

Analisis dan Perancangan Perangkat Lunak Kompresi Citra Menggunakan Algoritma *Fast Fourier Transform* (FFT)

Rima Lestari¹, Marihat Situmorang², Maya Silvi Lydia³

Program Studi SI Ilmu Komputer, FASILKOM-TI USU

Jalan Universitas No. 9 Kampus USU Medan 20155

¹rima_tary@yahoo.com

²marihatsitumorang63@gmail.com

³maya_marzuki@yahoo.com

Abstrak— Kecepatan pengiriman informasi dalam bentuk perpaduan data teks, *audio* maupun citra secara nyata merupakan aspek penting dalam pertukaran informasi. Kecepatan pengiriman ini sangat bergantung kepada ukuran dari informasi tersebut. Pada umumnya informasi yang berupa citra akan membuat file menjadi lebih besar sehingga mempengaruhi kecepatan proses pengiriman informasi. Salah satu solusi untuk masalah di atas adalah dengan melakukan kompresi data citra sebelum ditransmisikan. Kompresi data adalah mengurangi ukuran file atau meminimalkan kebutuhan memori untuk merepresentasikan sebuah file digital. Pada penelitian ini dilakukan kompresi file citra dengan teknik *lossy* menghasilkan file citra hasil kompresi lebih kecil dari file semula karena adanya data yang dihilangkan. Data yang dihilangkan tersebut tidak akan terlihat oleh kasat mata manusia secara kualitas. Teknik kompresi yang tepat untuk keperluan diatas adalah transformasi citra, dimana yang digunakan adalah Transformasi Fast Fourier. Hasil kompresi dengan perangkat lunak yang dibangun menunjukkan ukuran file citra yang dihasilkan berkurang rata-rata untuk format BMP sebesar 87,02 % dan format JPG sebesar 41,11 %.

Kata Kunci— Pengolahan Citra Digital, Kompresi Citra, Algoritma *Fast Fourier Transform* (FFT).

I. PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang pesat telah menjadi peran yang sangat penting untuk pertukaran informasi yang cepat. Kecepatan pengiriman informasi dalam bentuk perpaduan teks, *audio* maupun citra secara nyata akan menjadi bagian utama dalam pertukaran informasi. Kecepatan pengiriman ini sangat bergantung kepada ukuran dari informasi tersebut. Salah satu solusi untuk masalah di atas adalah dengan melakukan kompresi data sebelum ditransmisikan.

Tujuan kompresi data adalah mengurangi ukuran file atau meminimalkan kebutuhan memori untuk merepresentasikan sebuah file digital. Pada umumnya file citra digital memiliki ukuran (*size*) yang lebih besar dibandingkan dengan file teks.

Salah satu syarat untuk penyimpanan atau transmisi sebuah file digital adalah ukuran file yang ramping. Oleh sebab itu, diperlukan teknik kompresi yang sesuai tujuan kebutuhan akan data yang dikompresi.

Pada penelitian ini dilakukan kompresi file citra yang tidak sensitif terhadap kehilangan beberapa bagian data dari file citra tersebut dan yang terpenting citra tersebut tidak berubah kualitasnya secara kasat mata. Diharapkan ukuran citra yang akan dihasilkan memiliki rasio kompresi yang lebih baik.

Metode *lossy* menghasilkan file citra hasil kompresi lebih kecil dari file semula karena adanya data yang hilang akibat kompresi, tetapi masih dapat ditolerir oleh penglihatan manusia. Teknik kompresi yang tepat untuk keperluan diatas adalah transformasi citra, dimana yang digunakan adalah Transformasi Fast Fourier.

Pada penelitian Riyanto, Sugeng [10]. Penelitian tersebut bertujuan untuk mengetahui persamaan DFT dan FFT dengan resolusi yang lebih tinggi yaitu 1/10 Hz, kemudian mengimplementasikan persamaan tersebut ke dalam suatu program.

Pada penelitian ini dilakukan kompresi file citra dengan menggunakan teknik kompresi *lossy* dengan algoritma *Fast Fourier Transform* (FFT) serta membandingkan hasil kompresinya dengan parameter rasio kompresi.

Berdasarkan latar belakang yang telah diuraikan diatas maka penelitian ini diberi judul “Analisis dan Perancangan Perangkat Lunak Kompresi Citra Menggunakan Algoritma *Fast Fourier Transform* (FFT)”.

B. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

- 1) Bagaimana merancang sebuah perangkat lunak yang dapat melakukan kompresi file citra dengan

menggunakan Algoritma *Fast Fourier Transform* (FFT).

- 2) Bagaimana menghitung rasio hasil kompresi algoritma diatas.

C. Batasan Masalah

Adapun yang menjadi batasan masalah dalam penelitian ini adalah:

- 1) File citra yang dikompresi adalah citra digital *true color* 24 bit dengan format citra yang dikompresi adalah JPG dan BMP.
- 2) Parameter perbandingan hasil kompresi adalah rasio kompresi antara ke dua format citra di atas.
- 3) Tidak membahas perubahan resolusi citra hasil kompresi.
- 4) Tools bahasa pemrograman yang digunakan adalah *Microsoft Visual Basic 6.0*[®].
- 5) Tools pembuatan Hasil Pengujian menggunakan *Crystal Report 8.5*[®].

