



**ANALISIS HISTOGRAM PADA CITRA DIGITAL YANG
DIMAMPATKAN DENGAN METODE KUANTISASI**

SKRIPSI

oleh
Darul Afandi
NIM 111810101041

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**



**ANALISIS HISTOGRAM PADA CITRA DIGITAL YANG
DIMAMPATKAN DENGAN METODE KUANTISASI**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Matematika (S1) dan mencapai gelar Sarjana Sains

oleh

Darul Afandi
NIM 111810101041

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PERSEMBAHAN

Dengan penuh rasa syukur kehadiran Allah SWT yang Maha Pengasih dan Penyayang, serta Shalawat serta Salam kepada Nabi Muhammad SAW, dengan setulus hati skripsi ini saya persembahkan untuk:

- ❖ Ayahanda Suroto dan Ibunda Abidah, yang telah memberikan cinta, kasih sayang, nasehat, dan do'a yang tiada henti serta kerja keras dengan penuh pengorbanan.
- ❖ Saudara kembar saya Bahrul Afandi yang selalu memberikan semangat dan keceriaan dalam menuntut ilmu.
- ❖ Guru-guru sejak taman kanak-kanak sampai perguruan tinggi yang memberikan ilmu yang bermanfaat sebagai bekal dalam menjalani kehidupan.
- ❖ Saudara seangkatan KRAMAT'11 yang telah memberikan cerita serta pengalaman hidup tersendiri.
- ❖ Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMAN 1 Jombang, SMPN 1 Jombang, SDN Puton 1, RA.Thoyib Hidayah sebagai tempat menimba ilmu.

MOTTO

Jangan pernah menyerah kepada siapapun atau terhadap apapun, karena keajaiban terjadi setiap hari.

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan”.

(Terjemahan QS. Al-Insyirah: 5-6)

Jangan lupa bahagia dengan bersyukur.

“Maka nikmat Tuhan mu yang manakah yang kamu dustakan?”

(Terjemahan QS. Ar-Rahman: 77)

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Darul Afandi

NIM : 111810101041

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Analisis Histogram pada Citra Digital yang Dimampatkan dengan Metode Kuantisasi” adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan skripsi ini belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan isinya sesuai dengan skripsi ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenar-benarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Desember 2016
Yang menyatakan,

Darul Afandi
NIM 111810101041

SKRIPSI

**ANALISIS HISTOGRAM PADA CITRA DIGITAL YANG
DIMAMPATKAN DENGAN METODE KUANTISASI**

oleh

Darul Afandi
NIM 111810101041

Pembimbing :

Dosen Pembina Utama : Ahmad Kamsyakawuni, S.Si., M.Kom.

Dosen Pembimbing Anggota : M. Ziaul Arif, S.Si., M.Sc.

PENGESAHAN

Skripsi berjudul “Analisis Histogram pada Citra Digital yang Dimampatkan dengan Metode Kuantisasi” telah diuji dan dishkan pada:

Hari :

Tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,

(Dosen Pembimbing Utama)

Sekretaris,

(Dosen Pembimbing Anggota)

Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP. 197211291998021001

M. Ziaul Arif, S.Si., M.Sc.
NIP. 198501112008121002

Anggota I,

Anggota II,

Kusbudiono, S.Si., M.Si.
NIP. 197704302005011001

Kosala Dwidja Purnomo, S.Si., M.Si.
NIP. 196908281998021001

Mengesahkan

Dekan,

Drs. Sujito, Ph.D.

NIP.196102041987111001

RINGKASAN

Analisis Histogram pada Citra Digital yang Dimampatkan dengan Metode Kuantisasi; Darul Afandi, 111810101041; 2016: 71 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan teknologi saat ini begitu pesat, salah satunya adalah perkembangan teknologi dalam pengolahan citra digital. Semakin berkembangnya teknologi tersebut maka kualitas dari suatu citrapun berkembang pula dari yang dulunya hitam putih/biner kini citra lebih berwarna dan bisa diolah sesuai keinginan.

Citra digital dalam layar ditampilkan dalam bentuk kotak-kotak kecil yang sangat halus dimana didalamnya terdiri dari kumpulan dari warna-warna. Kotak-kotak kecil yang terdapat dalam citra digital disebut dengan *pixel*.. Masing-masing *pixel* pada citra mempunyai nilai kuantitatif antara 0 sampai 255, untuk citra berwarna terdiri dari 3 warna. Tingginya kualitas dari citra yang dihasilkan oleh mesin pengolahan citra juga mempengaruhi besarnya ukuran memori yang ada dalam citra digital tersebut. Oleh karena itu pemampatan citra sangatlah penting untuk efisiensi waktu dalam pengiriman dan penyimpanan citra tersebut. Pemampatan dengan Metode Kuantisasi dilakukan dengan cara memampatkan derajat keabuan citra menjadi 7 bit, 6 bit, 5 bit, 4 bit, 3 bit, 2 bit, dan 1 bit.

Penelitian pemampatan dilakukan pada citra berwarna ataupun citra keabuan. Untuk mengetahui kualitas citra hasil dari pemampatan tersebut digunakan Histogram, nilai MSE, dan Nilai PSNR. Besarnya nilai PSNR dipengaruhi oleh besar kecilnya nilai MSE dimana nilai MSE dipengaruhi oleh besar kecilnya bit pemampatan. Semakin besar nilai MSE pada citra hasil pemampatan tersebut maka nilai dari PSNR akan semakin kecil.

PRAKATA

Puji Syukur kehadiran Allah SWT atas segala limpahan rahmat dan karunia-Nya yang telah diberikan sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Histogram pada Citra Digital yang Dimampatkan dengan Metode Kuantisasi”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan dan kerjasama dari berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Pembimbing Utama serta Dosen Pembimbing Akademik, dan Bapak M. Ziaul Arif, S.Si., M.Sc., selaku Dosen Pembimbing Anggota yang telah meluangkan waktunya untuk memberikan bimbingan, pengarahan bagi penulis;
2. Bapak Kusbudiono, S.Si., M.Si., sebagai dosen penguji utama dan Bapak Kosala Dwidja Purnomo, S.Si., M.Si., sebagai dosen penguji anggota yang telah memberikan kritikan dan saran kepada penulis;
3. Bapak Suroto sekeluarga yang senantiasa memberikan do'a serta dukungan moril maupun materiil selama di Jember;
4. Seluruh dosen serta karyawan Jurusan Matematika
5. Rafi', Rimbi, Jefri, Haki, Yulio, Emil, dan teman-teman KRAMAT'11 yang tidak disebutkan satu-persatu semoga persaudaraan kita tetap terjalin sampai kapanpun

Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat dalam perkembangan ilmu pengetahuan.

Jember, Desember 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN	ii
MOTTO	iii
PERNYATAAN	iv
PEMBIMBING	v
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Citra Digital	4
2.2 Elemen Citra Digital	5
2.2.1 Warna	5
2.2.2 Kontras	5
2.2.3 Kecerahan	6
2.2.4 Kontur	6
2.3 Tipe-tipe Citra Digital	6
2.3.1 Citra Hitam-putih (<i>biner/monochrom</i>)	6
2.3.2 Citra Keabuan (<i>Citra Grayscale</i>)	7
2.3.3 Citra Berwarna (RGB)	8

2.4 Format File Citra	9
2.4.1 Bitmap (.bmp).....	9
2.4.2 JPEG/JPG (.jpg)	9
2.4.3 GIF (.gif).....	10
2.4.4 PNG (.png)	10
2.5 Pemampatan Citra Digital	10
2.5.1 <i>Lossless Compression</i>	10
2.5.2 <i>Lossy Compression</i>	11
2.6 Kriteria Kompresi Citra	11
2.7 Metode Kuantisasi	12
2.8 Software MATLAB dan GUI	13
2.8.1 MATLAB	13
2.8.2 GUI (<i>Graphical User Interface</i>)	15
2.9 Nisbah Citra	18
2.10 Kualitas Kompresi Citra Menggunakan MSE dan PSNR	18
2.11 Histogram	19
Bab 3. METODE PENELITIAN	22
3.1 Data Penelitian	22
3.2 Langkah-langkah Penelitian	22
Bab 4. HASIL DAN PEMBAHASAN	24
4.1 Hasil	24
4.1.1 Kompresi Citra menggunakan Metode Kuantisasi	24
4.1.2 Hasil <i>running</i> Program dan Program <i>Graphic User interface</i> (GUI)	32
4.2 Pembahasan	44
4.2.1 Analisis Pada Citra Hasil Kompresi Metode Kuantisasi.....	44
BAB 5. PENUTUP	47
5.1 Kesimpulan.....	47
5.2 Saran.....	47
DAFTAR PUSTAKA	48
LAMPIRAN	49

DAFTAR GAMBAR

	Halaman
2.1 Citra Digital.....	4
2.2 Contoh Citra Biner	7
2.3 Citra Grayscale	8
2.4 Citra Berwarna	9
2.5 Tampilan <i>software</i> Matlab	14
2.6 Tampilan untuk perintah <i>Guide</i>	16
2.7 Tampilan awal GUI.....	16
2.8 Citra RGB	20
2.9 Histogram <i>Red</i>	20
2.10 Histogram <i>Green</i>	21
2.11 Histogram <i>Blue</i>	21
3.1 Skema langkah-langkah penelitian	23
4.1 Contoh foto <i>tanaman.jpg</i>	24
4.2 (a) Foto asli, (b) foto hasil pemampatan 7 bit	33
4.3 (a) Foto asli, (b) foto hasil pemampatan 6 bit	35
4.4 (a) Foto asli, (b) foto hasil pemampatan 7 bit	37
4.5 (a) Foto asli, (b) foto hasil pemampatan 6 bit	38
4.6 Tampilan GUI program kompresi Metode Kuantisasi.....	41
4.7 Tampilan program setelah proses pemampatan	42
4.8 (a) Histogram <i>Red</i> , (b) Histogram <i>Green</i> , dan (c) Histogram <i>Blue</i> pada citra hasil kompresi <i>koala.jpg</i>	43

DAFTAR TABEL

	Halaman
2.1 Kuantisasi Skala keabuan.....	13
4.1 Tabel pengelompokan derajat keabuan komponen <i>red</i>	26
4.2 Tabel pengelompokan derajat keabuan komponen <i>green</i>	26
4.3 Tabel pengelompokan derajat keabuan komponen <i>blue</i>	27
4.4 Tabel pembagian kelompok komponen <i>red</i>	28
4.5 Tabel pembagian kelompok komponen <i>green</i>	29
4.6 Tabel pembagian kelompok komponen <i>blue</i>	29
4.7 Tabel pengkodean kembali derajat keabuan komponen <i>red</i>	30
4.8 Tabel pengkodean kembali derajat keabuan komponen <i>green</i>	30
4.9 Tabel pengkodean kembali derajat keabuan komponen <i>blue</i>	31
4.10 Tabel MSE dan PSNR pada setiap komponen 7 bit foto <i>laut.jpg</i>	35
4.11 Tabel MSE dan PSNR pada setiap komponen 6 bit foto <i>laut.jpg</i>	36
4.12 Tabel MSE dan PSNR pemampatan 7 bit dan 6 bit citra keabuan.....	39
4.13 Nisbah citra.....	40
4.14 Tabel nilai rata-rata hasil pemampatan citra berwarna.....	45
4.15 Tabel nilai rata-rata hasil pemampatan citra keabuan.....	45

BAB 1.PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini begitu pesat, salah satunya adalah perkembangan teknologi dalam pengolahan citra digital. Semakin berkembangnya teknologi tersebut maka kualitas dari suatu citrapun berkembang pula dari yang dulunya hitam putih/biner kini citra lebih berwarna dan bisa diolah sesuai keinginan. Banyak orang yang memanfaatkan berkembangnya teknologi pengolahan citra digital sebagai salah satu cara untuk mendokumentasikan kejadian disekitar mereka, salah satunya adalah seorang fotografer. Seorang fotografer tentunya akan memanfaatkan perkembangan dari teknologi pengolahan citra digital demi mendapatkan gambar atau foto yang memiliki kualitas yang tinggi.

