

KOMPRESI FILE GAMBAR DAN EKSPLORASI SOFTWARE
KOMPRESI

LAPORAN TUGAS AKHIR

*Diajukan Sebagai Salah Satunya Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika*



Disusun Oleh :

Nama : Fadli Yusra

No.Mahasiswa : 04 523 359

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

LEMBAR PENGESAHAN PEMBIMBING

KOMPRESI FILE GAMBAR DAN EKSPLORASI SOFTWARE

KOMPRESI

LAPORAN TUGAS AKHIR

Disusun oleh

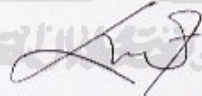
Nama : Fadli Yusra

No. Mahasiswa : 04 523 359

Yogyakarta, Januari 2011

Telah Diterima Dan Disetujui Dengan Baik Oleh :

Dosen Pembimbing



(Ami Fauziah, ST., MT.,)

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

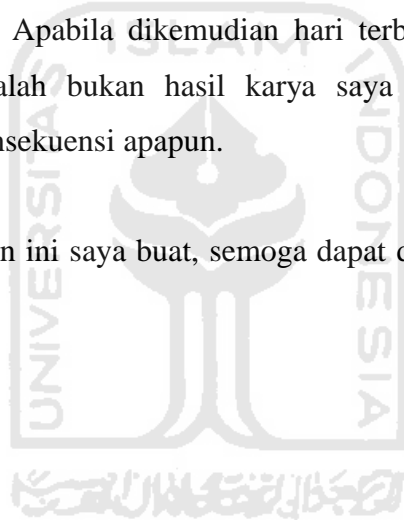
Saya yang bertanda tangan dibawah ini,

Nama : Fadli Yusra

No. Mahasiswa : 04 523 359

Menyatakan bahwa seluruh komponen dan isi dalam laporan Tugas Akhir ini adalah hasil karya sendiri. Apabila dikemudian 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, Januari 2011

(Fadli Yusra)

**LEMBAR PENGESAHAN PENGUJI
KOMPRESI FILE GAMBAR EKSPLORASI SOFTWARE KOMPRESI
TUGAS AKHIR**

Disusun oleh:

Nama : Fadli Yusra

No.Mahasiswa : 04 523 359

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia.
Yogyakarta, Januari 2011

Tim Penguji

Ami Fauziah, ST., M.T.

Ketua

Anggota I

Affan Mahtarami, S.Kom., M.T.

Anggota II

Beni Suranto, S.T.

Mengetahui,

Ketua Program Studi Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

PERSEMBAHAN



**UNTUK AKU DAN SEGENAP RIAK-RIAK KEHIDUPAN YANG
MENYERTAI KU**

MOTTO



Ilmu pengetahuan tanpa agama adalah pincang

'Einstein'

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu 'alaikum Wr. Wb

Dengan mengucapkan Alhamdulillah, puji dan syukur kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir, dengan judul “ **Kompresi File Gambar dan Eksplorasi Software Kompresi** ”. Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para kerabat, serta pengikutnya hingga hari kiamat nanti.

Penyusunan tugas akhir merupakan sebagian upaya untuk memenuhi syarat kelulusan studi serta syarat untuk memperoleh gelar sarjana dari Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia Yogyakarta.

Dalam pembuatan tugas akhir ini, penulis mendapat banyak bimbingan dan pengarahan serta bantuan yang sangat bermanfaat dari berbagai pihak. Oleh karena itu dalam kesempatan ini, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT , Tuhan bagi seluruh alam yang melimpahkan rahmat dan karuniannya sehingga penulis selalu diberi kesehatan dan kemudahan selama pembuatan tugas akhir ini.
2. Kedua orang tua, yang telah memberikan seluruh do'a dan restu, serta dorongan sehingga penulis dapat menyelesaikan studi dengan baik.
3. Bapak Ir. Gumbolo HS,. M.Sc selaku Dekan Fakultas Teknik Industri, Universitas Islam Indonesia.
4. Bapak Yudi Prayudi, S.Si.,M.kom., selaku Ketua Jurusan Teknik Informatika, Universitas Islam Indonesia.

5. Ibu Ami Fauziah, ST, MT selaku dosen pembimbing yang telah memberikan pengarahan, bimbingan, serta masukan selama pelaksanaan tugas akhir dan penulisan laporan.

6. Seluruh keluarga terima kasih atas semangatnya yang telah diberikan.

Semoga dengan segala bantuan yang telah diberikan kepada penulis, akan mendapat pahala yang setimpal dari ALLAH SWT. Amin

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Untuk itu segala saran dan kritik yang sifatnya membangun sangat penulis harapkan guna penyempurnaan dimasa mendatang.

Akhir kata, semoga laporan tugas akhir ini dapat bermanfaat bagi kita semua. Amiin.

Wassalamu'alaikum Wr.Wb.



Yogyakarta, Januari 2011

Fadli Yusra

SARI

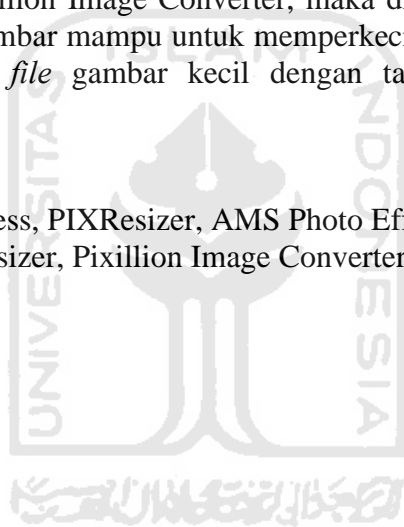
Dari berbagai jenis-jenis format *file* gambar juga terdapat banyak metode penyimpanan kompresi gambar yang berguna untuk menyimpan suatu format *file* gambar yang telah berhasil dikompresi.

Kompresi gambar ini salah satu manfaat teknologi komputer yang mampu menjelaskan jenis-jenis *file* gambar, media penyimpanan *file* gambar dan cara kompresi *file* gambar. Ada 2 metode utama dalam kompresi gambar, yaitu metode *lossy* dan *lossless*.

Berdasarkan eksplorasi yang telah dilakukan dengan menggunakan 5 *software* yaitu PIXResizer, AMS Photo Effects, FastStone Image Viewer, VSO Image Resizer dan Pixillion Image Converter, maka diperoleh suatu kesimpulan bahwa kompresi *file* gambar mampu untuk memperkecil ukuran *file* gambar yang besar menjadi ukuran *file* gambar kecil dengan tanpa mengurangi kualitas gambar.

Kata kunci :

Kompresi, *Lossy*, *Lossless*, PIXResizer, AMS Photo Effects, FastStone Image Viewer, VSO Image Resizer, Pixillion Image Converter.



DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	iii
LEMBAR PENGESAHAN PENGUJI	iv
PERSEMBAHAN.....	v
MOTTO	vi
KATA PENGANTAR	vii
SARI.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB I TIPE FILE GAMBAR DAN KARAKTERISTIKNYA	1
1.1. Karakteristik File Gambar	1
1.2. Tipe Gambar.....	2
DAFTAR PUSTAKA	13
BAB II METODE KOMPRESI DATA GAMBAR	14
2.1. Gambar Tidak Dimampatkan.....	18
2.2. Gambar Dimampatkan	19
DAFTAR PUSTAKA	38
BAB III EKSPLORASI SOFTWARE KOMPRESI GAMBAR.....	39
3.1. Metodologi Kompresi File Gambar.....	39
3.2. PIXResizer.....	40
3.3. FastStone Image Viewer	41
3.4. VSO Image Resizer	43
3.5. AMS Photo Effects.....	46
3.6. Pixillion Image Converter	48
3.7. Uji Coba	50

DAFTAR PUSTAKA	64
BAB IV PENUTUP	65
4.1. Kesimpulan :	65
4.2. Saran.....	66



DAFTAR TABEL

Tabel 1.1 Perbandingan dan pendukung pada data gambar	9
Tabel 3.1 Data kompresi file gambar dengan pengaturan standar	39
Tabel 3.2 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .png	51
Tabel 3.3 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .bmp	51
Tabel 3.4 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .gif	52
Tabel 3.5 Perbandingan konversi gambar ukuran sedang dari .png ke .jpeg	53
Tabel 3.6 Perbandingan konversi gambar ukuran sedang dari .png ke .gif	54
Tabel 3.7 Perbandingan konversi gambar ukuran sedang dari .png ke .bmp	54
Tabel 3.8 Perbandingan konversi gambar ukuran besar dari .bmp ke .jpeg	55
Tabel 3.9 Perbandingan konversi gambar ukuran besar dari .bmp ke .png	56
Tabel 3.10 Perbandingan konversi gambar ukuran besar dari .bmp ke .gif	56
Tabel 3.11 Perbandingan konversi gambar dari .jpeg ke .png dengan pengaturan size	57
Tabel 3.12 Perbandingan konversi gambar dari .bmp ke .gif dengan pengaturan size	59
Tabel 3.13 Analisis kelebihan dan kekurangan software	61

DAFTAR GAMBAR

Gambar 2.1 Pengelompokkan Teknik DWT.....	17
Gambar 2.2 Derajat Keabuan.....	24
Gambar 2.3 Alur Algoritma Huffman.....	29
Gambar 3.1 Tampilan <i>Software</i> PIXResizer.....	40
Gambar 3.2 Tampilan <i>Software</i> FastStone Image Viewer.....	42
Gambar 3.3 Tampilan <i>Software</i> VSO Image Resizer	44
Gambar 3.4 Tampilan <i>Software</i> AMS Photo Effects	47
Gambar 3.5 Tampilan <i>Software</i> Pixillion Image Converter.....	49



BAB I

TIPE FILE GAMBAR DAN KARAKTERISTIKNYA

Multimedia merupakan sesuatu yang dahsyat dan hebat, sesuatu yang berkaitan dengan komputer, dengan gambar diam atau bergerak, dengan suara yang bagus dan kaya, serta pengguna yang dapat ikut serta berinteraksi. Multimedia pun sekarang sudah digunakan dalam berbagai hal, misalnya kartu ucapan selamat, album foto, konferensi, katalog bergambar, film, sampai pada internet.

Multimedia merupakan kombinasi teks, seni, suara, gambar, animasi dan video yang disampaikan dengan komputer atau dimanipulasi secara digital dan dapat disampaikan dan/atau dikontrol secara interaktif (Vaughan, 2004).

1.1. Karakteristik *File* Gambar

a. Format

Format adalah perintah untuk pengaturan penyusunan data pada disk. Pada DOS merupakan perintah untuk membentuk sektor pada disket atau harddisk sehingga dapat digunakan untuk menyimpan data.

b. *File Size*

File adalah arsip yang disimpan dalam suatu media yang terdiri dari kumpulan karakter lalu didokumentasikan. Dengan karakter yang dimiliki, maka *file* menyimpan ukuran dengan satuan ukuran pada komputer, yaitu bit.

c. EXIF Data

EXIF adalah singkatan dari *Exchangeable Image File* akan tetapi sering disalah pahami dengan Exposure Information. EXIF data selalu melekat pada foto yang diambil menggunakan kamera digital. Informasi EXIF tersebut meliputi nama *file*, tanggal pemotretan, kecepatan rana, diafragma, kamera model, panjang focus lensa, ISO, eksposur bias dan besar *file*.

d. Bit Depth

Bit Depth atau kedalaman bit adalah kemampuan pixel dalam menyimpan bit atau informasi, dalam hal ini akan berpengaruh terhadap banyaknya warna yang dapat ditampilkan. Semakin besar jumlah bit yang digunakan untuk mempresentasikan suatu titik, semakin banyak warna dan atau bayangan abu-abu yang dapat dibuat.

e. Dimensi

Dimensi berarti parameter atau pengukuran yang dibutuhkan untuk mendefinisikan sifat-sifat suatu objek, yaitu panjang, lebar dan tinggi. Dalam fotografi, dimensi berarti jumlah tempat yang dibutuhkan oleh suatu array untuk mengidentifikasi sebuah elemen.

f. Rasio

Rasio adalah suatu perbandingan antara jumlah suatu peristiwa dengan jumlah peristiwa yang lain. Suatu jumlah peristiwa sebagai pembilang (numerator) dengan jumlah peristiwa yang lain sebagai penyebut (denominator).

1.2. Tipe Gambar

1.2.1. Bitmap

Bitmap digunakan untuk foto realistik dan gambar kompleks yang membutuhkan detail yang halus. Bit merupakan elemen primitif dalam dunia digital. Bit menunjuk

pada angka biner, sedangkan map merupakan matriks 2 dimensi dari bit ini. Bitmap berarti matriks sederhana dari titik-titik kecil yang membentuk sebuah image dan ditampilkan di layar komputer atau dicetak. Elemen gambar dikenal sebagai pels atau secara umum disebut piksel.

Sebuah bitmap merupakan matriks data yang mendeskripsikan karakteristik dari semua pixel yang menyusun sebuah gambar. Warna terindeks (*indexed color*) terbatas sampai 256 warna yang dapat juga merupakan 256 warna sembarang dari himpunan 16,7 juta (24 bit) warna. Setiap warna menggunakan 24 bit nilai RGB. Setiap *file* gambar berisi palet warnanya sendiri yang merupakan daftar dari 256 warna yang dipilih (atau 16 warna dalam palet yang lebih kecil).

Gambar dapat dikatakan mempunyai warna terindeks karena data warna gambar yang sebenarnya untuk setiap piksel merupakan indeks dari palet tersebut. Setiap data pixel adalah nomor yang menetapkan satu dari warna palet, misalnya “warna nomor 80”. 80 merupakan indeks dari palet. Artinya warna ke-80 ada dalam daftar warna palet. Palet disimpan dalam *file* dengan gambarnya.

Indeks biasanya bernilai 4 bit (16 warna) atau 8 bit (256 warna) untuk setiap piksel sehingga penyimpanannya akan lebih kecil daripada penyimpanan 24 bit untuk setiap piksel. Meskipun begitu, angka 8 bit hanya dapat berisi nilai dari 0 sampai 255. Oleh sebab itu, hanya ada 256 warna yang dapat berada dalam palet.

File gambar juga dapat berisi palet yang merupakan tabel dari warna 24 bit yang sudah dipilih atau sebesar 3 byte dari RGB untuk setiap warna pada palet (768 byte untuk 256 warna). Warna RGB pertama dalam tabel adalah indeks 0, warna RGB kedua adalah indeks 1, dan seterusnya sehingga hanya ada 256 warna di palet.

