menghitung nilai *Height(h1)* atau tinggi gambar hasil.
- 5) **sh := w1/w**, menghitung skala pembesaran gambar ke arah horizontal
- 6) **sv := h1/h**, menghitung skala pembesaran gambar ke arah vertikal
- 7) **pf24bit**, jika iya lakukan langkah 8 sampai 22, jika tidak lakukan langkah 23 sampai 33
- 8) Tentukan koordinat asal nilai $x:=0$ to $w-1$ dan $y:=0$ to $h-1$
- 9) **Bi[x,y]:=PC[3*x]**, hitung nilai koordinat *Blue* titik hasil
- 10) **Gi[x,y]:=PC[3*x+1]**, hitung nilai koordinat *Green* titik hasil
- 11) **Ri[x,y]:=PC[3*x+2]**, hitung nilai koordinat *Red* titik hasil
- 12) Tentukan koordinat hasil nilai $x1:=0$ to $w1-1$ dan $y1:=0$ to $h1-1$
- 13) **xAsal := x1/sh**, hitung nilai koordinat sumbu x asal
- 14) **yAsal := y1/sv**, hitung nilai koordinat sumbu y asal
- 15) Periksa $xAsal < 0$ or $xAsal > w-1$ or $yAsal < 0$ or $yAsal > h-1$
- 16) Jika ya maka isi **R0[x1,y1] := 255, G0[x1,y1] := 255, B0[x1,y1] := 255**, lanjutkan proses selanjutnya. Jika tidak jalankan langkah 17 sampai 22
- 17) isi **R0[x1,y1] := Ri[x,y], G0[x1,y1] := Gi[x,y], B0[x1,y1] := Bi[x,y]**.

- 18) $y1 = 0$ to $h1-1$, hitung nilai koordinat y hasil.
- 19) $x1 = 0$ to $h1-1$, hitung nilai koordinat x hasil.
- 20) $PH[3*x1] := B0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 21) $PH[3*x1+1] := G0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 22) $PH[3*x1+2] := B0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 23) Tentukan nilai koordinat asal $x:=0$ to $w-1$ dan $y:=0$ to $h-1$
- 24) Tentukan nilai $ki[x,y]:=pc[x]$
- 25) Tentukan nilai koordinat hasil $x1:=0$ to $w1-1$ dan $y1:=0$ to $h1-1$
- 26) $xAsal := x1/sh$, hitung nilai koordinat sumbu x asal
- 27) $yAsal := y1/sv$, hitung nilai koordinat sumbu y asal
- 28) Periksa $xAsal < 0$ or $xAsal > w-1$ or $yAsal < 0$ or $yAsal > h-1$
- 29) Jika ya maka isi $k0[x1,y1] := 255$, lanjutkan proses selanjutnya. Jika tidak jalankan langkah 30 sampai 33
- 30) isi $k0[x1,y1] := Ri[x,y]$.
- 31) $y1 = 0$ to $h1-1$, hitung nilai koordinat y hasil.
- 32) $x1 = 0$ to $h1-1$, hitung nilai koordinat x hasil.
- 33) $PH[x1] := k0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 34) **Hasil**, nilai picture hasil.
- 35) **Selesai**, penanda akhir dari semua proses.

4.2.7 Flowchart pengaturan ukuran gambar pada interface

Setelah pola strimin dibentuk, maka langkah selanjutnya adalah menghadirkannya ke dalam *interface* program yang ada. Dalam hal ini ukuran pola strimin yang sudah dibentuk akan ditampilkan dalam *interface* dengan ukuran 800 pixel untuk lebar gambar dan tinggi gambar menyesuaikan dengan perbandingan ukuran tinggi gambar aslinya. Ketentuan ini berlaku untuk gambar dengan format pixel 8 bit dan 24 bit. Setelah masing-masing gambar dipisahkan antara yang 8 bit dan 24 bit maka pola strimin yang telah dibuat dimasukkan kedalam *interface* yang ada. *Flowchart* pengaturan ukuran gambar pada *interface* dapat dilihat pada gambar 4.6



Gambar 4.6 Flowchart pengaturan ukuran gambar pada interface

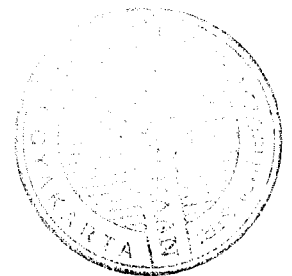
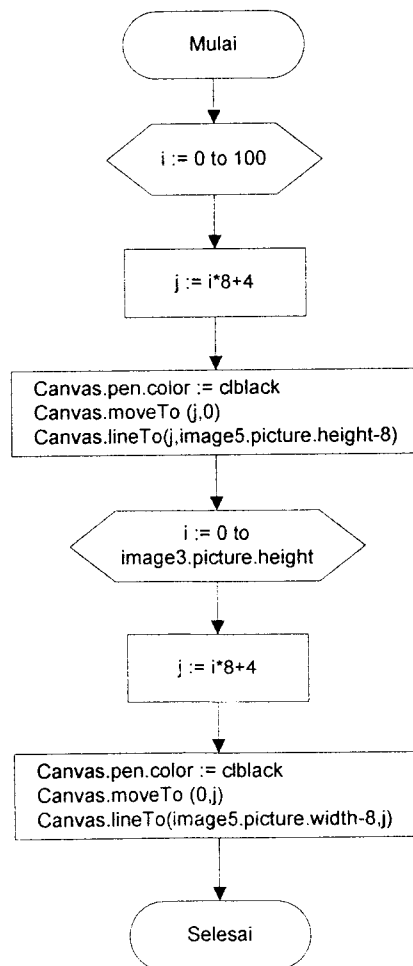
Pada Gambar 4.6, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flowchart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **File .bmp**, membuka *file* gambar yang akan diproses.
- 3) **w1:=800**, menetapkan nilai *Width (w1)* atau lebar gambar hasil
- 4) **h1 := Round(w1/AR)**, menghitung nilai *Height(h1)* atau tinggi gambar hasil.
- 5) **sh := w1/w**, menghitung skala pembesaran gambar ke arah horizontal
- 6) **sv := h1/h**, menghitung skala pembesaran gambar ke arah vertikal
- 7) **pf24bit**, jika iya lakukan langkah 8 sampai 22, jika tidak lakukan langkah 23 sampai 33
- 8) Tentukan koordinat asal nilai $x:=0$ to $w-1$ dan $y:=0$ to $h-1$
- 9) **Bi[x,y]:=PC[3*x]**, hitung nilai koordinat *Blue* titik hasil
- 10) **Gi[x,y]:=PC[3*x+1]**, hitung nilai koordinat *Green* titik hasil
- 11) **Ri[x,y]:=PC[3*x+2]**, hitung nilai koordinat *Red* titik hasil
- 12) Tentukan koordinat hasil nilai $x1:=0$ to $w1-1$ dan $y1:=0$ to $h1-1$
- 13) **xAsal := x1/sh**, hitung nilai koordinat sumbu x asal
- 14) **yAsal := y1/sv**, hitung nilai koordinat sumbu y asal
- 15) Periksa **xAsal < 0 or xAsal > w-1 or yAsal < 0 or yAsal > h-1**
- 16) Jika ya maka isi **R0[x1,y1] := 255, G0[x1,y1] := 255, B0[x1,y1] := 255**, lanjutkan proses selanjutnya. Jika tidak jalankan langkah 17 sampai 22
- 17) isi **R0[x1,y1] := Ri[x,y], G0[x1,y1] := Gi[x,y], B0[x1,y1] := Bi[x,y]**.

- 18) $y1 = 0$ to $h1-1$, hitung nilai koordinat y hasil.
- 19) $x1 = 0$ to $h1-1$, hitung nilai koordinat x hasil.
- 20) $PH[3*x1] := B0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 21) $PH[3*x1+1] := G0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 22) $PH[3*x1+2] := B0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 23) Tentukan nilai koordinat asal $x:=0$ to $w-1$ dan $y:=0$ to $h-1$
- 24) Tentukan nilai $ki[x,y]:=pc[x]$
- 25) Tentukan nilai koordinat hasil $x1:=0$ to $w1-1$ dan $y1:=0$ to $h1-1$
- 26) $xAsal := x1/sh$, hitung nilai koordinat sumbu x asal
- 27) $yAsal := y1/sv$, hitung nilai koordinat sumbu y asal
- 28) Periksa $xAsal < 0$ or $xAsal > w-1$ or $yAsal < 0$ or $yAsal > h-1$
- 29) Jika ya maka isi $k0[x1,y1] := 255$, lanjutkan proses selanjutnya. Jika tidak jalankan langkah 30 sampai 33
- 30) isi $k0[x1,y1] := Ri[x,y]$.
- 31) $y1 = 0$ to $h1-1$, hitung nilai koordinat y hasil.
- 32) $x1 = 0$ to $h1-1$, hitung nilai koordinat x hasil.
- 33) $PH[x1] := k0[x1,y1]$, isi picture hasil dengan hasil perhitungan.
- 34) Hasil, nilai picture hasil.
- 35) Selesai, penanda akhir dari semua proses.

4.2.8 Flowchart menggambar grid

Ukuran pola strimin yang dibuat diatur menjadi 100 pixel untuk lebar gambar yang diinputkan dengan tinggi gambar yang menyesuaikan dengan perbandingan ukuran gambar aslinya. Setelah menghitung nilai i , diteruskan dengan mencari nilai j dengan cara memasukkan nilai i yang sudah ada kemudian dikalikan 8 dan ditambah 4. langkah selanjutnya adalah menggambar garis vertikal. Hal yang sama dilakukan untuk menggambar garis horizontal hanya saja banyaknya garis horizontal yang digambar menyesuaikan dengan perbandingan ukuran aslinya. *Flowchart* menggambar grid strimin dapat dilihat pada gambar 4.7



Gambar 4.7 Flowchart menggambar grid.

Pada Gambar 4.7, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flow chart* tersebut :

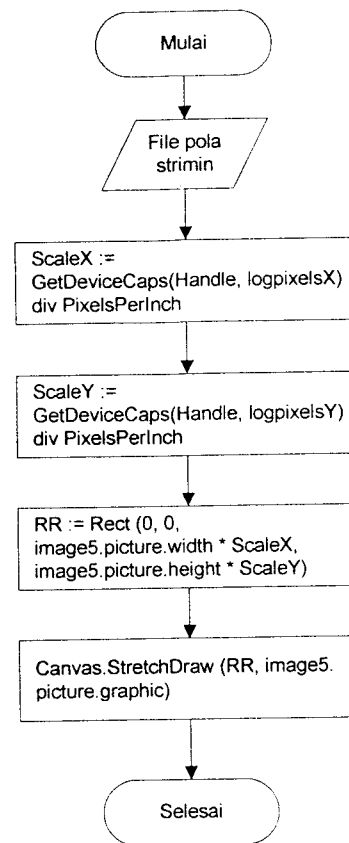
- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) **i:=0 to 100**, untuk memberikan nilai awal i sampai dengan 100.
- 3) **j:=i*8+4**, untuk menghitung letak 100 garis vertikal yang akan digambar mulai dari kiri ke kanan. 8 merupakan koefisien (1 kotak strimin = 8 pixel), 4 adalah setengah dari kotak strimin agar garis yang digambar terletak di sisi kanan kotak strimin
- 4) **canvas.pen.color := clblack, canvas.moveTo (j,0),
canvas.lineTo (j,image5.picture.height-8)**,
untuk menggambar garis vertikal sebanyak 100 garis dari kiri ke kanan menggunakan property **pen** dengan garis berwarna hitam digambar pada image5 (image yang akan tampak pada antarmuka program).
- 5) **i:=0 to image3.picture.height**, untuk memberikan nilai awal i pada pembuatan garis horizontal sebanyak n sesuai dengan perbandingan tinggi gambar pada image3 (image yang berisi gambar yang telah dikenai operasi pengontrasan, tidak tampak pada antarmuka program).
- 6) **j:=i*8+4**, untuk menghitung banyak dan letak garis horizontal yang akan digambar mulai dari atas ke bawah. 8 merupakan koefisien (1 kotak strimin = 8 pixel), 4 adalah setengah dari kotak strimin agar garis yang digambar terletak di sisi bawah kotak strimin.
- 7) **canvas.pen.color := clblack, canvas.moveTo (j,0),
canvas.lineTo (j,image5.picture.height-8)**,

untuk menggambar garis horizontal sebanyak n garis dari atas ke bawah menggunakan property **pen** dengan garis berwarna hitam digambar pada `image5` (image yang akan tampak pada antarmuka program).