Citra digital dalam layar ditampilkan dalam bentuk kotak-kotak kecil yang sangat halus dimana didalamnya terdiri dari kumpulan dari warna-warna. Kotak - kotak kecil yang terdapat dalam citra digital disebut dengan *pixel* yang setiap *pixel* dalam citra memiliki nilai warna yang disebut dengan derajat keabuan. Masing-masing *pixel* pada citra mempunyai nilai kuantitatif antara 0 sampai 255, untuk citra berwarna terdiri dari 3 warna.

Tingginya kualitas dari citra yang dihasilkan oleh mesin pengolahan citra juga mempengaruhi besarnya ukuran memori yang ada dalam citra digital tersebut. Semakin tinggi kualitas dari citra tersebut, maka semakin besar pula ukuran file citra tersebut. File citra tersebut akan membutuhkan banyak memori apabila citra tersebut disimpan dalam mesin penyimpanan seperti *hard disk* ataupun alat penyimpan data lainnya. Begitupun untuk mentransfer atau mengirim citra tersebut, akan membutuhkan waktu yang sedikit lebih lama dalam prosesnya belum lagi dengan sinyal internet yang tidak selalu lancar. Oleh karena itu kompresi gambar atau citra sangatlah penting untuk efisiensi waktu dalam pengiriman dan penyimpanan citra tersebut.

Penggunaan kompresi citra pertama kali dilakukan pada 1921 untuk gambar koran yang dikirim melalui kabel dari New York ke London. Pengiriman data tersebut mengurangi waktu dari satu minggu menjadi kurang dari 3 jam dengan menyeberangi samudra atlantik. Selanjutnya kompresi citra digital dengan menggunakan teknik komputer dimulai tahun 1964, yaitu sebuah citra bulan yang berasal dari pesawat Jet Propulsion yang ditransmisikan oleh Pesawat Ranger-7 seperti yang dikemukakan oleh Kwatno (2011).

Menurut Salomon (2007), pemampatan ialah proses perubahan sekumpulan data menjadi suatu bentuk kode untuk mendapatkan ukuran yang lebih kecil. Ada banyak metode yang bisa digunakan dalam pemampatan data atau citra digital, salah satunya adalah metode kuantisasi. Metode kuantisasi bekerja dengan mengurangi jumlah intensitas warna, sehingga jumlah *bit* yang digunakan untuk mempresentasikan citra menjadi berkurang, sehingga ukuran file menjadi lebih kecil. Metode kuantisasi termasuk kedalam *lossy* compression, dimana ada informasi data yang hilang sehingga sulit untuk didekompresi atau dikembalikan seperti semula.

Penelitian tentang pemampatan citra dengan pendekatan kuantisasi pernah dilakukan oleh Kwatno (2011), yang menggunakan citra biner atau hitam-putih sebagai obyek penelitiannya. Dalam penelitiannya didapatkan kapasitas rata-rata ukuran kompresi foto berkurang menjadi 57,63 % dari foto semula. Ada juga penelitian yang dilakukan oleh Kurniawan (2015), yang berjudul Perancangan Aplikasi Kompresi Citra Berwarna Menggunakan Metode Kuantisasi. Dari penelitian tersebut didapatkan hasil yaitu kompresi yang dilakukan dengan cara menurunkan tingkat intensitas RGB secara seragam, mengakibatkan bit berkurang dan file citra pun akan semakin kecil. Dari dua penelitian tersebut didapatkan hasil yang sama yaitu ukuran dari file citra tersebut menjadi lebih kecil namun tidak dilakukan uji kualitas pada citra hasil yang telah dikompresi menggunakan Metode Kuantisasi. Oleh sebab itu penelitian kali ini akan menganalisis apakah terjadi perubahan kualitas pada citra yang dimampatkan dengan metode kuantisasi menggunakan histogram, MSE dan PSNR.

1.2 Perumusan Masalah

Permasalahan yang akan dikaji dalam penelitian kali ini adalah sebagai berikut:

- a. Bagaimana memampatkan citra menggunakan metode kuantisasi;
- b. Berapa besar hasil pemampatan yang didapatkan dalam pemampatan tersebut;
- c. Berapa nilai kualitas citra hasil pemampatan metode kuantisasi.

1.3 Tujuan Penelitian

Tujuan yang diinginkan dalam penelitian ini adalah

- a. untuk mengetahui penggunaan metode kuantisasi dalam proses pemampatan citra;
- b. untuk mengetahui berapa besar hasil pemampatan tersebut;
- c. untuk mengetahui nilai kualitas pada citra setelah dimampatkan.

1.4 Manfaat penelitian

Manfaat dari penelitian mengenai analisa kompresi citra digital menggunakan metode kuantisasi ini adalah sebagai berikut:

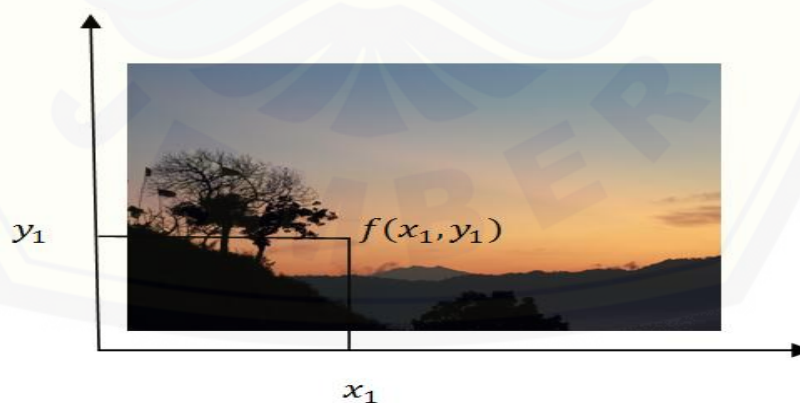
- a. dapat meminimalkan ukuran dari file foto atau citra sehingga dapat menghemat ruang penyimpanan pada media penyimpanan;
- b. mempercepat waktu transfer dalam proses pentransferan citra digital tersebut;
- c. serta mengetahui terjadinya perubahan derajat keabuan dan kualitas pada citra setelah dimampatkan dengan metode kuantisasi.

BAB 2. TINJAUAN PUSTAKA

2.1 Citra Digital

Sebuah citra digital adalah kesatuan dari sekumpulan beberapa kotak-kotak persegi dari sebuah titik yang diatur dalam m baris dan n kolom. Dimana $m \times n$ mempresentasikan nilai dari resolusi gambar, dan titik yang ada dalam gambar disebut sebagai *pixel* (Salomon, 2007). Secara umum citra dibagi menjadi dua macam, yaitu citra kontinu dan diskrit. Citra kontinu adalah citra yang dihasilkan dari sistem optik yang menerima sinyal analog, misalnya mata manusia dan kamera analog. Sedangkan citra diskrit adalah citra yang dihasilkan melalui proses digitalisasi terhadap citra kontinu dan direpresentasikan sebagai sebuah matriks yang masing-masing elemennya mempresentasikan nilai identitas.

Citra digital dapat dinyatakan dalam matriks 2 dimensi $f(x,y)$ dimana x dan y merupakan koordinat *pixel* dalam matriks dan f merupakan derajat intensitas *pixel* tersebut. Nilai $f(x,y)$ dari suatu citra digital berada pada selang keabuan $[0,G-1]$. Skala keabuan 0 biasanya digunakan untuk menyatakan intensitas hitam, sementara intensitas bernilai $G-1$ menyatakan warna putih (Arymurthy dan Setiawan, 1992).



Gambar 2.1 Citra Digital

Jika kita perhatikan dengan seksama atau citra tersebut kita perbesar maka akan nampak titik-titik kecil berbentuk segi empat yang menyusun citra tersebut, titik-titik berbentuk persegi empat itulah yang disebut dengan *pixel*. *Pixel* adalah

bagian terkecil dari sebuah citra. Pada kenyataannya, untuk menggambarkan satu titik pada citra dibutuhkan 2 *pixel*. Satu *pixel* digunakan untuk mempresentasikan terang gelap, sedangkan satu *pixel* lain digunakan untuk mempresentasikan warna pada titik yang sama (Lestari *et al*, 2013). Dalam sebuah citra, jumlah satuan panjang dari *pixel* tersebut akan menentukan resolusi dari sebuah citra, semakin banyak *pixel* yang mewakili dalam citra maka semakin tinggi nilai resolusinya serta citra yang dihasilkanpun semakin halus.

2.2 Elemen Citra Digital

Citra digital tersusun atas beberapa elemen dasar, dimana elemen-elemen tersebut perlu diperhatikan dalam pengkompresan citra. Elemen-elemen tersebut diantaranya adalah sebagai berikut:

2.2.1 Warna

Warna merupakan persepsi yang dirasakan oleh mata manusia terhadap panjang gelombang cahaya yang dipantulkan oleh obyek. Jumlah warna berbeda yang dapat ditampilkan dalam setiap citra ditentukan oleh banyaknya bit pada setiap *pixel*. Ukuran kedalaman warna pada citra dalam komputer terbagi menjadi tiga, yaitu: 8, 16, dan 24 bit untuk setiap *pixel*. Sebuah citra 24 bit menyediakan resolusi warna yang optimal, dengan menggunakan 8 bit untuk setiap komponen warna sehingga terdapat 256 level warna merah, hijau, dan biru (Wijaya dan Prijono, 2007).

2.2.2 Kontras

Kontras menyatakan sebaran terang dan gelap di dalam sebuah citra. Citra dengan kontras yang baik, komposisi dari terang dan gelapnya tersebar secara merata dan citra dengan kontras yang rendah biasanya sebagian besar komposisinya adalah terang atau sebagian besar gelap. Citra dengan kontras yang baik memperlihatkan jangkauan nilai keabuan yang lebar tanpa ada nilai keabuan yang mendominasi.

Citra dengan kontras tinggi memiliki nilai jangkauan nilai keabuan yang lebar, tetapi terdapat area yang lebar yang didominasi oleh warna gelap dan area yang lebar yang didominasi oleh warna terang (Munir, 2004). Citra dengan

kontras rendah dapat diperbaiki kualitasnya dengan operasi peregangan kontras. Melalui operasi ini, nilai-nilai keabuan pixel akan merentang dari 0 sampai 255 (pada citra 8-bit).

2.2.3 Kecerahan

Kecerahan adalah intensitas cahaya yang dipancarkan oleh *pixel* dari sebuah citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah *pixel* didalam citra adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia dapat menyesuaikan dengan tingkat kecerahan mulai dari yang terendah sampai yang paling tinggi. Kecerahan gambar dapat diperbaiki dengan menambahkan atau mengurangi sebuah konstanta kepada atau dari setiap *pixel* didalam citra. Nilai *pixel* hasil pengubahan mungkin lebih kecil atau sama dengan derajat keabuan minimum (0) atau lebih besar sama dengan derajat keabuan maksimum. Karena itu, *pixel* tersebut perlu dilakukan clipping ke nilai keabuan minimum atau ke nilai keabuan maximum (Munir, 2004)

2.2.4 Kontur

Kontur merupakan salah satu elemen dari citra yang ditimbulkan oleh perubahan intensitas pada *pixel-pixel* yang bertetangga. Karena adanya perubahan intensitas inilah kita dapat mendeteksi tepi dari sebuah citra.