File terindeks menyimpan 24 bit untuk setiap warna palet, tetapi tidak untuk setiap piksel. Setiap piksel hanya menyimpan indeks 4 bit atau 8 bit untuk menetapkan warna palet yang mana yang digunakan.

Aplikasi pengolah gambar akan memperlihatkan palet untuk gambar terindeks dan dapat memodifikasi warna palet. Ada bermacam-macam cara untuk menciptakan

palet dan untuk memilih warna pilihan yang mungkin akan dimasukkan. Pemilihan palet akan mempengaruhi kualitas gambar.

Bitmap dapat dibuat dengan program paint atau yang lainnya. Bitmap dapat ditangkap dari layar monitor dan kemudian disimpan melalui program paint atau aplikasi lainnya. Gambar bitmap juga dapat ditangkap dengan menggunakan pemindai, kamera digital, atau penangkap gambar yang dapat mendigitalkan gambar. Selain itu, gambar bitmap bisa didapatkan dari *ClipArt* yang sudah disediakan oleh perangkat lunak tertentu atau dari pembelian CD yang berisi *ClipArt*. Ada banyak perangkat lunak pengolah bitmap, antara lain adalah GIMP dan Adobe Photoshop.

Format *File* Bitmap

Format *file* gambar dalam arti yang standar berarti mengorganisasikan dan menyimpan gambar, baik itu fotografi atau gambar lainnya. *File* gambar sebenarnya terdiri dari data piksel atau vektor yang dikonversi (di-*raster*) ke piksel ketika ditampilkan di layar monitor.

Piksel penyusun gambar diatur sebagai *grid* (kolom dan baris) dengan setiap piksel yang terdiri dari bilangan yang merepresentasikan besarnya kecerahan (*brightness*) dan warna.

Ada banyak format *file* gambar yang masing-masing mempunyai kelebihan dan kekurangan serta mempunyai kesesuaian dengan fungsinya. Ada banyak format *file* untuk gambar bitmap, tergantung dari kepentingan dan kegunaannya masing-masing.

a. JPEG

JPEG adalah singkatan dari *Joint Photographic Experts Group*. Kebanyakan JPEG merupakan *file* gambar dengan pemampatan *lossy* (ekstensi *file* dengan sistem operasi DOS adalah JPEG). Sudah banyak kamera digital menyimpan gambarnya dengan format JPEG yang mendukung

8 bit per warna (*red, green, blue*), totalnya menjadi 24 bit, dan menghasilkan *file* yang lumayan kecil. Kelemahan JPEG adalah kualitas gambar yang akan menurun, bertambah kasar, dan pecah jika *file* gambar di-*edit* dan disimpan berulang-ulang. Sebagai catatan, gambar fotografi yang akan di-*edit* lagi sebaiknya disimpan dalam format non-JPEG yang *lossless*.

b. EXIF

EXIF adalah singkatan dari *Exchangeable Image File Format* yang merupakan algoritma yang tergabung dalam perangkat lunak JPEG yang digunakan di dalam sebagian besar kamera. Tujuannya adalah untuk merekam dan menstandarisasi pertukaran data antara kamera digital, perangkat lunak peng-*editan*, dan penampil. Gambar secara individu direkam dan diberi masukan data, seperti pengesetan kamera, tanggal dan jam, kecepatan *shutter*, ukuran gambar, pemampatan, nama kamera, informasi warna, dan lain-lain. Ketika gambar dilihat atau di-*edit* dengan perangkat lunak pengeditan gambar, semua informasi tersebut ditampilkan.

c. TIFF

TIFF adalah singkatan dari *Tagged Image File Format* yang merupakan format yang fleksibel dan normalnya menyimpan 8 bit atau 16 bit perwarna (*red, gree, blue*) untuk total 24 bit (menggunakan ekstensi *file* TIFF) dan 48 bit (menggunakan ekstensi *file* TIF). TIFF menggunakan metode pemampatan *lossy* dan *lossless*. Beberapa kamera digital menyimpan gambar dengan format TIFF dan menggunakan algoritma pemampatan LZW yang *lossless*.

Format gambar TIFF jarang didukung oleh *browser* web. TIFF banyak digunakan untuk standar *file* fotografi untuk kepentingan percetakan. TIFF dapat menangani perangkat khusus pewarnaan, seperti CMYK yang digunakan oleh tinta percetakan secara khusus.

Paket perangkat lunak OCR (*Optical Character Recognition*) biasanya menghasilkan gambar TIFF untuk hasil pemindaian teks.

d. RAW

RAW merujuk pada keluarga *raw image format* yang secara opsional ada pada beberapa kamera digital. Sesuai dengan namanya, *file* RAW ini masih mentah, belum diproses dan belum dapat digunakan dengan *editor* grafis bitmap atau dicetak. Format *file* ini menggunakan metode pemampatan *lossless* atau *nearly-lossless* dan menghasilkan ukuran *file* yang lebih kecil daripada format TIFF ketika diproses dengan ukuran penuh dari kamera yang sama.

Format RAW tidak mempunyai standar dan tidak terdokumentasikan serta berbeda-beda antara satu pabrik kamera dengan pabrik kamera lainnya.

e. PNG

PNG merupakan singkatan dari *Portable Network Graphics*. Format *file* yang bersifat gratis dan open source ini dirancang sebagai pengganti GIF. Format *file* PNG mendukung *truecolor* (16 juta warna), sementara GIF hanya mendukung 256 warna.

Format PNG yang *lossless* merupakan format yang baik untuk pengeditan gambar, sedangkan format *lossy* (seperti JPG) merupakan format yang baik untuk distribusi hasil fotografi karena *file* JPG lebih kecil daripada *file* PNG. Banyak browser lama yang tidak mendukung format *file* PNG, tetapi *browser* sekarang (Mozilla Firefox dan Internet Explorer 7) sudah mendukung penggunaan format PNG termasuk 8 bit transparan penuh.

PNG pengganti GIF yang bebas paten dapat menggantikan beberapa penggunaan umum dari TIFF. Gambar dengan warna terindeks, *grayscale*, dan *truecolor* sudah didukung, ditambah juga dengan *alpha channel* yang bersifat opsional. PNG dirancang untuk bekerja dengan baik pada aplikasi penglihat secara on-line, seperti *world wide web*, sehingga benar-benar dapat *di-stream*.

PNG termasuk tangguh karena menyediakan pemeriksaan integritas *file* dan pendeteksi sederhana bagi kesalahan pengiriman yang umum. Di samping

itu. PNG dapat menyimpan data gamma dan *chromacity* untuk peningkatan kecocokan warna pada platform yang heterogen.

f. GIF

GIF merupakan singkatan dari *Graphics Interchange Format*. Format *file* GIF terbatas sampai 8 bit palet atau 256 warna. Hal ini membuat format GIF sesuai untuk penyimpanan gambar dengan warna yang sedikit, seperti diagram sederhana, bentuk bangun, logo dan gambar bergaya kartun. Format GIF mendukung animasi dan sampai sekarang masih digunakan secara luas untuk menampilkan efek animasi.

Format GIF menggunakan pemampatan *lossless* yang efektif ketika daerahnya luas dan hanya mempunyai warna tunggal, tetapi tidak efektif untuk gambar detail atau gambar dengan banyak gradasi halus.

g. BMP

Format *file* BMP (Windows bitmap) menangani *file* garis dalam sistem operasi Microsoft Windows. Biasanya *file* BMP tidak termampatkan sehingga ukurannya besar. Keuntungan menggunakan format BMP adalah sederhana, diterima secara luas dan digunakan pada aplikasi-aplikasi Windows.

h. DIB

Kelebihan dari format DIB adalah bersifat fleksibel karena dapat digunakan untuk menyimpan gambar dengan 1, 4, 8, 16 dan 24 bit per piksel. Di samping itu proses untuk membaca informasi warna piksel di dalam format DIB lebih mudah dan dapat dilakukan secara tepat. DIB adalah singkatan dari *Device Independent Bitmap*, aplikasi yang digunakan pada kernel Windows yang fungsinya hampir sama seperti BMP (Windows Bitmap)

i. PSD

Ini adalah format khusus untuk Photoshop. Format ini menyimpan informasi yang mana nanti akan berguna saat akan kembali melakukan *pengedit-an* lagi pada Photoshop. Informasi tersebut berupa *layer*, *channel extra*, *file info* dan sebagainya. Karena format ini memang khusus dibuat untuk

Photoshop, maka wajar jika tidak semua program bisa membuka dan menggunakannya. Tapi itu tidak menjadi masalah besar, karena dapat menyimpan format ini dengan mengubah ke format lainnya.

j. TGA

TGA adalah singkatan dari TrueVision's Targa. Dengan format Targa ini dapat membuat video dengan cara menggabungkan gambar-gambar tersebut. Format ini biasanya dipakai oleh para profesional yg bergerak dalam bidang aplikasi video untuk PC. Format ini sudah jarang dipakai, ini populer pada awal '80-an hingga pertengahan '98. Gambar ini juga digunakan untuk gambar-gambar pada *game* dikarenakan format TGA mendukung teknik transparasi.

k. PNT

PNT adalah format yg dipakai oleh MacPaint. Program ini mirip dengan MicsrosoftPaint pada Windows. MacPaint termasuk yang pertama dalam mengaplikasikan grafis di komputer, sekarang MacPaint tidak dikembangkan lagi. Program ini dibuat oleh Bill Atkinson, beliau adalah salah satu tokoh di Silicon Valley.

Ada empat format *file* yang umum digunakan, terutama untuk pencetakan, pemindaian dan internet, yaitu TIFF, PNG, JPEG dan GIF. Meskipun demikian, TIFF tidak dapat digunakan untuk *browser* internet.

Semua aplikasi *editor* grafis, seperti Adobe Photoshop atau GIMP, mendukung format *file* tersebut yang secara umum mendukung dan menyimpan gambar pada mode warna seperti yang ditunjukkan Tabel berikut.

Tabel 1.1 Perbandingan dan pendukung pada data gambar

	Mode Warna, Bit per Piksel
TIFF	<p>RGB – 24 atau 48 bit, <i>Grayscale</i> – 8 atau 16 bit, <i>Indexed color</i> – 1 sampai 8 bit, <i>Line Art</i> (bilevel) – 1 bit</p>
TIFF	<p>Untuk <i>file</i> TIFF, banyak aplikasi yang tidak memampatkan atau menggunakan pemampatan LZW (<i>lossless</i>, tetapi kurang efektif untuk gambar dengan 24 bit warna).</p>
PNG	<p>RGB – 24 atau 48 bit, <i>Grayscale</i> – 8 atau 16 bit, <i>Indexed color</i> – 1 sampai 8 bit, <i>Line Art</i> (bilevel) – 1 bit</p> <p>PNG menggunakan pemampatan ZIP yang <i>lossless</i> dan sedikit lebih efektif daripada LZW. PNG merupakan format baru yang gratis dan bersifat serbaguna.</p>
JPEG	<p>RGB – 24 bit, <i>Grayscale</i> – 8 bit</p> <p>JPEG selalu menggunakan pemampatan JPG yang <i>lossy</i>, tetapi derajat pemampatannya dapat dipilih. Jika kualitasnya bagus, ukuran <i>file</i>-nya besar, sedangkan jika kualitasnya rendah, ukuran <i>file</i>-nya kecil.</p>
GIF	<p><i>Indexed color</i> – 1 sampai 8 bit,</p> <p>GIF menggunakan pemampatan LZW yang <i>lossless</i>, efektif untuk warna terindeks. <i>File</i> GIF tidak mempunyai informasi tentang dpi untuk keperluan pencetakan.</p>

1.2.2. Vektor

Vektor didefinisikan dengan suatu formula. Kebanyakan sistem authoring multimedia menyediakan sarana untuk menggambar objek berbentuk vektor, seperti garis, segiempat, oval, polygon dan gambar kompleks yang dibuat dari objek-objek tersebut dan teks. Vektor merupakan garis sederhana yang dideskripsikan oleh lokasi dua ujung titik. Ada banyak perangkat lunak pengolah vektor, di antaranya adalah InkScape dan CorelDraw.

Format *File* Vektor.

Kebalikan dari format gambar bitmap, format gambar vektor berisi deskripsi geometri yang dapat di-render dengan halus pada ukuran tampilan yang diinginkan. *File* format vektor dapat berisi data bitmap. *File* grafik 3D secara teknis merupakan format vektor dengan data *pixel texture mapping* pada permukaan objek vektor semu yang dilengkungkan untuk sesuai dengan sudut pandangan perspektif.

Pada beberapa titik, semua grafik vektor harus dirasterisasi untuk ditampilkan pada monitor digital. Meskipun demikian, gambar vektor juga dapat ditampilkan pada teknologi CRT yang analog.

Plotter merupakan printer yang menggunakan data vektor bukan data piksel untuk menggambar suatu grafik.

a. CGM

CGM adalah singkatan dari *Computer Graphics Metafile* yang merupakan format *file* untuk 2D grafik vektor, grafik bitmap dan teks yang didefinisikan oleh ISO/IEC 8623. Semua elemen garis dapat dispesifikasikan dengan *file* sumber berbasis teks yang dapat dikompilasi ke dalam *file* biner atau satu dari dua representasi teks.

CGM membuat pertukaran data grafik untuk representasi komputer dari informasi grafis 2D yang tidak tergantung dari aplikasi tertentu, sistem, platform atau perangkat. CGM sudah banyak diadopsi pada gambar teknik dan perancangan profesional, tetapi saat ini sudah mulai tergantikan dengan format lain, seperti SVG dan DXF.

b. SVG

SVG adalah singkatan dari *Scalable Vektor Graphics*, yang merupakan standar terbuka yang diciptakan dan dikembangkan oleh *World Wide Web Consortium* untuk memenuhi kebutuhan yang serbaguna dan berubah-ubah, scriptable, serta mempunyai semua kegunaan untuk web dan lain-lain.

Format SVG tidak mempunyai metode pemampatannya sendiri, tetapi grafik SVG dapat dimampatkan dengan menggunakan aplikasi lain, seperti gzip.

c. PICT

File PICT (atau *Picture*) merupakan format yang umum digunakan pada OS Mac dan aplikasi pengolah layout, serta digunakan untuk melakukan transfer *file* antar aplikasi tersebut. Format PICT sangat efektif untuk melakukan kompresi *file* yang mempunyai area warna yang solid dan luas. Selain itu, dalam aplikasi Adobe InDesign mendukung gambar RGB PICT dengan resolusi yang bervariasi dan di-*embed* pada gambar QuickTime.

