

RANCANG BANGUN APLIKASI KOMPRESI DAN DEKOMPRESI PADA CITRA DIGITAL MENGGUNAKAN METODE *HUFFMAN*

Kharisma Mahesa¹, Karpen²

*Program Studi Teknik Informatika, STMIK Amik Riau, Pekanbaru
Jl. Purwodadi Indah Km. 10 Panam Pekanbaru-Riau
E-mail : kharismamahesa@stmik-amik-riau.ac.id*

Abstract

The rapid development of information technology and currently it's possible to send information quickly. Speed transmission of information is certainly very dependent on the size of the capacity of such information. The large size of the capacity of information could result in the slow process of delivery. Beside that, the size of a large information capacity would require a very large storage space. The solution to this problem is to compress the information before sending in order to measure the capacity of such information becomes smaller. While reverting to forms the initial compression is to perform decompression technique. Some digital image compression techniques have been developed as block-coding, encoding and more. Digital image compression techniques continue to be developed with the aim to compress the image data as small as possible, but at the moment none of the decompression image data is lost. Huffman method is one method that is able to compress and decompress the digital image. Image compression is the process to minimize the number of bits that represents an image so that the image data capacity becomes smaller. Huffman method is a method of digital image compression is lossless compression method that is by doing the encoding in the form of bits to represent character data. Each type of characters contained in an image is calculated and sort by the fewest amount to a large amount in the form of binary trees and code for each character. The existing data will be replaced with the code bits based on a binary tree. Data that has been turned into a bit of code stored as a result of compression.

Keywords: Compression, Decompression, Digital Image, Huffman, Lossless

Abstrak

Pesatnya perkembangan teknologi dan informasi saat ini sangat memungkinkan untuk melakukan pengiriman informasi dengan cepat. Kecepatan pengiriman informasi tentunya sangat bergantung kepada besarnya ukuran kapasitas informasi tersebut. Besarnya ukuran kapasitas informasi dapat mengakibatkan lambatnya proses pengiriman. Disamping itu pula, ukuran kapasitas informasi yang besar akan membutuhkan ruang penyimpanan yang sangat besar. Solusi dari permasalahan ini adalah dengan melakukan kompresi pada informasi sebelum pengiriman agar ukuran kapasitas dari informasi tersebut menjadi lebih kecil. Sedangkan untuk mengembalikan hasil kompresi ke bentuk awal adalah dengan melakukan teknik dekompresi. Beberapa teknik kompresi citra digital telah dikembangkan seperti block-coding, encoding dan lainnya. Teknik kompresi citra digital terus dikembangkan dengan tujuan untuk mengompres hingga sekecil mungkin data citra, namun pada saat dekompresi tidak satupun data citra yang hilang. Metode Huffman adalah salah satu metode yang mampu melakukan kompresi dan dekompresi pada citra digital. Kompresi citra adalah proses untuk meminimalisasi jumlah bit yang mempresentasikan suatu citra sehingga kapasitas data citra menjadi lebih kecil. Metode Huffman adalah salah satu metode kompresi citra digital bersifat lossless yang merupakan metode kompresi dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili data karakter. Setiap jenis karakter yang terdapat didalam sebuah citra dihitung dan mengurutkan berdasarkan jumlah paling sedikit ke jumlah yang besar kedalam bentuk pohon biner dan memberi kode untuk setiap karakter. Data yang ada akan digantikan dengan kode bit berdasarkan pohon biner. Data yang telah berubah menjadi kode bit disimpan sebagai hasil kompresi.

Kata kunci: Kompresi, Dekompresi, Citra Digital, Huffman, Lossless

© 2016 Jurnal PROCESSOR.