2.3 Tipe-tipe Citra Digital

Sutoyo *et al.* (2009), menyatakan bahwa dalam citra digital yang disimpan dalam sebuah komputer dibedakan menjadi tiga, yaitu: citra hitam-putih, citra keabuan, dan citra berwarna.

2.3.1 Citra Hitam-putih (*Biner/monochrom*)

Dalam sebuah citra *Biner/monochrom* hanya terdapat dua warna saja, yaitu warna hitam dan warna putih. Dalam bentuk matriksnya, citra biner hanya memiliki dua elemen nilai kemungkinan, yaitu 0 untuk warna hitam dan 1 untuk warna putih.



Gambar 2.2 Contoh Citra Biner

Citra biner dapat digambarkan dengan palet warna, misalkan suatu citra memiliki derajat keabuan (A) 256, maka nilai tengahnya adalah 128, maka untuk mengubah citra tersebut menjadi citra biner adalah dengan cara sebagai berikut: jika $A < 128$ maka $A=0$ yang mempresentasikan warna hitam, jika tidak $A=1$ dan jika $A > 128$ maka $A=1$ yang mempresentasikan warna putih.

2.3.2 Citra Keabuan (Citra *Grayscale*)

Nilai keabuan dari sebuah citra digital dinyatakan bergerak dari hitam ke putih. Semisal terdapat sebuah citra dengan derajat keabuan dari 0 hingga 255, dengan 0 menyatakan hitam dan 255 menyatakan putih. Nilai antara 0 sampai 255 menyatakan nilai keabuan dari sebuah citra yang terletak diantara warna hitam dan putih. Contoh dari citra *grayscale* dapat dilihat pada Gambar (2.3), dimana masih dapat dibedakan antara daerah pada citra yang memiliki warna hitam pekat, agak hitam, agak putih sampai ke warna putih.



Gambar 2.3 Citra *grayscale*

2.3.3 Citra Berwarna (RGB)

Citra berwarna atau citra RGB ini biasa disebut juga dengan citra spektral atau *truecolor*. Setiap *pixel* pada citra warna merupakan kombinasi dari tiga warna dasar yaitu, *red*, *green*, dan *blue* (RGB) (Ginting, 2010). Dalam sebuah citra RGB 24-bit, nilai RGB masing-masing berkisar antara 0-255. Sebagai contoh representasi warna pada sebuah *pixel* antara lain, RGB [255,0,0] menghasilkan warna merah, RGB [0,255,0] menghasilkan warna biru, dan RGB [0,0,255] menghasilkan warna hijau (Kaur dan Kaur, 2013). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Dengan demikian setiap *pixel* mempunyai kombinasi warna sebanyak $2^8 \cdot 2^8 \cdot 2^8 = 16.777.216$. Itulah sebabnya format ini dinamakan *truecolor* karena mempunyai jumlah warna yang cukup banyak.

Penyimpanan citra *truecolor* didalam memori berbeda dengan citra grayscale. Setiap *pixel* dari citra grayscale 256 gradasi warna diwakili oleh 1 *byte*, sedangkan 1 *pixel* citra *true color* diwakili oleh 3 *byte*, dimana masing-masing *byte* mewakili warna merah (*red*), hijau (*green*), dan biru (*blue*) (Munir, 2002). Berikut contoh citra berwarna:



Gambar 2.4 Citra Berwarna

2.4 Format File Citra

Terdapat beberapa tipe *file* citra yang biasa digunakan, antara lain:

2.4.1 Bitmap (.*bmp*)

Bitmap merupakan format citra digital yang umum digunakan dan merupakan format *standart windows*. Tipe *.bmp* rata-rata memiliki ukuran yang cukup besar. Tipe *.bmp* menyimpan data citra digital tanpa melakukan pemrosesan terhadap citra tersebut. Dalam (*web*) internet, tipe bitmap sangat jarang dijumpai karena ukuran *file* yang relatif besar. Ada beberapa *software* yang berbasis bitmap diantaranya adalah *Adobe Photoshop*, *Paint*, dan *Corel Photo Paint*.

2.4.2 JPEG/JPG (.*jpeg* / *.jpg*)

Format file *Joint Photographic Exprest Group* (JPEG) atau biasa disingkat JPG terkenal karena ukurannya yang mini dibanding dengan format-format citra lainnya. Mampu menayangkan warna dengan kedalaman 24-bit *true color*. JPG mendukung warna RGB, CMYK (*Cyan*, *Magenta*, *Yellow*, *Black*) dan Grayscale (Ginting, 2010). *File* JPG cocok digunakan untuk gambar yang memiliki banyak warna, misalnya foto wajah dan pemandangan dan tidak cocok digunakan untuk gambar yang hanya memiliki sedikit warna seperti kartun atau komik.

2.4.3 GIF (.gif)

GIF adalah singkatan *Graphics Interchange Format* yang merupakan format gambar yang umum digunakan di internet. Tipe GIF merupakan salah satu format citra digital yang mendukung kompresi lossless, yaitu kompresi yang tidak menghilangkan informasi citra digital. Format ini tidak mengurangi kualitas citra digital dibandingkan dengan aslinya, namun kekurangan terbesarnya adalah terbatasnya kedalaman warna yang didukung yaitu hanya sampai 256 warna.

2.4.4 PNG (.png)

Format file *Portable Network Graphic* (PNG) PNG hampir sama dengan format GIF, bahkan format PNG ini didesain untuk menggantikan format GIF, dengan beberapa peningkatan. Format PNG menggunakan kompresi lossless untuk mengkompresi citra digital.

2.5 Pemampatan Citra Digital

Pemampatan citra berwarna (*color image compression*) adalah proses mengurangi atau menghilangkan data pada citra berwarna. Sifat kompresi pada citra dapat dibedakan menjadi dua tipe, yaitu *lossless* dan *lossy*. *Lossless* adalah pemampatan citra yang bersifat tanpa merugi, sedangkan *lossy* adalah pemampatan citra yang bersifat merugi (Gonzales, 2002).

2.5.1 Lossless compression

Lossless compression merupakan pemampatan citra dimana data hasil kompresi dapat didekompresi lagi dan hasilnya tepat sama seperti data sebelum proses kompresi. Namun rasio kompresi citra metode ini sangat rendah. Banyak aplikasi yang membutuhkan metode kompresi tanpa cacat, seperti aplikasi radiografi, kompresi citra hasil diagnosa medis atau gambar satelit, dimana kehilangan gambar sekecil apapun akan menyebabkan hasil yang tidak diharapkan. Contoh dari metode ini adalah *Run Length Encoding* (RLE), *Entropy Encoding* (Huffman), dan *Adaptive Dictionary Based* (LWZ) (Kurniawan, 2015).

2.5.2 Lossy Compression

Lossy compression merupakan pemampatan citra dimana hasil dekompresi citra yang terkompresi tidak sama dengan citra aslinya karena ada informasi citra yang hilang, tetapi masih bisa ditolerir oleh persepsi mata. Mata tidak dapat membedakan perubahan kecil pada citra tersebut. Metode ini menghasilkan rasio kompresi yang lebih tinggi dibanding metode *Lossless*. Contoh dari metode ini adalah *Color Reduction*, *chroma subsampling*, dan *transform coding* seperti Transformasi wavelet (Sutoyo *et al.*, 2009). Untuk menyederhanakan perhitungan pada pemampatan citra berwarna biasanya diubah menjadi *grayscale* (l) dengan mengambil rata-rata dari r (*layer red*), g (*layer green*), dan b (*layer blue*), sehingga dapat dituliskan :

$$l = \frac{r+g+b}{3} \quad (2.1)$$

2.6 Kriteria Kompresi Citra

Sutoyo *et al.* (2009), menyatakan terdapat empat kriteria yang digunakan untuk mengukur suatu proses pemampatan citra, yaitu:

a. Waktu kompresi dan dekompresi

Proses kompresi adalah proses mengkodekan citra sehingga diperoleh citra dengan representase kebutuhan memori yang minimum. Citra terkompresi tersimpan dalam bentuk tertentu, misalnya JPEG (*Joint Photographic Expert Group*). Sedangkan proses dekompresi adalah proses untuk menguraikan citra yang dimampatkan untuk dikembalikan lagi menjadi citra semula. Algoritma pemampatan yang baik adalah algoritma yang membutuhkan waktu untuk kompresi atau dekompresi paling cepat.

b. Kebutuhan Memori

Metode kompresi yang baik adalah metode yang mampu memampatkan file citra menjadi file yang berukuran paling minimal. Algoritma pemampatan yang baik akan menghasilkan memori yang dibutuhkan untuk menyimpan hasil kompresi yang berkurang secara berarti, biasanya semakin besar presentase

pemampatan, semakin kecil kebutuhan memori yang diperlukan sehingga kualitas citra semakin berkurang. Dan sebaliknya, semakin kecil presentase pemampatan pada citra, maka hasil dari pemampatan lebih bagus.

c. Kualitas pemampatan

Metode kompresi yang baik adalah yang mampu mengembalikan citra ke bentuk semula tanpa kehilangan informasi apapun. Walaupun ada informasi yang hilang akibat pemampatan, sebaiknya hal itu ditekan seminimal mungkin. Biasanya semakin berkualitas hasil pemampatan, semakin besar memori yang dibutuhkan. Sebaliknya semakin jelek kualitas citra hasil pemampatan, semakin kecil kebutuhan memori yang harus disediakan.

d. Format keluaran

Format citra hasil pemampatan yang baik adalah yang cocok dengan kebutuhan pengiriman dan penyimpanan data.

2.7 Metode Kuantisasi

Metode kuantisasi merupakan salah satu dari *lossy compression* yang prinsip kerjanya menghilangkan beberapa bagian dari sebuah citra yang dianggap tidak begitu penting, metode kuantisasi bekerja dengan mengurangi derajat keabuan sehingga jumlah bit yang dibutuhkan untuk merepresentasikan citra menjadi berkurang (Munir, 2002). Oleh karena jumlah bit berkurang maka ukuran *file* menjadi lebih kecil. Dengan berkurangnya intensitas warna pada suatu citra maka ada informasi yang hilang dari citra tersebut. Citra yang dikompresi dengan metode kuantisasi akan sulit untuk didekompresi kembali karena ada informasi yang hilang. Proses kuantisasi adalah proses membagi-bagi warna kontinu menjadi G buah warna. Biasanya G diambil dari perpangkatan 2.

$$G = 2^m \quad (2.2)$$

dimana :

G = derajat keabuan

m = bilangan bulat positif

Lebih jelasnya akan ditunjukkan pada Tabel 2.1 sebagai berikut.

Tabel 2.1 Kuantisasi skala keabuan

Skala Keabuan	Rentang Nilai Keabuan	Pixel Depth
2^1 (2 nilai)	0,1	1 bit
2^2 (4 nilai)	0 sampai 3	2 bit
2^3 (8 nilai)	0 sampai 7	3 bit
2^4 (16 nilai)	0 sampai 15	4 bit
2^5 (32 nilai)	0 sampai 31	5 bit
2^6 (64 nilai)	0 sampai 63	6 bit
2^7 (128 nilai)	0 sampai 127	7 bit
2^8 (256 nilai)	0 sampai 255	8 bit

Semakin besar kedalaman *pixel* yang dipakai untuk merepresentasikan suatu gambar, semakin bagus gambar digital yang akan dihasilkan karena jumlah warna yang dapat direpresentasikan semakin banyak. Pada umumnya, gambar *grayscale* memiliki 256 level dan membutuhkan kapasitas 8 bit / 1 *byte* untuk tiap *pixel* (Sutoyo *et al*, 2009). Berikut adalah algoritma metode kuantisasi:

- Buat histogram citra semula (citra yang akan dimampatkan).
- Tentukan n buah kelompok didalam histogram sedemikian hingga setiap kelompok mempunyai $\frac{P}{n}$ buah *pixel*. P adalah jumlah kelompok *pixel* dari citra tersebut, n adalah banyaknya rentang bit kompresi.
- Nyatakan setiap kelompok dengan derajat keabuan 0 sampai $n-1$. Setiap *pixel* didalam kelompok dikodekan kembali dengan nilai derajat keabuan yang baru.