D. Tujuan Penelitian

Tujuan penelitian ini adalah :

- 1) Membangun sebuah perangkat lunak kompresi file citra dengan menggunakan teknik *Lossy* dengan Algoritma *Fast Fourier Transform* (FFT).
- 2) Untuk mengetahui rasio kompresi file citra.

E. Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

- 1) Sebagai bahan referensi mengenai algoritma *Fast Fourier Transform* (FFT) pada proses kompresi file citra.
- 2) Mengetahui format citra yang paling baik dikompresi dengan algoritma *Fast Fourier Transform* (FFT).

F. Metodologi Penelitian

Penelitian ini dilakukan dengan metodologi sebagai berikut:

- 1) Studi Literatur
Mempelajari tentang citra digital, teknik kompresi data dan algoritma *Fast Fourier Transform* (FFT).
- 2) Analisa dan Perancangan
Pada tahap ini dilakukan analisa proses kerja kompresi file citra menggunakan algoritma *Fast Fourier Transform* (FFT) dan selanjutnya dilakukan perancangan sistem yang meliputi perancangan proses kerja sistem, perancangan *interface*, serta algoritma pemrograman.
- 3) Implementasi
Tahap ini digunakan mengimplementasikan hasil analisa dan perancangan sistem ke dalam bahasa pemrograman agar bisa digunakan untuk kompresi file citra.

- 4) Pengujian

Tahap ini digunakan untuk menguji coba sistem hasil rasio kompresi yang dibuat untuk mengetahui apakah program sudah berjalan sesuai dengan tujuan penelitian atau tidak, serta kelebihan dan kelemahannya.

- 5) Dokumentasi dan Laporan Akhir

Tahap ini dilakukan dokumentasi hasil analisa dan perancangan yang digunakan untuk menuangkan hasil penelitian tersebut ke dalam laporan akhir dalam bentuk skripsi.

II. TINJAUAN PUSTAKA

A. Citra

Citra (*image*) adalah gambar yang terletak pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pemantulan cahaya ini ditangkap oleh alat-alat optik antara lain layaknya mata pada manusia atau hewan, alat sensor cahaya, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek tersebut dapat terekam atau tersimpan kedalam format digital maupun analog. [9].

Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal video seperti gambar pada monitor televisi, dan bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Citra sebagai salah satu komponen multimedia memegang peranan penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

Citra yang digunakan dalam pengolahan citra oleh komputer adalah citra dalam bentuk digital. Citra digital dapat dibuat di komputer dengan bantuan perangkat lunak grafis, seperti Paint Brush, PhotoShop, dan sebagainya. Selain itu, citra digital dapat juga dihasilkan dari citra analog yang melalui proses *scanning*. Proses *scanning* memerlukan perangkat keras khusus yang disebut *scanner*.

Citra digital adalah sebagai keluaran suatu sistem perekam data dapat bersifat optik, bersifat analog atau bersifat digital. Menurut presisi yang digunakan untuk menyatakan nilai keabuan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra yaitu citra kontinu-kontinu, kontinu-diskrit, diskrit-kontinu dan diskrit-diskrit yang dimana label pertama menyatakan presisi dari titik koordinat pada bidang citra sedangkan label kedua menyatakan presisi nilai keabuan atau warna. Kontinu dinyatakan dengan presisi angka tak terhingga, sedangkan diskrit dinyatakan dengan presisi angka terhingga.[12].

Untuk mengubah citra yang bersifat kontinu menjadi citra digital diperlukan proses pembuatan kisi-kisi arah horizontal dan vertikal, sehingga diperoleh gambar dalam bentuk array dua dimensi. Proses disebut sebagai proses digitalisasi atau *sampling*.

Proses yang dilakukan selanjutnya dalam konversi tersebut diatas adalah proses kuantisasi. Dalam proses ini tingkat keabuan setiap *pixel* dinyatakan dengan suatu harga integer. Batas-batas harga integer atau besarnya daerah tingkat keabuan yang digunakan untuk menyatakan tingkat keabuan pixel akan menentukan resolusi kecerahan dari gambar yang diperoleh. Kalau digunakan tiga bit untuk menyimpan harga integer tersebut, maka akan diperoleh sebanyak delapan tingkat keabuan. Makin besar jumlah tingkat keabuan yang digunakan makin baik gambar yang akan diperoleh, karena kontinuitas dari tingkat keabuan akan semakin tinggi sehingga mendekati citra aslinya.