1. Pendahuluan

Pesatnya perkembangan teknologi informasi dan komunikasi pada saat ini sangat memungkinkan manusia untuk melakukan pengiriman data dan informasi dengan cepat. Informasi tidak hanya disajikan dalam bentuk teks saja, melainkan data dan informasi dapat juga berupa gambar, suara dan video yang lebih dikenal dengan multimedia. Pada era teknologi informasi ini manusia tidak dapat dipisahkan dengan multimedia. Citra digital merupakan salah satu komponen dari multimedia yang berperan penting sebagai bentuk informasi visual. Pada umumnya representasi citra digital akan membutuhkan ruang penyimpanan yang cukup besar. Semakin besar kapasitas dari citra digital tersebut maka akan semakin besar pula ruang penyimpanan yang dibutuhkan. Disisi lain, kebanyakan citra digital mengandung duplikasi data. Duplikasi data pada citra digital dapat berarti dua hal. Pertama, besar kemungkinan suatu piksel dengan piksel disebelahnya memiliki intensitas yang sama, sehingga setiap piksel akan memboroskan ruang penyimpanan. Kedua, citra banyak mengandung bagian (*region*) yang sama, sehingga bagian yang sama ini tidak perlu dikodekan berulang kali karena akan mengalami redundansi data.

Kompresi adalah cara untuk meminimalkan ruang penyimpanan dalam merepresentasikan citra digital. Prinsip dasar dari kompresi adalah dengan cara mengurangi duplikasi data yang terdapat didalam citra digital tersebut. Sehingga ruang penyimpanan yang dibutuhkan untuk merepresentasikan citra digital menjadi lebih sedikit dibandingkan representasi awal. Sedangkan dekompresi adalah proses pengembalian citra digital hasil kompresi kedalam bentuk semula. Ada dua jenis kompresi pada citra digital, yaitu kompresi tipe *lossy* dan kompresi tipe *lossless*. Kompresi tipe *lossy* adalah kompresi yang dimana hasil dari proses tersebut akan mengakitnya kualitas citra digital yang dihasilkan akan jauh lebih rendah dari pada kualitas citra digital sebelum dikompresi. Sedangkan kompresi tipe *lossless* adalah kompresi yang dimana hasil dari proses tersebut akan mengakitnya kualitas citra digital yang dihasilkan tidak menurun. Salah satu contoh kompresi tipe *lossless* adalah dengan menggunakan metode Huffman. Masalah utama yang sering ditemui dalam penyimpanan data dalam bentuk citra digital adalah besarnya kapasitas ruang penyimpanan yang diperlukan. Hal akan mengakitnya lebih cepat penuhnya memory penyimpanan data dikomputer. Untuk mengatasi hal ini maka setiap data citra digital harus dilakukan kompresi sehingga akan memperkecil kapasitas citranya namun tidak mengurangi kualitas dari citra itu sendiri.

Dari uraian yang sudah dijelaskan di atas tersebut, maka perlu adanya pemecahan masalah atau solusi, salah satunya adalah dengan melakukan rancang bangun aplikasi kompresi dan dekompresi terhadap citra digital. Adanya rancang bangun aplikasi kompresi dan dekompresi citra digital ini, akan memaksimalkan ruang penyimpanan dengan mempertahankan kualitas citra digital yang tetap baik, mempercepat dalam proses pemanggilan citra digital dan mempercepat dalam proses penyimpanannya.

2. TinjauanPustaka

2.1 Pengertian Kompresi

Kompresi adalah suatu teknik pemampatan data sehingga diperoleh file dengan ukuran yang lebih kecil daripada ukuran aslinya. Kompresi bekerja dengan mencari pola-pola perulangan pada data dan menggantinya dengan sebuah penanda tertentu. Metode kompresi (*compression*) atau metode dengan pemampatan data pada dasarnya dapat dikelompokkan ke dalam dua kelompok besar, yaitu: 1. Metode Lossless dan 2. Metode Lossy [1]. 1. Metode Lossless, metode Lossless merupakan metode dimana data kompresi adalah kelas dari algoritma data kompresi yang memungkinkan data yang asli dapat disusun kembali dari data kompresi. Kompresi data lossless digunakan dalam berbagai aplikasi seperti format ZIP dan GZIP. Lossless juga sering digunakan sebagai komponen dalam teknologi kompresi data lossy. Kompresi Lossless digunakan ketika sesuatu yang penting pada kondisi asli. Beberapa format gambar seperti BMP (*Bitmap*), PNG (*Portable Network Graphics*) atau GIF (*Graphics Interchange Format*) hanya menggunakan kompresi lossless, sedangkan yang lainnya seperti TIFF (*Tagged Image Format File*)