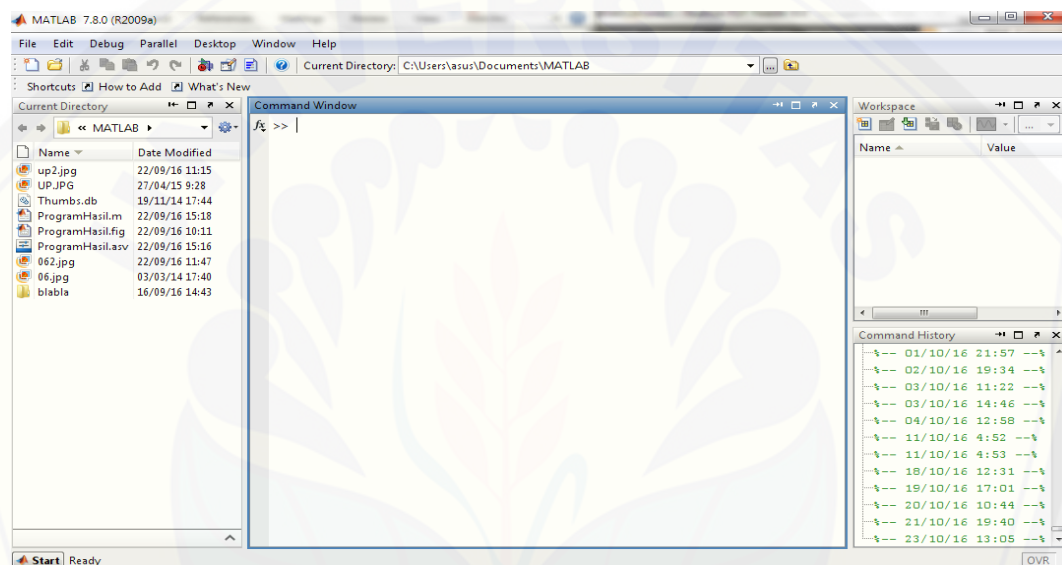
2.8 Software MATLAB dan GUI

2.8.1 MATLAB

MATLAB merupakan suatu program komputer yang bisa membantu memecahkan berbagai masalah matematis yang kerap kita temui dalam bidang teknis. Kita bisa memanfaatkan kemampuan MATLAB untuk menemukan solusi dari berbagai masalah numerik secara cepat, mulai hal yang paling dasar, misalkan sistem 2 persamaan dengan 2 variabel hingga yang kompleks, seperti

mencari akar-akar polinomial, interpolasi dari sejumlah data, perhitungan dengan matriks, pengolahan sinyal, dan metoda numerik. Salah satu aspek yang sangat berguna dari MATLAB ialah kemampuannya untuk menggambarkan berbagai jenis grafik, sehingga kita bisa memvisualisasikan data dan fungsi yang kompleks.

Setelah membuka *Software* Matlab dengan cara *double click* pada ikon Matlab di layar komputer maka akan muncul tampilan seperti Gambar 2.5 sebagai berikut.



Gambar 2.5 Tampilan *software* Matlab

Tampilan tersebut adalah tampilan *default* pada dekstop Matlab dan bisa diubah sesuai keinginan dengan cara klik menu Dekstop kemudian pilih Dekstop Layout.

Ahlersten K. (2012) menyatakan bahwa pada tampilan awal tersebut terdapat 4 *windows*, yaitu:

a. *Command Window*

Bagian ini adalah bagian terpenting dari Matlab karena dari sinilah kita memasukkan perintah untuk memberitahukan program tentang apa yang harus dilakukan.

b. *Current Directory*

Pada bagian ini sama seperti halnya menu *Explore* pada komputer yang menampilkan folder ataupun *file* yang ada pada folder tersebut.

c. *Work Space*

Merupakan daftar dari semua variabel yang ada. Tampilannya akan terlihat kosong sebelum kita memasukkan variabel yang ada. Bisa digunakan untuk melihat dan merubah variabel yang kita inginkan.

d. *Command History*

Bagian ini akan memunculkan *Command* atau perintah sebelumnya yang kita masukan pada Matlab.

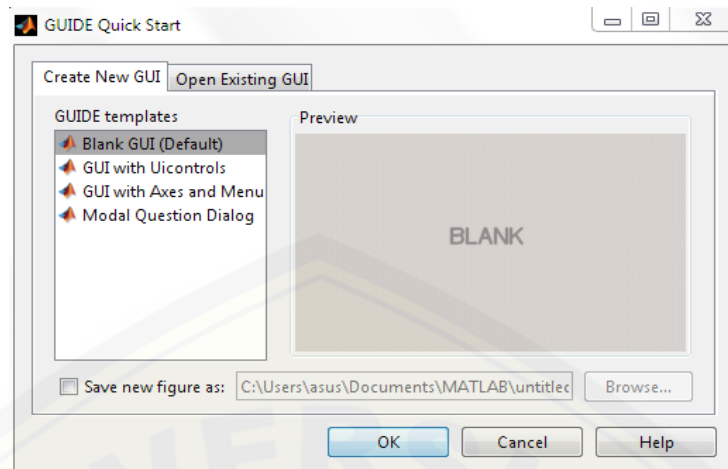
Apabila telah selesai menggunakan *software* Matlab, klik tanda x berwarna merah pada bagian pojok kanan atas untuk keluar dari *software* Matlab.

2.8.2 GUI (*Graphical User Interface*)

Graphical User Interface atau biasa disingkat dengan GUI merupakan salah satu menu yang ada pada *software* MATLAB yang mengandung perintah, atau komponen program yang mempermudah pengguna dalam menjalankan sebuah program dalam MATLAB. Dalam GUI juga menyediakan berbagai komponen interaktif, termasuk menu, toolbar dan tabel. Untuk membuat GUI pada MATLAB akan dijelaskan sebagai berikut.

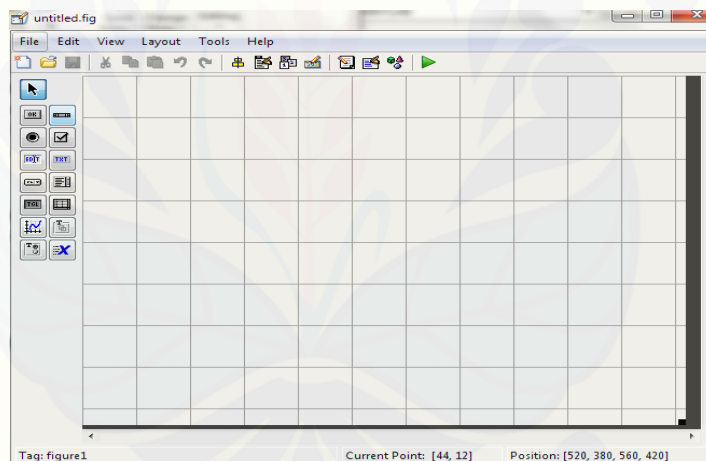
a. Jalankan *software* Matlab

b. Ketik Guide pada *Command Window*, maka akan muncul tampilan seperti Gambar 2.6 sebagai berikut.



Gambar 2.6 Tampilan Untuk perintah Guide

- c. Selanjutnya pilih Blank GUI (Default) dan klik OK, maka akan muncul tampilan seperti Gambar 2.7 berikut.



Gambar 2.7 Tampilan awal GUI

Dalam GUI terdapat beberapa menu, diantaranya:

1. *Select*

Berbentuk seperti anak panah yang berfungsi untuk memilih menu yang ada pada GUI.

2. *Push Button*

Push Button biasa digunakan untuk menjalankan suatu fungsi akan dieksekusi. Saat GUI dijalankan, *Push Button* di klik untuk menjalankan fungsi tersebut.

3. *Radio Button*

Radio button digunakan untuk memilih atau menandai pilihan dari beberapa pilihan yang ada (bisa lebih dari satu).

4. *Edit Text*

Edit Text digunakan untuk input data yang dimasukkan ke dalam program.

5. *Pop-up Menu*

Pop-up Menu biasa digunakan untuk menu atau pilihan. Biasanya diisi lebih dari 1 pilihan.

6. *Toggle Button*

Toggle button memiliki fungsi yang sama dengan pushbutton. Perbedaannya adalah saat pushbutton ditekan, maka tombol akan kembali pada posisi semula jika tombol mouse dilepas, sedangkan pada toggle button, tombol tidak akan kembali ke posisi semula, kecuali kita menekannya kembali.

7. *Axes*

Axes berfungsi menampilkan grafik atau gambar (image). *Axes* tidak masuk dalam *UIControl*, tetapi dapat diprogram agar pemakai dapat berinteraksi dengan *axes* dan obyek grafik yang ditampilkan melalui *axes*.

8. *Button Group*

Button Group berfungsi menyatukan beberapa radio button. Jika salah satu radio button dipilih, yang lain otomatis dikosongi.

9. *Slider*

Slider berfungsi memberi input nilai tanpa menggunakan keyboard, kita dapat mengatur sendiri nilai maksimum, minimum, serta sliderstep. Caranya dengan menggeser slider secara vertical maupun horizontal ke nilai yang diinginkan. Nilai default slider adalah 0 sampai 1.

10. *Check Box*

Check Box berfungsi menyediakan beberapa pilihan mandiri atau tidak bergantung dengan pilihan-pilihan lainnya.

11. *Static Text*

Static Text biasa digunakan untuk memberi keterangan pada GUI. Bisa juga digunakan untuk menampilkan output.

12. *Listbox*

Mirip dengan Pop-up Menu tetapi daftar menu ditampilkan dalam bentuk list.

13. *Table*

Table berfungsi menampilkan tabel di GUI pada software Matlab.

14. *Panel*

Panel biasa digunakan sebagai background atau tempat mendesain GUI. Bisa juga untuk mengelompokan.

15. *ActiveX Control*

2.9 Nisbah Citra

Nisbah citra adalah ukuran presentase citra yang telah berhasil dimampatkan. Secara matematis nisbah citra ditulis sebagai berikut:

$$\text{Nisbah} = \left[\frac{\text{Kapasitas foto sebelum} - \text{Kapasitas foto sesudah}}{\text{Kapasitas foto sebelum}} \right] \times 100\% \quad (2.3)$$

Semakin tinggi presentase dari kompresi citra maka ukuran citra yang dihasilkan semakin kecil dan sebaliknya semakin sedikit presentase hasil nisbah kompresi citra maka hasil dari kompresi citra masih berukuran besar.

2.10 Kualitas Kompresi Citra Menggunakan MSE dan PSNR

Kompresi pada citra digital tentunya akan menghilangkan beberapa informasi pada citra tersebut. Banyak sedikitnya informasi yang hilang akan berpengaruh terhadap kualitas citra hasil kompresi tersebut. Semakin banyak informasi yang hilang maka kualitas dari citra kompresi akan banyak mengalami penurunan sehingga mengakibatkan kualitas citra hasil kompresi menjadi kurang baik, sedangkan apabila semakin sedikit informasi yang hilang maka kualitas dari citra hasil kompresi yang baik. Begitupun dengan kapasitas memori, untuk citra dengan kualitas hasil kompresi yang baik tentunya ukuran dari file citra hasil kompresi tersebut lebih besar dibanding dengan citra hasil kompresi dengan kualitas yang tidak baik. Untuk menghitung nilai kualitas dari citra hasil kompresi

digunakan besaran PSNR (*Peak Signal to Noise Ratio*) yang dinyatakan dalam satuan *decibel* (db). Metode kompresi dikatakan baik apabila menghasilkan nilai MSE (*Mean Square Error*) yang kecil dan nilai PSNR yang tinggi. Untuk menghitungnya dapat digunakan persamaan berikut ini.