8) **Selesai**, penanda akhir dari semua proses.

4.2.9 Flowchart prosedur cetak hasil

Proses pencetakan hasil pengolahan gambar menjadi pola strimin dilakukan dengan menghitung skala *pixel* dalam nilai x dan y , dimana x mewakili nilai *width* dan y mewakili nilai *height* gambar. Proses pencetakan bisa dilakukan langsung melalui printer yang aktif di komputer pengguna atau disimpan terlebih dalam *file microsoft office document image writer*. *Flowchart* cetak hasil strimin dapat dilihat pada gambar 4.8



Gambar 4.8 Flowchart mencetak gambar pola strimin.

Pada Gambar 4.8, dapat dijelaskan pemakaian simbol-simbol yang terdapat pada *flow chart* tersebut :

- 1) **Mulai**, penanda awal proses dikerjakan.
- 2) *File* pola strimin yang akan dicetak.
- 3) Hitung pixel yang ada pada gambar pada posisi sumbu x kemudian periksa perangkat alat cetak yang aktif di komputer pengguna.
- 4) Hitung pixel yang ada pada gambar pada posisi sumbu y kemudian periksa perangkat alat cetak yang aktif di komputer pengguna.
- 5) Jalankan **rect (0,0,image5.picture.width*scaleX, image5.picture.height*scaleY**

- 6) **canvas.stretchdraw (RR, image5.picture.graphic)**, gambar dengan metode stretchdraw sesuai ukuran pada RR di image5.
- 7) **Selesai**, penanda akhir dari semua proses.

4.3 Rancangan Antarmuka

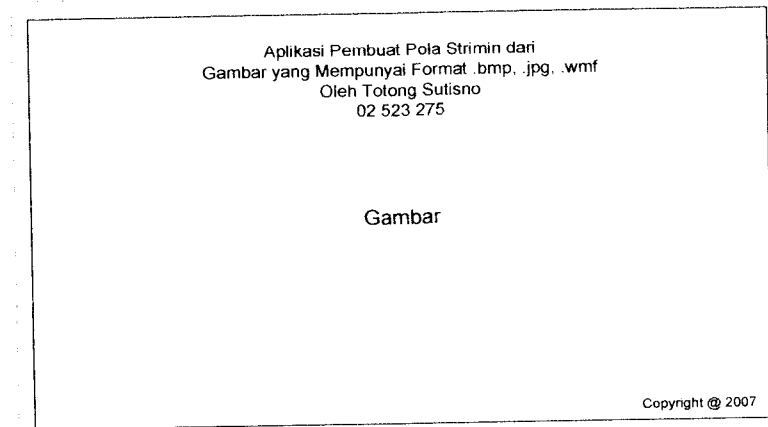
Perancangan antarmuka dimaksudkan untuk menggambarkan desain tampilan dari sistem, ilustrasi dari rancangan *interface* terhadap sistem yang akan diaplikasikan.

Rancangan antarmuka dari perangkat lunak untuk aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0. yang dibutuhkan diantaranya adalah splash awal, form halaman depan, form halaman utama, splash penutup dan form lainnya yang dibutuhkan kemudian.

4.3.1 Perancangan Antarmuka Splash awal

Merupakan *splash screen* saat program pertama kali dijalankan oleh pengguna. Tujuan dari adanya form *splash screen* adalah memberitahukan kepada pengguna bahwa *icon/shortcut* dari program berhasil di klik dan pengguna menunggu beberapa saat sebelum masuk ke halaman depan.

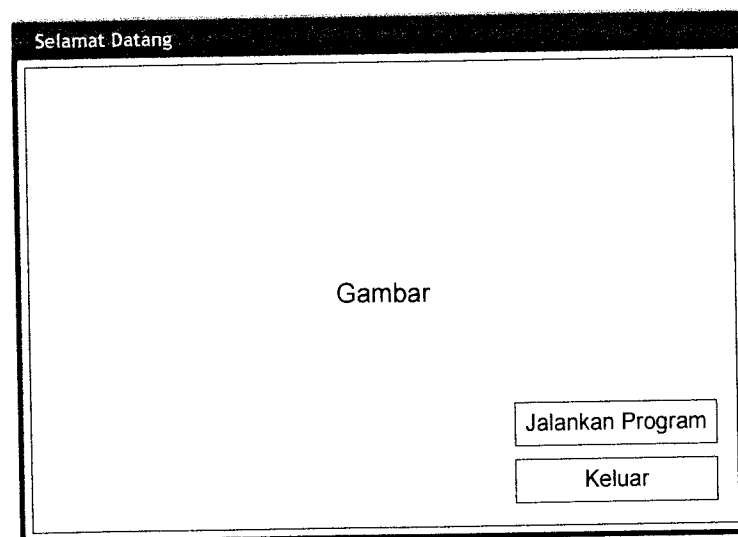
Dalam *splash screen* ini juga ditampilkan informasi nama aplikasi yang dijalankan, nama pembuat aplikasi dan tahun pembuatan aplikasi. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.9



Gambar 4.9 Antarmuka splash screen program

4.3.2 Perancangan Antarmuka Halaman Depan

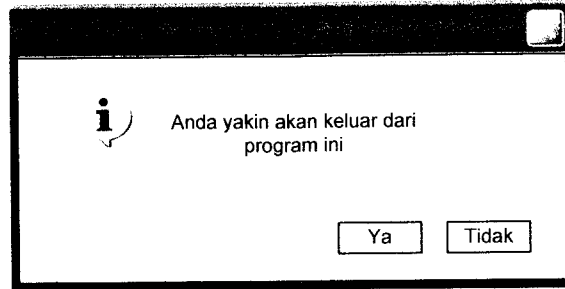
Merupakan *form* pertama yang dijumpai ketika program aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0 dijalankan. Pada *form* ini terdapat satu gambar yang berisi sekilas tentang program dan dua buah tombol yaitu tombol jalankan program, yang berfungsi untuk masuk ke program utama dan tombol keluar yang berfungsi untuk keluar dari program. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.10



Gambar 4.10 Antarmuka halaman depan

4.3.3 Perancangan Antarmuka konfirmasi keluar dari program

Merupakan dialog box konfirmasi yang bertujuan untuk memastikan pilihan pengguna untuk keluar dari program. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.11



Gambar 4.11 Antarmuka konfirmasi keluar dari program

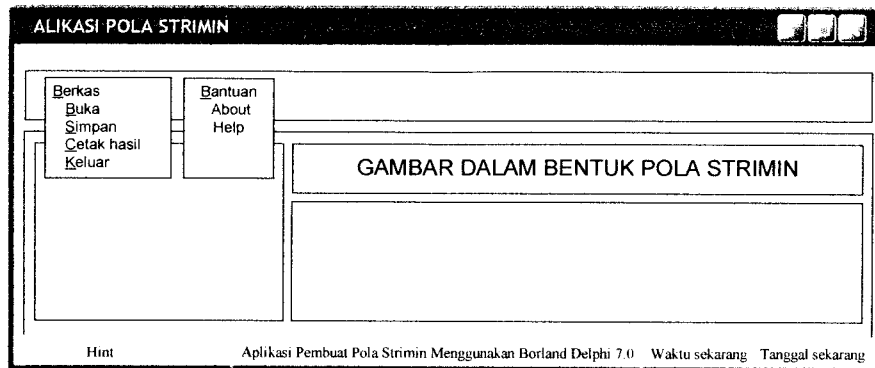
4.3.4 Perancangan Antarmuka menu utama program

Merupakan *form* utama ketika program aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0 dijalankan. Pada *form* ini terdapat menu bantuan yang berisi *file* panduan menjalankan program dan satu buah menu utama yang berisi :

- a. **Buka**, berfungsi untuk membuka dan memasukkan gambar yang akan diolah ke dalam program. Adapun format gambar yang dapat dibuka dan diolah menjadi pola strimin adalah gambar dengan format .bmp, .jpg dan .wmf.
- b. **Simpan**, berfungsi untuk menyimpan *file* gambar yang telah diolah menjadi gambar pola strimin dengan nama *file* disesuaikan dengan keinginan pengguna namun tetap dalam format .bmp.
- c. **Cetak hasil**, berfungsi untuk mencetak hasil pengolahan gambar menjadi pola strimin.

- d. **Keluar**, berfungsi untuk keluar dari program.
- e. **About**, berisi data pembuat program aplikasi
- f. **Help**, berisi panduan menjalankan program aplikasi

Gambar rancangan antarmukanya dapat dilihat pada gambar 4.12



Gambar 4.12 Antarmuka menu utama program

4.3.5 Perancangan Antarmuka form utama program

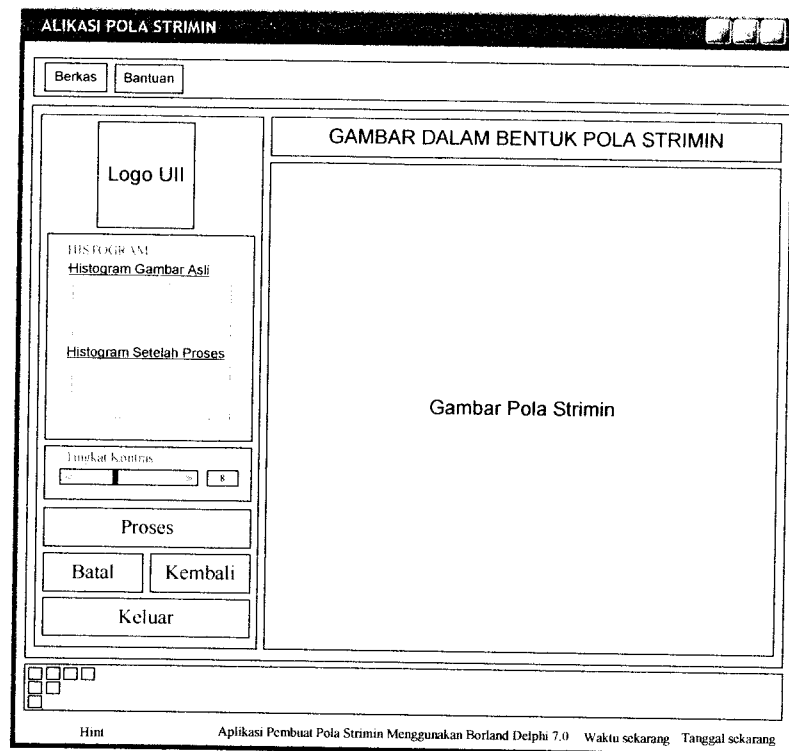
Merupakan *form* utama ketika program aplikasi pembuat pola strimin dari suatu gambar menggunakan borland delphi 7.0 dijalankan. Pada *form* ini terdapat beberapa menu dan tombol, diantaranya :

- a. *Image* yang berisi logo UII, apabila menu utama buka di klik untuk memasukkan *file* gambar yang akan diolah, maka *file* gambar sebelum diolah akan masuk ke *image* dan menimpa logo UII.
- b. *Panel* histogram citra, berfungsi untuk menampilkan histogram citra setelah diolah menjadi pola strimin.
- c. *Scrollbar* dan *edit*, berfungsi untuk memasukkan besarnya peningkatan kontras yang diinginkan oleh pengguna. Di sini sistem membatasi nilai tingkat kontras citra dari 1 sampai dengan 20

dikerenakan tingkat kontras tiap gambar yang berbeda. Cara memasukkan nilai tingkat citra dapat langsung di ketikkan di kotak edit atau dengan cara menggeser kursor yang ada di *scrollbar* ke kanan atau ke kiri.