dan PNG dapat menggunakan metode lossy atau lossless. Metode lossless menghasilkan data yang identik atau sama dengan data aslinya. Misalnya pada citra atau gambar dimana metode ini menghasilkan hasil yang tepat sama dengan citra semula, pixel per pixel sehingga tidak ada informasi yang hilang akibat adanya proses kompresi. Namun demikian ratio kompresi (Rasio kompresi yaitu, ukuran file yang dikompresi dibanding yang tidak terkompresi dari file) dengan metode ini masih sangat rendah. Metode ini cocok untuk kompresi citra yang mengandung informasi penting yang tidak boleh rusak akibat kompresi, misalnya gambar hasil diagnosa medis. Contoh metode lossless adalah metode run-length, Huffman, delta dan LZW (*Lempel-Ziv-Welch.*) 2. Metode Lossy, Lossy kompresi adalah suatu metode untuk mengkompresi data dan men-dekompresinya, data yang diperoleh mungkin berbeda dari yang aslinya tetapi cukup dekat perbedaannya. Lossy kompresi ini paling sering digunakan untuk mengkompres data multimedia (Audio dan gambar statis). Sebaliknya, kompresi lossless diperlukan untuk data teks dan file, seperti catatan bank, artikel teks dll. Format kompresi lossy mengalami generation loss, yaitu jika dilakukan secara berulang kali kompresi dan dekompresi file akan menyebabkan kehilangan kualitas citranya secara progresif. hal ini berbeda dengan kompresi data lossless. ketika pengguna yang menerima file terkompresi secara lossy (misalnya untuk mengurangi waktu download) file yang diambil dapat sedikit berbeda dari yang asli di level Bit ketika tidak dapat dibedakan oleh mata dan telinga manusia untuk tujuan paling praktis.

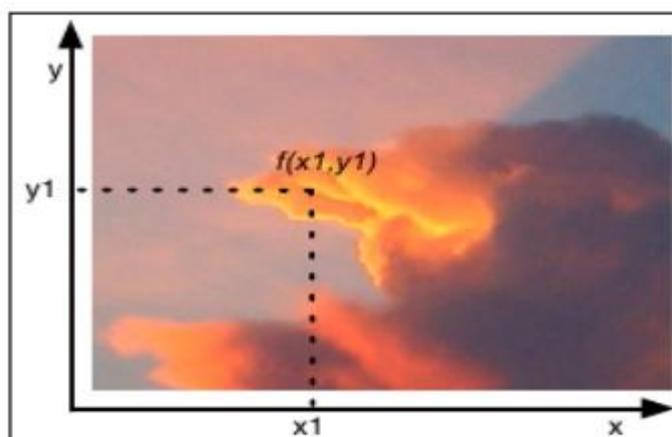
Proses kompresi yang dilakukan terhadap sebuah citra memberikan manfaat yang sangat besar, misalnya dalam industri multimedia saat ini. Diantara manfaat tersebut adalah, 1. Pada proses pengiriman data (*data transmission*) pada saluran komunikasi data. Citra yang telah dikompresi membutuhkan waktu pengiriman yang lebih cepat (singkat) dibandingkan dengan citra yang tidak dikompresi. Hal ini dapat dilihat pada aplikasi pengiriman gambar lewat *fax*, *videoconferencing*, pengiriman data medis, pengiriman gambar dari satelit luar angkasa, pengiriman gambar via telepon genggam, download gambar dari internet, dan sebagainya. 2. Selain manfaat pada proses pengiriman data, kompresi citra juga bermanfaat pada saat dilakukan proses penyimpanan data (*data storing*) di dalam media sekunder (*storage*). Sebuah citra yang telah dilakukan kompresi atau pemampatan akan membutuhkan ruang memori di dalam media storage yang lebih sedikit (kecil) jika dibandingkan dengan citra yang tidak dilakukan proses kompresi.