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2 \quad (2.4)$$

dan

$$PSNR = 10 \log_{10} \frac{(255)^2}{MSE} \quad (2.5)$$

dengan:

MSE : *Mean Square Error*

PSNR : *Peak Square to Noise Ratio*

M,N : dimensi citra

I(x,y) : nilai *pixel* citra asli

I'(x,y) : nilai *pixel* citra kompresi

2.11 Histogram

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu didalam citra. Histogram juga dapat menunjukkan banyak hal tentang kecerahan dan kontras dari sebuah citra. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra (Munir, 2002). Secara matematis histogram citra dihitung dengan rumus sebagai berikut.

$$h_i = \frac{n_i}{n}, i = 0,1,\dots,L-1 \quad (2.6)$$

yang dalam hal ini,

n_i = jumlah *pixel* yang memiliki derajat keabuan i

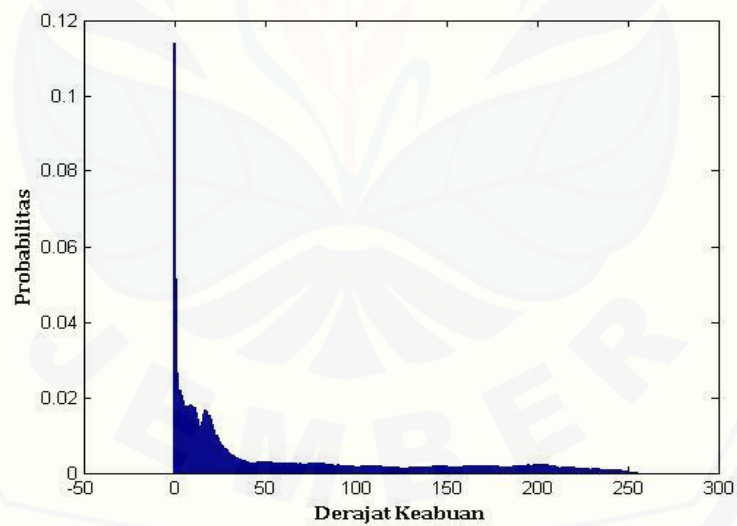
n = jumlah seluruh *pixel* di dalam citra

L = derajat keabuan

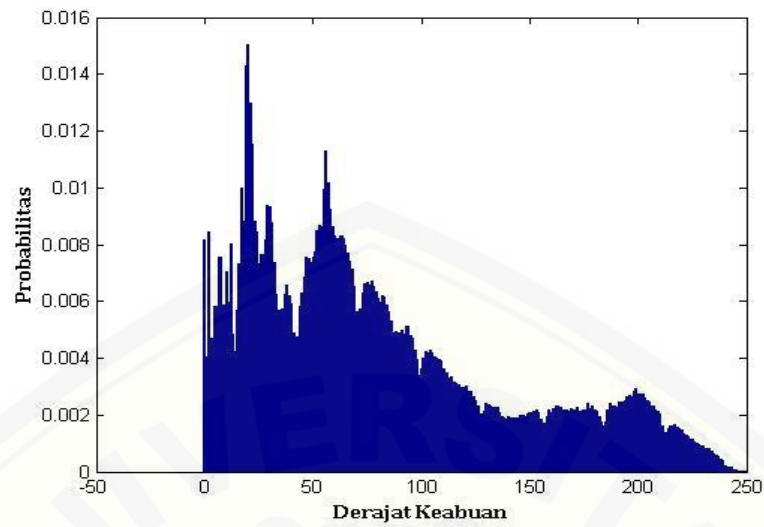
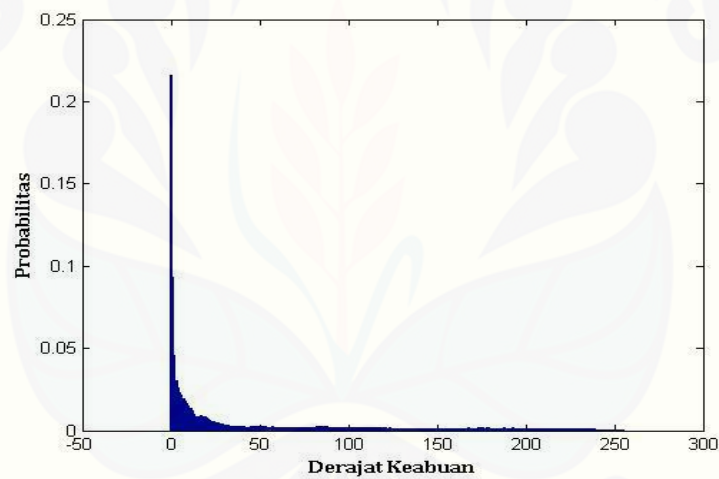
Khusus untuk citra berwarna, histogramnya dibuat pada setiap kanal RGB, seperti contoh berikut.



Gambar 2.8 Citra RGB



Gambar 2.9 Histogram *Red*

Gambar 2.10 Histogram *Green*Gambar 2.11 Histogram *Blue*

Pada histogram *red*, *green*, dan *blue* diatas, pada sumbu x terdapat nilai 0-300 yang menunjukkan rentang nilai derajat keabuan pada citra dan pada sumbu y menunjukkan nilai probabilitas derajat keabuan pada citra tersebut.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Pada penelitian ini data yang akan digunakan dalam penelitian ini adalah citra digital dengan format *.jpg*, *.bmp*, dan *.png*. Citra tersebut terdiri dari citra berwarna dan citra keabuan. Citra keabuan didapatkan dari citra berwarna yang diubah menjadi citra keabuan.

3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur

Studi literatur yang dilakukan pada penelitian kali ini adalah dengan mempelajari materi tentang Metode Kuantisasi dari penelitian-penelitian yang pernah dilakukan sebelumnya.

b. Pembuatan Program Pemampatan

Pembuatan program dilakukan untuk mengetahui waktu kompresi, perubahan intensitas warna *red*, *green*, dan *blue* pada citra tersebut, serta perubahan kualitas pada setiap komponen *red*, *green*, dan *blue*. Pembuatan program menggunakan *software* MATLAB2009a versi 7.8.0.

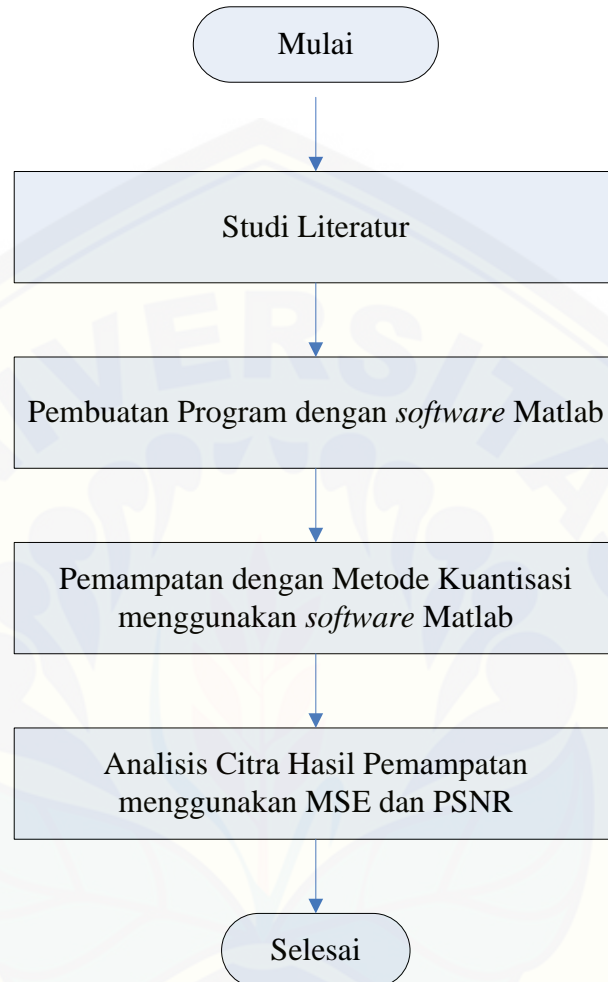
c. Pemampatan dengan Metode Kuantisasi

Pemampatan dilakukan berdasarkan bit kompresinya, yaitu dengan merubah citra menjadi 7 bit, 6 bit, 5 bit, 4 bit, 3 bit, 2 bit, dan 1 bit.

d. Analisis Histogram Citra Hasil Pemampatan

Analisis histogram dilakukan untuk mengetahui terjadinya perubahan pada RGB citra pada citra berwarna dan nilai *Grayscale* pada citra keabuan setelah dimampatkan. Analisis dilakukan dengan membandingkan intensitas warna pada histogram citra sebelum dan setelah dimampatkan serta menggunakan nilai MSE

dan PSNR untuk mengetahui kualitas citra hasil pemampatan Metode Kuantisasi tersebut. Skema dari metode penelitian bisa dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Skema langkah-langkah penelitian.

BAB 5. PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian tentang Analisa Pemampatan metode kuantisasi dengan menggunakan 3 foto berwarna dan 3 foto keabuan dalam skripsi ini adalah:

- a. pemampatan dilakukan dengan menggunakan Metode Kuantisasi dengan cara memampatkan citra tersebut dengan menjadikannya 7 bit, 6 bit, 5 bit, 4 bit, 3 bit, 2 bit, dan 1 bit.
- b. menggunakan citra berwarna dan keabuan didapatkan hasil, bahwa ukuran *file* citra hasil kompresi bergantung pada besarnya bit kompresi, semakin besar bit kompresi yang digunakan maka akan menghasilkan nilai nisbah yang kecil, dan semakin kecil bit kompresi yang digunakan maka akan menghasilkan nilai nisbah yang besar.
- c. untuk mengetahui nilai kualitas pada citra hasil pemampatan digunakan Histogram, MSE, dan PSNR. Dari penelitian yang dilakukan didapatkan hasil bahwa besarnya selisih antara derajat keabuan awal dengan derajat keabuan baru menyebabkan besarnya nilai MSE, hal ini menyebabkan nilai PSNR / kualitas citra hasil pemampatan menjadi kecil.

5.2 Saran

Penelitian tentang analisis kualitas citra bisa dilakukan pada metode-metode pemampatan lainnya, baik metode *Lossless Compression* ataupun *Lossy Compression*. Dengan demikian akan diketahui metode mana yang menghasilkan citra hasil kompresi dengan kualitas yang baik.

DAFTAR PUSTAKA

- Ahlersten, K. 2012. *An Intoduction to Matlab*, Denmark: The eBook company.
- Arymurthy, A.M dan Setiawan, S. 1992. *Pengantar Pengolahan Citra*, PT. Elex Media Komputindo.
- Ginting, A. M. 2010. Implementasi Algoritma Huffman pada kompresi citra BMP. Skripsi. Medan: Universitas Sumatera Utara.
- Gonzales R. C. dan Woods R. E.2002. *Digital Image Processing Third Edition*. Pearson Education International.
- Kaur, M. dan Kaur, G. 2013 A survey of Lossless and Lossy Image Compression Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*. **3**, 323-326.
- Kuatno, N. 2011. “Kompresi Dengan Modifikasi Metode Kuantisasi pada Foto Digital”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember
- Kurniawan, F. 2015. Perancangan Aplikasi Kompresi Citra Berwarna Menggunakan Metode Kuantisasi. *Informasi dan Teknologi Ilmiah (INTI)*. **5**, 155-159.
- Lestari, M. Situmorang, M. dan Lydia, M. S. 2012. Analisis dan Perancangan Perangkat Lunak Kompresi Citra Menggunakan Algoritma Fast Fourier Transform (FFT). *Program Studi SI Ilmu Komputer*. **1**, 1-8.
- Munir, R. 2002. *Diktat Kuliah Pengolahan Citra Edisi Kedua*. Bandung: Informatika.
- Munir, R. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Salomon, D. 2007. *Data Compression The Complete Reference*. 4th Edition. London : Springer-Verlag.
- Sutoyo, T. M, Edy. S, Vincent, N, Oky D. W. 2009. *Teori Pengolahan Citra*. Semarang : Andi.
- Wijaya, M. Ch. dan Prijono, A. 2007. *Pengolahan Citra Digital Menggunakan Matlab*. Jakarta: Informatika.