Gambar PICT tidak mendukung separasi warna serta tidak tergantung pada peralatan tertentu. Untuk perangkat non-Mac, *file* format PICT ini memerlukan program Quicktime untuk membuka dan menggunakannya.

Format *File* Vektor yang lain.

d. ODG (Open Document Graphics), merupakan standar terbuka,

e. EPS (Encapsulated PostScript),

f. PDF (Portable Document Format),

g. SWF (Small Web Format),

- h. WMF/EMF (*Windows Metafile/Enhanced Metafile*), dan
- i. XPS (XML Paper Specification).



DAFTAR PUSTAKA

Binanto, Iwan. 2010. Multimedia Digital Dasar Teori plus Pengembangannya.
Yogyakarta: Andi Offset.

Pengertian Bitmap Available at <http://id.wikipedia.org/wiki/Bitmap>

Pengertian istilah Available at

http://muhammadarif.web.id/pdf_file/arief_resolusi_desain.pdf

Pengertian istilah Available at <http://www.total.or.id>

Pengertian Vektor Available at <http://masbadar.com/2009/06/02/pengertian-gambar-berformat-vector-dan-gambar-berformat-bitmap/>



BAB II

METODE KOMPRESI DATA GAMBAR

Pada umumnya representasi data digital membutuhkan memori yang besar, di sisi lain kebanyakan data misalnya gambar mengandung duplikasi. Duplikasi ini dapat berarti dua hal. Pertama, besar kemungkinan suatu piksel dengan piksel lain tetangganya memiliki intensitas yang sama, sehingga penyimpanan setiap piksel memboroskan tempat. Kedua, gambar banyak mengandung bagian (*region*) yang sama, sehingga bagian yang sama ini tidak perlu dikodekan berulang kali. Saat ini, kebanyakan aplikasi menginginkan representasi dengan memori yang lebih sedikit. Pemampatan data atau kompresi data bertujuan meminimalkan kebutuhan memori untuk merepresentasikan data digital. Prinsip umum yang digunakan pada proses kompresi adalah mengurangi duplikasi data sehingga memori untuk merepresentasikan menjadi lebih sedikit daripada representasi data digital semula.

Data digital yang telah dikompresi dapat dikembalikan ke bentuk data digital semula (dekompresi) dimana hal ini tergantung pada aplikasi *software* yang mendukung aplikasi *software* tersebut. Ketika suatu aplikasi mampu menghilangkan atau mengkompresi data yang tidak dibutuhkan, maka aplikasi tersebut juga mampu mengembalikan data yang dihilangkan tersebut sehingga menjadi data digital semula (dekompresi). Namun terdapat juga suatu aplikasi yang dapat mengkompresi namun ketika dekompresi dapat menggunakan aplikasi lain, contohnya aplikasi winzip dan aplikasi winrar.

Ada empat pendekatan yang digunakan pada kompresi suatu data, yaitu:

1. Pendekatan statistik.

Kompresi didasarkan pada frekuensi kemunculan derajat keabuan piksel di dalam seluruh bagian.

Contoh metode : *Huffman Coding*.

2. Pendekatan ruang.

Kompresi didasarkan pada hubungan spasial antara piksel-piksel di dalam suatu kelompok yang memiliki derajat keabuan yang sama dalam suatu daerah gambar atau data.

Contoh metode : *Run-Length Encoding*.

3. Pendekatan kuantisasi.

Kompresi dilakukan dengan mengurangi jumlah derajat keabuan yang tersedia.

Contoh metode : Kompresi Kuantisasi (*Arithmetic Coding*).

4. Pendekatan fractal.

Kompresi dilakukan pada kenyataan bahwa kemiripan bagian-bagian di dalam gambar dapat dieksploitasi dengan suatu matriks transformasi.

Contoh metode : *Fractal Image Compression (Arithmetic Coding)*.

Hal-hal penting dalam kompresi gambar:

1. *Scalability/Progressive Coding/Embedded Bitstream*.

- a. Adalah kualitas dari hasil proses pengkompresian gambar karena manipulasi bitstream tanpa adanya dekompresi atau rekompresi.
- b. Biasanya dikenal pada metode *lossless*.
- c. Contohnya pada saat *preview image* sementara image tersebut di *download*. Semakin baik *scalability*, makin bagus *preview image*.

Tipe-tipe scalability adalah:

a. *Quality Progressive*

Dimana gambar dikompres secara perlahan-lahan dengan penurunan kualitasnya.

b. *Resolution Progressive*

Dimana gambar dikompresi dengan mengkode resolusi gambar yang lebih rendah terlebih dahulu baru kemudian ke resolusi yang lebih tinggi.

c. *Component Progressive*

Dimana image dikompresi berdasarkan komponennya, pertama mengkode komponen gray baru kemudian komponen warnanya.

2. *Region of Interest Coding*

Daerah-daerah tertentu di-encode dengan kualitas yang lebih tinggi daripada yang lain.

3. *Meta Information*

Gambar yang dikompres juga dapat memiliki *meta information* seperti statistic warna, tekstur, small preview image dan author atau *copyright information*.

Dalam kompresi gambar terdapat suatu standar pengukuran *error* (galat) kompresi. Ada dua jenis pengukuran, yaitu:

1. MSE (*Mean Square Error*)

Adalah sigma dari jumlah *error* antara gambar hasil kompresi dan gambar asli. MSE memiliki rumus sebagai berikut:

$$MSE = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N [f(i, j) - g(i, j)]^2 \dots\dots\dots (2.1)$$

Dimana: (x, y) adalah nilai piksel pada gambar asli
 (x', y') adalah nilai piksel pada gambar hasil kompresi
 M, N adalah dimensi gambar

2. *Peak Signal to Noise Ratio (PSNR)*

Adalah untuk menghitung peak error.

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$

Jika nilai MSE semakin rendah, maka hasil semakin baik. Sebaliknya, jika nilai PSNR semakin tinggi, maka hasil semakin baik.

Algoritma dalam kompresi dan dekompresi.

Algoritma umum untuk kompresi gambar adalah:

1. Menentukan bitrate dan toleransi distorsi gambar dari inputan user.
2. Pembagian data gambar ke dalam bagian-bagian tertentu sesuai dengan tingkat kepentingan yang ada (*classifying*). Menggunakan salah satu teknik DWT (*Discrete Wavelet Transform*) yang akan mencari frekuensi nilai piksel masing-masing lalu menggabungkannya jadi satu dan mengelompokkannya sebagai berikut:

LL	HL
LH	HH

(a) Single Level Decomposition

LL	HL	HL
LH	HH	
LH		HH

(a) Two Level Decomposition

LL	HL	HL	HL
LH	HH		
LH		HH	HH
LH			

(c) Three Level Decomposition

Gambar 2.1 Pengelompokkan Teknik DWT

Keterangan: LL: *Low Low Frequency (most importance)*

HL: *High Low Frequency (lesser importance)*

LH: *Low High Frequency (more lesser importance)*

HH: *High High Frequency (most less importance)*

3. Pembagian bit-bit di dalam masing-masing bagian yang ada bit (*bit allocation*).
4. Lakukan kuantisasi (*quantization*).
 - a. Kuantisasi Skalar: data-data dikuantisasi sendiri-sendiri.
 - b. Kuantisasi Vektor: data-data dikuantisasi sebagai suatu himpunan nilai-nilai vektor yang diperlukan sebagai suatu kesatuan.
5. Lakukan pengenkodingan untuk masing-masing bagian yang sudah dikuantisasi tadi dengan menggunakan teknik *entropy coding* (Huffman dan Aritmatik) dan menuliskannya ke dalam *file* hasil.

Sedangkan algoritma umum untuk dekompresi adalah sebagai berikut:

1. Baca data hasil kompresi menggunakan *entropy encoder*.
2. Dekuantisasi data
3. *Rebuild Image*.

2.1. Gambar Tidak Dimampatkan.

Pada komputer grafis, bitmap (disebut juga *grafik raster*) merupakan struktur data yang merepresentasikan kubus piksel atau titik warna yang dapat dilihat dengan monitor, kertas atau media penampil lainnya. Gambar bitmap disimpan dalam *file* gambar dengan beragam format. Bitmap secara teknis ditandai dengan panjang dan

lebar gambar dalam satuan piksel dan dengan jumlah bit per piksel (kedalaman warna yang menentukan jumlah warna yang dapat ditampilkan).

Pada gambar bitmap, *file* gambar mendefinisikan dengan tepat warna dari setiap piksel di dalam gambar. Sebagai contoh, pendefinisian sebuah *file* gambar berukuran 400 x 400 piksel membutuhkan 24 bit per piksel untuk 160.00 piksel atau sebesar 480.00 byte. Ukuran ini termasuk besar sehingga membutuhkan cara untuk memampatkan *file*.

Kedalaman warna 24 bit per piksel berarti satu piksel mengandung tiga elemen warna, yaitu *red*, *green*, *blue* (RGB) yang masing-masing mempunyai nilai sebesar 8 bit ($3 \times 8 = 24$ bit).

Ukuran *file* sebesar 480.000 byte dari sebuah *file* gambar yang berukuran 400 x 400 piksel dengan kedalaman 24 bit didapatkan dari perhitungan sebagai berikut:

$$400 \times 400 \times 3 = 480.000 \text{ byte}$$

Perhitungan di atas merupakan perhitungan yang mencari ukuran *file* untuk gambar bitmap yang tidak dimampatkan. Perhitungan tersebut dapat dirumuskan menjadi:

$$\text{Ukuran} = \text{panjang} * \text{lebar} * n/8, \text{ dengan } n \text{ adalah kedalaman warna.}$$

2.2. Gambar Dimampatkan

Pemampatan gambar merupakan aplikasi dari pemampatan data pada gambar digital. Tujuannya adalah untuk mengurangi data gambar yang berlebih supaya dapat disimpan atau dikirim secara efisien. Ada dua macam pemampatan gambar, yaitu metode *lossless* dan *lossy*.

2.2.1. Metode *Lossy*

Metode pemampatan *lossy* terutama digunakan untuk gambar natural, seperti foto pada suatu aplikasi dengan detail yang kurang jelas, namun masih dapat diterima dan dianggap wajar. Pemampatan *lossy* yang menghasilkan perbedaan yang tidak terlihat oleh mata manusia dapat disebut *visually lossless*. Keuntungan metode *lossy* dibandingkan metode *lossless* adalah metode *lossy* dapat menghasilkan *file* termampatkan yang lebih kecil dari *file* hasil pemampatan metode *lossless* (di beberapa kasus).

2.2.2. Metode *Lossless*

Pemampatan *lossless* lebih sering digunakan untuk gambar-gambar maya, seperti gambar teknik, ikon atau komik. Di samping itu, metode pemampatan *lossless* juga lebih sering digunakan untuk konten yang lebih mengutamakan mutu gambar, seperti pencitraan medis atau hasil pemindaian untuk kebutuhan pengarsipan. Pada kasus seperti ini, metode *lossy* tidak dapat digunakan karena metode pemampatan *lossy* akan lebih memunculkan ketidakjelasan gambar terutama ketika digunakan pada bit rate rendah.

Lossless data kompresi adalah kelas dari algoritma data kompresi yang memungkinkan data yang asli dapat disusun kembali dari data kompresi. *Lossless* data kompresi digunakan dalam berbagai aplikasi seperti format ZIP dan GZIP. *Lossless* juga sering digunakan sebagai komponen dalam teknologi kompresi data *lossy*. Kompresi *lossless* digunakan ketika sesuatu yang penting pada kondisi asli. Beberapa format gambar seperti PNG atau GIF hanya menggunakan kompresi *lossless*, sedangkan yang lainnya seperti TIFF dan MNG dapat menggunakan metode *lossy* atau *lossless*.

Metode *lossless* menghasilkan data yang identik dengan data aslinya, hal ini dibutuhkan untuk banyak tipe data, contohnya: *executable code*, *word processing*

files, tabulated numbers dan sebagainya. Misalnya pada gambar dimana metode ini akan menghasilkan hasil yang tepat sama dengan gambar semula, piksel per piksel sehingga tidak ada informasi yang hilang akibat kompresi. Namun rasio kompresi dengan metode ini sangat rendah.

Metode ini sangat cocok untuk kompresi gambar yang mengandung informasi penting yang tidak boleh rusak akibat kompresi, misalnya gambar hasil diagnosa medis. Contoh metode lossless adalah *Run-Length Coding, Variable Length Coding, Dictionary Based Coding* dan *Arithmetic Coding*.

a. Metode *Run-Length*.

Metode ini mengenali jumlah *run* pada suatu piksel dengan level intensitas sama dan berurutan dalam satu arah tertentu. Kemudian ciri tekstur dapat diperoleh dari SRE (*Short Run Emphasis*), LRE (*Long Run Emphasis*), GRU (*Grey Level Uniformity*), RLU (*Run Length Uniformity*) dan RPC (*Run Percentage*). Kelima ciri tersebut kemudian digunakan untuk klasifikasi dengan menggunakan metode LDA (*Linear Discriminant Analysis*) dan k-NN (*k-Nearest Neighbor*). Sedangkan gambar tekstur yang dianalisis berasal dari VisTex Database.

Metode *Run Length* menggunakan distribusi suatu piksel dengan intensitas yang sama secara berurutan dalam satu arah tertentu sebagai primitifnya. Masing-masing primitif didefinisikan atas panjang, arah, dan level keabuan. Panjang dari primitif tekstur pada arah yang berbeda dapat digunakan untuk menggambarkan suatu tekstur.

Analisis tekstur metode *Run Length* ini digunakan untuk membedakan gambar halus dan gambar kasar. Tekstur kasar menunjukkan banyak piksel tetangga yang memiliki intensitas yang sama, sedangkan tekstur halus menunjukkan sedikit piksel tetangga yang menunjukkan intensitas yang sama.