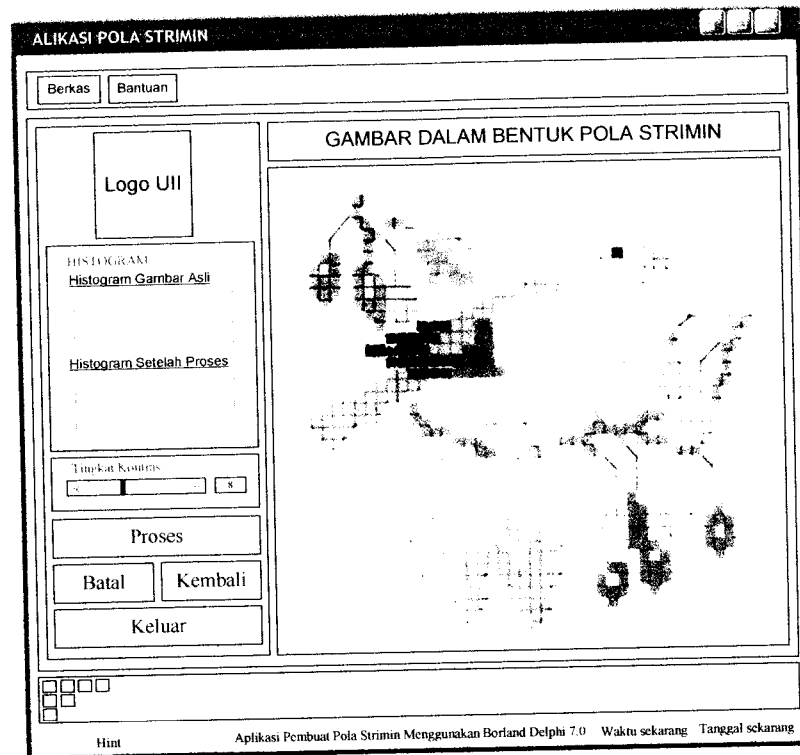
- d. Proses, tombol ini berfungsi untuk mengolah gambar menjadi pola strimin dengan syarat gambar sudah dimasukkan ke dalam program aplikasi, format gambar yang dapat di olah adalah .bmp, .jpg, .wmf dan nilai tingkat kontras sudah diisi antara 1 sampai dengan 20.
- e. Batal, tombol ini berfungsi untuk membatalkan semua perintah dan kembali ke halaman depan.
- f. Kembali, tombol ini berfungsi untuk membatalkan sebagian perintah dan kembali ke kondisi awal *form* tengah (kondisi *form* tengah nil).
- g. Keluar, tombol ini berfungsi untuk keluar dari program.
- h. Image, berfungsi sebagai tempat gambar yang sudah diolah menjadi pola strimin di tampilkan untuk dilihat oleh pengguna.
- i. *Statusbar*, dibagi menjadi 4 panels yang masing-masing diisi dengan hint, nama program aplikasi, waktu sekarang, tanggal sekarang.

Gambar rancangan antarmukanya dapat dilihat pada gambar 4.13



Gambar 4.13 Antarmuka form utama program

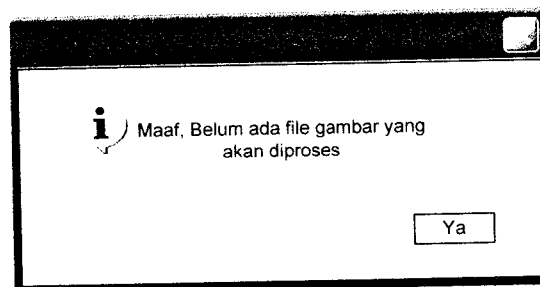
Tombol proses, tombol ini berfungsi untuk mengolah gambar menjadi pola strimin dengan syarat gambar sudah dimasukkan ke dalam aplikasi, format gambar yang dapat diolah adalah .bmp, .jpg, .wmf dan nilai tingkat kontras sudah diisi antara 1 sampai dengan 20. setelah tombol proses diklik akan tampak hasil gambar pola strimin seperti terlihat pada gambar 4.14



Gambar 4.14 Antarmuka gambar pola strimin

4.3.6 Perancangan Antarmuka informasi kesalahan proses

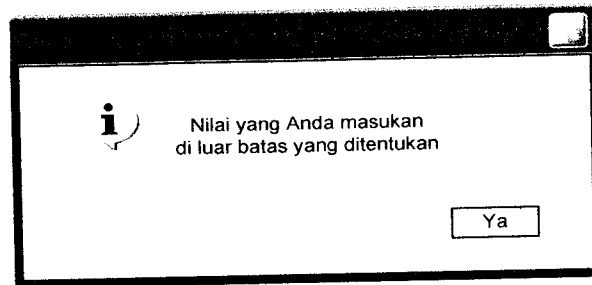
Merupakan dialog box informasi kesalahan yang bertujuan untuk memberitahukan pengguna bahwa *file* gambar yang akan diolah belum di masukkan. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.15



Gambar 4.15 Antarmuka informasi kesalahan proses

4.3.7 Perancangan Antarmuka informasi kesalahan input nilai kontras

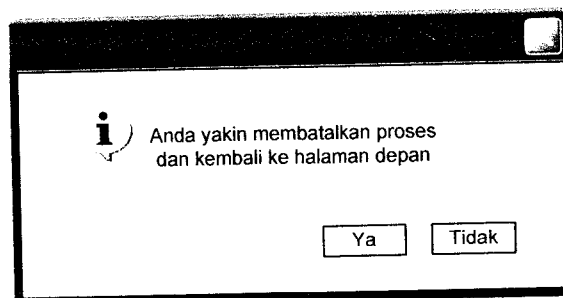
Merupakan dialog box informasi kesalahan yang bertujuan untuk memberitahukan pengguna bahwa nilai kontras yang dimasukkan kurang dari nilai minimal atau melebihi batas nilai maksimal yang telah ditentukan. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.16



Gambar 4.16 Antarmuka informasi kesalahan input nilai kontras

4.3.8 Perancangan Antarmuka informasi pembatalan

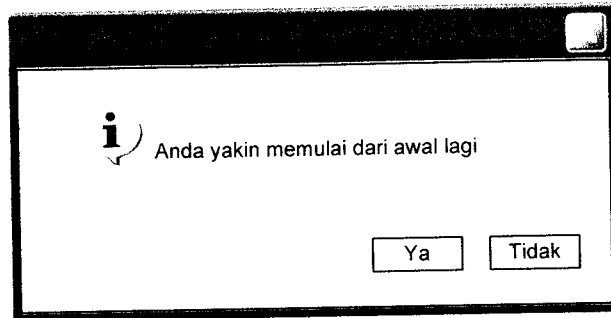
Merupakan dialog box informasi yang bertujuan untuk menanyakan kepada pengguna apakah proses dibatalkan dan akan kembali ke *form* halaman depan. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.17



Gambar 4.17 Antarmuka konfirmasi pembatalan

4.3.9 Perancangan Antarmuka konfirmasi kembali

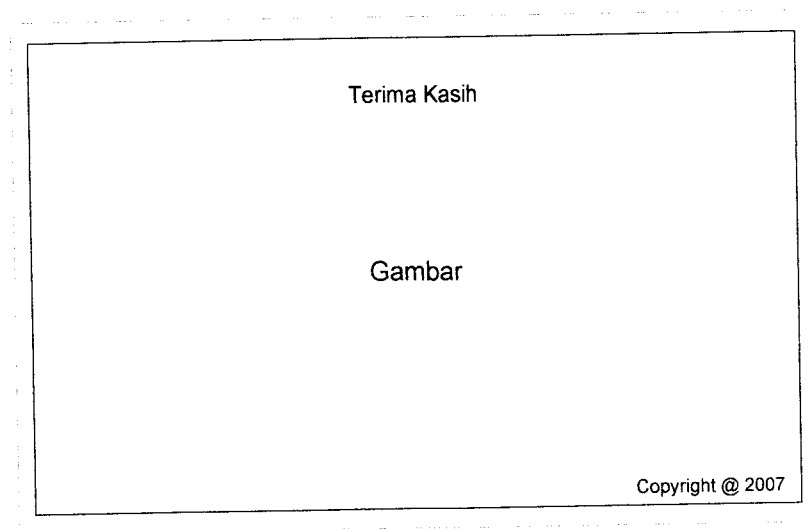
Merupakan dialog box informasi yang bertujuan untuk menanyakan kepada pengguna apakah proses dijalankan kembali serta memulai dari kondisi awal *form* utama. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.18



Gambar 4.18 Antarmuka konfirmasi kembali

4.3.10 Perancangan Antarmuka Halaman Depan

Merupakan *splash screen* untuk keluar saat tombol keluar atau menu keluar di klik. Pada bagian ini hanya berisi gambar dan ucapan terima kasih. Gambar rancangan antarmukanya dapat dilihat pada gambar 4.19



Gambar 4.19 Antarmuka splash screen keluar dari program

4.4 Rancangan Struktur Data

Beberapa perintah yang digunakan dalam pembuatan program pembuat pola strimin adalah sebagai berikut :

1. Perintah untuk memainkan *file* multimedia dengan format .wav mulai dari awal sampai selesai diulang berkali-kali.

```
sndPlaySound('DK.wav', SND_ASYNC or SND_LOOP);
```

2. Perintah untuk menghentikan perintah menjalankan *file* multimedia.

```
playsound(nil, 0, snd_async);
```

3. Perintah untuk mengeset tampilan *form* tanpa caption dengan menggunakan windowlong

```
SetWindowLong(Handle, gwl_style, GetWindowLong  
(handle, gwl_style) and NOT ws_caption);
```

4. Merupakan format *file* citra skala keabuan. Citra skala keabuan direpresentasikan sebagai citra berindeks 8 bit dengan komponen palet warna merah, hijau dan biru yang bernilai sama, sehingga menampilkan warna keabuan dari hitam sampai putih. Sintaxnya sebagai berikut :

```
Pf8bit;
```

5. Merupakan format *file* citra *true color* (16 juta warna).

```
Pf24bit;
```

6. *Incremen*, perintah untuk menjalankan perhitungan naik sebanyak k

```
Inc[k];
```

7. Komponen dalam Delphi yang berfungsi untuk menampilkan citra yang mempunyai format bitmap (.bmp), metafile (.wmf atau .emf), ikon (.ico), dan format citra jpeg (.jpg).

```
TImage;
```

8. *Type Property* dari *TImage*, digunakan untuk menyimpan data dari suatu citra dan juga untuk menampilkan citra tersebut.

`TPicture;`

9. Komponen dalam Delphi, dalam hal ini digunakan untuk menampilkan *file* gambar yang baru dibuka dengan tampilan *stretch*

`Image2;`

10. Komponen dalam Delphi, dalam hal ini digunakan untuk menampilkan *file* gambar yang telah dikenakan operasi kontras.

`Image3;`

11. Komponen dalam Delphi, dalam hal ini digunakan untuk menampilkan *file* gambar yang telah selesai diolah menjadi pola strimin dengan ukuran lebar 800 pixel dan tinggi menyesuaikan sesuai dengan perbandingan pada *image3*.

`Image5;`

12. Ukuran untuk lebar dengan satuan *pixel*

`Width;`

13. Ukuran untuk tinggi dengan satuan *pixel*

`Height;`

14. Perintah yang digunakan apabila pernyataan selesai dan memulai lagi dari langkah awal di tengah sebuah operasi yang belum selesai dijalankan.

`Break;`

15. *Property* dalam Delphi yang digunakan untuk memberikan warna.

`Color;`

16. *Property mode* yang digunakan untuk menggambar kotak.

`Rectangle;`

17. Digunakan untuk membuat/menggambar sesuatu baik dengan sendirinya maupun dengan parameter yang telah ditentukan.

`Create;`

18. Model yang digunakan windows untuk merepresentasikan warna dimana setiap warna dinyatakan dengan besarnya intensitas warna merah, hijau, biru.

`RGB (Red, Green, Blue);`

19. Kelas dalam VCL Delphi yang digunakan sebagai pembungkus fungsi-fungsi GDI (*Graphics Device Interface*).

`Canvas;`

20. *Property* dari kelas TCanvas dan juga merupakan intensitas dari kelas TPen. Pen digunakan untuk menulis atau menggambar dalam delphi

`Pen;`

21. Metode untuk menampilkan citra dan menggambar suatu objek yang diciptakan dari salah satu kelas anak TGraphic. Kelas anak dari TGraphic adalah TBitmap, TMetafile, TIcon, TJPEGImage.