LAMPIRAN

A. Matlab

1. Program Utama

Nama file : **kompresi_kuantitasi.m**

```
function varargout = kompresi_kuantitasi(varargin)
% KOMPRESI_KUANTITASI M-file for kompresi_kuantitasi.fig
%   KOMPRESI_KUANTITASI, by itself, creates a new
KOMPRESI_KUANTITASI or raises the existing
%   singleton*.
%
%   H = KOMPRESI_KUANTITASI returns the handle to a new
KOMPRESI_KUANTITASI or the handle to
%   the existing singleton*.
%
%
KOMPRESI_KUANTITASI('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in KOMPRESI_KUANTITASI.M with the
given input arguments.
%
%   KOMPRESI_KUANTITASI('Property','Value',...) creates a new
KOMPRESI_KUANTITASI or raises the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before kompresi_kuantitasi_OpeningFcn
gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to
kompresi_kuantitasi_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
kompresi_kuantitasi

% Last Modified by GUIDE v2.5 01-Nov-2016 14:37:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @kompresi_kuantitasi_OpeningFcn, ...
                  'gui_OutputFcn',  @kompresi_kuantitasi_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
```

```
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before kompresi_kuantitasi is made visible.
function kompresi_kuantitasi_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to kompresi_kuantitasi (see
VARARGIN)

% Choose default command line output for kompresi_kuantitasi
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes kompresi_kuantitasi wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = kompresi_kuantitasi_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all
properties.
function uitable1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```



```

% handles    empty - handles not created until after all
CreateFcns called
set(hObject,'data','');

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nama_file,nama_path]=uigetfile({'*.jpg'; '*.bmp'; '*.png'; '*.tif'},
...
    'Buka Citra');
if ~isequal(nama_file,0)
    handles.data=imread(fullfile(nama_path,nama_file));
    guidata(hObject,handles);
    axes(handles.axes1);
    imshow(handles.data);
    info=iminfo(fullfile(nama_path,nama_file));
    set(handles.edit1,'string',[nama_path,nama_file]);
    namadata=[{'Nama
File'},{'Ukuran'},{'Format'},{'Panjang'},{'Tinggi'},{'Tanggal'}];
    data=[{nama_file}; {num2str(floor(info.FileSize/1000)) '
Kb'}]; {info.Format}; {num2str(info.Width)};
{num2str(info.Height)};...

```

```

        {info.FileModDate(1:length(info.FileModDate)-9)}];
        set(handles.uitable1,'data',[namadata'
data'],'columnname','','rowname','numbered','columnwidth',{82,87})
    end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
matkomp=get(handles.figure1,'userdata');
set(handles.figure1,'CurrentAxes',handles.axes2);
set(handles.axes2,'nextplot','replace');
pop=get(handles.popupmenu1,'value');
set(handles.radiobutton1,'value',1)
set(handles.radiobutton2,'value',0)
ap=max(max(handles.data));
ap1=max(ap);
dbin=dec2bin(ap1);
kl=length(dbin);
if pop==1
    [data p fr]=histogram(handles.data(:,:,1));
    [data1 p1 fr2]=histogram(matkomp(:,:,1));
elseif pop==2
    [data p fr]=histogram(handles.data(:,:,2));
    [data1 p1 fr2]=histogram(matkomp(:,:,2));
else
    [data p fr]=histogram(handles.data(:,:,3));
    [data1 p1 fr2]=histogram(matkomp(:,:,3));
end
plot(data,p,'k','linewidth',2)
set(handles.axes2,'nextplot','add');
plot(data1,p1,'m','linewidth',2)
legend('Citra Asli','Citra Kompresi')
if max(p)>max(p1)
    t=max(p);
else
    t=max(p1);
end
[ms psnr]=mse(handles.data,matkomp,pop);
set(handles.text1,'string',['PSNR : ' num2str(psnr)])
set(handles.text2,'string',['MSE : ' num2str(ms)])
F=max(max(matkomp));
F1=max(F);
G=max(max(handles.data));
G1=max(G);
if F1~=G1
    if G1>200
        bnr=ones(1,kl);
    else
        bnr=ones(1,kl-1);
    end
    set(handles.axes2,'xlim',[0 biner2digit(bnr)],'ylim',[0
t],'xtick',[0 F1 biner2digit(bnr)],'Color',[1 1 1],'XColor',...

```

```

        [0 0 0], 'YColor', [0 0 0], 'xgrid', 'off', 'ygrid', 'off');
else
    bnr=ones(1,kl-1);
    set(handles.axes2, 'xlim', [0 biner2digit(bnr)], 'ylim', [0
t], 'xtick', [0 biner2digit(bnr)], 'Color', [1 1 1], 'XColor', ...
        [0 0 0], 'YColor', [0 0 0], 'xgrid', 'off', 'ygrid', 'off');
end
set(handles.radiobutton1, 'value', 1)
set(handles.uitable2, 'data', handles.data(:, :, pop), 'columnwidth', {4
8});
% Hints: contents = get(hObject, 'String') returns popupmenu1
contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes during object creation, after setting all
properties.
function uitable2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uitable2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
set(hObject, 'data', '')

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc
set(handles.popupmenu1, 'value', 1);
set(handles.radiobutton1, 'value', 1);
set(handles.radiobutton2, 'value', 0);

```

```

plh=get(handles.popupmenu2,'value');
if plh==1
    warndlg('Pilih Kompresi Bit Terlebih Dahulu','Peringatan');
    return
end
ap=max(max(handles.data));
ap1=max(ap);
dbin=dec2bin(ap1);
kl=length(dbin);
cek=0;
if kl-1<plh-1
    warndlg(['Citra Tidak Dapat Dikompresi ke ' num2str(plh-1) '
bit, karena ukuran maksimum piksel ' num2str(kl-1) '
bit'],'PERINGATAN')
    return
end
e=handles.data(:,:,1);
if (kl-2)<plh-1
    for i=1:kl
        if dbin(i)=='0'
            cek=1;
            break
        end
    end
end
if cek==0
    komp(:,:,1)=compres(handles.data,1,plh-1);
    komp(:,:,2)=compres(handles.data,2,plh-1);
    komp(:,:,3)=compres(handles.data,3,plh-1);
else
    komp=handles.data;
end
guidata(hObject,handles);
axes(handles.axes3);
imshow(komp);
[nama_file_simpan,path_simpan]=uiputfile(...
    {'*.jpg','File citra(*.jpg)';
    '*.jpg','Citra JPEG(*.jpg)';...
    '*.*','Semua File (*.*)'},...
    'Menyimpan File Citra Hasil Kompres JPEG');
if~isequal(nama_file_simpan,0)
    imwrite(komp,fullfile(path_simpan,nama_file_simpan));
    info=imfinfo(fullfile(path_simpan,nama_file_simpan));
    namadata=[{'Nama
File'},{'Ukuran'},{'Format'},{'Panjang'},{'Tinggi'},...
    {'Tanggal'}}];
    data=[{nama_file_simpan}; {[num2str(floor(info.FileSize/1000)) '
Kb']}; {info.Format}; {num2str(info.Width)};
{num2str(info.Height)};...
    {info.FileModDate(1:length(info.FileModDate)-9)}}];
    set(handles.uitable3,'data',[namadata'
data'],'columnname','','rowname','numbered','columnwidth',{82,87})
end
set(handles.figure1,'CurrentAxes',handles.axes2);
set(handles.axes2,'nextplot','replace');
[data p fr]=histogram(e);

```

```

[data1 p1 fr2]=histogram(komp(:,:,1));
plot(data,p,'k','linewidth',2)
set(handles.axes2,'nextplot','add');
plot(data1,p1,'m','linewidth',2)
legend('Citra Asli','Citra Kompresi')
if max(p)>max(p1)
    t=max(p);
else
    t=max(p1);
end
[ms psnr]=mse(e,komp,1);
set(handles.text1,'string',['PSNR : ' num2str(psnr)])
set(handles.text2,'string',['MSE : ' num2str(ms)])
F=max(max(komp));
F1=max(F);
G=max(max(handles.data));
G1=max(G);
if F1~=G1
    if G1>200
        bnr=ones(1,k1);
    else
        bnr=ones(1,k1-1);
    end
    set(handles.axes2,'xlim',[0 biner2digit(bnr)'],'ylim',[0
t],'xtick',[0 F1 biner2digit(bnr)'],'Color',[1 1 1],'XColor',...
[0 0 0],'YColor',[0 0 0],'xgrid','off','ygrid','off');
else
    bnr=ones(1,k1-1);
    set(handles.axes2,'xlim',[0 biner2digit(bnr)'],'ylim',[0
t],'xtick',[0 biner2digit(bnr)'],'Color',[1 1 1],'XColor',...
[0 0 0],'YColor',[0 0 0],'xgrid','off','ygrid','off');
end
set(handles.radiobutton1,'value',1)
set(handles.uitable2,'data',handles.data(:,:,1),'columnwidth',{48}
,'columnname','numbered','rowname','numbered');
set(handles.figure1,'userdata',komp);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.edit1,'string','')
set(handles.text1,'string','PSNR :')
set(handles.text2,'string','MSE :')
set(handles.uitable1,'data',[],'rowname','','columnname','');
set(handles.uitable2,'data',[],'rowname','','columnname','');
set(handles.uitable3,'data',[],'rowname','','columnname','');
set(handles.popupmenu1,'value',1)
set(handles.popupmenu2,'value',1)
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',0);
cla(handles.axes1,'reset');
set(handles.axes1,'Color',...

```

```

        [0.941 0.941 0.941], 'XColor', [0.941 0.941 0.941], ...
        'YColor', [0.941 0.941 0.941]);
set(handles.axes1, 'Xtick', [], ...
    'Ytick', []);
cla(handles.axes2, 'reset');
set(handles.axes2, 'Color', ...
    [1 1 1], 'XColor', [0 0 0], ...
    'YColor', [0 0 0]);
set(handles.axes2, 'xlim', [1 10], 'ylim', [1 10], 'Xtick', [1:10], ...
    'Ytick', [1:10], 'xgrid', 'on', 'ygrid', 'on');
cla(handles.axes3, 'reset');
set(handles.axes3, 'Color', ...
    [0.941 0.941 0.941], 'XColor', [0.941 0.941 0.941], ...
    'YColor', [0.941 0.941 0.941]);
set(handles.axes3, 'Xtick', [], ...
    'Ytick', []);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1, 'value', 1);
set(handles.radiobutton2, 'value', 0);
pop=get(handles.popupmenu1, 'value');
set(handles.uitable2, 'data', handles.data(:, :, pop), 'columnwidth', {4
8});
% Hint: get(hObject, 'Value') returns toggle state of radiobutton1

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1, 'value', 0);
set(handles.radiobutton2, 'value', 1);
k=get(handles.figure1, 'userdata');
pop=get(handles.popupmenu1, 'value');
set(handles.uitable2, 'data', k(:, :, pop), 'columnwidth', {48});
% Hint: get(hObject, 'Value') returns toggle state of radiobutton2

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all
properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2. Program Kompresi