Untuk melakukan ekstraksi ciri dengan menggunakan metode *Run Length* gambar aras keabuan dengan matriks (f, g) harus ditransformasikan terlebih dahulu ke dalam matriks *Grey Level Run Length* (GLRL), $(GLRL, g)$

$$(f, g) \longrightarrow (GLRL, g) \dots\dots\dots (2.2)$$

Elemen matriks dari GLRL $(GLRL, g)$ menghitung banyaknya primitif dengan panjang g dan level keabuan f . Jumlah dari primitif dapat diperoleh dengan persamaan:

$$= \sum_{f=0}^{255} \sum_{g=1}^{255} (GLRL, g) \dots\dots\dots (2.3)$$

Adapun ciri dari tekstur dapat diperoleh dari persamaan-persamaan berikut ini:

1. *Short Run Emphasis* (SRE)

SRE mengukur distribusi *short run*. SRE sangat tergantung pada banyaknya *short run* dan diharapkan bernilai besar pada tekstur halus.

$$= \sum_{f=0}^{255} \sum_{g=1}^{255} \frac{(GLRL, g)}{g} \dots\dots\dots (2.4)$$

2. *Long Run Emphasis* (LRE)

LRE mengukur distribusi *long run*. LRE sangat tergantung pada banyaknya *long run* dan diharapkan bernilai besar pada tekstur kasar.

$$= \sum_{f=0}^{255} \sum_{g=1}^{255} (GLRL, g) \dots\dots\dots (2.5)$$

3. *Grey Level Uniformity* (GLU)

GLU mengukur persamaan nilai derajat keabuan di seluruh gambar dan diharapkan bernilai kecil jika nilai derajat keabuan serupa di seluruh gambar.

$$= \sum_{i,j} \frac{1}{n} \cdot (\sum_{k=0}^{255} (f_{i,j,k})^2) \dots\dots\dots (2.6)$$

4. *Run Length Uniformity* (RLU)

RLU mengukur persamaan panjangnya run diseluruh gambar dan diharapkan bernilai kecil jika panjangnya *run* serupa di seluruh gambar.

$$= \sum_{i,j} \frac{1}{n} \cdot (\sum_{k=1}^{255} (f_{i,j,k})^2) \dots\dots\dots (2.7)$$

5. *Run Percentage* (RPC)

RPC mengukur keserbasamaan dan distribusi *run* dari sebuah gambar pada arah tertentu. RPC bernilai paling besar jika panjangnya run adalah 1 untuk semua derajat keabuan pada arah tertentu.

$$\frac{\sum_{i,j} \sum_{k=1}^{255} (f_{i,j,k})}{\sum_{i,j} \sum_{k=0}^{255} (f_{i,j,k})} \dots\dots\dots (2.8)$$

0°	1	2	3
0	2	3	3
2	1	1	1
3	0	3	0

Citra aras keabuan (,)

	1	2	3	4
0	4	0	0	0
1	1	0	1	0
2	3	0	0	0
3	3	1	0	0

Run Length (,)

Gambar 2.2 Derajat Keabuan

Pengenalan Pola.

Pengenalan pola merupakan ilmu mengenai diskripsi atau klasifikasi dari hasil ekstraksi ciri yang memetakan suatu fitur, yang merupakan ciri utama suatu objek (yang dinyatakan dalam sekumpulan bilangan-bilangan) ke suatu kelas yang sesuai. Salah satu metode pengenalan pola yaitu metode statistik.

Model statistik didefinisikan sebagai sebuah keluarga dari fungsi kerapatan peluang bersyarat kelas $Pr (|)$, yakni peluang vektor fitur jika diberikan kelas . Metode ini dapat dilakukan dengan pendekatan supervised maupun pendekatan *unsupervised*. Pendekatan *supervised* (dengan pengarahan) menyediakan training set untuk mengerahkan atau memberi informasi atau pengetahuan tentang kelas-kelas yang ada. Contoh pendekatan *supervised* adalah metode k-NN, Bayes, Linear *Discriminant Analysis* dan lain-lain.

Pendekatan *unsupervised* (tanpa pengarahan) tidak menyediakan *training set*. Informasi yang disediakan adalah jumlah klaster yang ada. Sehingga

pengelompokkan dilakukan sepenuhnya berdasarkan karakteristik data. Contoh dari pendekatan ini adalah k-Mean.

Linear Discriminant Analysis (LDA)

Tujuan utama dari analisis diskriminan adalah untuk memperoleh kaidah matematis, yang dikenal dengan fungsi diskriminan, yang dapat digunakan untuk memisahkan kelompok objek yang berbeda, seperti kelompok air dan pasir. Fungsi diskriminan ditentukan oleh parameter statistik yang tergambar dari populasi ciri objek pada kelas yang telah diketahui. Vektor ciri yang telah diperoleh dari objek yang akan diklasifikasikan dipergunakan sebagai masukan. Keluarannya biasanya bernilai skalar yang dapat digunakan untuk menentukan kelas yang paling memungkinkan. Fungsi diskriminan menetapkan permukaan keputusan dari - dimensi yang memisahkan kelas-kelas distribusi ciri pada -dimensi ruang ciri.

k-Nearest Neighbor (k-NN)

Metode k-NN merupakan pengembangan dari estimasi kerapatan non-parametrik. Ekspresi umum dari estimasi fungsi kerapatan peluang non-parametrik dapat dituliskan dengan rumus sebagai berikut:

$$f(x) = \frac{1}{n} \sum_{i=1}^k \frac{1}{V_k} \dots \dots \dots (2.9)$$

- dimana: V_k = volume yang melingkupi x
- n = jumlah total sampel
- k = jumlah sampel di dalam

Keuntungan utama dari metode k-NN adalah metode ini memberikan pendekatan yang sangat sederhana untuk pemilahan optimal Bayes. Apabila terdapat suatu data masukan yang jumlahnya n buah dimana k buah dari kelas C_1 dan dilakukan pemilahan sampel n yang tidak diketahui, maka dapat digambarkan suatu

bola (*hyper*) dengan volume disekitar . Bila volume ini mengandung -buah data, dimana -buah diantaranya dari kelas , maka fungsi kebolehjadian dapat didekati dengan metode k-NN dengan $P(x_i) = \frac{1}{N}$. Dengan cara yang sama, fungsi kerapatan tak-kondisional diestimasi oleh $P(x_i) = \frac{1}{N}$. Sedangkan prior-nya didekati dengan rumus $P(x_i) = \frac{1}{N}$. Setelah dikumpulkan, pemilah Bayes menjadi:

$$P(x_i) = \frac{P(x_i)}{P(x_i)} = \frac{P(x_i)}{P(x_i)} = \dots \dots \dots (2.10)$$

Pada metode k-NN, penggunaan nilai yang besar memiliki keuntungan, yaitu menyediakan informasi probabilistik. Akan tetapi, pengambilan yang terlalu besar akan merusak lokalitas estimasi dan juga meningkatkan beban komputasi.

b. Metode *Variable Length*.

Sejak entropi menunjukkan informasi yang terkandung dalam sumber informasi, ini mengarah kita pada sebuah keluarga metode *coding* yang umumnya dikenal sebagai metode pengkodean entropi. Seperti yang dijelaskan sebelumnya, *Variable Length* adalah salah satu metode yang cukup terkenal. Di sini ada tiga metode yang akan dibahas, yaitu Algoritma Huffman, Algoritma Shannon-Fano dan Algoritma Adaptive Huffman.

Algoritma Shannon-Fano.

Algoritma Shannon-Fano dikembangkan oleh Shannon (Bell Labs) dan Robert Fano (MIT) pada tahun 1949. Algoritma ini merupakan salah satu algoritma pertama yang tujuannya adalah membuat *codebook* dengan redundansi minimum. Ide dasar dari membuat *codebook* dengan *Variablecode Length*, seperti Huffman *codes* yang

ditemukan beberapa tahun kemudian. *Codebook* yang dihasilkan diperoleh dari pembuatan pohon Shannon-Fano dilakukan secara *top-down*.

Secara umum algoritma Shannon-Fano adalah sebagai berikut:

1. Hitung masing-masing frekuensi kemunculan simbol.
2. Urutkan frekuensi kemunculan simbol dari yang terbesar ke yang terkecil dan jumlahkan seluruh frekuensi kemunculan simbol, kemudian masukkan ke sebuah node.
3. Bagi menjadi dua buah node dengan jumlah frekuensi kemunculan simbol yang sama besar, dan berikan kode '0' untuk bagian kiri dan kode '1' untuk bagian kanan.
4. Ulangi langkah kedua, hingga tidak ada node yang tidak dapat dibagi lagi.
5. Substitusi simbol-simbol yang ada dengan kode yang telah dihasilkan.

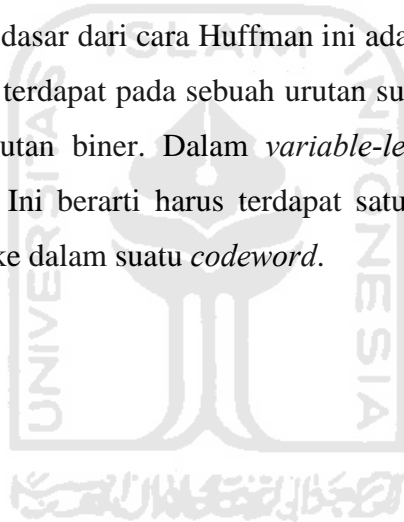
Pada dasarnya proses *coding* dengan algoritma ini membutuhkan data akan frekuensi jumlah kemunculan suatu karakter pada setiap pesan. Dan cara kerja dari algoritma ini sama persis dengan Huffman. Algoritma ini membentuk sebuah pohon, kemudian meng-*encoding*-nya dan yang terakhir adalah mengembalikannya atau *decoding*. Hanya saja perbedaan yang mendasar terdapat pada pembuatan pohon. Pembuatan pohon pada Shannon-Fano dibuat berdasarkan proses *top-down*, sementara Huffman *bottom-up*. Sebuah pohon Shannon-Fano dibangun sesuai dengan spesifikasi yang dirancang untuk mendefinisikan tabel kode yang efektif. Tiga prinsip utama yang mendasari algoritma ini adalah:

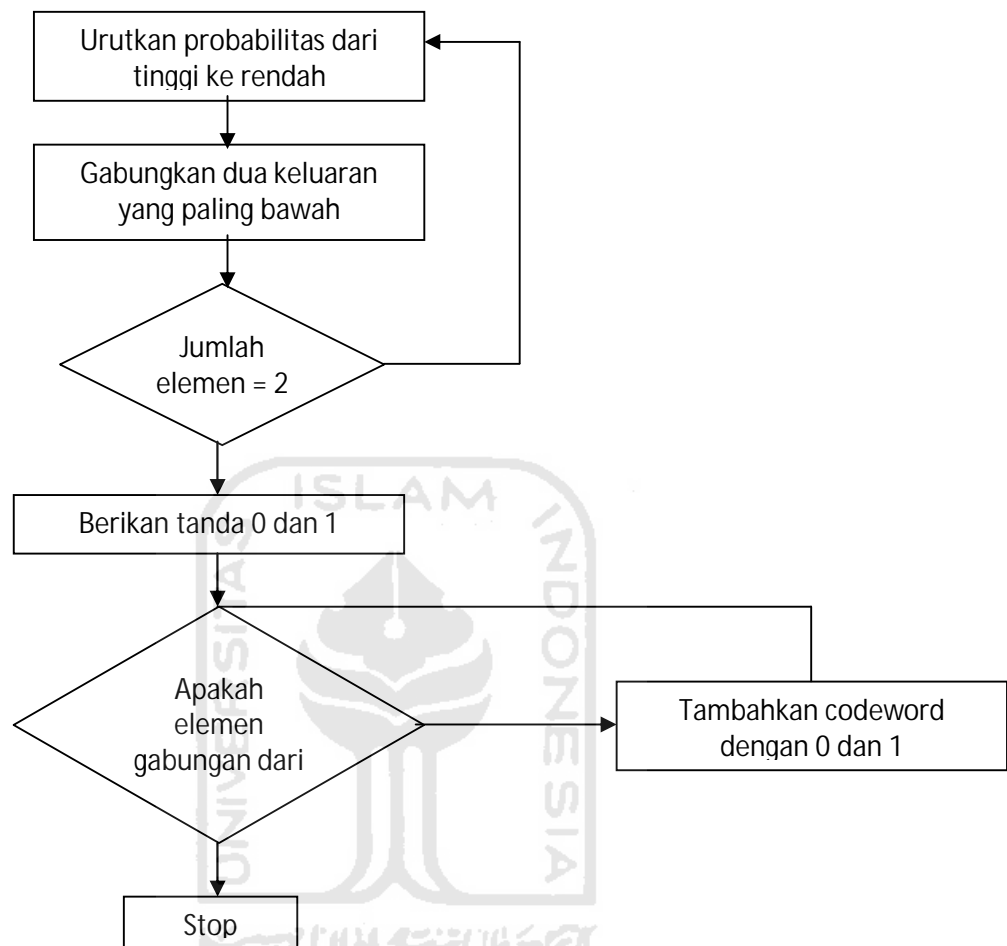
1. Simbol yang berbeda memiliki kode yang berbeda.
2. Kode untuk simbol yang sering muncul memiliki jumlah bit yang lebih sedikit dan sebaliknya simbol yang jarang muncul memiliki kode dengan jumlah bit yang lebih besar.
3. Walaupun berbeda jumlah bit-nya tetepai kode harus tetap dikodekan secara pasti (tidak ambigu).

Algoritma Huffman.

Pertama kali diperkenalkan oleh David A. Huffman lewat makalahnya pada tahun 1952, metode ini menarik sejumlah besar penelitian dan telah diadopsi dalam aplikasi penting dan komersial, seperti mesin faks, JPEG dan MPEG. Tidak seperti Shannon-Fano yang memakai *top-down*, langkah-langkah algoritma Huffman *Coding* dijelaskan dengan cara *bottom-up*.

Dalam Algoritma Huffman, panjang blok dari keluaran sumber dipetakan dalam blok biner berdasarkan panjang variable. Cara seperti ini juga disebut sebagai *fixed to variable-length coding*. Ide dasar dari cara Huffman ini adalah memetakan mulai dari simbol yang paling banyak terdapat pada sebuah urutan sumber sampai dengan yang jarang muncul menjadi urutan biner. Dalam *variable-length coding*, sinkronisasi merupakan suatu masalah. Ini berarti harus terdapat satu cara untuk memecahkan urutan biner yang diterima ke dalam suatu *codeword*.