Seluruh tahapan proses konversi diatas dikenal sebagai konversi analog ke digital yang biasanya akan menyimpan hasil prosesnya pada memori citra. Sebaliknya sebagai hasil suatu proses pengolahan citra digital, kadang-kadang perlu mengeluarkan gambar dari memori citra ke bentuk peragaan pada monitor televisi atau ke bentuk cetak foto. Proses konversi kebalikan ini dikenal sebagai konversi digital ke analog. Citra monokrom atau citra hitam-putih merupakan citra satu kanal, dimana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam ke putih, x menyatakan variabel baris atau garis jelajah dan y menyatakan variabel kolom atau posisi pixel digaris jelajah. Sebaliknya citra berwarna dikenal juga citra multi spektral, dimana warna citra biasanya dinyatakan dalam tiga komponen warna yaitu merah, hijau dan biru yang sering disebut sebagai RGB, citra berwarna $\{f_{merah}(x,y), f_{hijau}(x,y), f_{biru}(x,y)\}$ merupakan fungsi harga vektor tingkat keabuan merah, hijau dan biru.

B. Pengolahan Citra

Ada tiga bidang studi utama yang menangani pengolahan data dalam bentuk gambar dan citra yaitu [7]

- 1) Grafika komputer
- 2) Pengolahan citra
- 3) Pengenalan pola

Karena pengenalan pola sering juga merupakan bagian dari pengolahan citra seperti misalnya pada proses klasifikasi, maka perbedaan tiga bidang studi tersebut berubah menjadi:

- 1) Grafika komputer
- 2) Pengolahan citra
- 3) Visi komputer

Dalam penerapan pengolahan citra digital tujuannya adalah untuk mengolah citra suatu objek agar dapat memperoleh informasi mengenai objek itu sendiri karena hanya citra digital yang dapat diolah oleh komputer maka citra dari suatu objek harus dikonversi ke citra digital yang merupakan representasi dari objek itu sendiri. Sistem

pengolahan citra digital secara umum mencakup tiga operasi dasar yang terdiri dari :

- 1) Digitalisasi (*Digitalization*)
- 2) Pengolahan (*Processing*)
- 3) Peragaan (*Display*)

Oleh karena itu maka sistem pengolahan citra digital mempunyai beberapa komponen dasar, yang terdiri atas :

- 1) *Digitizer*
- 2) *Image Processor*
- 3) *Storage*
- 4) Alat peraga

C. Format File Citra BMP

Format file BMP merupakan format standar sistem operasi windows dalam IBM OS/2. Format ini mendukung mode warna dari bitmap mode hingga RGB mode.

- 1) Format file ini menyimpan datanya secara terbalik, yaitu dari bawah keatas.
- 2) Citra dengan resolusi warna 8 bit, lebar citra harus merupakan kelipatan dari 4 bila tidak maka pada saat penyimpanan akan ditambahkan beberapa byte pada data hingga merupakan kelipatan dari 4.
- 3) Citra dengan resolusi warna 24 bit, urutan 3 warna dasar adalah biru, hijau, merah (RGB).
- 4) Ukuran file bitmap ini bisa sangat besar hingga *Megabytes*.

D. Format File Citra JPEG

Format file *Joint Photographic Expert Group* (JPEG) atau yang biasa disingkat JPG meningkat pesan penggunaannya. Format ini terkenal karena ukurannya yang mini dibandingkan dengan format-format citra lainnya. JPG mendukung mode warna RGB, CMYK dan Grayscale, tetapi tidak mampu menampilkan citra dengan latar belakang transparan [3]. Format JPG menterjemahkan informasi tersebut menjadi komponen Luminance (komponen cahaya) dan 2 komponen Chromatic (komponen perubahan warna dari hijau ke merah dan biru ke kuning). Untuk kompresinya format file citra ini menggunakan kompresi JPG.

E. Mode Warna

Menampilkan sebuah citra pada layar monitor diperlukan lebih dari sekedar informasi tentang letak dari pixel-pixel pembentuk citra. Untuk memperoleh gambar yang tepat dibutuhkan juga informasi tentang warna-warna yang dipakai untuk menggambarkan sebuah citra digital [9]. Beberapa mode warna yang sering digunakan adalah sebagai berikut :

- 1) Bitmap mode memerlukan 1 bit data untuk menampilkan warna dan warna yang dapat ditampilkan hanya warna hitam dan putih (monokrom).
- 2) *Indexed Color Mode*, mengurutkan warna dalam jangkauan 0-255 (8 bit).
- 3) *Grayscale Mode*, menampilkan citra dalam 256 tingkat keabuan.

- 4) RGB Mode, menampilkan citra dalam kombinasi 3 warna dasar (*Red, Green, Blue*) tiap warna dasar memiliki intensitas warna 0-255 (8 bit).
- 5) CMYK Mode, menampilkan citra dalam kombinasi 4 warna dasar (cyan, magenta, yellow, black) tiap warna dasar memiliki intensitas warna 0-255 (8 bit).

Warna didefinisikan dengan memasukkan intensitas untuk setiap komponen dalam matriks. Tiap komponen memiliki matriksnya sendiri-sendiri dan matriks-matriks tersebut bisa dijumlahkan. Sebagai contoh, untuk menghasilkan merah saturasi sempurna, masukan (1,0,0) : 100% merah 0% hijau dan 0% biru. Pada saat ketiga komponen warna tersebut dikombinasikan dalam 100% saturasi (1,1,1) hasilnya adalah putih (seperti diperlihatkan berikut):

$$\begin{aligned} \text{Hijau (0,1,0) + Biru (0,0,1)} &= \text{Cyan (0,1,1)} \\ \text{Cyan (0,1,1) + Merah (1,0,0)} &= \text{Putih (1,1,1)} \end{aligned}$$

Formula sederhana berikut menterjemahkan model RGB ke model CMYK :

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

Formula tersebut hanya merupakan titik awal. Bagaimanapun pada prakteknya kalibrasi intensif dari piranti dibutuhkan karena pigmen warna khas umumnya tidak bekerja seperti yang diperkirakan dari perhitungan.