2.2 Pengertian Dekompresi

Kebalikan dari proses kompresi data yaitu proses dekompresi. Dekompresi adalah sebuah proses untuk mengembalikan data baru yang telah dihasilkan oleh proses kompresi menjadi data awal. Dekompresi yang menghasilkan data sama persis dengan data aslinya sebelum kompresi, maka data tersebut disebut *lossless compression*. Sebaliknya, jika hasil dekompresi menghasilkan data tidak sama persis dengan data aslinya sebelum dekompresi, karena ada data yang dihilangkan karena dirasa tidak terlalu penting tetapi tidak mengubah informasi yang dikandungnya, disebut *lossy compression* [2]. Setelah dilakukan proses kompresi terhadap sebuah gambar atau file, maka sebuah citra akan dapat dikembalikan ke dalam bentuk citra semula yaitu dengan melakukan dekompresi (*decompression*) pada data citra tersebut.

2.3 Elemen Citra Digital

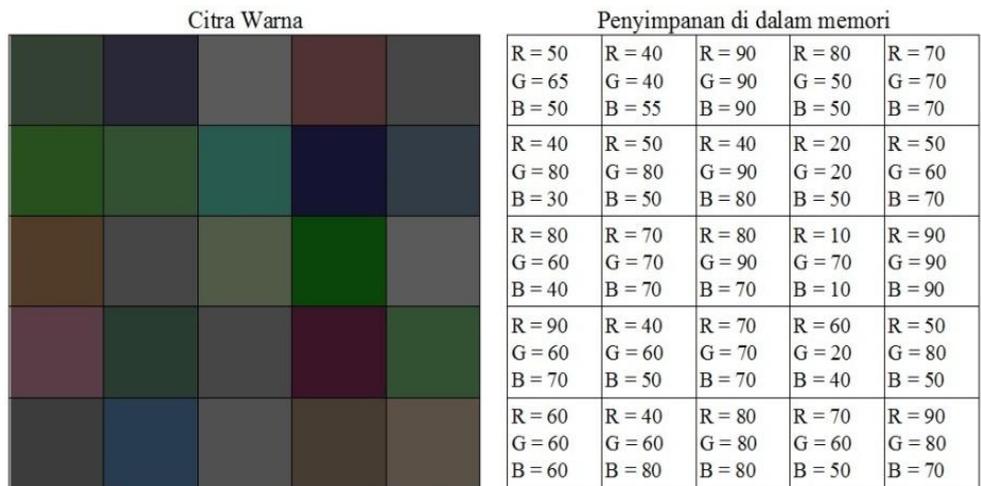
Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Sedangkan Citra digital dapat didefinisikan sebagai fungsi dua variable, $f(x,y)$, di mana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut. hal tersebut dilihat pada Gambar 1. Penggunaan teknologi yang menjadi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*).



Gambar 1. Citra digital $f(x,y)$.