Gambar 2.3 Alur Algoritma Huffman

Seperti yang disebutkan di atas, bahwa ide Algoritma Huffman adalah memilih panjang *codeword* dari yang paling besar probabilitasnya sampai dengan urutan *codeword* yang paling kecil probabilitasnya. Apabila kita dapat memetakan setiap keluaran sumber dari setiap probabilitas h ke sebuah *codeword* dengan panjang $1/h$ dan pada yang bersamaan dapat memastikan bahwa dapat dikodekan secara unik, kita dapat mencari rata-rata panjang kode (.) Algoritma Huffman dapat men-dekodekan secara unik dengan () minimum dan optimum pada keunikan dari kode-kode tersebut. Ada 3 cara penguraian kode Huffman, yaitu:

1. Menggunakan Pohon Huffman

Langkah-langkah yang dilakukan dalam penguraian kode menggunakan pohon Huffman adalah:

- a. Baca bit pertama dari stringer biner masukan
- b. Lakukan traversal pada pohon Huffman mulai dari akar sesuai dengan bit yang dibaca. Jika bit yang dibaca adalah 0, baca anak kiri, tetapi jika bit yang dibaca adalah 1, baca anak kanan.
- c. Jika anak dari pohon bukan merupakan daun, baca bit berikutnya dari stringer biner.
- d. Traversal hingga ditemukan daun.
- e. Pada daun tersebut, simbol ditemukan dan proses penguraian kode selesai.
- f. Proses penguraian kode dilakukan hingga seluruh string biner masukan diproses.

2. Menggunakan Tabel Kode Huffman

Cara kedua untuk menguraikan kode Huffman adalah dengan menggunakan tabel kode Huffman. Oleh karena kode Huffman disusun menggunakan kode awalan (*prefix code*), dapat dipastikan bahwa sebuah kode untuk sebuah simbol/karakter tidak boleh menjadi awalan dari simbol yang lain. Oleh karena itu, string biner yang berisi hasil enkripsi dapat dipisahkan dengan mudah berdasarkan setiap rangkaian bit-nya untuk diuraikan menjadi data semula. Yang perlu dilakukan hanyalah melihat setiap rangkaian bit yang ditemukan dalam string biner hasil enkripsi di dalam tabel kode Huffman.

Algoritma Huffman mempunyai kompleksitas waktu $(n) = (n \cdot H)$ ini karena dalam melakukan sekali proses iterasi pada saat penggabungan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar membutuhkan waktu

(\log ,) dan proses itu dilakukan berkali-kali sampai hanya tersisa satu buah pohon Huffman, yang berarti dilakukan sebanyak kali.

Sementara Algoritma metode Huffman adalah:

1. Buat data gambar yang berupa matriks tersebut menjadi vektor.
2. Tentukan frekuensi kemunculan tiap warna atau derajat keabuan.
3. Urutkan secara menaik warna atau nilai keabuan berdasarkan frekuensi kemunculannya atau peluang kemunculan piksel dalam gambar. Setiap nilai dinyatakan sebagai pohon bersimpul tunggal dan setiap simpul di-assign dengan frekuensi kemunculan nilai tersebut
4. Gabungkan 2 buah pohon yang mempunyai frekuensi kemunculan paling kecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua pohon penyusunnya. Frekuensi dengan nilai lebih kecil diletakkan di sisi kiri dan diberi bobot '0' sedangkan di sisi kanan diberi bobot '1'.
5. Ulangi langkah 4 sampai tersisa 1 pohon biner.
6. Telusuri pohon biner dari akar ke daun. Barisan bobot isi dari akar ke daun menyatakan kode Huffman untuk derajat keabuan atau warna yang bersesuaian.
7. Mengganti data dengan kode Huffman yang bersesuaian.
8. Menyimpan data lebar gambar, tinggi gambar, kode bit untuk tiap nilai, data warna yang terdapat di dalam gambar, dan data gambar yang sudah dikodekan ke dalam *file* hasil kompresi

Untuk mengembalikan data gambar terkompres menjadi data gambar aslinya, diperlukan suatu algoritma dekompresi yang merupakan kebalikan dari algoritma kompresi. Berikut algoritma dekompresi Huffman:

1. Baca *file* hasil kompresi dan data-datanya dimasukkan ke variable yang sesuai yaitu variable ukuran gambar, variable kode bit data, dan variable warna.

2. Baca data kode bit per bit dari kiri ke kanan dan dicocokkan dengan kode Huffman dari data warna yang didapat. Demikian seterusnya konversi dilakukan hingga data terakhir.
3. Rekonstruksi gambar dengan menggunakan data ukuran gambar, berarti data piksel berbentuk 1D dipenggal baris dan kolom sesuai ukuran gambar.

Metode *Adaptive* Huffman.

Metode *Adaptive* Huffman mengubah sifat statis metode Huffman menjadi dinamis atau adaptif. Sifat ini didapat dari pembaharuan pohon Huffman yang dilakukan terus selama proses kompresi dan dekompresi. Proses ini sebenarnya sama dengan pembuatan pohon baru, tetapi berdasarkan pohon Huffman yang telah ada. Jadi tidak bermula dari penghitungan peluang kemunculan.

Setiap kali dijumpai simbol yang sama maka peluang kemunculan ditambahkan dan pohon Huffman disesuaikan. Pohon yang baru ini digunakan untuk kompresi atau dekompresi simbol berikutnya. Jika simbol yang baru tidak terdapat pada pohon yang telah dibangun, maka simbol tersebut ditambahkan. Proses ini dilakukan terus hingga simbol yang terakhir dari pesan sumber.

Adaptive Huffman merupakan salah satu teknik *lossless* coding di mana teknik ini menentukan *mapping* dari pesan sumber menjadi *codeword* didasari pada perkiraan probabilitas pesan sumber. Kode bersifat adaptif, berganti sesuai dengan perkiraan optimalnya pada saat itu. Teknik *coding* ini kemudian diterapkan pada bagian *entropy coding* dari metode kompresi gambar JPEG.

Kemudian dapat dibuktikan bahwa *Adaptive* Huffman coding dapat digunakan sebagai *alternative entropy coding* pada metode kompresi JPEG. Faktor yang mempengaruhi rasio dan waktu kompresi adalah lebar data kuantitas. Sedangkan nilai PSNR gambar dipengaruhi oleh proses DCT dan kuantitas.

Rasio kompresi gambar yang dihasilkan oleh sistem ini bergantung pada lebar data yang dihasilkan oleh proses kuantisasi. Jika lebar dan jumlah data terlalu besar maka kode tetap yang dihasilkan oleh Adaptive Huffman akan menjadi panjang dan *codeword* yang dihasilkan akan semakin banyak sehingga rasio kompresi yang dihasilkan juga menjadi kecil, maka panjang kode tetap yang dihasilkan oleh Adaptive Huffman akan menjadi pendek dan jumlah *codeword* yang dihasilkan juga menjadi sedikit sehingga rasio kompresi juga menjadi tinggi dan waktu kompresi juga menjadi lebih pendek. Nilai rasio kompresi dari *file* yang dihasilkan oleh sistem secara umum sedikit lebih baik daripada JPEG standar.

Adapun Algoritma Adaptive Huffman sebagai berikut:

1. Menambahkan langkah Huffman (statis Huffman) dengan dua buah simpul. Satu simpul sebagai Huffman statis dan simpul yang lain sebagai daun. Tambahkan jumlah daun, jika bukan Huffman statis maka langsung menuju daun.
2. Jika pada blok tidak terdapat angka tertinggi, tukarkan dengan angka tertinggi pada blok.
3. Tambahkan jumlah dari simpul tersebut.
4. Periksa apakah simbol tersebut merupakan simpul akar. Jika bukan pergi ke simpul parent.

c. Dictionary Based Coding

Metode ini disebut juga metode LZW, metode kompresi yang bersifat *lossless* yang ditemukan oleh Lempel, Ziv, dan Welch pada tahun 1984. Metode ini menggunakan metode *dictionary* atau kamus. Secara umum algoritma kompresi LZW ini akan membentuk *dictionary* selama proses kompresinya berlangsung kemudian setelah selesai maka *dictionary* tersebut tidak ikut disimpan dalam *file* yang telah

terkompresi. Prinsip kompresi akan terjadi ketika besar bit untuk *dictionary* yang telah ditentukan menggantikan deretan karakter atau string yang terbentuk. Sedangkan dalam proses dekompresinya untuk memperoleh hasil yang sama dengan *file* sebelum dikompresi LZW akan membuat kembali *dictionary* selama proses dekompresinya berlangsung.

Prinsip umum kerja metode LZW adalah mengecek setiap karakter yang muncul kemudian menggabungkan dengan karakter selanjutnya menjadi sebuah string. Jika string baru tersebut tidak berada dalam *dictionary* atau belum diindekskan, maka string baru tersebut akan diindekskan ke dalam *dictionary*

Adapun algoritma dari metode LZW adalah:

1. Inisialisasi *input stream* yang mengandung karakter-karakter dasar.
2. Membaca karakter dari input stream, jika EOF maka menuju langkah 5.
3. Gabungkan karakter awal dengan karakter yang dibaca selanjutnya menjadi sebuah string.
4. Jika string baru ini tidak ada dalam *dictionary*, maka
 - a. Buat indeks baru untuk string baru tersebut.
 - b. Menuju langkah 2.
5. Jika string yang terbentuk telah memiliki indeks di dalam *dictionary* maka
 - a. gabungkan karakter sebelumnya dengan karakter yang dibaca atau string yang telah terbentuk menjadi sebuah string baru.
 - b. Menuju langkah 2.
6. Tulis output kode.

Dictionary

Dictionary melakukan inisialisasi untuk pertama kalinya terhadap karakter-karakter dasar huruf alphabet. Pembacaan input stream dilakukan per-byte karena

setiap byte memiliki alamat yang unik dalam memory, secara default *dictionary* 256 pointer awal yang dimulai dari 0-255 untuk pengkodean ASCII, sedangkan jumlah pointer untuk string yang terbentuk setelah membaca input stream dapat ditentukan dari besarnya bit yang dipakai yang dapat dirumuskan dengan rumus 2^{n-256} , di mana n menyatakan jumlah bit yang dipakai dan 256 menyatakan karakter-karakter ASCII yang telah diinisialisasi. Indeks untuk string baru yang terbentuk diperoleh dari nilai selanjutnya yang tersedia dalam memori.

Codewords

Codeword merupakan kode yang bersifat unik yang mewakili karakter atau string baru yang terbentuk setiap kali algoritma LZW memproses inputan, seperti yang telah disebutkan sebelumnya, jumlah dari string-string yang terdapat dalam *dictionary* ditentukan dari besarnya bit *dictionary* yang dipakai, di mana *codeword* mewakili setiap string-string tersebut, semakin banyak string yang dihasilkan maka semakin besar pula bit *dictionary* yang dibutuhkan untuk memberikan indeks bagi setiap string yang dihasilkan.

Pencarian dan Penginputan String.

Proses penambahan indeks dari string baru yang terbentuk akan dilakukan jika *dictionary* tidak penuh dan string tersebut belum ada di *dictionary*. Pencarian tempat bagi indeks pada indeks-indeks *array* yang masih kosong diperlukan untuk menentukan tempat yang sesuai bagi indeks untuk string baru yang terbentuk.

Dekompresi.

Dalam proses dekompresi, algoritma LZW tidak menyimpan *string table* yang berisi indeks-indeks dari setiap *codeword* yang dihasilkan dalam proses kompresi ke memori, akan tetapi menggunakan beberapa informasi yang telah disimpan sebelumnya antara lain 255 karakter ASCII, karakter pertama dari inputan dan *codeword* terakhir

Algoritma dari dekomprei metode LZW adalah:

1. Menginisialisasi seluruh *dictionary* yang mengandung karakter-karakter ASCII.
2. Baca kode (huruf atau *codeword*) pertama dari hasil kompresi LZW, kemudian output-kan karakter yang dibaca, kemudian simpan kode ke variable .
3. Baca kode selanjutnya
4. *Copy* kode yang dibaca ke variable entry
5. *Output*-kan kode (karakter)
6. Gabungkan value dari variable dengan karakter pertama dari entry kemudian simpan sebagai *dictionary*.
7. Isi variable dengan value dari variable entry.
8. Kembali ke langkah 3.

d. Metode *Arithmetic*.

Prinsip Metode *Arithmetic* diperkenalkan pertama kali oleh Peter Elias sekitar tahun 1960-an. Seperti halnya dengan metode Huffman, Metode *Arithmetic* juga membuat *codeword* untuk setiap simbol. Hanya saja, bila pada Metode Huffman kode setiap simbol berupa suatu bilangan integer unik. Pada Metode *Arithmetic* kode simbol berupa suatu interval bilangan pecahan.

Codeword pada bilangan *arithmetic* berada dalam suatu interval bilangan antara 0 dan 1. Interval ini diberikan atas dasar probabilitas kehadiran setiap simbol. Batas atas dan batas bawah dari interval tersebut merupakan parameter yang sangat penting untuk proses encoding dan decoding. Semakin tinggi frekuensi kemunculan suatu karakter maka semakin luas interval nilainya dan semakin sedikit representasi bit yang dibutuhkan untuk mewakili nilai tersebut. Sebaliknya semakin rendah

frekuensi kemunculan karakter maka semakin pendek intervalnya dan representasi bit yang dibutuhkan semakin besar.

Sama seperti Metode Huffman, Metode *Arithmetic* diawali dengan pembentukan tabel probabilitas setiap simbol. Berdasarkan tabel tersebut batas dan bawah interval setiap simbol akan dibuat.

Berikut adalah algoritma Metode *Arithmetic*:

```

LOW          0,0
HIGH         1,0
WHILE not end of input stream
  Get next CHARACTER
  RANGE=HIGH-LOW
  HIGH = LOW + RANGE*high range of
CHARACTER
  LOW = LOW + RANGE*low range of CHARACTER
END WHILE
Output LOW

```

Pada algoritma di atas, *LOW*, *HIGH* dan *RANGE* berturut-turut menyatakan batas bawah, batas atas dan jangkauan interval dari setiap simbol (*CHARACTER*). Pada tahap awal, *LOW* dan *HIGH* diinisialisasi dengan nilai '0' dan '1' kemudian pada proses selanjutnya kedua nilai tersebut diperbarui dengan rumus:

$$HIGH = LOW + RANGE * \text{high range of CHARACTER}$$

$$LOW = LOW + RANGE * \text{low range of CHARACTER}$$

DAFTAR PUSTAKA

Kompresi Citra Available at <http://angga212.wordpress.com/2009/01/06/kompresi-citra/>

Metode Kompresi Citra Available at

<http://nurilla7.wordpress.com/2008/11/05/kompresi-data-lossy-dan-lossless/>

Multimedia 2005/2006, Semester Genap. Kompresi Citra. Hlm 1-15

Nurhayati, Oky Dwi. 2005, Kompresi Citra. Hlm 1-33

Pengertian Dari File Format Gambar Available at

<http://prabelism.blogspot.com/2009/12/pengertian-dari-file-format-gambar.html>



BAB III

EKSPLOKASI SOFTWARE KOMPRESI GAMBAR

3.1. Metodologi Kompresi *File* Gambar

1. Kebutuhan *software*

Dalam melakukan eksplorasi *software* kompresi gambar, akan digunakan 5 *software* kompresi gambar yaitu PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter. Dari kelima *software* tersebut, empat diantaranya adalah freeware, yaitu PIXResizer, FastStone Image Viewer, VSO Image Resizer dan Pixillion Image Converter, serta satu diantaranya adalah shareware, yaitu AMS Photo Effects.