`Draw dan StretchDraw;`

22. Perintah yang digunakan untuk mengambil current dir (0,x).

`GetDir (0,x);`

23. Perintah ini digunakan untuk mengkonversi huruf menjadi huruf kapital.

`Uppercase;`

24. Perintah ini digunakan untuk menyamakan/menimpa *file* yang dituju (jpg).

`assign (jpg);`

25. Perintah ini digunakan untuk menscan/mengambil satu baris gambar.

`Scanline;`

26. Perintah ini digunakan untuk mengubah nilai string ke dalam bentuk pecahan.

```
StrToFloat;
```

27. Perintah ini digunakan untuk pembulatan angka dalam perhitungan.

```
Round;
```

28. Perintah ini digunakan untuk pembulatan angka hasil perhitungan menuju ke arah nol.

```
Floor;
```

29. Perintah ini digunakan untuk pembulatan angka dalam perhitungan ke arah nilai yang lebih besar atau pembulatan ke atas.

```
Ceil;
```

30. Perintah yang digunakan untuk memindahkan pointer gambar ke suatu tempat.

```
moveTo;
```

31. Perintah yang digunakan untuk menarik garis lurus dari posisi pointer sekarang ke tempat tertentu.

```
lineTo;
```

32. Perintah yang digunakan untuk menghitung skala x dan skala y.

```
scaleX;
```

```
scaleY;
```

33. Perintah yang digunakan untuk menentukan isi dimensi array.

```
Setlength;
```

34. Perintah yang digunakan untuk memulai dan mengakhiri dokumen.

```
beginDoc;
```

```
endoc;
```


35. Perintah yang digunakan untuk membaca data alat (yang aktif).

```
Getdevicecaps;
```

36. Perintah ini digunakan untuk membuat file temporari/file sementara.

```
Temp;
```

37. Perintah yang digunakan untuk mencoba proses perhitungan, jika error maka diarahkan ke perintah di bawah *except* dan jika benar maka menuju ke perintah *finally*.

```
Try;
```

```
Finally;
```

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Batasan implementasi pada perangkat lunak maupun perangkat keras yang digunakan adalah sebagai berikut :

A. Software yang digunakan untuk membuat aplikasi adalah :

1. Borland Delphi 7.0

Perangkat lunak ini digunakan karena memiliki kemampuan yang bisa digunakan untuk komputasi perhitungan, visualisasi, teknik dan pemrograman.

2. Sistem Operasi (Operating system)

Windows XP merupakan sistem operasi yang dapat digunakan untuk mengoperasikan sistem ini.

3. Adobe Photoshop CS

Perangkat lunak ini digunakan untuk mendukung pembuatan interface.

B. Spesifikasi minimal hardware yang digunakan untuk membuat aplikasi adalah :

1. Processor Intel Pentium III ke atas

2. RAM 256 MB

3. VGA 16 MB

4. *Free space disk* minimal 35 MB
5. Mouse dan keyboard
6. Printer berwarna.

5.2 Implementasi Perangkat Lunak

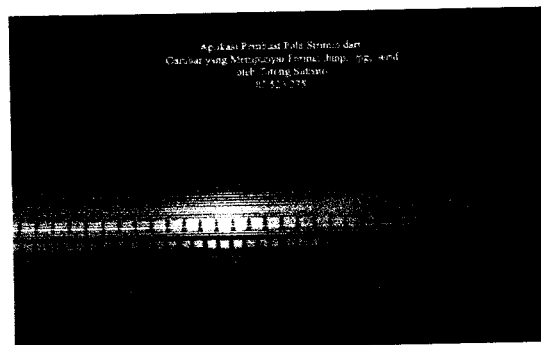
5.2.1 Pengembangan Antarmuka Perangkat Lunak

Pengembangan antarmuka perangkat lunak telah di desain sedemikian rupa sehingga mempermudah pengguna agar tidak mengalami kesulitan dalam menggunakan aplikasi ini.

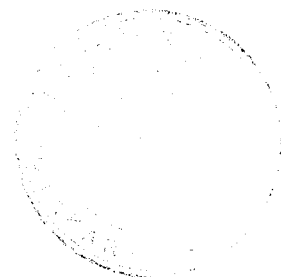
5.2.1.1 Form Splash Screen Awal

Merupakan *splash screen* saat program pertama kali dijalankan oleh pengguna. Tujuan dari adanya form *splash screen* adalah memberitahukan kepada pengguna bahwa *icon/shortcut* dari program berhasil di klik dan pengguna menunggu beberapa saat sebelum masuk ke halaman depan.

Dalam *splash screen* ini juga ditampilkan informasi nama aplikasi yang dijalankan, nama pembuat aplikasi dan tahun pembuatan aplikasi.



Gambar 5.1 Form Splash Screen Awal



5.2.1.2 Form Halaman Depan

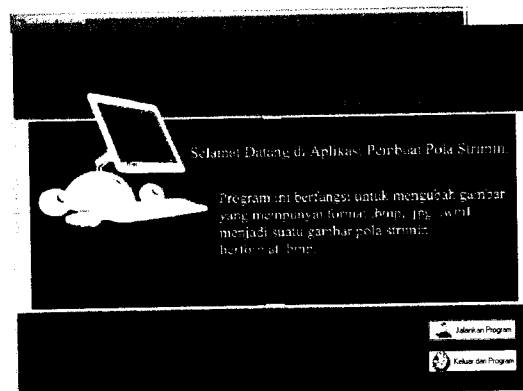
Merupakan *form* yang pertama kali muncul setelah *splash screen* selesai dijalankan. *Form* ini berisi selamat datang dan penjelasan singkat tentang program. *Form* ini berisi menu sebagai berikut :

1. Jalankan program

Apabila tombol menu 'Jalankan Program' di klik maka akan masuk ke *form* utama.

2. Keluar dari program

Apabila tombol menu 'Keluar Dari Program' di klik maka akan muncul dialog pesan untuk memastikan pilihan pengguna apakah keluar atau tidak. Apabila tombol 'Yes' di klik maka keluar dari program dan tombol 'No' di klik maka akan tetap berada di *form* halaman depan



Gambar 5.2 Form Halaman depan

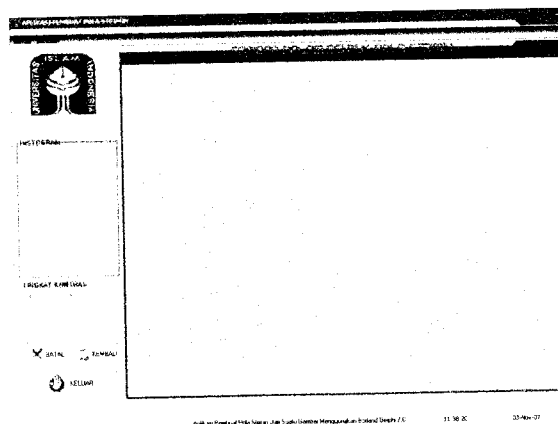
5.2.1.3 Form Utama

Merupakan *form* yang muncul setelah tombol 'Jalankan Program' di klik. *Form* ini berisi 2 buah menu sebagai berikut :

1. Menu Berkas, berisi :
 - a. Buka
 - b. Simpan
 - c. Cetak Hasil
 - d. Keluar
2. Menu Bantuan, berisi :
 - a. About
 - b. Help

Pada *form* ini juga terdapat 4 buah tombol sebagai berikut :

1. Tombol Proses
2. Tombol Batal
3. Tombol Kembali
4. Tombol Keluar

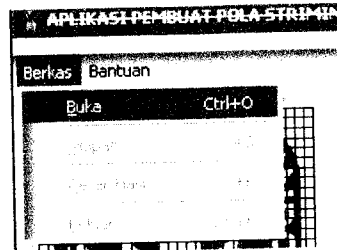


Gambar 5.3 Form Utama

Penjelasan kegunaan dari menu tersebut di atas adalah sebagai berikut :

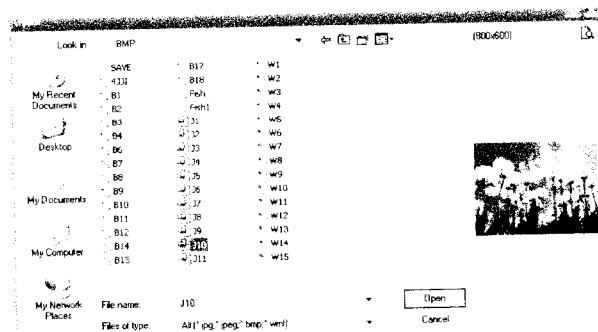
1. Berkas, berisi :

a. Buka



Gambar 5.4 Menu Buka Berkas

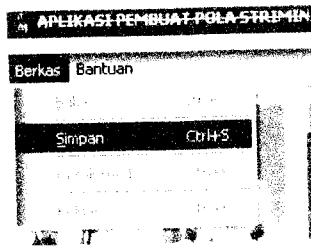
Apabila menu 'Buka' di klik maka akan tampil *form open* yang menunjukkan bahwa file gambar yang bisa di buka hanya gambar yang mempunyai format .bmp, .jpg, .wmf.



Gambar 5.5 Form Buka Berkas

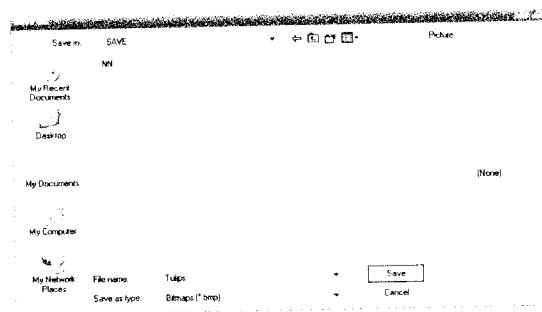
Setelah pilihan *open* di klik maka gambar baru bisa masuk ke program untuk dilakukan proses selanjutnya.

b. Simpan



Gambar 5.6 Menu Simpan Berkas

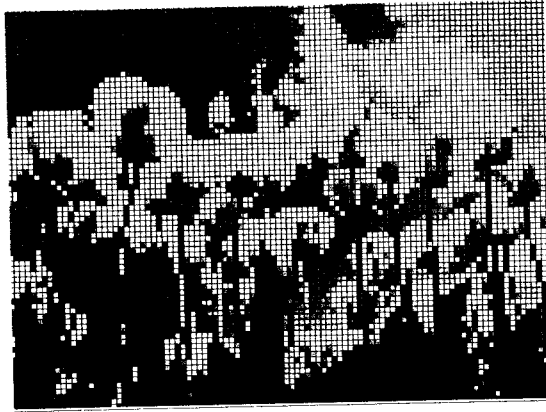
Apabila menu 'Simpan' di klik maka akan tampil *form save as* yang menunjukkan bahwa file gambar setelah di proses akan disimpan dalam direktori yang diinginkan pengguna dengan format gambar .bmp. besarnya ukuran gambar yang disimpan adalah 800 *pixel* untuk ukuran lebar dan ukuran tinggi gambar menyesuaikan dengan perbandingan gambar aslinya.



Gambar 5.7 Form Simpan Berkas

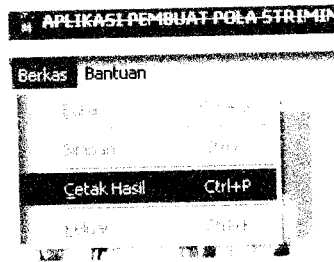
Setelah proses penyimpanan file gambar selesai maka program akan kembali ke halaman *form* utama. Setelah itu pengguna dapat melakukan pilihan-pilihan yang ada di *form* utama.

Contoh berkas gambar yang disimpan setelah proses perubahan menjadi pola strimin dapat dilihat pada gambar 5.8 berikut.



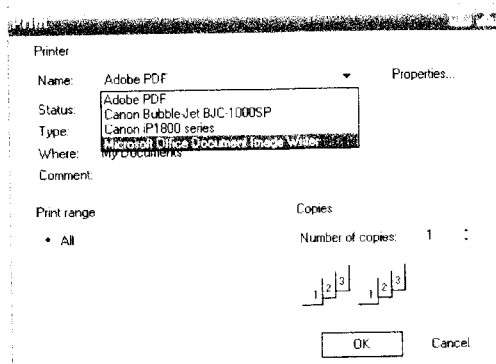
Gambar 5.8 Pola Strimin Gambar

c. Cetak Hasil



Gambar 5.9 Menu Cetak Hasil

Apabila menu 'Cetak Hasil' di klik maka akan tampil *form Print* yang menunjukkan bahwa file gambar akan dicetak pada printer yang aktif dalam komputer pengguna atau disimpan terlebih dahulu dalam berkas *adobe PDF* atau *MS Office Document Image Writer*.

