

BAB III PERANCANGAN SISTEM

1.1 Analisa Permasalahan

Serialisasi XML merupakan proses encoding sebuah obyek atau dokumen dalam bentuk binary (dalam tugas akhir ini berupa dokumen gambar ke dalam sebuah format dokumen XML yang well formed. Dari hasil proses serialisasi tersebut, maka terbentuk sebuah file XML baru dengan nama dan lokasi file sebagaimana ditentukan oleh pengguna saat dikonfirmasi oleh aplikasi. File XML yang telah dihasilkan akan memiliki format encoding *American Standard Code for Information Interchange* (ASCII) normal yang berasal dari class `Encoding.ASCII`. Hal ini dimaksudkan agar file XML dapat lebih bebas dimanfaatkan untuk pengembangan selanjutnya.

Serialisasi diharapkan akan memunculkan konsep konversi baru dokumen gambar, yang tidak lagi dari sudut pandang matriks layaknya pengolahan citra, ataupun sebagai sebuah dokumen biner secara umum. Tetapi dengan adanya serialisasi, dokumen gambar akan membentuk sebuah obyek baru dan tersimpan dalam bentuk dokumen XML yang divalidasi sebagai sebuah dokumen XML yang well formed.

1.2 Serialisasi Dokumen Gambar

Proses serialisasi dokumen gambar dimulai dari pemanggilan dokumen gambar ke bentuk blok memori. Serialisasi dokumen gambar yang outputnya akan dijadikan format XML, sebelumnya akan dimasukkan kedalam sebuah class tersendiri yang memiliki properti pengaturan agar dokumen gambar saat dideserialisasi mampu kembali ke bentuk semula dengan sempurna.

Tiap dokumen gambar akan dimasukkan dalam suatu class obyek. Kumpulan dari class obyek tersebut akan dimasukkan dalam suatu *array*. Saat proses serialisasi selesai dilakukan, maka hasil serialisasi akan diberi nama sesuai dengan permintaan dari pengguna untuk diproses selanjutnya dalam proses kompresi.



Gambar 3.1. Flowchart proses serialisasi

1.3 Kompresi Dokumen XML

Hasil dari proses serialisasi yang berupa dokumen XML selanjutnya akan dikompresi dengan menggunakan algoritma LZW. Dokumen XML dalam proses kompresi pada dasarnya dipandang sebagai sebuah dokumen teks normal atau *plain text* yang tidak memiliki *header* file tertentu seperti halnya dokumen berjenis *executable* ataupun dokumen yang berasal dari sebuah database.

Dalam kompresi yang dilakukan nanti akan menggunakan algoritma LZW yang diimplementasikan dalam sebuah DLL library yang dibuat menggunakan bahasa

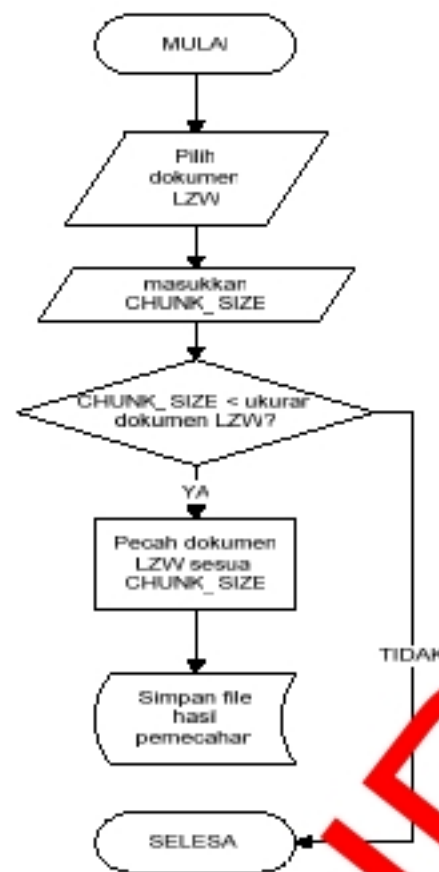
C.



Gambar 3.2. Flowchart proses kompresi

1.4 Pemecahan File

Hasil dari proses kompresi yang telah dilakukan sebelumnya berupa suatu file dengan ekstensi LZW. Selanjutnya akan dilakukan proses pemecahan file LZW menjadi beberapa file dengan ukuran yang lebih kecil sesuai dengan yang diinginkan oleh pengguna.