F. Representasi Warna Digital

Warna digital adalah kombinasi dari tiga warna primer yaitu merah, hijau, dan biru (RGB). Setiap warna dapat dispesifikasikan sebagai triplet nilai intensitas RGB, dengan setiap intensitas berkisar antara 0 sampai 255 [4] yaitu

$$\text{Red : RGB (255, 0,0).....(2.1)}$$

$$\text{Green : RGB (0, 255, 0).....(2.2)}$$

$$\text{Blue : RGB (0, 0, 255).....(2.3)}$$

Dari nilai triplet RGB persamaan (2.1) sampai (2.3) di atas dapat dikonversikan ke dalam nilai desimal seperti dibawah ini:

$$\text{Red : } 255 \cdot 256^0 + 0 \cdot 256^1 + 0 \cdot 256^2 = 255 + 0 + 0 = 255$$

.....(2.4)

$$\text{Green : } 0 \cdot 256^0 + 255 \cdot 256^1 + 0 \cdot 256^2 = 0 + 65,280 + 0$$

= 65,280(2.5)

$$\text{Blue : } 0 \cdot 256^0 + 0 \cdot 256^1 + 255 \cdot 256^2 = 0 + 0 + 16,711,680$$

= 16,711,680(2.6)

Rumus dasar mencari nilai RGB citra adalah:

$$R = \text{COLOR And RGB}(255, 0, 0).....(2.7)$$

$$G = (\text{COLOR And RGB}(0, 255, 0)) / 256).....(2.8)$$

$$B = ((\text{COLOR And RGB}(0, 0, 255)) / 256) / 256.....(2.9)$$

Dari persamaan (2.4) sampai (2.6) diatas, rumus RGB pada persamaan (2.7) sampai (2.9) menjadi:

$$\text{Nilai R} = c \text{ and } 255.....(2.10)$$

$$\text{Nilai G} = (c \text{ and } 65,280) / 256(2.11)$$

$$\text{Nilai B} = ((c \text{ and } 16,711,680) / 256) / 256(2.12)$$

G. Kompresi Data

Kompresi data dalam bidang ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana. Kompresi data dapat diartikan juga sebagai proses yang dapat mengubah sebuah aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dikompresi) yang memiliki ukuran yang lebih kecil [5].

Kompresi data sangat populer sekarang ini karena dua alasan yaitu :

- 1) Orang-orang lebih suka mengumpulkan data. Tidak peduli seberapa besar media penyimpanan yang dimilikinya. Akan tetapi cepat atau lambat akan terjadi kelebihan data (kekurangan memori penyimpanan).
- 2) Orang-orang benci menunggu waktu yang lama untuk memindahkan data. Misalnya ketika duduk di depan komputer untuk menunggu halaman *Web* terbuka atau mengunduh sebuah *file*.

Rasio kompresi data adalah ukuran persentase data yang telah berhasil dikompres [12]. Secara matematis rasio kompresi data ditulis sebagai berikut:

$$R = \frac{M_0 - M_1}{M_0} \times 100 \%$$

Dimana :

R = Rasio Kompresi

M_0 = Ukuran File Asli

M_1 = Ukuran File Terkompresi

Metode kompresi data dapat dikelompokkan dalam dua kelompok besar yaitu metode *lossless* dan metode *lossy* yaitu:

- 1) Metode *lossless*

Pada teknik ini tidak ada kehilangan data atau informasi. Jika data dikompres secara *lossless*, data asli dapat direkonstruksi kembali sama persis dari data yang telah dikompresi, dengan kata lain data asli tetap sama sebelum dan sesudah kompresi. Secara umum teknik *lossless* digunakan untuk penerapan yang tidak bisa mentoleransi setiap perbedaan antara data asli dan data yang telah direkonstruksi. Data

berbentuk tulisan misalnya *file* teks, harus dikompresi menggunakan teknik *lossless*, karena kehilangan sebuah karakter saja dapat mengakibatkan kesalahpahaman. *Lossless compression* disebut juga dengan *reversible compression* karena data asli bisa dikembalikan dengan sempurna. Akan tetapi rasio kompresinya sangat rendah, misalnya pada data teks, gambar seperti GIF dan PNG. Contoh metode ini adalah *Shannon-Fano Coding*, *Run Length Encoding*, *Arithmetic Coding* dan lain sebagainya.

2) Metode *lossy*

Pada teknik ini akan terjadi kehilangan sebagian informasi atau data. Data yang telah dikompresi dengan teknik ini secara umum tidak bisa direkonstruksi sama persis dari data aslinya. Di dalam banyak penerapan, rekonstruksi yang tepat bukan suatu masalah. Sebagai contoh, ketika sebuah *sample* suara ditransmisikan, nilai eksak dari setiap *sample* suara belum tentu diperlukan. Tergantung pada yang memerlukan kualitas suara yang direkonstruksi, sehingga banyaknya jumlah informasi yang hilang di sekitar nilai dari setiap *sample* dapat ditoleransi.

