BAB III

PERANCANGAN SISTEM

3.1 Metode Analisis

Analisis suatu sistem merupakan salah satu proses yang harus dilakukan dalam perancangan dan implementasi suatu perangkat lunak. Hal ini berguna untuk mengidentifikasi dan mengevaluasi permasalahan kesempatan, hambatan yang terjadi dan kebutuhan-kebutuhan yang diinginkan sehingga dapat diusulkan perbaikan-perbaikannya.

Metode analisis yang diperlukan sebagai bahan acuan adalah analisa dengan pendekatan terstrukur *(Structured Approach)* yang dilengkapi dengan alat (*tool*) dan teknik *(techniques)* yang membutuhkan pengembangan sistem, sehingga hasil dari analisa tersebut dapat dikembangan kedalam sebuah bentuk yang dapat diimplementasikan strukturnya dan dapat didefinisikan secara detail dan terperinci.

Sistem yang akan dirancang dan digunakan dalam prakiraan beban listrik dengan menggunakan bagan alir *(flowchart)* sebagai pernyataan algoritma pengujian dan pelatihan. Bagan alir *(flowchart)* digunakan untuk mengembangkan dan menggambarkan langkah-langkah algoritma dalam menentukan proses perhitungan pelatihan dan pengujian.

3.2 Analisis Sistem

Pada tahap analisis adalah suatu kegiatan untuk menentukan spesifikasi perangkat lunak yang diinginkan, sehingga terjadi komunikasi antara pembuat perangkat lunak dengan pemakai perangkat lunak. Komunikasi tersebut meliputi spesifikasi kemampuan atau fasilitas yang diinginkan, bentuk masukkan, proses-proses pengolahan data dan informasi yang diinginkan.

3.3 Analisis Kebutuhan

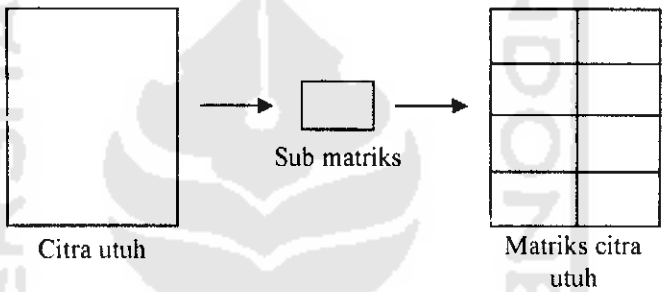
Analisis kebutuhan merupakan analisis yang dibutuhkan dalam membuat aplikasi prakiraan beban jangka pendek yang berupa *input*, *output*, fungsi-fungsi atau modus operasi yang dibutuhkan dan antar muka yang diinginkan.

3.3.1 Algoritma Block Dividing

- Simpan citra ke memori dengan membuat sebuah objek turunan dari komponen TImage, kemudian citra Bitmap *input* di-*copy* ke objek tersebut.
- Hitung jumlah sub matriks dari citra tersebut dengan cara melakukan pembagian bulat terhadap lebar dan tinggi citra dibagi dengan ukuran sub matriks yaitu 4 atau 8. Hitung pula sisa pembagiannya, jika ada sisa pembagian tinggi maka jumlah sub matriks secara menurun ditambah satu, demikian juga jika ada sisa pembagian lebar maka jumlah sub matriks secara mendatar ditambah satu.
- Salin nilai-nilai R, G, dan B dari tiap-tiap piksel ke dalam suatu variabel *array of record* yang mewakili satu sub matriks, masing-masing *record*

mewakili satu piksel. Jika isi sub matriks tidak memenuhi ukuran sub matriks maka dilakukan *padding*, yaitu mengisi dengan nilai piksel terakhir (piksel citra paling tepi).

- Salin sub matriks ke *array* yang menampung semua sub matriks, sehingga didapat sebuah *array* yang merupakan matriks citra secara utuh.

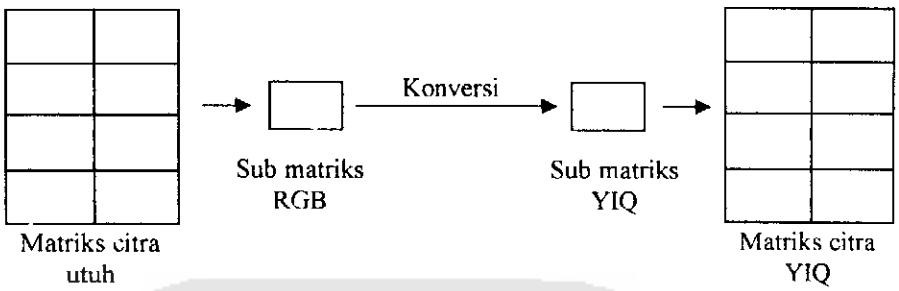


Gambar 3.1 Algoritma *Block Dividing*

3.3.2 Algoritma konversi RGB ke YIQ

Seperti dijelaskan pada Bab 2, untuk konversi gambar ke YIQ pada sebuah piksel adalah dengan menghitung nilai intensitas dari warna RGB sebuah piksel dan mengganti setiap nilai warna RGB tersebut dengan nilai YIQ. Algoritma konversi dari gambar berwarna ke *grayscale* adalah :

- Baca setiap *record* dalam sub matriks dan hitung nilai YIQ-nya.
- Ganti nilai *record* yang dihitung tersebut dengan nilai Y, I dan Q.
- Lakukan langkah 1 dan 2 pada semua *record* yang ada pada sub matriks.
- Lakukan langkah 1, 2, dan 3 pada semua sub matriks.