2. Kebutuhan data *file* gambar

Kebutuhan data *file* gambar pada eksplorasi *software* dengan pengaturan standar dan pengaturan bit rate dapat dilihat pada Tabel 3.1.

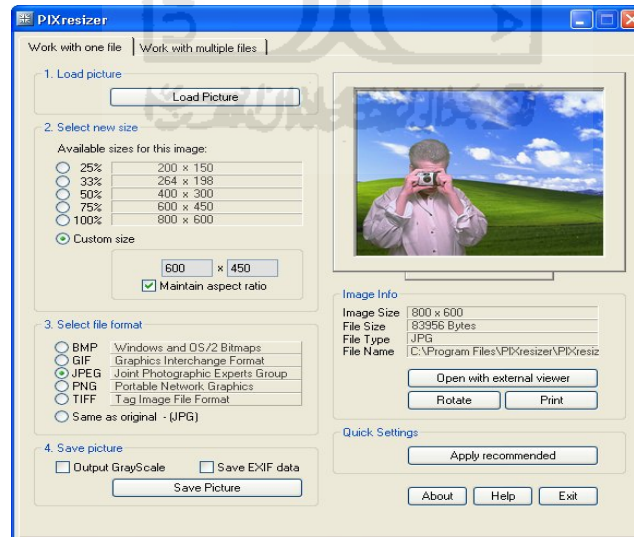
Tabel 3.1 Data kompresi *file* gambar dengan pengaturan standar

Jenis <i>File</i>	Ukuran <i>File</i>	Format Asli	Format Setelah Kompresi
Besar	5,14 MB	.bmp	.gif
Besar	5,14 MB	.bmp	.png
Besar	5,14 MB	.bmp	.jpeg
Sedang	774 KB	.png	.bmp
Sedang	774 KB	.png	.gif
Sedang	774 KB	.png	.jpeg
Kecil	27 KB	.jpeg	.gif

Kecil	27 KB	.jpeg	.bmp
Kecil	27 KB	.jpeg	.png

3.2. PIXResizer

PIXResizer adalah *freeware software* ini diterbitkan oleh Bluefive *Software*. PIXResizer adalah *software* untuk konversi gambar, sangat cocok digunakan untuk kompresi gambar yang akan diunggah di web atau *email*. *File* yang dikompresi disimpan di folder yang berbeda. Sehingga gambar asli tidak ikut terganti ukurannya. Konversi juga lebih cepat karena dapat memungkinkan mengubah ukuran beberapa *file* gambar dalam *batch mode*. PIXResizer dapat mengkonversi beberapa format gambar seperti JPEG, GIF, BMP, PNG and TIFF, memutar gambar, mengkonversi ke *grayscale* dan mengubah ukuran beberapa gambar dalam *batch mode*. PIXResizer juga memungkinkan untuk menyimpan informasi EXIF asli dari gambar. Cocok untuk *webmaster* dan fotografer digital. Tampilan utama *software* PIXResizer dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tampilan Software PIXResizer

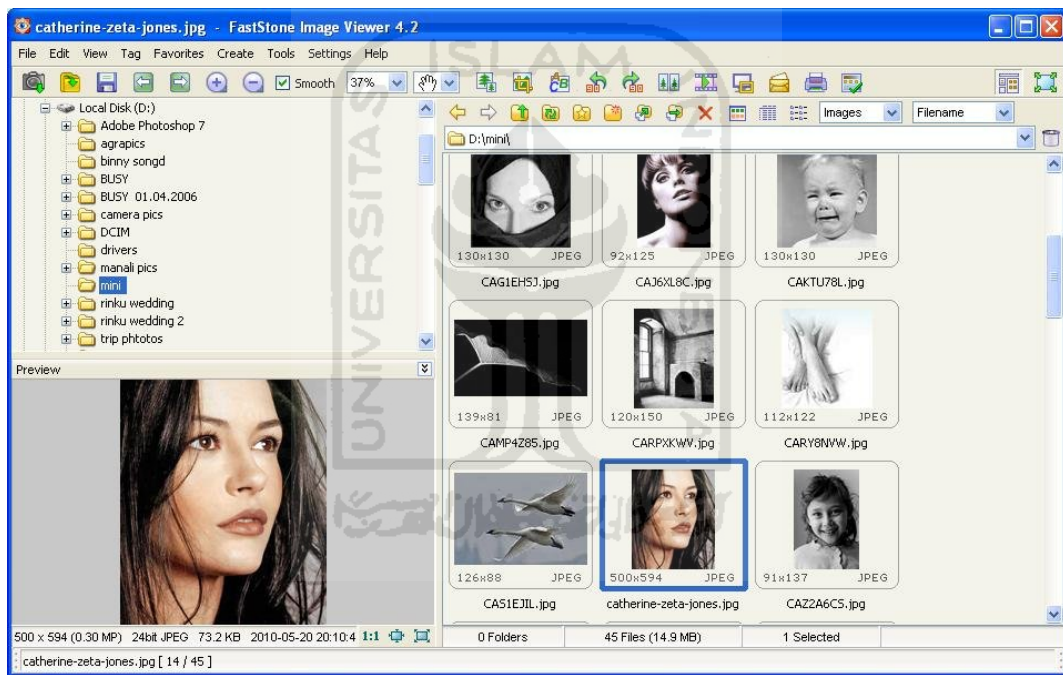
Cara penggunaan *Software* PIXResizer:

1. Install PIXResizer dengan cara *doubleclick* PIXResizer *setup* hasil *download*. Tunggu hingga proses instalasi selesai, *software* dapat langsung digunakan tanpa *restart*.
2. Klik *shortcut* PIXResizer untuk menjalankan *software*.
3. Klik tab *Load Picture* lalu cari *file* yang akan dikonversi melalui jendela yang terbuka.
4. Pilih hasil konversi ukuran yang diinginkan, ada 5 opsi (25%, 33%, 50%, 75% dan 100%), dan 1 opsi *custom size*.
5. Pilih jenis format yang akan digunakan dengan cara memilih *opsi* di bagian kiri *Select File Format*. Misalnya pilih *PNG*.
6. Pilih tempat untuk menyimpan *file* hasil konversi dengan cara klik *Tab Select Picture* di bawahnya ada 2 opsi tambahan, *Output Grayscale*, gambar akan disimpan dalam warna *grayscale*, dan *Save EXIF Data*, metadata dari gambar tersebut akan tetap tersimpan.
7. Bagian kanan ada opsi *Open with external viewer*, gambar akan dibuka pada jendela baru, opsi *Rotate*, berfungsi untuk memutar gambar, dan opsi *Print*, yaitu untuk mencetak gambar.
8. Di bagian kanan bawah, ada menu *Quick Settings* dan opsi *Apply Recommended* yaitu *default setting* dari *software* PIXResizer.
9. Selesai dan *file* siap digunakan.

3.3. FastStone Image Viewer

FastStone Image Viewer adalah freeware. *Software* ini diterbitkan oleh FastStone Soft. FastStone Image Viewer adalah *software* konversi gambar yang cepat, stabil, dan mudah digunakan. Memiliki fitur-fitur yang baru, termasuk tampilan gambar, manajemen, perbandingan, *red-eye removal*, *email*, mengubah ukuran, *cropping*, *retouching* dan penyesuaian warna. Menyediakan akses cepat ke

informasi EXIF, dapat melihat *thumbnail* dan fungsi utama melalui *toolbar* tersembunyi yang muncul saat *mouse* menyentuh keempat tepi layar. Mendukung beberapa format data gambar seperti BMP, JPEG, JPEG 2000, *animated* GIF, PNG, PCX, PSD, EPS, TIFF, WMF, ICO dan TGA. Serta beberapa tipe data di kamera digital seperti CRW, CR2, NEF, PEF, RAF, MRW, ORF, SRF, ARW, SR2, RW2 dan DNG. Tampilan utama *Software* FastStone Image Viewer dapat dilihat pada gambar 3.2.



Gambar 3.2 Tampilan *Software* FastStone Image Viewer

Cara penggunaan *software* FastStone Image Viewer:

1. Install FastStone Image Viewer dengan cara *doubleclick* FastStone Image Viewer *setup* hasil *download*. Tunggu hingga proses instalasi selesai, *software* dapat langsung digunakan tanpa *restart*.

2. Klik *shortcut* FastStone Image Viewer untuk menjalankan *software*.
3. Klik tab *File* lalu cari *file* yang akan dikonversi melalui jendela yang terbuka.
4. Pilih pengaturan gambar yang diinginkan. Misalnya ingin mengubah ukuran, maka pilih tab *Edit* lalu pilih opsi *Resize/Resample*, atau dapat menggunakan *hotkey* yaitu Ctrl+R.
5. Dalam opsi *Resize/Resample*, ada 4 opsi yaitu, mengubah ukuran berdasarkan *Pixels*, *Percent*, *Print Size*, atau *Standard*. Pilihan yang standar adalah *Width* x *Height*-nya dengan 400 x 300.
6. Pada opsi tersebut juga disediakan *Preserve Aspect Ratio*, yang artinya apakah gambar tersebut ingin diubah ukurannya secara proporsional atau tidak.
7. Setelah selesai mengatur ukuran gambar, pilih OK. Akan muncul notifikasi jendela baru, untuk konfirmasi lokasi *file* gambar data yang ingin disimpan.
8. Selesai dan *file* siap digunakan.

3.4. VSO Image Resizer

VSO Image Resizer adalah *freeware*. *Software* ini diterbitkan oleh FastStone Soft. VSO Image Resizer adalah *software* yang ringkas, tetapi sangat berdayaguna. Pada *software* ini dapat mengatur gambar dengan mudah. Hanya dengan menyeleksi data gambar (jika ingin memilih beberapa gambar, dengan menekan tombol *shift*) kemudian atur resolusi gambar yang diinginkan dalam *pixel* atau persentase dan tekan tombol OK. *Software* ini dapat mengubah ukuran dengan cepat dan mudah. Mendukung beberapa format gambar, yaitu JPEG, JPG, CR2 (Canon RAW), NEF (Nikon RAW), DNG (Adobe), GIF, ICO, PNG, PSD (Photoshop), TIFF, ARW, BMP, CRW, MEF, CUT, DCM, DIB, DICOM, EMF, JP2, JPE, MRW, ORF, PBM, PCD, PCX, PEF, PPM, PSP, PSPIMAGE, RAF, RGB, SGI, SR2, SRF, TGA, TIF, WMF. Dan beberapa keluaran format data gambar, yaitu BMP, JPG, GIF, PNG dan TIF. Ada fitur tambahan pada *software* ini, yaitu dapat menggunakan *watermark*

untuk proteksi terhadap data gambar, atau dapat menambahkan *copyright* (hak cipta) dengan format seperti penulisan di html, juga dapat menambahkan transparansi, memanipulasi gambar dengan menjadikannya *sepia*. Atau menambahkan *border*. Informasi EXIF dan XMP juga bisa tetap terjaga. Tampilan utama *Software* VSO Image Resizer dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tampilan *Software* VSO Image Resizer

Cara penggunaan *Software* VSO Image Resizer 2.1:

1. Install VSO Image Resizer dengan cara *doubleclick* VSO Image Resizer *setup* hasil *download*. Tunggu hingga proses instalasi selesai, *software* dapat langsung digunakan tanpa *restart*.
2. Klik *shortcut* VSO Image Resizer untuk menjalankan *software*.

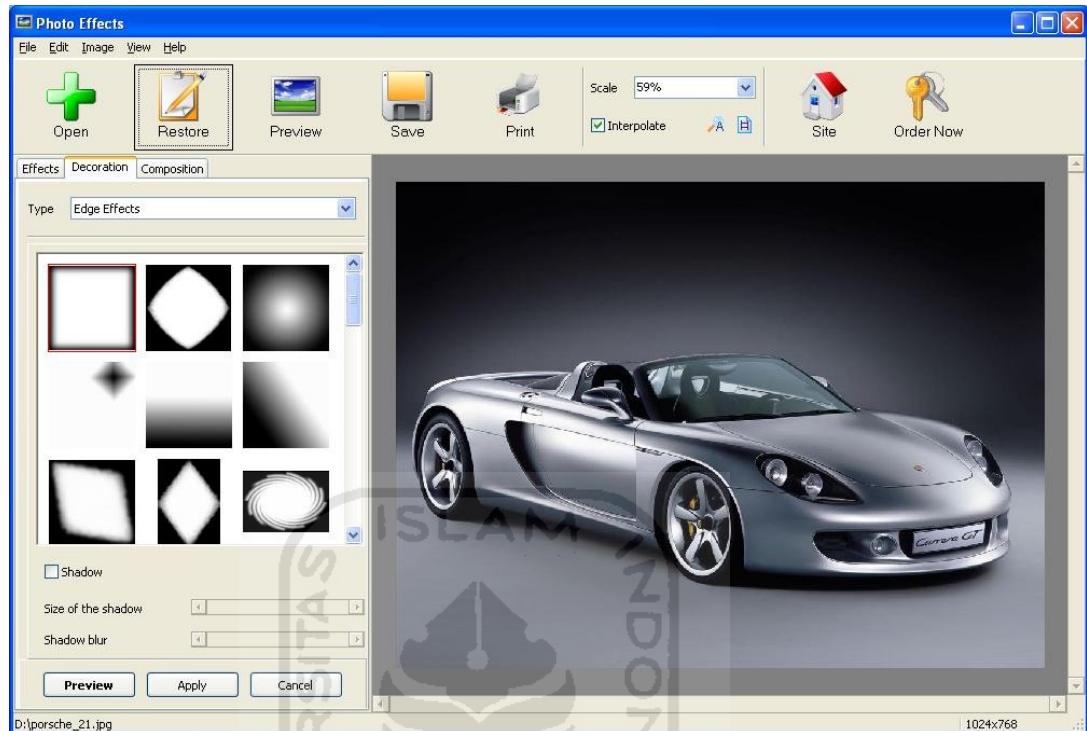
3. Klik tab *Files* lalu cari *file* yang akan dikonversi melalui jendela yang terbuka.
4. Kemudian tekan tombol *Next* untuk menuju ke langkah selanjutnya.
5. Dalam tab *Options* ada beberapa menu untuk mengedit gambar mulai dari opsi *Profile*, yaitu ukuran yang akan digunakan. Misal ingin menggunakan ukuran *Email* (1024x768).
6. Kemudian juga bisa atur sendiri *Width* dan *Height* dengan ukuran *pixel* atau persentase.
7. Kemudian pilih opsi *Mode*, diberi beberapa pilihan, mulai dari *Fit*, *Stretch*, *Center*, *Crop*, dan *Retarget*. Kemudian dibawah opsi tersebut ada *Keep Aspect Ratio*, yaitu menjaga agar bentuk dari *file* tersebut tetap proporsional. Atau bisa *uncheck*, jika tidak mementingkan bentuk proporsionalnya.
8. Pada tab *Generals*, terdapat beberapa opsi mulai dari *Filtering*, untuk mengatur kualitas gambar dari yang kualitas paling baik, yaitu *Cubic* (*slow, best quality*), hingga kualitas yang rendah yaitu *Fast Linear* (*fastest, lowest quality*).
9. Bisa juga mengatur opsi *Policy*, mulai dari *Always Resize*, *Only Enlarge*, maupun *Only Reduce*.
10. Pada Tab *Effect* dapat juga memberi *watermark*, untuk menjaga supaya gambar tidak dibajak. *Invert Color*, untuk mengubah atau menukar warna, *Convert to Grayscale*, untuk mengubah warna menjadi hitam putih (gradasi), Efek Sephia, atau *Add Border*, untuk menambahkan garis/bingkai.
11. Pada menu Tab *Misc*, bisa mengolah informasi metadata EXIF dan XMP (jika diperlukan). Jadi informasi tentang data gambar tetap terjaga.
12. Pada Tab *Publish*, dapat mengatur apakah ingin dikirim melalui *email* atau tidak.
13. Pada Opsi *Actions*, diberi beberapa pilihan mulai dari *create copies*, jika ingin duplikasi gambar, *resize original*, jika ingin mengubah ukuran pada *file* asli, *move*, jika ingin memindah gambar, *compress ZIP*, jika ingin mengompresi