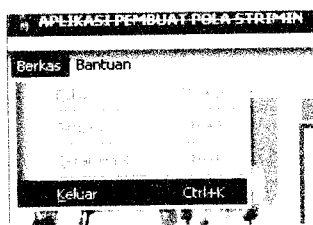


Gambar 5.10 Form Cetak Hasil

Setelah proses pencetakan *file* pola strimin selesai maka program akan kembali ke halaman *form* utama. Setelah itu pengguna dapat melakukan pilihan-pilihan yang ada di *form* utama.

Contoh hasil cetak untuk file input berkas gambar .bmp, .jpg, .wmf dapat dilihat pada lampiran A halaman A-1, lampiran B halaman B-1, lampiran C halaman C-1. Proses mencetak hasil menggunakan printer canon iP1800 series

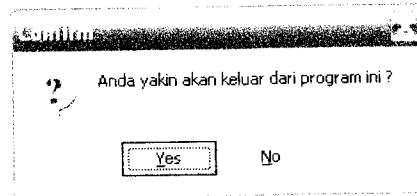
d. Keluar



Gambar 5.11 Menu Keluar Program

Apabila menu 'Keluar' di klik maka akan muncul dialog pesan konfirmasi untuk memastikan pilihan pengguna apakah keluar atau tidak. Apabila tombol 'No' di klik maka akan tetap berada di *form* utama

program dan apabila tombol 'Yes' di klik maka akan muncul *splash screen* keluar dan keluar dari program.

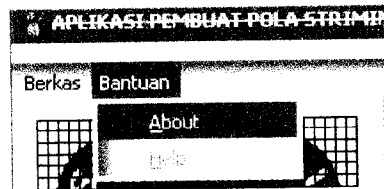


Gambar 5.12 Form Konfirmasi Keluar Program

2. Bantuan, berisi :

a. *About*

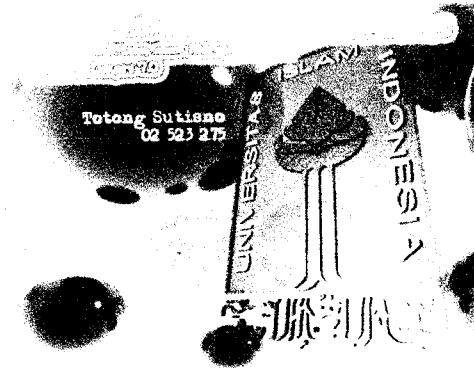
Form about merupakan *form* yang berisi tentang identitas dari pembuat aplikasi pembuat pola strimin dari gambar yang mempunyai format .bmp, .jpg, .wmf menggunakan borland delphi 7.0.



Gambar 5.13 Menu About

Apabila menu '*About*' di klik maka akan tampil *form about*. Selama *form about* aktif maka *form utama* tidak bisa dijalankan, dan setelah *form about* ditutup maka *form utama* dapat di jalankan kembali.

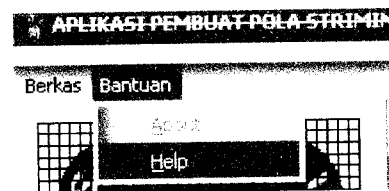
Penje



Gambar 5.14 Form About

b. *Help*

Berisi panduan dan penjelasan untuk menjalankan program aplikasi pembuat pola strimin.

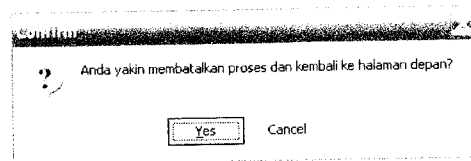


Gambar 5.15 Menu Help

Apabila menu '*Help*' di klik maka akan tampil *file help*. Selama *file help* aktif maka *form* utama tidak bisa dijalankan, dan setelah *file help* ditutup maka *form* utama dapat di jalankan kembali.

2. Tombol Batal

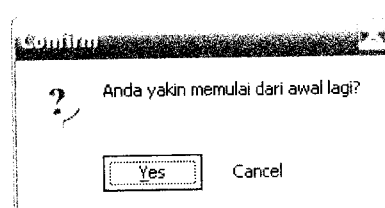
Apabila tombol 'Batal' di klik maka akan muncul dialog pesan konfirmasi untuk memastikan pilihan pengguna. Apabila tombol 'Yes' diklik maka program akan kembali ke *form* halaman depan. Dan apabila tombol 'cancel' diklik maka program akan tetap berada di *form* utama.



Gambar 5.17 Form Konfirmasi Pembatalan

3. Tombol Kembali

Apabila tombol 'Kembali' di klik maka akan muncul dialog pesan konfirmasi untuk memastikan pilihan pengguna. Apabila tombol 'Yes' diklik maka program akan kembali pada keadaan sebelum tombol proses atau menu buka dijalankan, tetapi jika tombol 'cancel' yang diklik maka program akan kembali pada keadaan sebelum tombol 'Kembali' diklik.



Gambar 5.18 Form Konfirmasi Kembali

4. Tombol Keluar

Apabila tombol 'Keluar' di klik maka akan muncul dialog pesan konfirmasi untuk memastikan pilihan pengguna apakah keluar atau tidak. Apabila tombol 'No' di klik maka akan tetap berada di *form* utama program dan apabila tombol 'Yes' di klik maka akan muncul *splash screen* keluar dan keluar dari program.



Gambar 5.19 Tombol Keluar

5.2.1.4 Form Splash Screen Keluar

Merupakan *splash screen* saat keluar dari program aplikasi pembuat pola strimin dari suatu gambar. Dalam *splash screen* ini juga ditampilkan ucapan terima kasih telah menggunakan program ini dan juga tahun pembuatan program aplikasi pembuat pola strimin dari suatu gambar.



Gambar 5.20 Form Splash Screen Keluar

5.2.2 Implementasi Prosedural

5.2.2.1 Prosedur Buka

Prosedur buka berfungsi untuk memberikan perintah kepada program yang dibuat untuk membuka file gambar yang akan diubah menjadi pola strimin. Dalam prosedur ini juga terdapat *sintax* untuk mengkonversi format gambar jpg dan wmf menjadi format gambar bmp. Prosedur ini terdapat pada procedure `buka1click(sender:Tobject)` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur buka adalah sebagai berikut :

```

procedure TTengah.Buka1Click(Sender: Tobject);
var  jpg:tjpegimage;
     wmf:tmetafile;
begin
  GetDir(0,x);
  if OPD1.Execute then
  begin
    if uppercase(rightstr(OPD1.FileName,3))='WMF' then
    begin
      image1.Picture.LoadFromFile(x+'\fish.bmp');
      wmf:=TMetafile.Create;
      wmf.LoadFromFile(OPD1.FileName);
      image2.Picture.Bitmap.Height:=wmf.Height;
      image2.Picture.Bitmap.Width:=wmf.Width;
      image1.Picture.Bitmap.Height:=wmf.Height;
      image1.Picture.Bitmap.Width:=wmf.Width;
      image2.Picture.Bitmap.Canvas.Draw(0,0,wmf);
      image1.Picture.Bitmap.Canvas.Draw(0,0,wmf);
    end
  else
    if uppercase(rightstr(OPD1.FileName,3))='JPG' then
    begin
      jpg:=TJpegImage.Create;
      jpg.LoadFromFile(OPD1.FileName);
      image2.picture.Bitmap.Assign(jpg) ;
      image1.Picture.Bitmap.Assign(jpg);
    end
  else
  begin
    image2.Picture.loadfromfile(OPD1.filename);
    image1.Picture.loadfromfile(OPD1.filename);
  end;
  suibutton1.Enabled:=true;
end;
end;

```

5.2.2.2 Prosedur Tingkat Kontras

Prosedur tingkat kontras berfungsi untuk memberikan perintah batasan tingkat kontras kepada program yang dibuat untuk membatasi tingkat kontras dengan nilai minimal 1 dan nilai maksimal 20. Prosedur ini terdapat pada procedure `suiEdit1Change(sender:TObject)` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur `suiEdit1Change` adalah sebagai berikut :

```

procedure TTengah.suiEdit1Change(Sender: TObject);
var i : integer;
begin
  if SuiEdit1.Text <> '' then
  begin
    i := strtoint(SuiEdit1.Text);
    if (i >= SuiScrollbar1.min) and (i<=SuisScrollbar1.Max)
  then
    begin
      SuisScrollbar1.Position:=i;
      suibutton1.Enabled:=true;
    end
  else
    begin
      messagedlg('Nilai yang Anda masukan di luar batas yang
ditentukan',mtwarning,
      [mbok],0);
      suibutton1.Enabled:=false;
    end;
  end;
end;

```

5.2.2.3 Prosedur Peningkatan Kontras

Prosedur tingkat kontras berfungsi untuk memberikan perintah peningkatan nilai kontras suatu gambar. Idenya adalah membuat warna yang agak terang menjadi lebih terang dan warna yang agak gelap menjadi lebih gelap. Prosedur ini terdapat pada procedure `suiButton1Click(Sender: TObject)` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur `suiButton1Click` adalah sebagai berikut :

```

begin
  if (Image2.Picture.Bitmap.PixelFormat = pf8bit)
  then
    begin
      P := 128;
      G := StrToFloat(SuiEdit1.Text);
      for i:= 0 to Image2.Picture.Height-1 do
        begin
          PC := Image2.Picture.Bitmap.ScanLine[i];
          PH := Image1.Picture.Bitmap.ScanLine[i];
          for j:= 0 to Image2.Picture.Width-1 do
            begin
              temp := Round(G*(PC[j]-P)+P);
              if (temp > 255) then
                temp := 255;
              if (temp < 0) then
                temp := 0;
              PH[j] := temp;
            end;
          end;
        end;
      end;

      if (Image2.Picture.Bitmap.PixelFormat = pf24bit)
      then
        begin
          P := 128;
          G := StrToFloat(SuiEdit1.Text);
          for i:= 0 to Image2.Picture.Height-1 do
            begin
              PC := Image2.Picture.Bitmap.ScanLine[i];
              PH := Image1.Picture.Bitmap.ScanLine[i];
              for j:= 0 to Image2.Picture.Width-1 do
                begin
                  temp := Round(G*(PC[3*j]-P)+P);
                  if (temp > 255) then
                    temp := 255;
                  if (temp < 0) then
                    temp := 0;
                  PH[3*j] := temp;
                  temp := Round(G*(PC[3*j+1]-P)+P);
                  if (temp > 255) then
                    temp := 255;
                  if (temp < 0) then
                    temp := 0;
                  PH[3*j+1] := temp;
                  temp := Round(G*(PC[3*j+2]-P)+P);
                  if (temp > 255) then
                    temp := 255;
                  if (temp < 0) then
                    temp := 0;
                  PH[3*j+2] := temp;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```


5.2.2.4 Prosedur pengaturan ukuran gambar pada strimin

Prosedur ukuran gambar pada strimin berfungsi untuk memberikan perintah pembuatan kotak strimin sebanyak 100 kotak ke arah horizontal. Adapun banyak kotak pada arah vertikal diatur secara otomatis sesuai dengan perbandingan dengan tinggi gambar aslinya. Prosedur ini terdapat pada procedure `suiButton1Click(Sender: TObject)` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur `suiButton1Click` dapat dilihat pada lampiran D halaman D-1 sampai dengan D-3.

5.2.2.5 Prosedur pengaturan ukuran gambar pada antarmuka

Prosedur ukuran gambar pada antarmuka berfungsi untuk memberikan perintah penampakan pola strimin pada pengguna. Prosedur ini terdapat pada procedure `suiButton1Click(Sender: TObject)` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur `suiButton1Click` dapat dilihat pada lampiran E halaman E-1 dan E-2.

5.2.2.6 Prosedur Pembuatan Grid

Prosedur pembuatan grid berfungsi untuk memberikan perintah pembuatan garis vertikal dan horizontal atau kotak grid. Garis vertikal sebanyak 101 garis ke arah horizontal atau 100 kolom. Adapun banyak garis horizontal pada arah vertikal diatur secara otomatis sesuai dengan perbandingan dengan tinggi gambar aslinya. Prosedur ini terdapat pada procedure `NggambarGrid` yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur `NggambarGrid` adalah sebagai berikut:

```

Procedure TTengah.NggambarGrid;
var i,j:integer;

begin
  for i:=0 to 100 do
  begin
    if i<=1 then
      j:=i*4
    else
      j:=i*8-4;
      with image5 do
      begin
        Canvas.Pen.Color := clBlack;
        Canvas.MoveTo(j, 0);
        Canvas.LineTo(j,image5.Picture.Height-8);
      end
    end;
    for i:=0 to image3.Picture.Height do
    begin
      if i<=1 then
        j:=i*4
      else
        j:=i*8-4;
        with image5 do
        begin
          Canvas.Pen.Color := clBlack;
          Canvas.MoveTo(0,j);
          Canvas.LineTo(image5.Picture.Width-8,j);
        end
      end;
    end;
  end;
end;

```

5.2.2.7 Prosedur Cetak

Prosedur cetak berfungsi untuk memberikan perintah mencetak pola strimin. Prosedur ini terdapat pada procedure CetakHasil1Click(Sender: TObject) yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur CetakHasil1Click adalah sebagai berikut :

```

procedure TTengah.CetakHasil1Click(Sender: TObject);
var
  ScaleX, ScaleY: Integer;
  RR: TRect;
begin
  if PrintDialog1.Execute then
  begin
    with Printer do
    begin
      BeginDoc;
      try
        ScaleX := GetDeviceCaps(Handle, logPixelsX) div
PixelsPerInch;
        ScaleY := GetDeviceCaps(Handle, logPixelsY) div
PixelsPerInch;