Nama file : compres.m

```

function [A minim maks h]=compres(B,rgb,bag)
A=double(B(:,:,rgb));
[m n]=size(A);
[b p frek]=histogram(A);
biit=zeros(1,bag+1);
biit(1)=1;
bagi=biner2digit(biit);
a=mod(length(b),bagi);
nb=floor(length(b)/bagi);
minim(1)=min(b);
for i=1:bagi
    h(i)=nb;
end
if a~=0
    for i=1:a
        h(i)=h(i)+1;
    end
end
jml=h(1);
for i=1:bagi-1
    maks(i)=b(jml);
    minim(i+1)=b(jml+1);
    jml=jml+h(i+1);
end
maks(i+1)=b(jml);

```

```

for i=1:m
    for j=1:n
        for z=1:length(minim)
            if A(i,j)>=minim(z) && A(i,j)<=maks(z)
                A(i,j)=z-1;
            end
        end
    end
end
end
A=uint8(A);

```

3. Program histogram

Nama file : histogram.m

```

function [data P frek]=histogram(dat)
dat=double(dat);
[j k l]=size(dat);
n=j*k;
D=sort(dat(1:n));
k=1;b=0;
for i=2:n
    if D(i-1)==D(i)
        k=k+1;
    else
        b=b+1;
        data(b)=D(i-1);
        P(b)=k/n;
        frek(b)=k;
        k=1;
    end
end
end
b=b+1;
data(b)=D(n);
frek(b)=k;
P(b)=k/n;

```

4. Program MSE dan PSNR

Nama file : mse.m

```

function [ms psnr]=mse(a,b,rgb)
mat=double(a(:,:,rgb));
mat1=double(b(:,:,rgb));
[m n]=size(mat);
T=(mat-mat1).^2;
jml=sum(sum(T));
A=max(max(mat));
ms=jml/(m*n);
psnr=20*log10(A/sqrt(ms));

```

5. Program biner ke digit

Nama file : biner2digit.m

```

function biner=biner2digit(m)
n=length(m);
jml=0;
for k=1:n

```




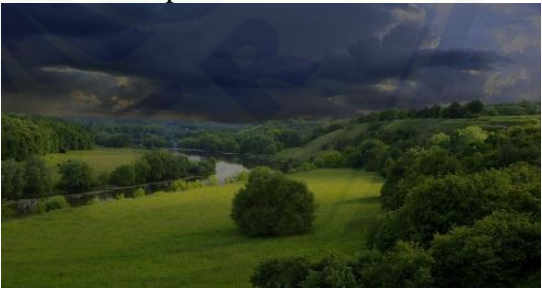


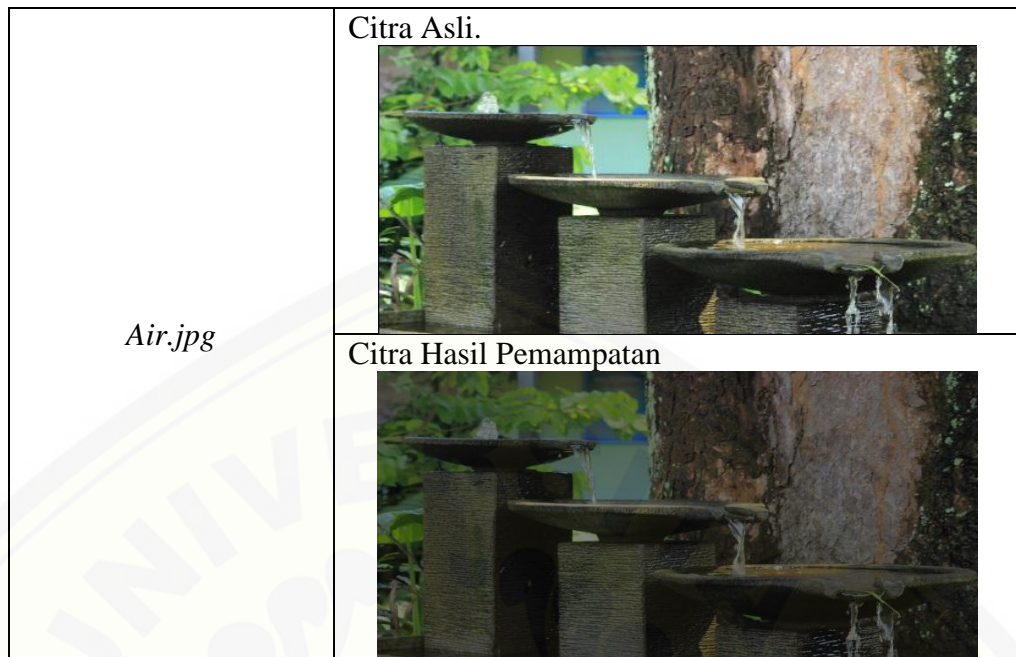
```
a(k)=2^(n-k);  
jml=jml+m(k)*a(k);  
end  
biner=jml
```



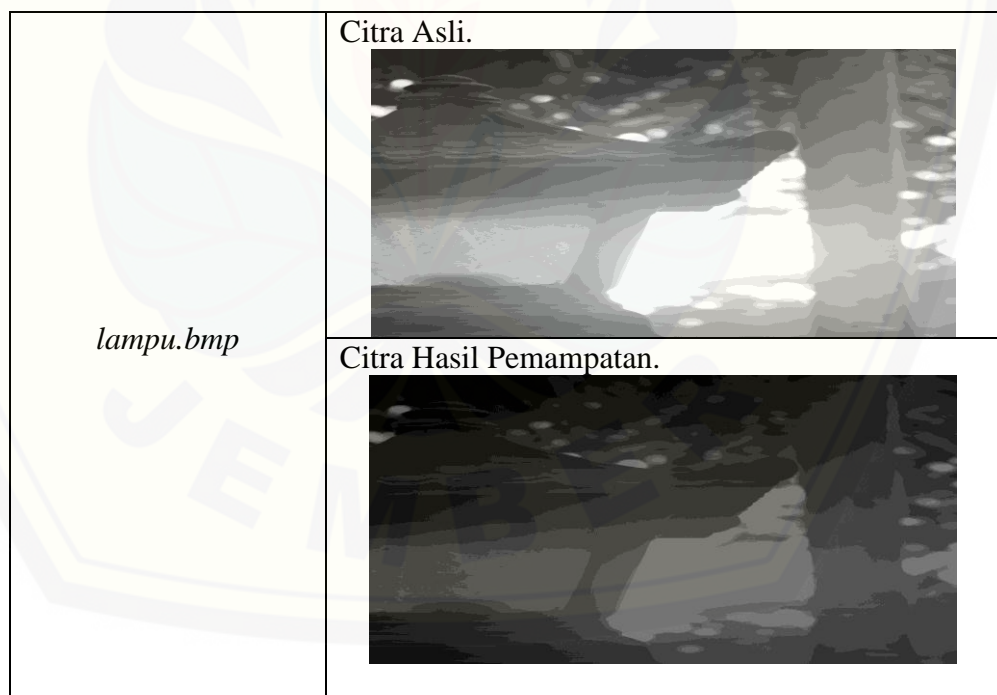
B. Citra Asli dan Citra Hasil Pemampatan

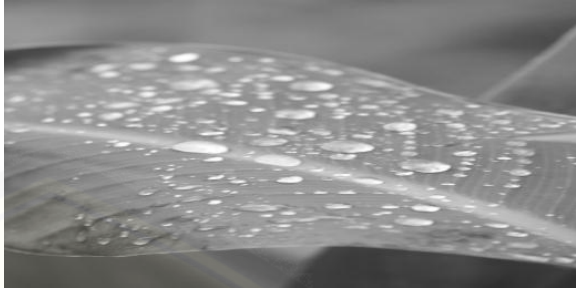
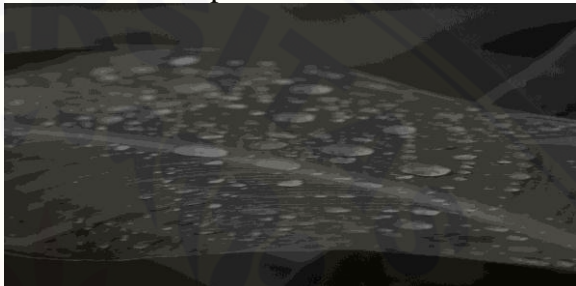


1). Citra Berwarna Pemampatan 7 bit

<p><i>pantai.bmp</i></p>	<p>Citra Asli.</p> 
	<p>Citra Hasil Pemampatan.</p> 
<p><i>hutan.png</i></p>	<p>Citra Asli.</p> 
	<p>Citra Hasil Pemampatan.</p> 


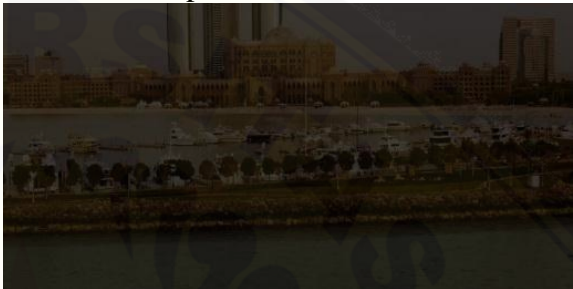

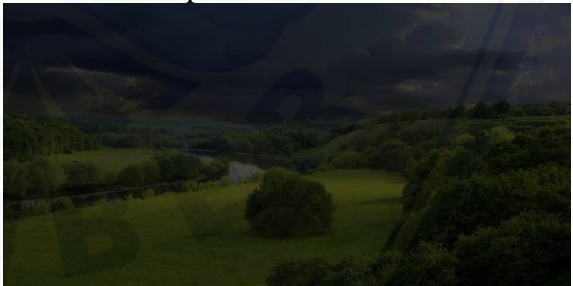


Citra Keabuan Pemampatan 7 bit.