Biasanya teknik ini membuang bagian-bagian data yang sebenarnya tidak begitu berguna, tidak begitu dirasakan, tidak begitu dilihat sehingga manusia masih beranggapan bahwa data tersebut masih bisa digunakan walaupun sudah dikompresi. Misalnya pada gambar dan MP3. Contoh metode ini adalah *color reduction*, *chroma subsampling*, dan *transform coding* seperti DWT, DCT, FFT dan lain-lain. *Lossy compression* disebut juga *irreversible compression* karena data asli mustahil untuk dikembalikan seperti semula [1]. Kelebihan teknik ini adalah rasio kompresi yang tinggi dibanding metode *lossless*.

Keuntungan dari metode *lossy* atas *lossless* adalah dalam beberapa kasus metode *lossy* menghasilkan file kompresi yang lebih kecil dibandingkan dengan metode *lossless*. Metode *lossy* sering digunakan untuk mengkompresi suara, gambar dan *video* karena data tersebut dimaksudkan kepada *human interpretation* dimana pikiran dapat dengan mudah “mengisi bagian-bagian yang kosong” atau melihat kesalahan yang sangat kecil atau inkonsistensi. Sedangkan *lossless* digunakan untuk mengkompresi data untuk diterima ditujukan dalam kondisi asli seperti dokumen teks. *Lossy* akan mengalami *generation loss* pada data sedangkan pada *lossless* tidak terjadi karena data yang hasil dekompresi sama dengan data asli.

H. Dekompresi

Sebuah data yang sudah dimampatkan tentunya harus dapat dikembalikan lagi ke bentuk aslinya, prinsip ini dinamakan penimampatan. Untuk dapat merubah data yang

termampatkan diperlukan cara yang berbeda seperti pada waktu proses kompres dilaksanakan. Jadi pada saat penimampatkan catatan *header* yang berupa *byte-byte* tersebut terdapat catatan isi mengenai isi dari file tersebut [5].

Catatan *header* akan menuliskan kembali mengenai isi dari file tersebut, jadi isi dari file sudah tertulis oleh catatan *header* sehingga hanya tinggal menuliskan kembali pada saat proses dekompres. Proses dekompresi sempurna (kembali ke bentuk aslinya) [11]. Secara umum proses kompresi dan dekompresi dapat dilihat pada Gambar 1.



Gambar 1 Alur kompresi *Lossy*

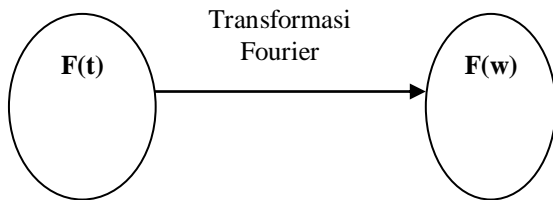
I. Transformasi Fourier

Proses penting dalam *Digital Signal Processing* (DSP) adalah menganalisis suatu sinyal input maupun output untuk mengetahui karakteristik sistem fisis tertentu. Proses analisis dan sintesis dalam domain waktu memerlukan analisis cukup panjang dengan melibatkan turunan dari fungsi, yang dapat menimbulkan ketidaktepatan hasil analisis. Analisis dan sintesis sinyal akan lebih mudah dilakukan pada domain frekuensi, karena besaran yang paling menentukan suatu sinyal adalah frekuensi. Oleh karena itu, untuk dapat bekerja pada domain frekuensi dibutuhkan suatu formulasi yang tepat sehingga proses manipulasi sinyal sesuai dengan kenyataan. Salah satu teknik untuk menganalisis sinyal adalah mentransformasikan (alih bentuk) sinyal yang semula analog menjadi diskret dalam domain waktu, dan kemudian diubah ke dalam domain frekuensi. Hal ini dilakukan dengan mengalikan sinyal diskret dengan suatu fungsi kernel [10].

Transformasi Fourier dua dimensi dipergunakan untuk menghitung spektrum energi citra pada domain frekuensi. Perbaikan penampilan citra dan koreksi linear dapat dilakukan dengan filter komponen-komponen frekuensi. Pilihan jenis filter tergantung pada frekuensi guling dari peralatan sistem optik dan faktor linear yang menyebabkan kualitas citra mengalami degradasi. Setelah itu transformasi Fourier balik pada komponen-komponen frekuensi akan mengembalikan citra terkoreksi ke domain spasial. Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekuensi.

FFT 2D adalah DFT 2D dengan teknik perhitungan yang cepat dengan memanfaatkan sifat periodikal dari transformasi Fourier. Transformasi Fourier dua dimensi dipergunakan untuk menghitung spektrum energi citra pada domain frekuensi. Perbaikan penampilan citra dan koreksi linear dapat dilakukan dengan filter komponen-komponen frekuensi. Pilihan jenis filter tergantung pada frekuensi guling (1 gelombang) dari peralatan sistem optik dan faktor linear yang menyebabkan

kualitas citra mengalami degradasi (penurunan mutu kualitas warna). Setelah itu transformasi Fourier balik pada komponen-komponen frekuensi akan mengembalikan citra terkoreksi ke domain spasial. Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekuensi.