        RR := Rect(0, 0, Image5.picture.Width * scaleX,
Image5.Picture.Height * ScaleY);
        Canvas.StretchDraw(RR, Image5.Picture.Graphic);
      finally
        EndDoc;
      end;
    end;
  end;
end;
end;

```

5.2.2.8 Prosedur Contoh Warna

Prosedur contoh warna berfungsi untuk mengambil sampel warna dari pola strimin yang telah dibuat serta menggabungkan beberapa warna yang mempunyai nilai RGB berdekatan. Prosedur ini terdapat pada Procedure TTengah.contohwarna yang terdapat pada unit Tengah. *Listing* selengkapnya prosedur contoh warna dapat dilihat pada lampiran F halaman F-1 dan F-2

BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pengujian Sistem

Pengujian sistem dilakukan untuk memeriksa kekompakan komponen sistem yang telah diimplementasikan. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian perlu dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan yang mungkin masih terjadi. Pengujian sistem termasuk juga pengujian program secara menyeluruh. Kumpulan program yang telah diintegrasikan perlu diuji kembali untuk melihat apakah suatu program dapat menerima input data dengan baik, dapat memproses data dengan baik dan memberikan *output* seperti yang diharapkan.

Untuk mengetahui kinerja dari aplikasi yang dibangun, maka perlu dilakukan pengujian dengan :

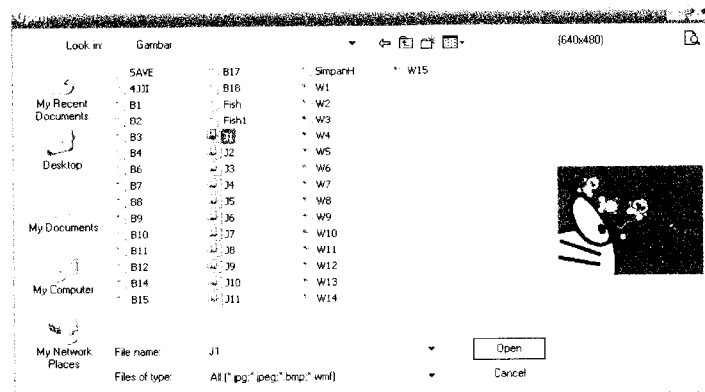
1. Data normal yaitu keadaan *input* dan proses yang benar.
2. Data tidak normal pada aplikasi akan dicoba dengan menjalankan program dengan tidak pada keadaan standar atau proses yang tidak berurutan.

6.1.1 Pengujian Data Normal

Pengujian dengan data normal dimaksudkan untuk mengetahui hasil *output* dari aplikasi yang dibangun telah sesuai dengan yang diharapkan. Namun, apabila pengisian dengan data normal *output* yang dihasilkan masih salah atau

tidak sesuai dengan harapan maka masih terdapat kesalahan dalam implementasi baik kesalahan algoritma, kesalahan logika, kesalahan asumsi atau yang lainnya.

6.1.1.1 Pengujian Input Gambar



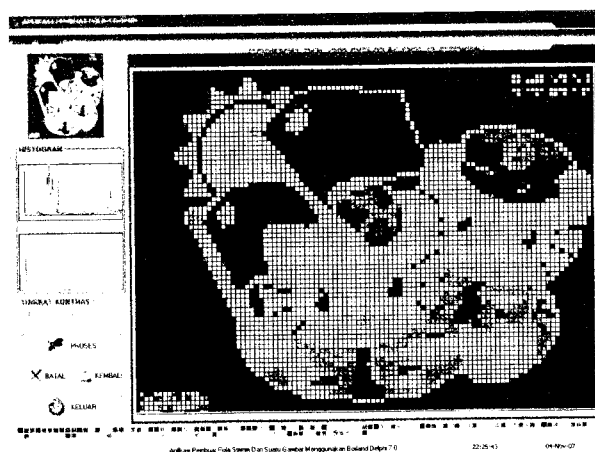
Gambar 6.1 Pembatasan File Gambar yang Dapat Diinputkan

Pada *form* ini terlihat bahwa file gambar yang dapat diinputkan dan diolah menjadi pola strimin hanya gambar yang mempunyai format .bmp, .jpg, .wmf.

6.1.1.2 Pengujian Dengan Input Gambar .bmp

Pada *form* ini proses pengolahan *file* gambar yang mempunyai format .bmp menjadi gambar pola strimin melalui pengujian pada tingkat kontras 1 sampai 20 dapat dilakukan. Pengujian dilakukan pada gambar yang mempunyai format .bmp yang mempunyai ukuran 800 x 600 *pixel*. Hasil dari pengolahan *file* .bmp dapat dilihat pada lampiran A halaman A-1.

Pengujian dengan data normal untuk proses pembuatan pola strimin dari gambar yang mempunyai format .bmp dapat dilakukan seperti pada gambar 6.2

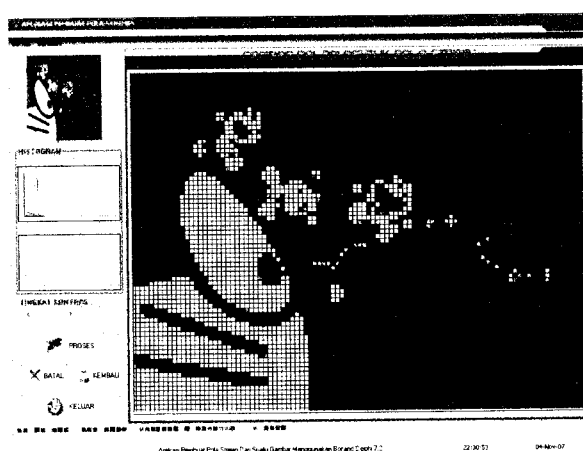


Gambar 6.2 Pengujian menggunakan File Gambar Dengan format bmp

6.1.1.3 Pengujian Dengan Input Gambar .jpg

Pada *form* ini proses pengolahan *file* gambar yang mempunyai format .jpg menjadi gambar pola strimin melalui pengujian pada tingkat kontras 1 sampai 20 dapat dilakukan. Pengujian dilakukan pada gambar yang mempunyai format .jpg yang mempunyai ukuran 640 x 480 *pixel*. Hasil dari pengolahan *file* .jpg dapat dilihat pada lampiran B halaman B-1.

Pengujian dengan data normal untuk proses pembuatan pola strimin dari gambar yang mempunyai format .jpg dapat dilakukan seperti pada gambar 6.3

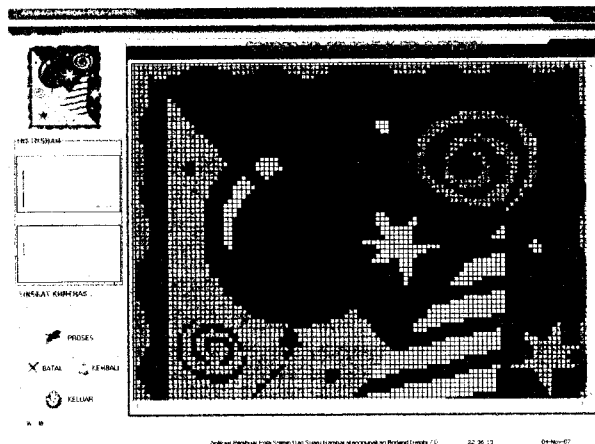


Gambar 6.3 Pengujian menggunakan File Gambar Dengan format jpg

6.1.1.4 Pengujian Dengan Input Gambar .wmf

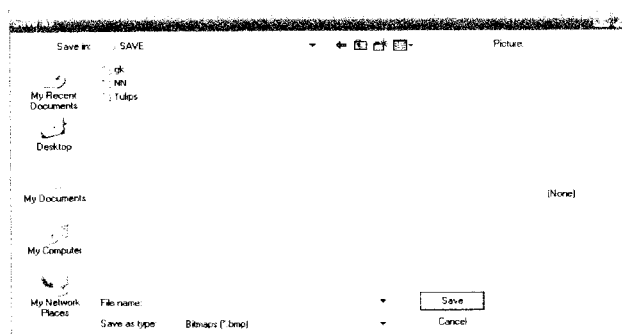
Pada *form* ini proses pengolahan *file* gambar yang mempunyai format .wmf menjadi gambar pola strimin melalui pengujian pada tingkat kontras 1 sampai 20 dapat dilakukan. Pengujian dilakukan pada gambar yang mempunyai format .wmf yang mempunyai ukuran 1760 x 1977 *pixel*. Hasil dari pengolahan *file* .wmf dapat dilihat pada lampiran C halaman C-1.

Pengujian dengan data normal untuk proses pembuatan pola strimin dari gambar yang mempunyai format .wmf dapat dilakukan seperti pada gambar 6.4



Gambar 6.4 Pengujian menggunakan File Gambar Dengan format wmf

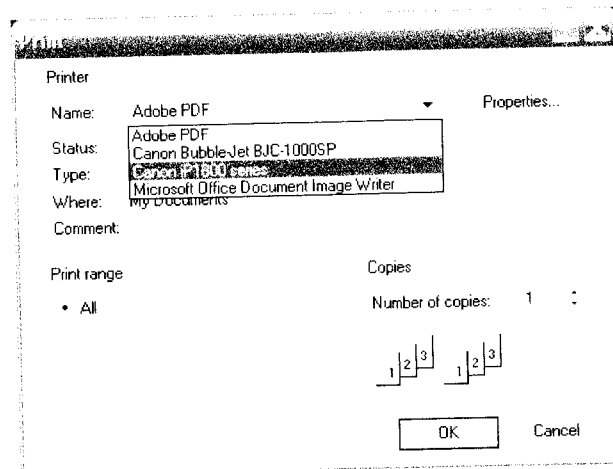
6.1.1.5 Pengujian Penyimpanan Gambar



Gambar 6.5 Penyimpanan Gambar Pola Strimin

Pada *form* ini proses penyimpanan *file* gambar setelah diolah menjadi pola strimin dan disimpan dalam *file* bmp dapat dilakukan.

6.1.1.6 Pengujian Mencetak Gmbar



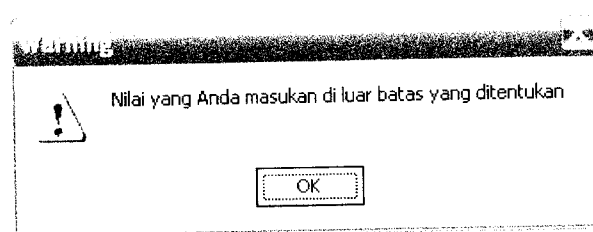
Gambar 6.6 Pencetakan Gambar Pola Strimin

Pada *form* ini proses pencetakan *file* gambar setelah diolah menjadi pola strimin dan dicetak menggunakan printer yang aktif di komputer dapat dilakukan. Hasil pengujian pencetakan menggunakan printer Canon iP1800 Series dapat dilihat pada lampiran A, B, C halaman A-1, B-1 dan C-1

6.1.2 Pengujian Data Tidak Normal (Proses yang tidak berurutan)

Pengujian dengan data tidak normal dimaksudkan untuk mengetahui reaksi sistem bila diberikan proses yang salah atau tidak lengkap dan untuk mengetahui akibat dari kesalahan proses tersebut.

6.1.2.1 Tingkat Kontras



Gambar 6.7 Kesalahan Tigkat Kontras

Sistem hanya bisa menjalankan proses apabila tingkat kontrasnya bernilai mulai dari 1 sampai dengan 20. Apabila ingin melakukan proses tetapi tingkat kontras bernilai kurang dari 1 atau lebih dari 20 maka akan muncul kotak pesan “Nilai yang Anda masukkan di luar batas yang ditentukan”.

6.2 Analisis Hasil Pengujian Sistem

Hal ini dimaksudkan untuk mengetahui tentang reaksi dari sistem terhadap proses yang diberikan baik secara normal maupun tidak normal.

6.2.1 Pengujian Dengan Data Normal

Pengujian dengan data normal untuk proses pembatasan format file gambar yang bisa diinputkan, pembuatan pola strimin dari gambar yang mempunyai format .bmp, .jpg, .wmf, penyimpanan gambar pola strimin dan

proses pencetakan berhasil dilakukan. Adapun hasil pencetakan dapat dilihat pada lampiran A, B, C pada halaman A-1, B-1, C-1.

Dari hasil proses pencetakan yang ada di lampiran dapat dilihat bahwa gambar yang dihasilkan dari input gambar dengan format bmp, jpg dan wmf sudah relatif bagus. Dengan input gambar berbagai ukuran tetapi tidak mempengaruhi kualitas gambar yang dihasilkan dan juga sudah mereduksi banyak warna.

6.2.2 Pengujian Dengan Data Tidak Normal (Proses yang tidak berurutan)

Dari hasil pengujian data tidak normal (pengujian dengan tingkat kontras) apabila tingkat kontras kurang dari 1 atau lebih dari 20 maka sistem akan menolak proses yang dilakukan dan muncul peringatan seperti pada Gambar 6.7

BAB VII

SIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi dan pengujian terhadap aplikasi pembuat pola strimin, maka dapat disimpulkan bahwa :

1. Aplikasi dapat membantu dalam proses pembuatan pola strimin
2. Aplikasi dapat digunakan untuk membuat pola strimin dari *input* gambar yang mempunyai format .bmp, .jpg, .wmf berbagai ukuran.
3. Hasil pengolahan gambar menjadi pola strimin dapat dicetak langsung atau disimpan terlebih dahulu
4. Aplikasi dapat membantu proses pembuatan kerajinan sulam strimin dengan detail gambar yang baik.

7.2 Saran

Penulis sadar bahwa aplikasi yang dibangun masih jauh dari sempurna oleh karena itu saran untuk pengembangan selanjutnya adalah :

1. Mengembangkan aplikasi dengan lebih dinamis dan interaktif serta dengan tampilan atau *Graphic User Interface(GUI)* yang lebih *user friendly*.
2. Aplikasi dapat dikembangkan dengan menambahkan format gambar yang bisa diproses dan memberikan informasi besar kain yang dibutuhkan dari gambar yang telah di proses.
3. Aplikasi memiliki fasilitas dimana *user* bisa menginputkan jumlah kotak strimin yang diinginkan, jumlah warna yang diinginkan dan adanya proses perubahan warna sesuai yang dikehendaki pengguna.

DAFTAR PUSTAKA

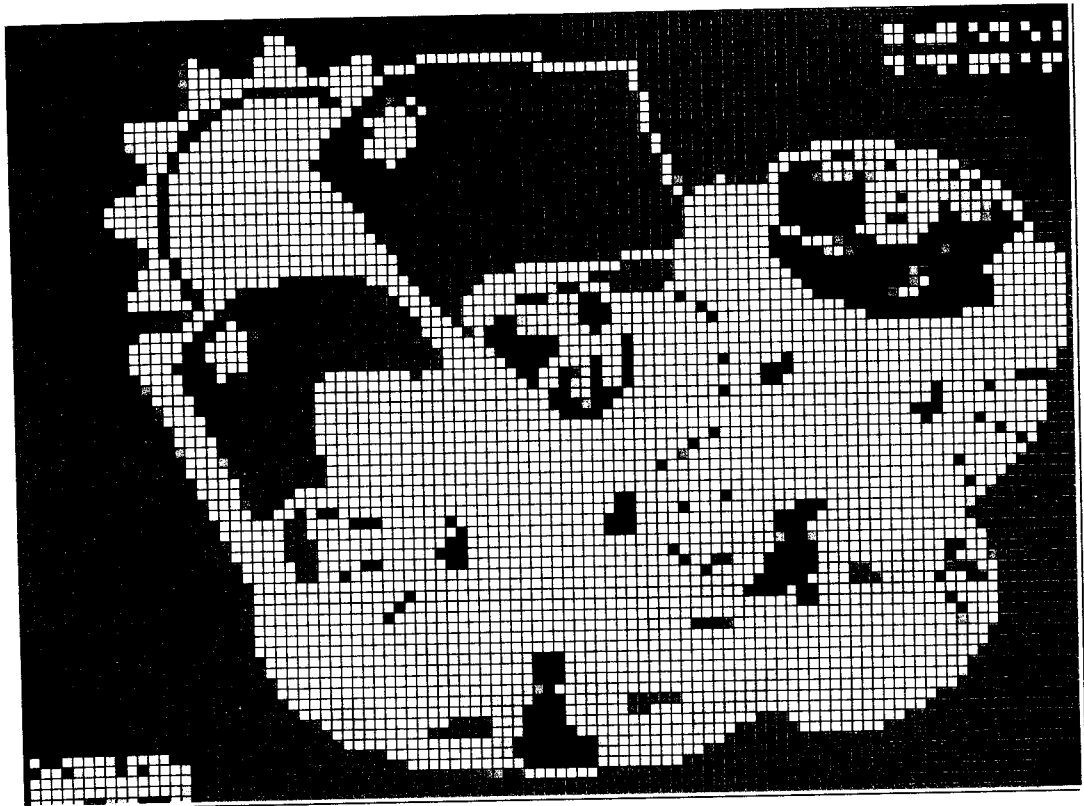
- [ACH05] Achmad, Balza. Ir., M.Sc.E. Firdausy, Kartika. ST., MT. *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Yogyakarta : Ardi Publishing, 2005.
- [AHM05] Ahmad, Usman. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta : Graha Ilmu, 2005.
- [MAD03] Madcoms. *Pemrograman Borland Delphi 7.0 Jilid 1*. Yogyakarta : Penerbit Andi, 2003.
- [MUN04] Munir, Rinaldi. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung : Penerbit Informatika, 2004.
- [PRA03] Pranata, A. *Pemrograman Borland Delphi 6 Edisi 4*. Yogyakarta : Andi, 2003.
- [PRA05] Prayudi, Yudi. SSi., M.Kom. *Bahan Kuliah Pengolahan Citra*. Yogyakarta : 2005.
- [WAH03] Wahana Komputer. *Tip dan Trik Pemrograman Delphi 7.0*. Yogyakarta : Penerbit Andi, 2003.

LAMPIRAN

LAMPIRAN A

HASIL CETAK MENGGUNAKAN PRINTER CANON iP1800 Series

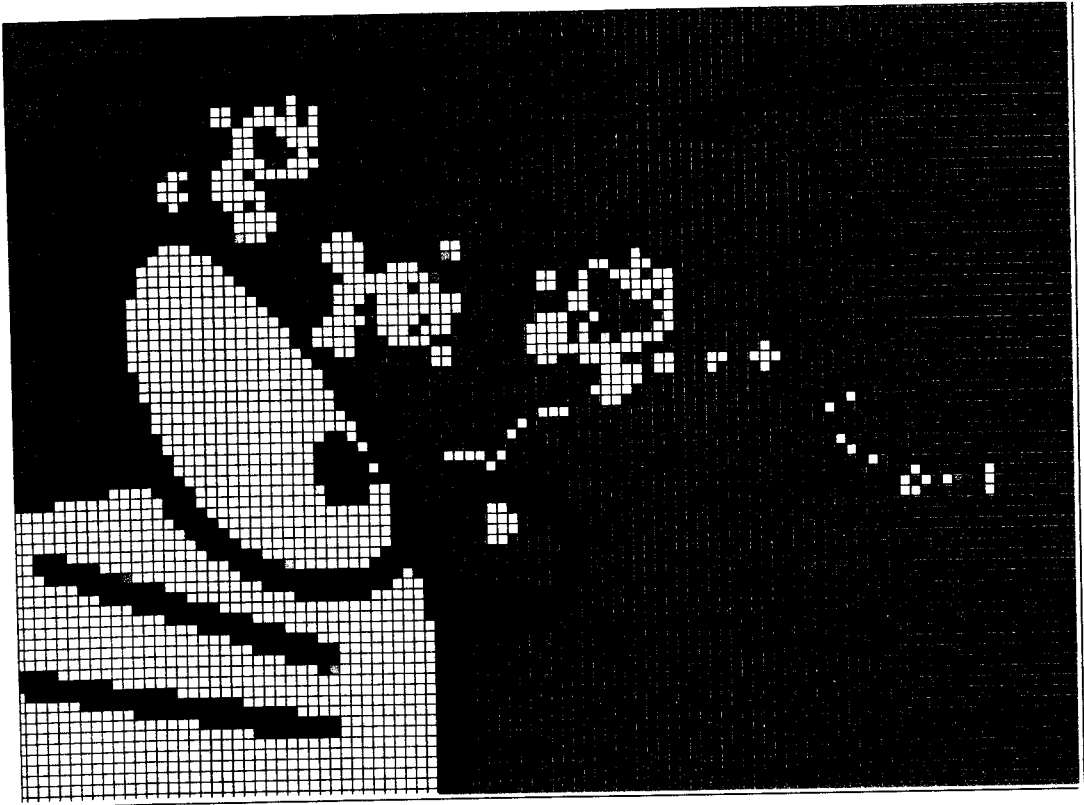
UNTUK FILE INPUT .BMP BERUKURAN 800 x 600 PIXEL



LAMPIRAN B

HASIL CETAK MENGGUNAKAN PRINTER CANON iP1800 Series

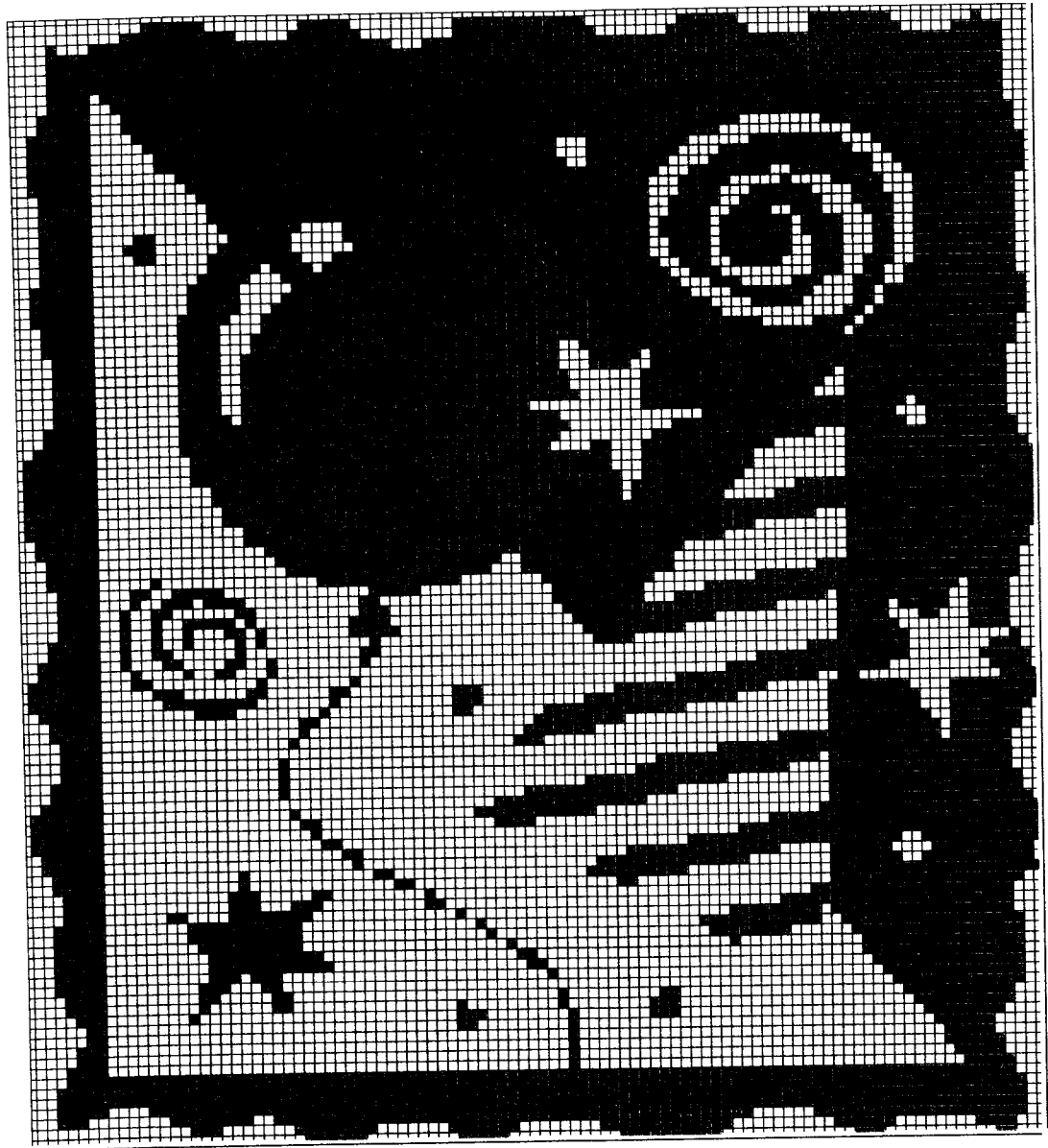
UNTUK FILE INPUT .JPG BERUKURAN 640 x 480 PIXEL



LAMPIRAN C

HASIL CETAK MENGGUNAKAN PRINTER CANON iP1800 Series

UNTUK FILE INPUT .WMF BERUKURAN 464 x 457 PIXEL



LAMPIRAN D

Prosedur pengaturan ukuran gambar pada strimin