dalam bentuk ZIP, dan *create PDF*, jika ingin membuat gambar tersebut dalam format PDF.

14. Pada opsi *Destination*, dapat memulih lokasi dimana *file* tersebut dapat disimpan.
15. Selesai, dan *file* siap digunakan.

3.5. AMS Photo Effects

AMS Photo Effects adalah *shareware*, yang artinya diberi waktu untuk mencoba program tersebut, dengan waktu yang ditentukan, setelah masa itu habis, harus membayar untuk menggunakan *software* tersebut. AMS Photo Effects diterbitkan oleh AMS Software. Aplikasi ini dapat digunakan untuk memanipulasi gambar, dengan ada beberapa keterbatasan. *Software* ini juga dapat untuk mengubah ukuran atau *crop* gambar, atau mengkonversi gambar ke beberapa format gambar dalam satu waktu. *Software* ini juga dapat memberi *border* (garis tepi), dan bingkai pada gambar. Tentunya dengan *frame*/bingkai yang telah disediakan oleh *software* ini. *A built-in image preview* membantu untuk melihat hasil gambar yang sedang diolah. Efek *fading* dan transparansi membantu untuk mengkreasi beberapa gaya bingkai, dan komposisi untuk gambar. Fitur yang paling bagus dalam *software* ini adalah *on-screen templates* dan *the effect samples*, yang membantu dalam proses membuat *frame*, dan *shading*, menjadi lebih mudah dan cepat. *Software* juga dapat mengubah atau menggerakkan gambar ke beberapa arah, untuk memberi kesan 3D. Tampilan utama *software* AMS Photo Effects dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan *Software AMS Photo Effects*

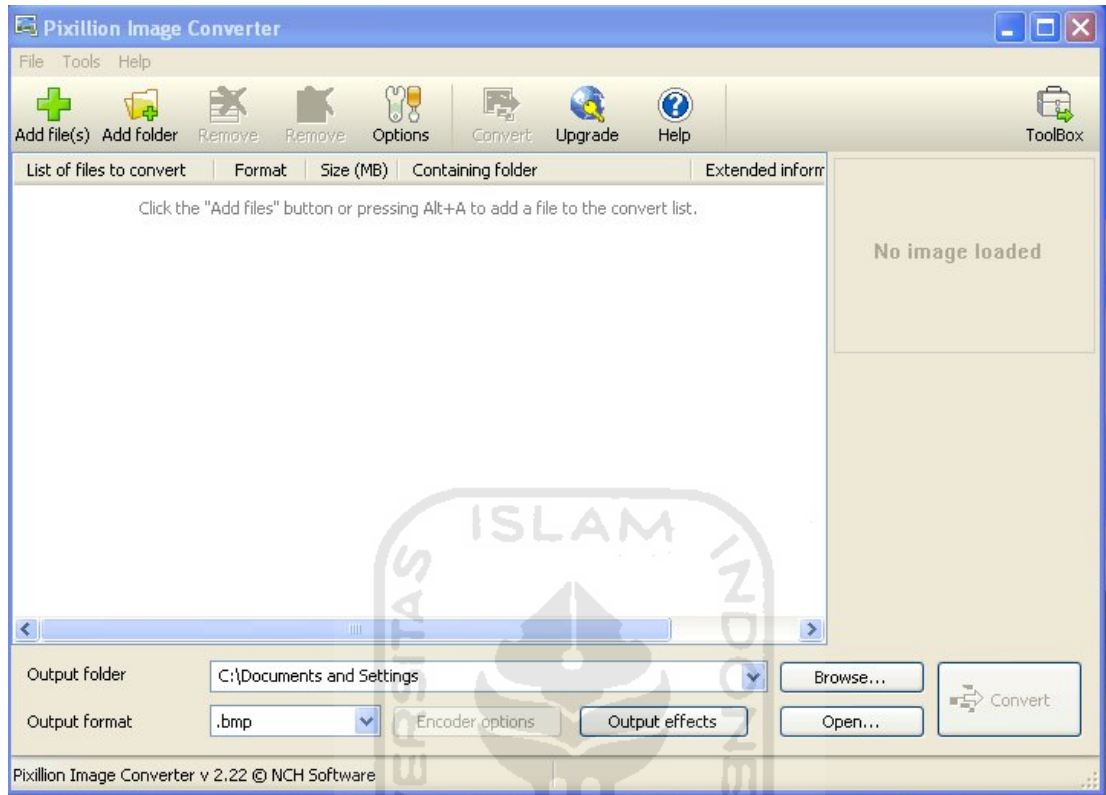
Cara penggunaan *Software AMS Photo Effects*:

1. Install AMS Photo Effects dengan cara *doubleclick* AMS Photo Effects *setup* hasil *download*. Tunggu hingga proses instalasi selesai, *software* dapat langsung digunakan tanpa *restart*.
2. Klik *shortcut* AMS Photo Effects untuk menjalankan *software*.
3. Klik tab *Files* lalu pilih opsi *Open* untuk mencari *file* yang akan dikonversi melalui jendela yang terbuka. Atau dapat dengan menekan tombol hijau di kanan atas.
4. Kemudian tekan tombol *Open* untuk menuju ke langkah selanjutnya.
5. Dalam tab *Effects* ada beberapa opsi untuk mengedit gambar mulai dari opsi *Name*, yang terdiri dari *Name Effects*, *Illumination effects*, *Traditional Effects*, *Distortion Effects*, dan *Artistic Effects*.

6. Setelah itu dapat mengatur posisi efek warna pada opsi *Accent*. Mulai dari *All Images, Only Edges, Only Center, Oval Edges, Oval Center, Top Part, dan Bottom Part*. Jika coba yang *All Images* berarti semua warna pada gambar diubah.
7. Dapat bisa memilih warna pada sampel warna yang disediakan, atau ingin memilih warna lain, dengan menandai/*checkbox* pada *Use Manual Settings*. Di sini dapat pula mengatur warna dan *brightness*(pencahayaan)
8. Setelah selesai pengaturan tadi, dapat melihat hasilnya dengan menekan tombol *Preview* yang ada di menu atas. Kemudian tekan *Save* jika semua sudah selesai pengaturan.
9. Selesai, dan *file* siap digunakan.

3.6. Pixillion Image Converter

Pixillion Image Converter adalah *freeware*. Aplikasi ini diterbitkan oleh NCH Software. Pixillion Image Converter adalah aplikasi untuk konversi gambar. Aplikasi ini dapat mendukung beberapa format gambar yaitu PNG, GIF, JPG, JPEG, BMP, JP2, PSD, TGA, CX, .TIF, TIFF, WBMP, RAS, PGF, PPM, DNG, RAF, CRW, CR2, KDC, DCR, MRW, NEF, ORF, PTX, PEF, ARW, SRF, X3F, ERF, MEF, MOS, RAW, R3D. Dan konversi format gambar berupa PNG, GIF, JPG, BMP, JP2, PSD, TGA, PCX, TIF, WBMP, RAS, PGF, PPM. Kelebihan lain dalam *software* ini yaitu pada saat menginstal *software* ini, dapat juga mengikutsertakan *Express Burn Disc Burning Software*, untuk menyimpan foto atau gambar ke dalam *CD* atau *DVD*. Selain itu juga dapat mengikutsertakan *PhotoStage Slideshow Creator*, aplikasi ini untuk *slideshow* gambar yang sudah dimanipulasi. Tampilan utama *software* Pixillion Image Converter dapat dilihat pada Gambar 3.5.



Gambar 3.5 Tampilan Software Pixillion Image Converter

Cara penggunaan *software* Pixillion Image Converter

1. Install Pixillion Image Converter dengan cara *doubleclick* Pixillion Image Converter *setup* hasil *download*. Tunggu hingga proses instalasi selesai, *software* dapat langsung di gunakan tanpa *restart*.
2. Klik *shortcut* Pixillion Image Converter untuk menjalankan *software*.
3. Dapat kombinasi atau mengikutsertakan program lainnya seperti *Express Burn Disc Burning Software*, dan atau *PhotoStage Slideshow Creator*.
4. Klik tab *Files* lalu pilih opsi *Add Images* cari *file* yang akan dikonversi melalui jendela yang terbuka. Atau dapat dengan menekan tombol hijau di kanan atas.

5. Kemudian tekan tombol *Edit Images* untuk menuju ke langkah selanjutnya. Dan akan muncul jendela baru yaitu *Photo Pad Editor*.
6. Dalam tab *Home* ada beberapa opsi untuk mengedit gambar yaitu opsi *Crop*, dan *Resize*.
7. Dalam tab *Insert* ada beberapa opsi untuk menambahkan gambar, efek *mosaic* dan tulisan.
8. Dalam tab *Color* dapat mengatur warna dan pencahayaan, mulai dari *Brightness*, *Contrast*, *Exposure*, *Auto Levels*, *Saturation*, *Hue*, *Tint*, *Temperature*, *Negative*, *Sepia*, dan *Black and White*.
9. Dalam tab *Effects* dapat mengatur efek-efek lain, mulai dari *Blur*, *Sharpen*, *Local Focus*, *Red-Eye*, dan *Blemish*.
10. Setelah selesai bisa klik *Save*
11. Selesai dan *file* siap digunakan.
12. Tunggu hingga proses selesai.

Perbedaan antara konversi dengan kompresi:

Konversi adalah suatu sistem atau perangkat yang berfungsi mengubah sesuatu kedalam lingkungan lain, misalnya dari digital ke analog.

Kompresi adalah pemampatan ukuran data dengan suatu cara tertentu, sehingga menghemat media yang digunakan.

3.7. Uji Coba

3.7.1. Format .jpeg konversi ke format .png

Hasil Ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image

Converter dari format .jpeg menjadi format .png tetapi pada saat melakukan *convert* dari format .jpeg menjadi format .png tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .png

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.jpeg	.png	.png	.png	.png	.png
<i>File Size</i>	27KB	215KB	274KB	302KB	275KB	271KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	24	32	24	32	24	24
Dimension (pixel)	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336
Ratio	1 : 1	1 : 8	1 : 10	1 : 11	1 : 10	1 : 10

3.7.2. Format .jpeg konversi ke format .bmp

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .jpeg menjadi format .bmp tetapi pada saat melakukan *convert* dari format .jpeg menjadi format .bmp tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .bmp

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.jpeg	.bmp	.bmp	.bmp	.bmp	.bmp
<i>File Size</i>	27KB	588KB	441KB	588KB	588KB	441KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada

Bit Depth	24	32	24	32	24	24
Dimension (pixel)	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336
Ratio	1 : 1	1 : 21,7	1 : 16,3	1 : 21,7	1 : 21,7	1 : 16,3

3.7.3. Format .jpeg konversi ke format .gif

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .jpeg menjadi format .gif tetapi pada saat melakukan *convert* dari format .jpeg menjadi format .gif tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.4.

Tabel 3.4 Perbandingan konversi gambar ukuran kecil dari .jpeg ke .gif

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.jpeg	.gif	.gif	.gif	.gif	.gif
<i>File Size</i>	27KB	69.0KB	88.4KB	67.0KB	90.5KB	60.7KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	24	8	8	8	8(Auto)	8
Dimension (pixel)	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336	448 x 336
Ratio	1 : 1	1 : 2,5	1 : 3,2	1 : 2,5	1 : 3,3	1 : 2,2

3.7.4. Format .png konversi ke format .jpeg

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .png menjadi format .jpeg tetapi pada saat melakukan *convert*

dari format .png menjadi format .jpeg tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.5.

Tabel 3.5 Perbandingan konversi gambar ukuran sedang dari .png ke .jpeg

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.png	.jpeg	.jpeg	.jpeg	.jpeg	.jpeg
<i>File Size</i>	774KB	78.3KB	81.0KB	78.2KB	254KB (image quality 100%)	248KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	24	24	24	24	24
Dots per inch(dpi)		96	300	300	300	300
Dimension (pixel)	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472
Ratio	1 : 1	9,8 : 1	9,5 : 1	9,8 : 1	3 : 1	3,1 : 1

3.7.5. Format .png konversi ke format .gif

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .png menjadi format .gif tetapi pada saat melakukan *convert* dari format .png menjadi format .gif tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.6.