Gambar 2 Transformasi Fourier

dimana :

F(t) adalah fungsi dalam domain waktu

F(w) adalah fungsi dalam domain frekuensi

Pada dasarnya citra adalah fungsi 2D sehingga transformasi Fourier yang digunakan adalah transformasi Fourier 2D. Pada beberapa citra hasil yang didapat dari transformasi Fourier adalah titik-titik putih pada sebelah pojok kiri atas atau dipojok kanan bawah pada gambar *magnitude*-nya, ini disebabkan terjadinya nilai dominan pada suatu frekuensi. Hal ini menyebabkan kesulitan dalam melakukan analisis terhadap frekuensi suatu citra. Untuk mengurangi dominansi tersebut digunakan fungsi logaritma yang dikenakan pada *magnitude*.

J. Discrete Fourier Transform

Discrete Fourier Transform adalah metode untuk mentransformasi sinyal dari domain waktu ke domain frekuensi.[2].Persamaan DFT dapat ditulis sebagai berikut:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Dimana:

N = banyak piksel citra

x = nilai piksel

e = *natural number* (2.718281828459045...)

k = indeks dalam domain frekuensi = 0, 1, ..., N-1

n = indeks dalam domain waktu = 0, 1, ..., N-1,

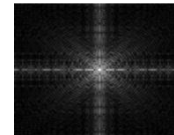
j = konstanta *fourier*

Sedangkan *Invers Discrete Fourier Transform (IDFT)* diberikan seperti persamaan:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}$$

Dalam pengolahan citra digital lebih dikenal sebagai 2D-DFT (*2 Dimensional Discrete Fourier Transform*) karena citra suatu bentuk dua dimensi. Sehingga 2D-DFT dalam

pengolahan citra digital memindahkan suatu informasi citra dari domain spasial kedalam domain frekuensi, yaitu dengan merepresentasikan citra spasial sebagai *magnitude* dan *phase*. *Magnitude* merepresentasikan seberapa banyak komponen frekuensi dalam citra tersebut. Sedangkan *phase* merepresentasikan letak dimana frekuensi tersebut dalam citra tersebut. Bentuk *phase* dalam *image processing* dapat dilihat pada Gambar 2.6.



Gambar 3 DFT pada *Image Processing*

K. Fast Fourier Transform

Pada penelitian Prima [8]. Penelitian ini bertujuan untuk mendapatkan hasil gambar sesuai dengan kebutuhan pengguna proses perbaikan kualitas gambar yang berbasiskan Transformasi Fourier. Jika terdapat suatu gambar yang belum memenuhi keinginan dari pengguna, maka dilakukanlah suatu perbaikan kualitas gambar tersebut yang berbasiskan transformasi Fourier.

Citra pada umumnya digambarkan dalam dimensi spasial antara amplitude dengan posisi spasial dua dimensi. Dalam operasi analisis citra, seringkali dilakukan operasi konvolusi pada matriks (citra) yang berdimensi besar. Hal ini menjadi salah satu penghambat karena ketika dilakukan operasi konvolusi, maka dibutuhkan resource computer yang sangat besar [6].

Algoritma DFT standar tidak bagus jika digunakan pada citra berukuran besar, karena proses komputasinya memakan waktu yang lama. Untuk mempercepat proses transformasi, digunakan algoritma FFT. Algoritma ini sangat cepat mengerjakan matriks yang berukuran besar. Perbedaan kompleksitas waktu DFT dengan FFT adalah $O(N^2)$: $O(N^2 \log N)$. Dengan FFT proses komputasi biasa direduksi dari N^2 menjadi $N \log 2N$. Misalnya dengan menggunakan DFT kita akan melakukan transformasi sebanyak $N=1024$ titik, maka kita memerlukan perkalian sebanyak $N^2 = 1.048.567$. Sedangkan dengan menggunakan FFT perkalian yang diperlukan sebanyak $N \log 2N = 5120$ perkalian.

Prinsip dasar FFT adalah menguraikan penghitungan N -titik DFT menjadi penghitungan DFT dengan ukuran yang lebih kecil dan memanfaatkan periodisitas dan simetri dari bilangan kompleks W_N^{kn} .

Dimana:

W = fungsi windows

kn = nilai indeks waktu

N = nilai piksel

Pada contoh proses *filtering* dilakukan melalui *masking* terhadap komponen frekuensi yang ditentukan. Agar tercipta karakteristik high-pass filter, maka komponen frekuensi rendah (koefisien frekuensi yang berada pada bagian pojok dari citra hasil FFT) di-*masking* menggunakan nilai 0. Demikian pula sebaliknya untuk memunculkan sifat *low-pass filter*, komponen frekuensi tinggi (koefisien frekuensi yang berada pada bagian tengah dari citra hasil FFT) dibuat menjadi 0 melalui perkalian dengan *mask low-pass* [3].