Gambar 3.3. Flowchart proses split

1.5 Penggabungan File

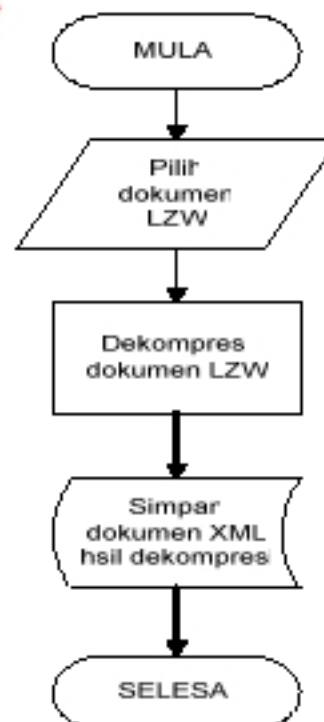
Proses penggabungan file dilakukan jika pengguna telah memiliki semua file-file yang dihasilkan dari proses pemecahan file pada aplikasi pertama. Proses penggabungan file ini menghasilkan file LZW sama seperti file LZW yang digunakan sebagai input pada proses pemecahan file pada aplikasi pertama, sehingga nantinya dari file ini dapat dilakukan proses selanjutnya yaitu proses dekompresi.



Gambar 3.4. Flowchart proses merge file

1.6 Dekompresi Dokumen

Proses dekompresi dilakukan setelah proses penggabungan file selesai dan menghasilkan file LZW yang sempurna. Hasil dari proses dekompresi ini adalah file XML yang sama dengan file XML yang dihasilkan oleh proses serialisasi yang dilakukan pada aplikasi pertama.



Gambar 3.5. Flowchart proses dekompresi

1.7 Deserialisasi Dokumen

Proses deserialisasi yang merupakan lanjutan dari proses dekompresi, dilakukan jika proses dekompresi telah berjalan dengan baik dan benar sesuai dengan rencana. Proses deserialisasi dilakukan untuk mendapatkan file-file gambar sesuai dengan file-file gambar yang digunakan sebagai input untuk proses serialisasi pada aplikasi pertama sehingga pengguna dapat menampilkan serta menyimpan file-file gambar yang telah didapatkan.



Gambar 3.6. Flowchart proses deserialisasi

1.8 Perancangan Antarmuka Sistem

Dalam melakukan perancangan antarmuka ini diharapkan mampu memenuhi aspek-aspek sebagai berikut :

1. Mudah dibaca dan dimengerti oleh pengguna.
2. Sederhana.
3. Mengurangi terjadinya kesalahan dari pengguna.

Untuk memberikan gambaran awal tentang antar muka aplikasi serta menjaga konsistensi dari desain aplikasi, maka perlu dibuat suatu rancangan masukan dan keluaran aplikasi.

1.8.1 Aplikasi Administrator

Aplikasi ini digunakan oleh administrator website untuk melakukan serialisasi, kompresi dan split file gambar majemuk. Selanjutnya hasil dari proses split akan di upload oleh administrator website.

A. Form Serialisasi

Form serialisasi digunakan untuk melakukan serialisasi dari beberapa file gambar yang dipilih menjadi suatu file XML yang ditentukan oleh pengguna.



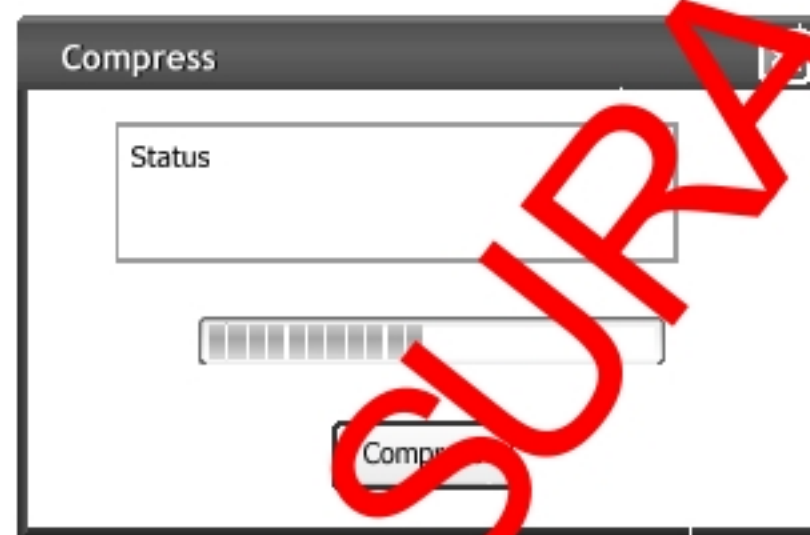
Gambar 3.7. Rancangan antarmuka form serialisasi

1. *View Image* : Menampilkan gambar yang akan diserialisasi.

2. Open : Tombol ini digunakan untuk membuka file-file gambar yang akan diserialisasi
3. Proses : Tombol ini digunakan untuk menjalankan proses serialisasi

B. Form Kompresi

Form kompresi digunakan untuk melakukan kompresi pada file XML yang dihasilkan dari proses serialisasi yang telah dilakukan pada Form Serialisasi



Gambar 3.8 Rancangan antarmuka form kompresi

1. Status : Menampilkan status dari proses kompresi.
2. Compress : Menjalankan proses kompresi.

C. Form Split

Form split ini digunakan untuk memecah file LZW hasil keluaran dari proses kompresi menjadi beberapa bagian dimana ukuran dari setiap file dapat ditentukan oleh pengguna.