Citra digital mempunyai beberapa elemen-elemen, elemen tersebut adalah: 1. Kecerahan (*Brightness*), Kecerahan (*Brightness*) merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) didalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya, 2. Kontras (*Contrast*) Kontras (*Contrast*) menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata, 3. Kontur (*Contour*) Kontur (*Contour*) adalah keadaan yang ditimbulkan oleh perubahan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek didalam citra. 4. Warna Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek. 5. Bentuk (*Shape*) Bentuk (*Shape*) adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan property intrinsic utama untuk sistem visual mata manusia. 6. Tekstur (*Texture*) Tekstur (*Texture*) dicirikan sebagai distribusi spasial dari derajat keabuan didalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cakup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

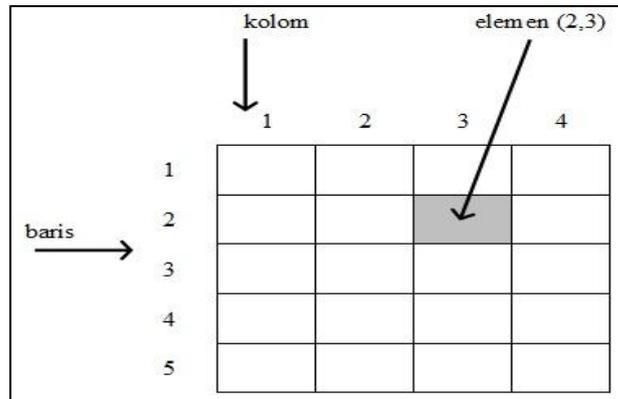
Selain citra digital, juga ada citra warna (*true color*). Dimana setiap piksel pada citra warna memiliki warna yang merupakan kombinasi dari tiga warna dasar (*RGB = Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $28 \cdot 28 \cdot 28 = 224 = 16$ juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam. Penyimpanan citra *true color* didalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra *true color* diwakili oleh 3 byte, dimana masing-masing byte mempresentasikan warna merah (*Red*), hijau (*Green*), dan biru (*Blue*). Citra digital berbeda dengan citra *color*, meskipun masing-masing citra memiliki 3 (tiga) warna dasar. Pada citra warna, sesuai dengan jumlah warna yang ada maka mempunyai ukuran yang berbeda satu dengan yang lain, dalam hal penyimpanan data di dalam memori sistem komputer. Untuk lebih jelasnya hubungan antara citra warna dengan kapasitas penyimpanan yang diperlukan untuk setiap warna yang ada, dapat dilihat pada Gambar 2. sebagai berikut :



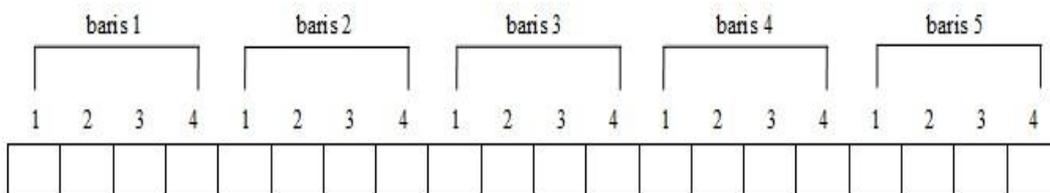
Gambar 2. Penyimpanan citra warna di dalam memori .

2.4 Matrik.

Matriks adalah struktur penyimpanan data didalam memori utama yang setiap individu elemennya diacu dengan menggunakan dua buah indeks (yang biasanya dinotasikan dengan indeks baris dan indeks kolom). Matriks digambarkan sebagai sekumpulan kotak yang tersusun berjajar pada setiap baris dan kolom. Susunan bujur sangkar itu sering dinamakan juga dengan *grid*. Dimana menggambarkan sebuah matriks yang berukuran 5 x 4 yang berarti matriks ini terdiri dari 5 baris dan 4 kolom .



Gambar 3. Matrik dengan 5 baris dan 4 kolom



Gambar 4. Representasi matriks 5 x 4 didalam memori sebagai larik

Karena matriks sebenarnya adalah sebuah larik, maka konsep umum dari larik juga berlaku untuk matriks yaitu: 1. Kumpulan elemen yang bertipe sama, tipe elemen matriks dapat berupa tipe dasar (*integer, real, Boolean, char* dan *string*), atau tipe terstruktur seperti *record*. 2. Setiap elemen data dapat diakses secara langsung jika indeksnya (baris dan kolom) diketahui, yang dalam hal ini indeks menyatakan posisi relative didalam kumpulannya. 3. Merupakan struktur data yang statis, artinya jumlah elemennya sudah dideklarasikan terlebih dahulu dan tidak bisa diubah selama pelaksanaan program. Dari penjelasan tersebut dapat disimpulkan bahwa setiap piksel yang terdapat didalam sebuah citra digital dapat diwakilkan dengan notasi disetiap pikselnya dan digambarkan kedalam bentuk sebuah matriks.