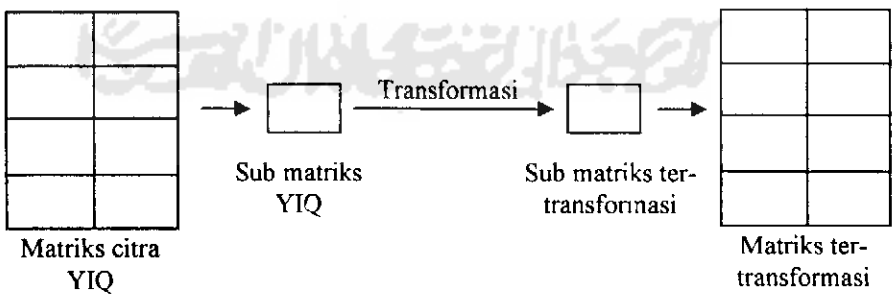


Gambar 3.2 Konversi RGB ke YIQ

Dalam proses dekompresi, dipakai struktur algoritma yang sama untuk mengkonversi nilai YIQ ke RGB.

3.3.3 Algoritma Transformasi Metode Hadamard

Untuk transformasi dengan metode Hadamard, cara yang digunakan sama yaitu dengan mengambil satu sub matriks dari matriks citra utuh, kemudian mentransformasi sub matriks tersebut dan mengembalikannya ke matriks citra.



Gambar 3.3 Transformasi Hadamard

3.3.4 Algoritma penyimpanan ke File

Setelah semua proses diatas selesai dilakukan maka matriks citra perlu disimpan ke file sementara (*temporary*) terlebih dahulu agar dapat dikodekan dengan program kompresi Huffman.

Algoritma penyimpanan ke file adalah sebagai berikut:

- Siapkan file output
- Ambil satu sub matriks
- Hitung *offset*, yaitu letak penyimpanan data di dalam file
- Isi buffer2, yaitu array yang berisi data 2 byte dan setelah itu buffer1 yang berisi data 1 byte.
- Arahkan *pointer* file ke posisi *offset*
- Tulis buffer2 ke file diikuti oleh buffer1
- Ulangi untuk sub-sub matriks berikutnya

Data disimpan dengan format *non-interleaved*, yaitu semua komponen warna digabungkan, sehingga susunannya berurutan seperti berikut:

Y_1	I_1	Q_1	Y_2	I_2	Q_2	Y_3	I_3	Q_3	Y_4	I_4	Q_4	Y_5	I_5	Q_5

Setelah data disimpan kedalam file temporary, maka selanjutnja data dikodekan menggunakan program kompresi Huffman dengan beberapa keterangan tambahan pada header file. Daftar keterangan ini tertera pada Tabel 3.1 di bawah.

Tabel 3.1 Keterangan tambahan pada file hasil kompresi

Offset	Jenis keterangan	Ukuran (byte)
1	Ukuran *Block*	1
2	Kompresi	1
3	Faktor Kuantisasi	8
11	Lebar citra	2
13	Tinggi citra	2

3.4 Analisis Kebutuhan Antar Muka *(Interface)*

Kebutuhan terhadap antar muka (*interface*) yang akan dibuat sebaiknya harus bersifat *user friendly* (ramah dengan pengguna / mudah dalam pengoperasiannya) dan *interaktif* artinya *user* (sebagai pengguna akhir) dapat menggunakan aplikasi yang dibuat semudah mungkin sehingga *user* dalam mengoperasikannya mudah.

Ramah disini juga dapat diartikan apabila pengguna mengalami kesalahan pada proses pemasukan data yang akan diproses, maka aplikasi tersebut tidak langsung terjadi *error* atau *hang* malah sebaliknya aplikasi tersebut langsung mengembalikan kepada proses masukan data tersebut. Ramah pengguna disini juga dapat diartikan, dengan sedikit belajar (*trainning*) memahami aplikasi yang ada, pengguna langsung bisa menjalankan aplikasi.

Suatu aplikasi dirancang agar berguna untuk meringankan pengguna dalam segala tugasnya, maka aplikasi itu juga dirancang memperbaiki kesalahan dan memperbaiki kesalahan yang dilakukan oleh pengguna dengan melakukan umpan balik (*feedback*) pada sistem aplikasi itu sendiri.

3.5 Analisis Kebutuhan Perangkat Keras *(Hardware)*

Untuk membuat aplikasi ini dibutuhkan spesifikasi perangkat keras minimal sebagai berikut :

1. PC *(Personal Computer)* dengan kecepatan minimal AMD K6-II 350 MHz dengan RAM minimal 32 MB atau diatasnya.
2. Monitor SVGA dengan resolusi minimal 800 x 600

3.6 Analisis Kebutuhan Perangkat Lunak *(Software)*

Perangkat lunak *(software)* yang digunakan dalam perancangan aplikasi ini harus memiliki kemampuan yang baik dan beroperasi dengan stabil dalam waktu yang lama.

Perangkat lunak yang dipergunakan antara lain :

1. Sistem Operasi minimal : Microsoft Windows 98
2. Borland Delphi 6.0

3.7 Analisis Kebutuhan Proses

Analisis kebutuhan merupakan analisis yang dibutuhkan dalam membuat aplikasi.

1. Masukan (Input)
    1. Pemilihan file yang akan di kompres..
    2. Tampilan file terpilih pada proses kompresi.
    3. Faktor Kuantisasi.
    4. Jenis matrik yang akan digunakan.
2. Keluaran (Output)
    1. Penyimpanan hasil keluaran proses kompresi.
    2. Tampilan file hasil proses kompresi.
    3. Running time proses yang dilakukan
3. Proses
    1. Blok Dividing
    2. Metode Hadamard

3. Kuantitasi

4. Pengkodean dengan metode Huffman