Gambar 3.9. Rancangan antarmuka form kompresi

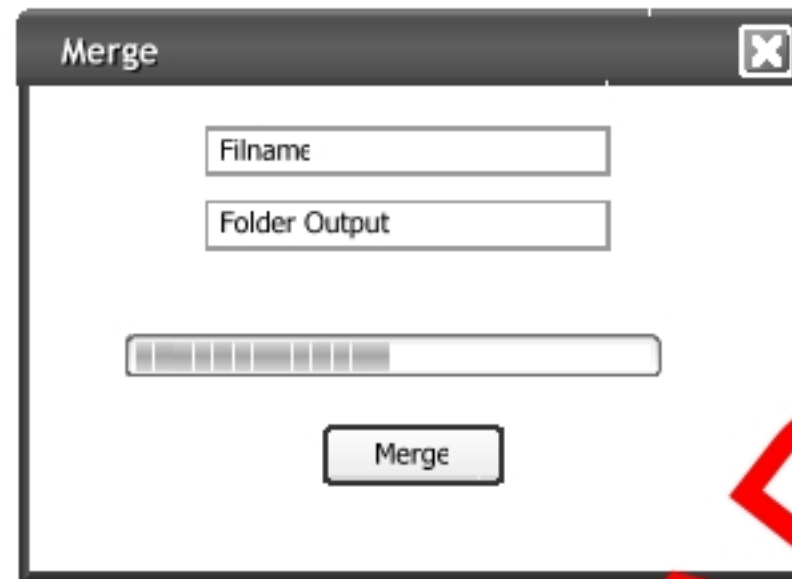
1. Filename : Inputan nama file yang akan di split.
2. Folder output : Inputan letak file-file hasil proses split akan diletakkan.
3. Chunk Size : Inputan ukuran file-file hasil proses split.
4. Split : Menjalankan proses split

1.8.2 Aplikasi Pengguna

Aplikasi ini digunakan oleh pelanggan website yang telah melakukan pembelian gambar untuk melakukan deserialisasi, dekompresi dan merge file gambar majemuk sehingga pengguna dapat menampilkan dan menyimpan file gambar yang telah dibelinya.

A. Form Merge

Form merge ini digunakan untuk menggabungkan kembali file-file hasil dari proses split yang dilakukan pada aplikasi administrator. Proses ini menghasilkan file LZV yang nantinya akan digunakan sebagai input pada proses dekompresi.

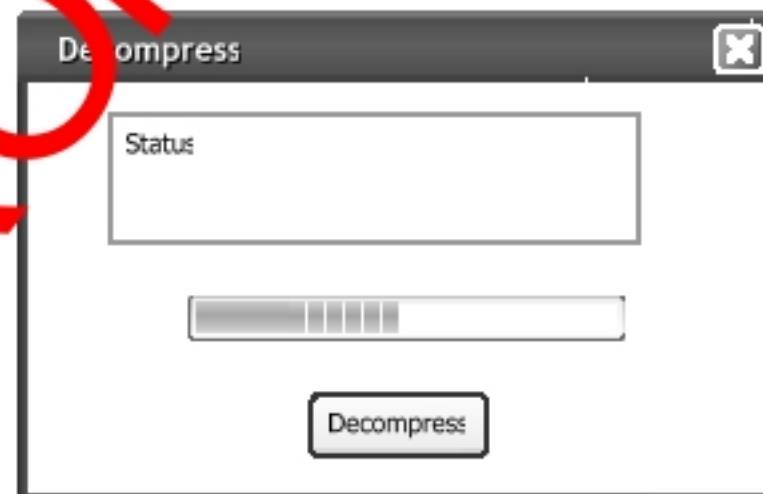


Gambar 3.10. Rancangan antarmuka form merge

1. Filename : Nama file yang akan di-merge
2. Folder output : Folder letak file-file hasil proses merge akan diletakkan.
3. Merge : Menjalankan proses merge

B. Form Dekompresi

Form dekompresi untuk melakukan dekompresi pada file LZW hasil dari proses merge. Hasil dari proses ini adalah file XML yang nantinya akan digunakan untuk proses deserialisasi.



Gambar 3.11. Rancangan antarmuka form dekompresi

1. Status : Menampilkan status dari proses dekompresi.
2. Decompress : Menjalankan proses dekompresi.

C. Form Deserialisasi

Form deserialisasi untuk melakukan deserialisasi dari file XML yang dihasilkan dari proses dekompresi menjadi file-file gambar yang dapat dilihat dan disimpan oleh pengguna.



Gambar 3.12. Tampilan antarmuka form deserialisasi

1. View Image : Menampilkan gambar yang telah dideserialisasi.
2. Deserialize : Tombol ini digunakan untuk membuka file LZW dan melakukan proses deserialisasi.
3. Save : Tombol ini digunakan untuk menyimpan file-file gambar.