III. ANALISIS DAN PERANCANGAN

A. Analisis

Dalam penelitian ini dilakukan analisis kompresi citra dengan algoritma transformasi FFT 2 dimensi. Transformasi Fourier 2 dimensi dipergunakan untuk menghitung spektrum energi citra pada domain frekuensi yang menghasilkan koefisien transformasi. Untuk melakukan kompresi, nilai piksel yang memiliki frekuensi tinggi dikuantisasi karena kurang berpengaruh terhadap representasi citra pada domain spatialnya. Untuk mengembalikan citra terkoreksi ke domain spatialnya, lakukan transformasi Fast Fourier Balik (IFFT) pada komponen-komponen frekuensinya.

- 1) Menghitung nilai RGB citra.
 Nilai R = $c \text{ Mod } 256$
 Nilai G = $(c \text{ and } 65280) / 256$
 Nilai B = $(c \text{ and } 16711680) / 65280$
 Dimana :

c adalah nilai piksel citra

- 2) Menghitung Nilai Grayscale Citra

$$f_0(x,y) = \left(\frac{f^R(x,y) + f^G(x,y) + f^B(x,y)}{3} \right)$$

Dimana :

$f^R(x,y)$ = nilai piksel merah

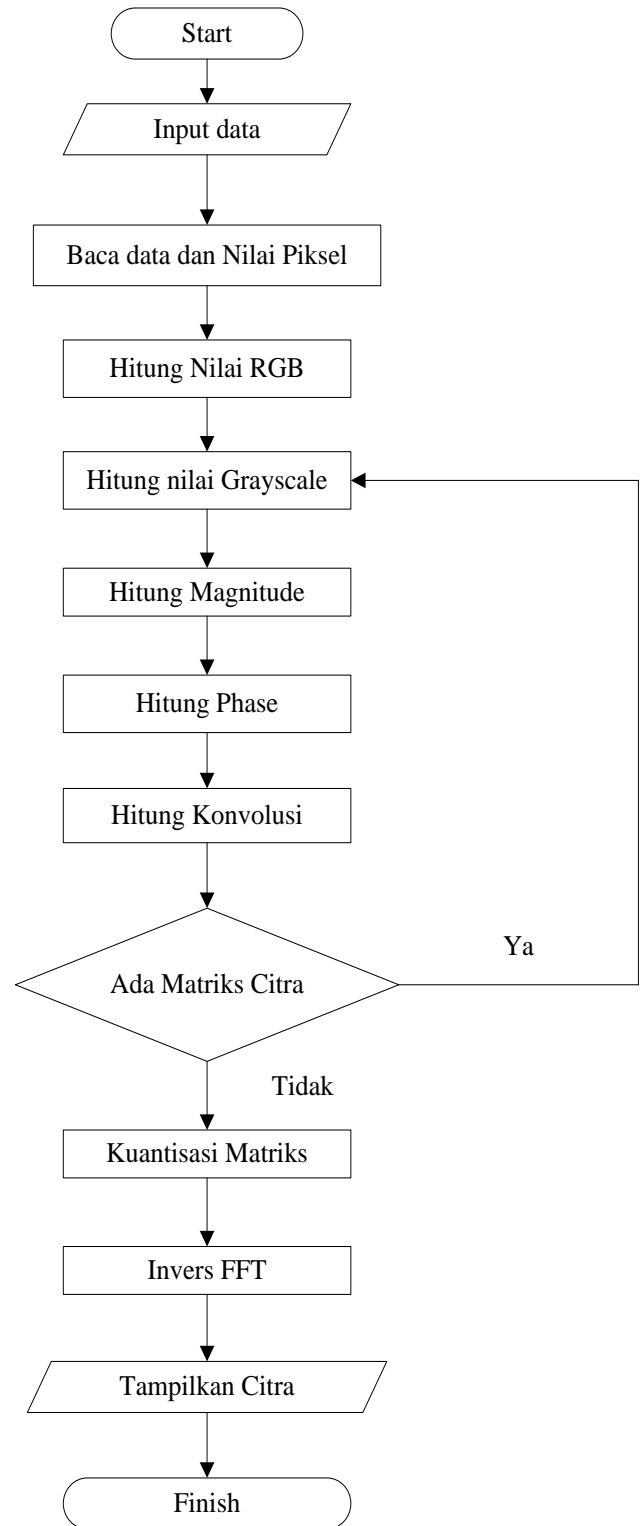
$f^G(x,y)$ = nilai piksel hijau

$f^B(x,y)$ = nilai piksel biru

- 3) Kompresi Citra

$$F(x,y) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$
- 4) Menghitung Nilai Magnitude yaitu mendapatkan Nilai Real.
- 5) Menghitung Nilai Phase yaitu mendapatkan Nilai Imajiner
- 6) Menghitung Konvolusi yaitu melakukan perkalian antar matriks.
- 7) Menghitung Kuantisasi Matriks yaitu Proses Threshold dengan cara jika nilai dibawah 0 menjadi 0 dan jika nilai diatas 255 menjadi 255.
- 8) Menghitung Nilai Invers FFT yaitu Nilai Normalisasi Grayscale Citra.