```
image3.Picture.Bitmap:=image1.Picture.Bitmap;
w1 := 100;
w := Image1.Picture.Width;
h := Image1.Picture.Height;
try
  AR := w/h
except
  showmessage('Maaf, Belum ada file gambar yang akan
diproses');
  exit;
end;
cetakhasil1.Enabled:=true;
simpan1.Enabled:=true;
h1 := round(w1/AR);
Sh := w1/w ;
Sv := h1/h;

Image3.Picture.Bitmap.Width := w1;
Image3.Picture.Bitmap.Height := h1;
if (Image1.Picture.Bitmap.PixelFormat = pf8bit)
then
  begin
    SetLength(Ki, w, h);
    SetLength(Ko, w1, h1);
    for y := 0 to h-1 do
      begin
        PC := Image1.Picture.Bitmap.ScanLine[y];
        for x := 0 to w-1 do
          Ki[x, y] := PC[x];
        end;
        for x1 := 0 to w1-1 do
          for y1 := 0 to h1-1 do
            begin
              xAsal := x1/Sh;
              yAsal := y1/Sv;
              if ((Floor(xAsal)<0) or (Ceil(xAsal)>w-1) or
(Floor(yAsal)<0) or (Ceil(yAsal)>h-1))
              then
                Ko[x1, y1] := 255
              else
                begin
                  x := Round(xAsal);
                  y := Round(yAsal);
                  Ko[x1, y1] := Ki[x, y];
                end
              end;
            end;
          for y1 := 0 to h1-1 do
            begin
```

```

        PH := Image3.Picture.Bitmap.ScanLine[y1];
        for x1 := 0 to w1-1 do
            PH[x1] := Ko[x1, y1];
        end;
    Ki := nil;
    Ko := nil;
end;
if (Image1.Picture.Bitmap.PixelFormat = pf24bit)
then
begin
    SetLength(Ri, w, h);
    SetLength(Gi, w, h);
    SetLength(Bi, w, h);
    SetLength(Ro, w1, h1);
    SetLength(Go, w1, h1);
    SetLength(Bo, w1, h1);
    for y := 0 to h-1 do
        begin
            PC := Image1.Picture.Bitmap.ScanLine[y];
            for x := 0 to w-1 do
                begin
                    Bi[x, y] := PC[3*x];
                    Gi[x, y] := PC[3*x+1];
                    Ri[x, y] := PC[3*x+2];
                end;
            end;
        for x1 := 0 to w1-1 do
            for y1 := 0 to h1-1 do
                begin
                    xAsal := x1/Sh;
                    yAsal := y1/Sv;
                    if ((Floor(xAsal)<0) or (Ceil(xAsal)>w-1) or
                        (Floor(yAsal)<0) or (Ceil(yAsal)>h-1))
                    then
                        begin
                            Ro[x1, y1] := 255;
                            Go[x1, y1] := 255;
                            Bo[x1, y1] := 255;
                        end
                    else
                        begin
                            x := Round(xAsal);
                            y := Round(yAsal);
                            Ro[x1, y1] := Ri[x, y];
                            Go[x1, y1] := Gi[x, y];
                            Bo[x1, y1] := Bi[x, y];
                        end
                    end;
                for y1 := 0 to h1-1 do
                    begin
                        PH := Image3.Picture.Bitmap.ScanLine[y1];
                        for x1 := 0 to w1-1 do
                            begin
                                PH[3*x1] := Bo[x1, y1];
                                PH[3*x1+1] := Go[x1, y1];

```

```
        PH[3*x1+2] := Ro[x1, y1];
    end;
end;
Ri := nil;
Gi := nil;
Bi := nil;
Ro := nil;
Go := nil;
Bo := nil;
end;
```

LAMPIRAN E

Prosedur pengaturan ukuran gambar pada antarmuka

```
image5.Picture.Bitmap:=image3.Picture.Bitmap;
w1 := 800;
w := Image3.Picture.Width;
h := Image3.Picture.Height;
AR := w/h;
h1 := round(w1/AR);
Sh := w1/w ;
Sv := h1/h;

Image5.Picture.Bitmap.Width := w1;
Image5.Picture.Bitmap.Height := h1;
if (Image3.Picture.Bitmap.PixelFormat = pf8bit)
then
begin
SetLength(Ki, w, h);
SetLength(Ko, w1, h1);
for y := 0 to h-1 do
begin
PC := Image3.Picture.Bitmap.ScanLine[y];
for x := 0 to w-1 do
Ki[x, y] := PC[x];
end;
for x1 := 0 to w1-1 do
for y1 := 0 to h1-1 do
begin
xAsal := x1/Sh;
yAsal := y1/Sv;
if ((Floor(xAsal)<0) or (Ceil(xAsal)>w-1) or
(Floor(yAsal)<0) or (Ceil(yAsal)>h-1))
then
Ko[x1, y1] := 255
else
begin
x := Round(xAsal);
y := Round(yAsal);
Ko[x1, y1] := Ki[x, y];
end
end;
for y1 := 0 to h1-1 do
begin
PH := Image5.Picture.Bitmap.ScanLine[y1];
for x1 := 0 to w1-1 do
PH[x1] := Ko[x1, y1];
end;
Ki := nil;
Ko := nil;
end;
if (Image3.Picture.Bitmap.PixelFormat = pf24bit)
```

```

then
  begin
    SetLength(Ri, w, h);
    SetLength(Gi, w, h);
    SetLength(Bi, w, h);
    SetLength(Ro, w1, h1);
    SetLength(Go, w1, h1);
    SetLength(Bo, w1, h1);
    for y := 0 to h-1 do
      begin
        PC := Image3.Picture.Bitmap.ScanLine[y];
        for x := 0 to w-1 do
          begin
            Bi[x, y] := PC[3*x];
            Gi[x, y] := PC[3*x+1];
            Ri[x, y] := PC[3*x+2];
          end;
        end;
      for x1 := 0 to w1-1 do
        for y1 := 0 to h1-1 do
          begin
            xAsal := x1/Sh;
            yAsal := y1/Sv;
            if ((Floor(xAsal)<0) or (Ceil(xAsal)>w-1) or
              (Floor(yAsal)<0) or (Ceil(yAsal)>h-1))
            then
              begin
                Ro[x1, y1] := 255;
                Go[x1, y1] := 255;
                Bo[x1, y1] := 255;
              end
            else
              begin
                x := Round(xAsal);
                y := Round(yAsal);
                Ro[x1, y1] := Ri[x, y];
                Go[x1, y1] := Gi[x, y];
                Bo[x1, y1] := Bi[x, y];
              end
            end;
          end;
        for y1 := 0 to h1-1 do
          begin
            PH := Image5.Picture.Bitmap.ScanLine[y1];
            for x1 := 0 to w1-1 do
              begin
                PH[3*x1] := Bo[x1, y1];
                PH[3*x1+1] := Go[x1, y1];
                PH[3*x1+2] := Ro[x1, y1];
              end;
            end;
          end;
        Ri := nil;
        Gi := nil;
        Bi := nil;
        Ro := nil;
        Go := nil;
        Bo := nil;
      end;
    end;
  end;

```

LAMPIRAN F

Prosedur Contoh Warna

```
Procedure Ttengah.contohwarna;
var i,j,k:integer;
    PC: PByteArray;
    merah,hijau,biru,mr,hj,br:byte;
    sama:boolean;

begin
  if n>0 then
    for i:=1 to n do
      begin
        warna[i].kotak.Destroy;
      end;
    n:=1;
    if (image3.Picture.Bitmap.PixelFormat = pf24bit) then
      begin
        PC := image3.Picture.Bitmap.ScanLine[0];
        warna[1].b:=PC[0];
        warna[1].h:=PC[1];
        warna[1].m:=PC[2];
        warna[1].kotak:=TPanel.Create(self);
        warna[1].kotak.Parent:=scrollbox2;
        warna[1].kotak.Width := 12;
        warna[1].kotak.Height:= 12;
        warna[1].kotak.Top:=3;
        warna[1].kotak.Left:=3;
        warna[1].kotak.Color:=rgb(warna[1].m,warna[1].h,warna[1].b);
        warna[1].kotak.Visible:=true;
        for i:= 0 to image3.Picture.Height-1 do
          begin
            PC := image3.Picture.Bitmap.ScanLine[i];
            for j:= 1 to image3.Picture.Width-1 do
              begin
                k:=0;
                repeat
                  inc(k);
                  biru:=PC[3*j];
                  hijau:=PC[3*j+1];
                  merah :=PC[3*j+2];
                  mr:= warna[k].M;
                  hj:= warna[k].H;
                  br:= warna[k].B;
                  If ((mr-10>merah) or (merah>mr+10)) or
                    ((hj-10>hijau) or (hijau>hj+10)) or
                    ((br-10>biru) or (biru>br+10)) then
                    begin
                      sama:=false;
                    end
                  else

```

```

begin
  sama:=true;
  break;
end;
until k=n;
if not sama then
begin
  inc(n);
  warna[n].m:=merah;
  warna[n].h:=hijau;
  warna[n].b:=biru;
  warna[n].kotak:=TPanel.Create(self);
  warna[n].kotak.Parent:=scrollbox2;
  warna[n].kotak.Width := 12;
  warna[n].kotak.Height:= 12;
  warna[n].kotak.Top:= ((n-1) div 80)*12+3;
  if (n mod 80 )=1 then
  warna[n].kotak.Left:=3
  else
  warna[n].kotak.Left:= warna [n-1].kotak.Left + 12;

warna[n].kotak.Color:=rgb(warna[n].m,warna[n].h,warna[n].b);
  warna[n].kotak.Visible:=true;
end;
end;
end;
end;
end;

```