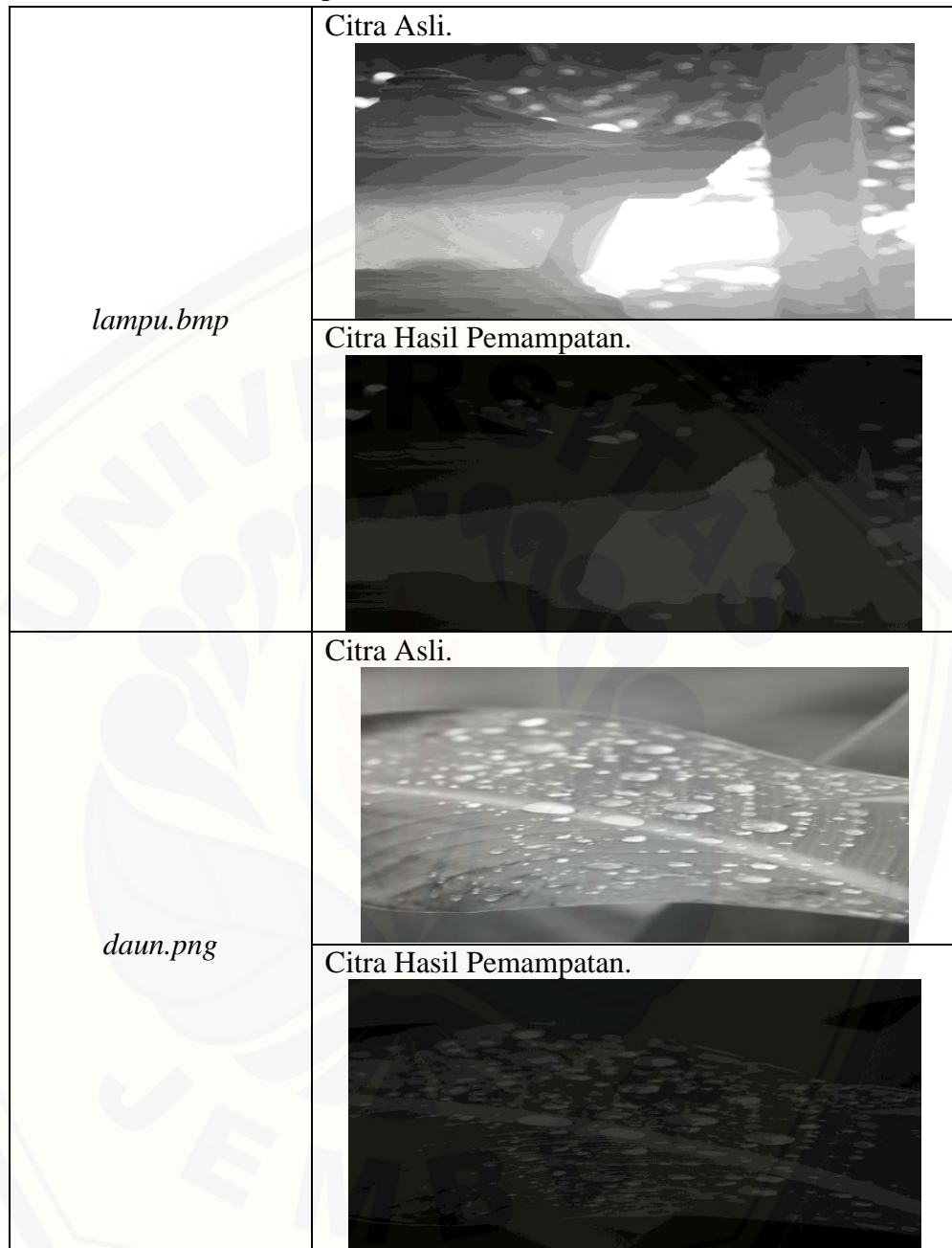


<p><i>daun.png</i></p>	<p>Citra Asli.</p> 
	<p>Citra Hasil Pemampatan.</p> 
<p><i>tenda.jpg</i></p>	<p>Citra Asli.</p> 
	<p>Citra Hasil Pemampatan.</p> 

3). Citra Berwarna Pemampatan 6 bit

<i>pantai.bmp</i>	Citra Asli. 
	Citra Hasil Pemampatan. 
<i>hutan.png</i>	Citra Asli. 
	Citra Hasil Pemampatan 

4). Citra Keabuan Pemampatan 6 bit

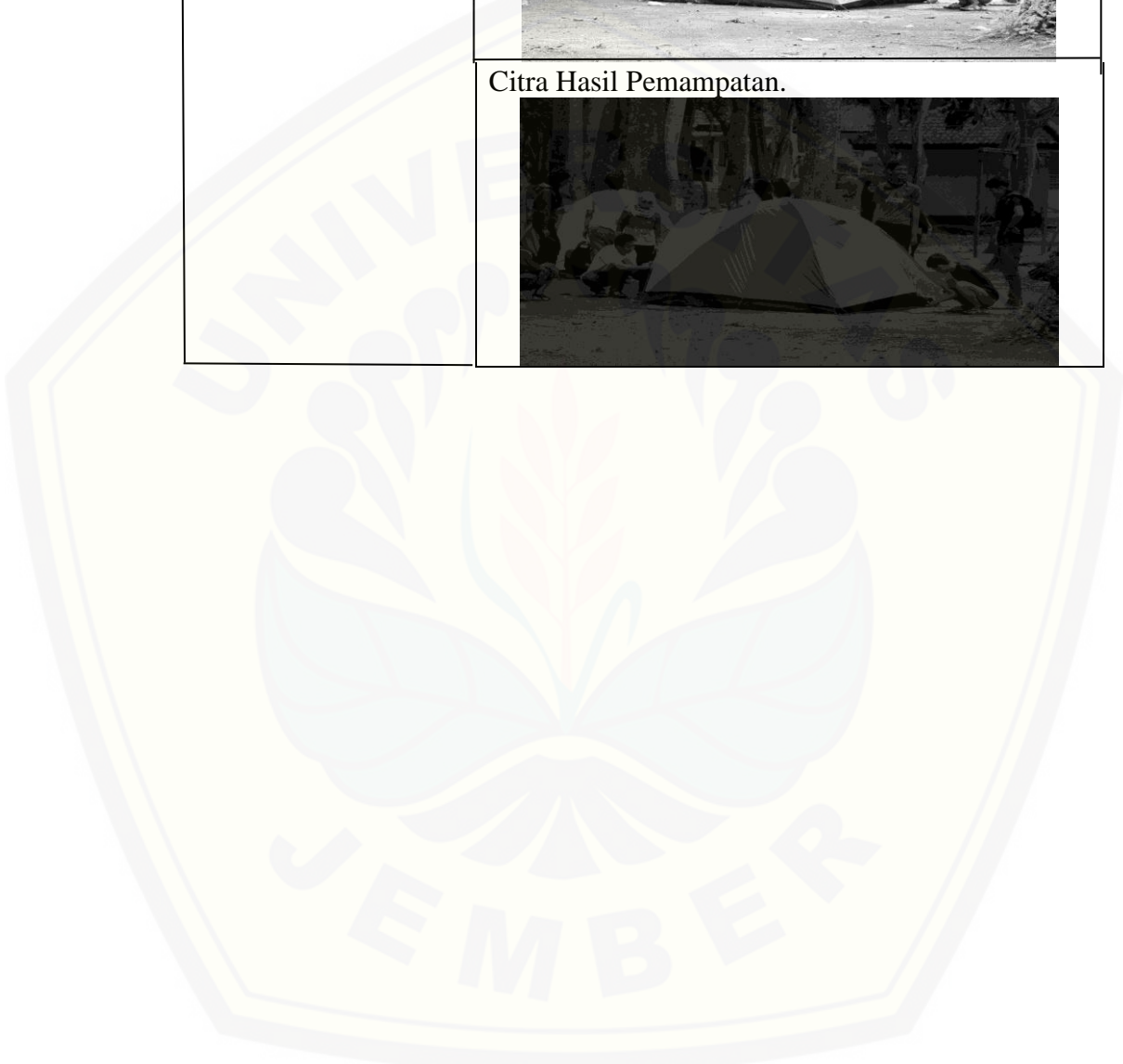


tenda.jpg

Citra Asli.



Citra Hasil Pemampatan.



G. Tabel hasil pemampatan foto keabuan *tenda.jpg*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE	PSNR	Nisbah (%)
1,4	1135	7	50	6659,2	9,8965	18,9
1,4	647	6	28	14983,4	6,3747	53,7
1,4	436	5	18	20394,7	5,0356	68,8
1,4	313	4	12	23412,2	4,4364	77,6
1,4	255	3	10	24999,1	4,1516	81,7
1,4	229	2	9	25813,8	4,0123	83,6
1,4	217	1	8	26241,3	3,9409	84,5

H. Tabel hasil pemampatan foto keabuan *lampu.png*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE	PSNR	Nisbah (%)
17,9	413	7	45	8101,1	9,0453	97,69
17,9	295	6	26	15245,8	6,2993	98,35
17,9	248	5	16	19570,4	5,2148	98,61
17,9	226	4	11	22219,9	4,6634	98,73
17,9	216	3	8	23610,1	4,3998	98,79
17,9	212	2	7,6	24323,3	4,2706	98,81
17,9	211	1	7	24678,5	4,2076	98,82

I. Tabel hasil pemampatan foto keabuan *daun.png*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE	PSNR	Nisbah (%)
7,3	1135	7	44	6659,2	9,8965	84,45
7,3	647	6	25	14983,4	6,3747	91,13
7,3	436	5	16	20394,7	5,0356	94,02
7,3	313	4	11	23412,2	4,4364	95,71
7,3	255	3	8	24999,1	4,1516	96,50
7,3	229	2	7,6	25813,8	4,0123	96,86
7,3	217	1	7	26241,3	3,9409	97,02

J. Tabel hasil pemampatan foto keabuan *sore.jpg*

Ukuran Foto Awal (kb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE	PSNR	Nisbah (%)
657	287	5	15	3452,8	12,749	56,31
657	252	4	12	3957,3	12,156	61,64
657	229	3	10	4223,3	11,875	65,14
657	219	2	8	4359,1	11,736	66,66
657	213	1	7	4421,3	11,675	67,57

C. Tabel hasil kompresi foto berwarna *Air.jpg*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE			PSNR			Nisbah (%)
				R	G	B	R	G	B	
12,3	940	7	173	3055,2	3568,0	2434,2	13,2803	12,6065	14,2672	92,3
12,3	618	6	95	6874,3	7870,7	5476,9	9,7585	9,1706	10,7454	94,9
12,3	449	5	61	9356,6	10652,1	7454,7	8,4196	7,8564	9,4065	96,3
12,3	357	4	42	10741,4	12200,0	8557,6	7,8202	7,2672	8,8073	97
12,3	312	3	35	11470,9	13014,9	9137,9	7,5348	6,9864	8,5223	97,4
12,3	291	2	30	11843,7	13430,1	9434,5	7,3959	6,85	8,3836	97,6
12,3	282	1	26	12029,8	13637,4	9583,0	7,3282	6,7835	8,3158	97,7

D. Tabel hasil kompresi foto berwarna *hutan.png*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE			PSNR			Nisbah (%)
				R	G	B	R	G	B	
6,3	427	7	27,3	2624,8	3644,3	2790,7	13,9398	12,5146	13,6736	93,22
6,3	274	6	15,6	5905,6	8199,6	6279,2	10,4181	8,9929	10,1518	95,65
6,3	169	5	9,8	8038,1	11160,3	8547,2	9,0792	7,654	8,8125	97,31
6,3	102	4	6,8	9228,5	12811,9	9812	8,4795	7,0546	8,2132	98,38
6,3	70	3	5,3	9852,6	13679,8	10479,7	8,1953	6,77	7,9273	98,88
6,3	61	2	4,6	10176,6	14120,9	10824,4	8,0548	6,6322	7,7657	99,03
6,3	59	1	4,3	10342,6	14352,6	10994,9	7,9845	6,5615	7,7189	99,06

E. Tabel hasil kompresi foto berwarna *pantai.bmp*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE			PSNR			Nisbah (%)
				R	G	B	R	G	B	
2,7	70	7	6,45	4901,3	3495	1882,5	11,2276	12,4894	13,6969	97,40
2,7	49	6	3,25	11028,2	7864	3803,5	7,7058	8,9675	10,6425	98,18
2,7	32	5	2,47	15010,2	10669,6	5493,1	6,3669	7,6425	9,0462	98,81
2,7	22	4	1,80	17232,7	12216	6439,3	5,7673	7,0547	8,3559	99,18
2,7	16	3	1,77	18387,1	13088,3	6982,8	5,4857	6,7551	8,004	99,40
2,7	13	2	1,35	18986,3	13538	7278,1	5,3464	6,6084	7,8242	99,51
2,7	11	1	1,27	19310	13759,9	7409,8	5,2729	6,5378	7,7463	99,59

F. Tabel kompresi *laut.jpg*

Ukuran Foto Awal (mb)	Ukuran Foto Kompresi (kb)	Kompresi Bit	Waktu (detik)	MSE			PSNR			Nisbah (%)
				R	G	B	R	G	B	
14,7	695	5	69	9857,5	11498,0	10356,6	8,1931	7,5246	7,9786	95,2
14,7	473	4	44	11316,2	13169,2	11889,2	7,5938	6,9352	7,3793	96,7
14,7	349	3	32	12083,6	14046,1	12695,2	7,3088	6,6552	7,0944	97,6
14,7	296	2	27	12479,9	14499,2	13106,1	7,1687	6,5173	6,9561	97,9
14,7	284	1	25	12687,1	14739,8	13322,0	7,0972	6,4459	6,8851	98