Tabel 3.6 Perbandingan konversi gambar ukuran sedang dari .png ke .gif

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.png	.gif	.gif	.gif	.gif	.gif
<i>File Size</i>	774KB	135KB	207KB	153KB	216KB (auto)	153KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	8	8	8	8	8
Dimension (pixel)	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472
Ratio	1 : 1	1 : 1,9	1 : 1,4	1 : 1,9	1 : 1,9	1 : 1,9

3.7.6. Format .png konversi ke format .bmp

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .png menjadi format .bmp tetapi pada saat melakukan *convert* dari format .png menjadi format .bmp tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.7.

Tabel 3.7 Perbandingan konversi gambar ukuran sedang dari .png ke .bmp

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.png	.bmp	.bmp	.bmp	.bmp	.bmp
<i>File Size</i>	774KB	1.48MB	1.11MB	1.48MB	1.48MB	1.48MB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	32	24	32	32	32
Dimension (pixel)	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472	826 x 472
Ratio	1 : 1	1 : 1,9	1 : 1,4	1 : 1,9	1 : 1,9	1 : 1,9

3.7.7. Format .bmp konversi ke format .jpeg

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .bmp menjadi format .jpeg tetapi pada saat melakukan *convert* dari format .bmp menjadi format .jpeg tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.8.

Tabel 3.8 Perbandingan konversi gambar ukuran besar dari .bmp ke .jpeg

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.bmp	.jpeg	.jpeg	.jpeg	.jpeg	.jpeg
<i>File Size</i>	5.14MB	338KB	338KB	338KB	827KB	819KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	24	24	24	24	24
Dots per inch(pixel)	-	96	300	300	300	96
Dimension (pixel)	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382
Field Ratio	1 : 1	15,2 : 1	15,2 : 1	15,2 : 1	6,2 : 1	6,2 : 1

3.7.8. Format .bmp konversi ke format .png

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .bmp menjadi format .png tetapi pada saat melakukan *convert* dari format .bmp menjadi format .png tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.9.

Tabel 3.9 Perbandingan konversi gambar ukuran besar dari .bmp ke .png

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.bmp	.png	.png	.png	.png	.png
<i>File Size</i>	5.14MB	748KB	782KB	914KB	792KB	914KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	32	24	32	24	32
Dimension (pixel)	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382
Ratio	1 : 1	6,8 : 1	6,5 : 1	5,6 : 1	6,4 : 1	5,6 : 1

3.7.9. Format .bmp konversi ke format .gif

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .bmp menjadi format .gif tetapi pada saat melakukan *convert* dari format .bmp menjadi format .gif tidak memasukkan nilai apapun pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.10.

Tabel 3.10 Perbandingan konversi gambar ukuran besar dari .bmp ke .gif

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.bmp	.gif	.gif	.gif	.gif	.gif
<i>File Size</i>	5.14MB	64.2 KB	258 KB	492 KB	246 KB	492 KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	8	8	8	8	8
Dimension (pixel)	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382	976 x 1382
Field Ratio	1 : 1	80 : 1	19,9 : 1	10,4 : 1	20,8 : 1	10,4 : 1

3.7.10. Format .jpeg konversi ke format .png

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .jpeg menjadi format .png dengan mengubah ukuran gambar pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.11.

Tabel 3.11 Perbandingan konversi gambar dari .jpeg ke .png dengan pengaturan size

	<i>File Asli</i>	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.jpeg	.png	.png	.png	.png	.png
<i>File Size</i>	27KB	66.8 KB	86.7 KB	96 KB	78.8 KB	96 KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	24	32	24	32	24	24
Dimension (pixel)	448 x 336	224 x 168	224 x 168	224 x 168	224 x 168	224 x 168
Field Ratio	1 : 1	1 : 0,4	1 : 0,3	1 : 0,2	1 : 0,34	1 : 0,2

Keterangan PIXResizer

1. Ukuran *file* menjadi 66.8 KB
2. Ukuran bit depth menjadi 32
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* PIXResizer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file size* menjadi 66.8 KB dan kenaikan bit depth menjadi 32.

Keterangan Fast Stone Image Viewer

1. Ukuran *file* menjadi 86.7 KB
2. Ukuran bit depth tetap 24
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file* size menjadi 86.7 KB dan bit depth tetap 24.

Keterangan VSO Image Resize

1. Ukuran *file* menjadi 96 KB
2. Ukuran bit depth menjadi 32
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file* size menjadi 96 KB dan kenaikan bit depth menjadi 32.

Keterangan AMS Photo Effects

1. Ukuran *file* menjadi 96 KB
2. Ukuran bit depth tetap 24
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file* size menjadi 96 KB dan bit depth tetap 24

Keterangan Pixillion Image Converter

1. Ukuran *file* menjadi 78.8 KB
2. Ukuran bit depth tetap 24
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file* size menjadi 96 KB dan kenaikan bit depth menjadi 32.

3.7.11. Format .bmp konversi ke format .gif

Hasil ujicoba menggunakan *software image* kompresi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects, dan Pixillion Image Converter dari format .bmp menjadi format .gif dengan mengubah ukuran gambar pada semua *software* tersebut. Tabel perbandingan dapat dilihat pada Tabel 3.12.

Tabel 3.12 Perbandingan konversi gambar dari .bmp ke .gif dengan pengaturan size

	<i>File</i> Asli	PIXResizer	Fast Stone Image Viewer	VSO Image Resizer	AMS Photo Effects	Pixillion Image Converter
General						
Format	.bmp	.gif	.gif	.gif	.gif	.gif
<i>File</i> Size	5.14MB	26.5 KB	27.5 KB	37.5 KB	27.9 KB	37.5 KB
EXIF Data	Ada	Ada	Ada	Ada	Tidak Ada	Tidak Ada
Bit Depth	32	8	8	8	8	8
Dimension (pixel)	976 x 1382	244 x 346	244 x 346	244 x 346	244 x 346	244 x 346
Field Ratio	1 : 1	198 : 1	191 : 1	140 : 1	188 : 1	140 : 1

Keterangan PIXResizer

1. Ukuran *file* menjadi 26.5 KB
2. Ukuran bit depth menjadi 8
3. Ukuran size berubah menjadi 244 x 346 pixel

Pada hasil uji coba menggunakan *software* PIXResizer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 244 x 346 pixel mengalami kenaikan *file* size menjadi 26.5 KB dan kenaikan bit depth menjadi 8.

Keterangan Fast Stone Image Viewer

1. Ukuran *file* menjadi 27.5 KB
2. Ukuran bit depth tetap 8
3. Ukuran size berubah menjadi 244 x 346 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 244 x 346 pixel mengalami kenaikan *file* size menjadi 27.5 KB dan bit depth tetap 8.

Keterangan VSO Image Resize

1. Ukuran *file* menjadi 37.5 KB
2. Ukuran bit depth menjadi 8
3. Ukuran size berubah menjadi 244 x 346 pixel

Pada hasil uji coba menggunakan *software* VSO Image Resize dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 244 x 346 pixel mengalami kenaikan *file* size menjadi 37.5 KB dan kenaikan bit depth menjadi 8.

Keterangan AMS Photo Effects

1. Ukuran *file* menjadi 96 KB
2. Ukuran bit depth tetap 24
3. Ukuran size berubah menjadi 224 x 168 pixel

Pada hasil uji coba menggunakan *software* Fast Stone Image Viewer dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 224 x 168 pixel mengalami kenaikan *file* size menjadi 96 KB dan bit depth tetap 24.

Keterangan Pixillion Image Converter

1. Ukuran *file* menjadi 37.5 KB
2. Ukuran bit depth tetap 8
3. Ukuran size berubah menjadi 244 x 346 pixel

Pada hasil uji coba menggunakan *software* Pixillion Image Converter dengan mengubah pengaturan size dari 448 x 336 pixel menjadi 244 x 346 pixel mengalami kenaikan *file* size menjadi 37.5 KB dan kenaikan bit depth menjadi 8.

Tabel 3.13 Analisis kelebihan dan kekurangan *software*

Keterangan	PIXResizer	Fast Stone Image Viewer	VSO Image Resize	AMS Photo Effects	Pixillion Image Converter
Kualitas gambar	Terkompresi dengan baik tetapi mengalami penurunan kualitas gambar yang signifikan	Terkompresi dengan baik tetapi tidak mengalami penurunan kualitas gambar yang signifikan	Terkompresi dengan baik tetapi mengalami penurunan kualitas gambar yang signifikan	Terkompresi dengan baik tetapi mengalami penurunan kualitas gambar yang signifikan	Terkompresi dengan baik tetapi mengalami penurunan kualitas gambar yang signifikan
Dukungan terhadap	JPEG, GIF, BMP, PNG dan TIFF	BMP, JPEG, JPEG 2000, <i>animated</i>	JPEG, JPG, CR2 (Canon	BMP, JPEG, GIF, PNG, TIF	PNG, GIF, JPG, JPEG, BMP, JP2,

format <i>file</i>		GIF, PNG, PCX, PSD, EPS, TIFF, WMF, ICO dan TGA, kamera digital (CRW, CR2, NEF, PEF, RAF, MRW, ORF, SRF, ARW, SR2, RW2 dan DNG).	RAW) , NEF(Nikon RAW), DNG (Adobe), GIF, ICO, PNG, PSD (Photoshop) , TIFF, ARW, BMP, CRW, MEF, CUT, DCM, DIB, DICOM, EMF, JP2, JPE, MRW, ORF, PBM, PCD, PCX, PEF, PPM, PSP, PSPIMAGE, RAF, RGB, SGI, SR2, SRF, TGA, TIF, WMF		PSD, TGA, .CX, .TIF, TIFF, WBMP, RAS, PGF, PPM, DNG, RAF, CRW, CR2, KDC, DCR, MRW, NEF, ORF, PTX, PEF, ARW, SRF, X3F, ERF, MEF, MOS, RAW, R3D. Dan konversi format gambar berupa PNG, GIF, JPG, BMP, JP2, PSD, TGA, PCX, TIF, WBMP, RAS, PGF, PPM
Kompresi ukuran <i>file</i> size	Terkompresi dengan baik, ukuran <i>file</i> mengecil	Terkompresi dengan baik, ukuran <i>file</i> mengecil tapi lebih besar dari PIXResizer	Terkompresi dengan baik, ukuran <i>file</i> mengecil tapi lebih besar dari PIXResizer	Terkompresi dengan baik, ukuran <i>file</i> mengecil tapi lebih besar dari PIXResizer	Terkompresi dengan baik, ukuran <i>file</i> mengecil tapi lebih besar dari PIXResizer
Penamaan <i>file</i>	Tidak ada(duplikasi <i>file</i> asli)	Ada fitur penamaan <i>file</i>	Ada fitur penamaan <i>file</i>	Tidak ada(duplikasi <i>file</i> asli)	Tidak ada(duplikasi <i>file</i> asli)

Sistem operasi yang Mendukung	Windows	Windows	Windows	Windows	Windows
Pengaturan aspect ratio	Tersedia dan dapat digunakan dengan baik	Tersedia dan dapat digunakan dengan baik	Tersedia dan dapat digunakan dengan baik	Tersedia dan dapat digunakan dengan baik	Tersedia dan dapat digunakan dengan baik (harus mengunduh dahulu)
Kemudahan pengoperasian <i>software</i> kompresi video	Mudah digunakan	Mudah digunakan	Mudah digunakan	Mudah digunakan	Mudah digunakan
Jenis <i>software</i>	freeware	Freeware	Shareware	Shareware	freeware



DAFTAR PUSTAKA

Converter Available at <http://total.or.id/info.php?kk=converter>

Compression Available at <http://total.or.id/info.php?kk=compression>

Pengertian Istilah Available at

http://muhammadarif.web.id/pdf_file/arief_resolusi_desain.pdf

AMS Photo Effects Software Available at <http://ams-photo-effects.software.informer.com/>

FastStone Image Viewer Software Available at <http://faststone-image-viewer.software.informer.com>

PIXResizer Software Available at <http://pixresizer.software.informer.com/>

VSO Image Resizer Software Available at <http://vso-image-resizer.software.informer.com/>

Pixillion Image Converter Software Available at <http://pixillion-image-converter.software.informer.com/>



BAB IV

PENUTUP

4.1. Kesimpulan

Setelah melakukan studi pustaka dan uji coba terhadap *software* kompresi dan konversi gambar dengan menggunakan PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter. Maka penulis dapat mengambil beberapa kesimpulan sebagai berikut :

- a. Terdapat banyak tipe-tipe *file* gambar, selama ini hanya mengenal beberapa tipe-tipe *file* gambar yaitu : JPEG, GIF, BMP, PNG and TIFF
- b. Dalam melakukan konversi *file* gambar terdapat banyak *software* untuk melakukan kompresi *file* gambar yang freeware dan shareware, diantara *software* yang di gunakan adalah PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter.
- c. Dalam melakukan eksplorasi pada *software* PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter, kompresi terhadap *file* gambar diharapkan tidak mengubah kualitas gambar dengan menentukan perubahan property yang tepat pada saat kompresi. Perubahan yang sangat terlihat adalah pada perubahan kualitas gambar. Peningkatan dan penurunan pada ukuran dan tipe *file* gambar sangat berpengaruh terhadap kualitas gambar.
- d. Sistem operasi yang mendukung pada aplikasi PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter adalah Windows.
- e. PIXResizer merupakan *software* kompresi dan konversi gambar yang paling sedikit mengenal tipe-tipe *file* gambar, sementara *Software* FastStone merupakan *software* yang paling berdayaguna dibanding yang lain, paling banyak terdapat fitur-fitur pendukung.

4.2. Saran

Berdasarkan uji coba yang telah dilakukan penulis, masih ada beberapa kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar hasil uji coba menjadi lebih baik. Oleh karena itu untuk uji coba ke depan disarankan sebagai berikut:

- a. Studi pustaka tidak hanya diambil dari referensi internet dan referensi buku.
- b. Gambar kompresi dan konversi yang dihasilkan oleh 5 *software* tersebut adalah gambar dengan hasil maksimal, namun tidak hanya terpaku pada 5 *software* tersebut. Lima *software* tersebut adalah PIXResizer, FastStone Image Viewer, VSO Image Resizer, AMS Photo Effects dan Pixillion Image Converter.
- c. Media yang digunakan untuk mengambil contoh gambar adalah kamera digital, akan tetapi memungkinkan juga menggunakan gambar hasil *Digital Imaging*.