B. Flowchart



Gambar 4 Flowchart Kompresi FFT

IV. IMPLEMENTASI



Gambar 5 Tampilan Menu Utama

HASIL PENGUJIAN PROGRAM
Tanggal: 11-July-2012

No.	Nama File Asli	Ext	Ukuran Awal	Nama File Hasil Kompresi	Ext	Ukuran Akhir	Rasio
Format : bmp							
1	Wave.bmp	bmp	5,760,054.00	wavehsi.bmp	bmp	118,999.00	97.93
2	pesawat.bmp	bmp	294,966.00	pesawathsi.bmp	bmp	8,937.00	96.97
3	Pohon.bmp	bmp	85,258.00	pohonhsi.bmp	bmp	34,873.00	59.10
4	jujuk.BMP	BMP	8,937.00	jujukhsi.bmp	bmp	8,383.00	6.20
						Rata Rata Rasio:	65.05
Format : JPG							
5	jembatan.JPG	JPG	141,243.00	jembatanshsi.jpg	jpg	29,591.00	79.05
6	burung.jpg	jpg	155,628.00	burunghsi.jpg	jpg	30,727.00	80.26
7	cat.jpg	jpg	93,676.00	cathsi.jpg	jpg	18,388.00	80.37
8	frog.jpg	jpg	299,679.00	froghsi.JPG	JPG	83,555.00	72.12
						Rata Rata Rasio:	38.97

Gambar 6 Tampilan Hasil Pengujian

V. KESIMPULAN

Berdasarkan pengujian yang dilakukan atas Analisis Dan Perancangan Perangkat Lunak Kompresi Citra Menggunakan Algoritma *Fast Fourier Transform* (FFT), maka diperoleh hasil sebagai berikut :

- 1) Hasil kompresi rata-rata untuk citra yang berformat JPG adalah 41,11 %
- 2) Hasil kompresi rata-rata untuk citra yang berformat BMP adalah 87,02 %

Dari data diatas dapat ditarik kesimpulan bahwa:

- 1) Aplikasi ini dapat melakukan kompresi citra dengan format JPG maupun BMP.
- 2) Ukuran file citra sebelum dan sesudah dikompresi mengalami perubahan menjadi kecil yang berarti bahwa ada data yang hilang selama proses kompresi tetapi secara kasat mata tidak terlihat oleh mata manusia.
- 3) Rasio hasil kompresi rata-rata untuk citra yang berformat BMP lebih tinggi daripada rasio hasil kompresi rata-rata untuk citra yang berformat JPG yang berarti bahwa rasio hasil kompresi rata-rata citra berformat BMP paling baik dikompresi dengan algoritma *Fast Fourier Transform* (FFT).

DAFTAR PUSTAKA

- [1] Abdullah, Muhammad Maulana. 2009. *Kompresi String Menggunakan Algoritma LZW dan HUFFMAN*. Bandung, Indonesia : Institut Teknologi Bandung.
- [2] Aini, Ossi., Hadi, M.Z.S., dan Assidiq, M.H. 2008. *Analisis Penggunaan Filter Pada Sistem Pengenalan Plat Nomor Menggunakan Phase Only Correlation (POC)*. Surabaya, Indonesia : Politeknik Negeri Surabaya.
- [3] Basuki, Achmad. Jozua F Palandi. Fatchurrochman. 2005. *Pengolahan Citra Digital Menggunakan Visual Basic*. Jakarta: Penerbit Graha Ilmu.
- [4] Gonzalez, Rafael. 2002. *Digital Image Processing*. USA: Addison-Wesley Publishing Co : University of Tennessee.
- [5] Gozali, Ferrianto, Mervyn. 2004. *Analisis Perbandingan Kompresi Data Dengan Teknik Arithmetic Coding Dan Run Length Encoding*. Jakarta,Indonesia : Universitas Trisakti.
- [6] Gusmayuda, Rizki Ayus. 2008. *Steganografi Pada Media Video Digital Dengan Menggunakan Metode FFT (Fast Fourier Transform) dan LSB (Least Significant Bit)*. Skripsi. Bandung,Indonesia : Universitas Komputer Indonesia.
- [7] Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Penerbit Informatika.
- [8] Prima,Rani Suci. 2010. *Perbaikan Kualitas Citra Berbasis Transformasi Fourier*. Skripsi. Medan, Indonesia : Universitas Sumatera Utara
- [9] Putra, Darma. 2010. *Teori Pengolahan Citra Digital*. Yogyakarta. Penerbit : ANDI.
- [10] Riyanto, Sugeng Atal. 2009. *Algoritma Fast Fourier Transform (FFT) Decimation In Time (Dit) dengan Resolusi 1/10 Hertz*. Yogyakarta, Indonesia : Universitas Negeri Yogyakarta.
- [11] Santi, Ririani. 2010. *Perancangan Perangkat Lunak Kompresi Citra Dengan Menggunakan Algoritma Run Length Encoding*. Skripsi.Medan, Indonesia : Universitas Sumatera Utara.
- [12] Sutoyo, T. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.