2.5 Metode Huffman

Algoritma kompresi Huffman dinamakan sesuai dengan nama penemunya yaitu David Huffman, salah seorang profesor dari institut terkemuka yaitu MIT (*Massachusetts Institute of Technology*). Kompresi Huffman merupakan algoritma kompresi lossless dan ideal untuk mengkompresi teks atau file program. Ini yang menyebabkan mengapa algoritma ini banyak dipakai dalam program kompresi. Kompresi Huffman termasuk dalam algoritma keluarga dengan variable codeword length. Ini berarti simbol individual (karakter dalam sebuah file teks sebagai contoh) digantikan oleh urutan bit yang mempunyai suatu panjang yang nyata (*distinct length*). Jadi simbol yang muncul cukup banyak dalam file akan memberikan urutan yang pendek sementara simbol yang jarang dipakai akan mempunyai urutan bit yang lebih panjang [3].

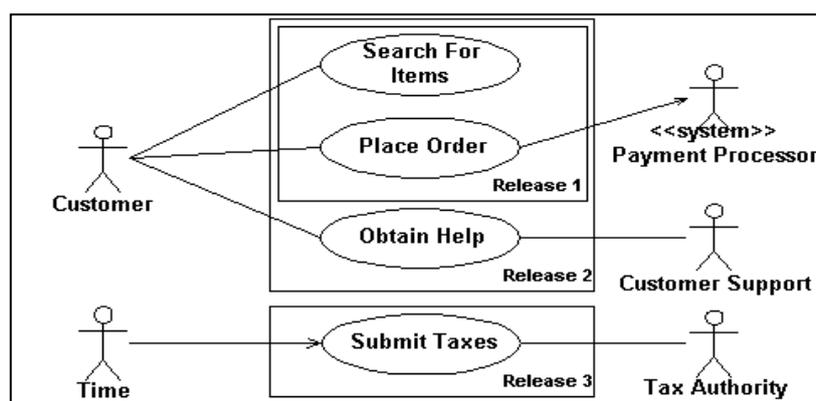
langkah-langkah metode kompresi atau pemampatan dengan menggunakan algoritma Huffman, adalah: 1. Data dianalisis dahulu dengan cara membuat tabel frekuensi kemunculan setiap simbol ASCII, tabel frekuensi tersebut memiliki atribut berupa simbol ASCII dan frekuensi. 2. Dua data yang memiliki frekuensi kemunculan paling kecil dipilih sebagai simpul pertama pada pohon Huffman. 3. Dari dua simpul ini dibuat simpul induk yang mencatat jumlah frekuensi dua simpul pertama. 4. Kemudian dua simpul tersebut dihapus dari tabel digantikan oleh simpul induk tadi. Simpul ini kemudian dijadikan acuan untuk membentuk pohon. 5. Langkah 3-5 dilakukan berulang-ulang hingga isi tabel tinggal satu saja, data inilah yang akan menjadi simpul bebas atau simpul akar. 6. Setiap simpul yang terletak pada cabang kiri (simpul dengan frekuensi lebih besar) diberi nilai 0 dan simpul yang terletak pada cabang kanan (simpul dengan frekuensi lebih kecil) diberi nilai 1. 7. Pembacaan dilakukan dari simpul akar kearah simpul daun dengan memperhatikan nilai setiap cabang.

2.6 UML (Unified Modeling Language)

Adalah suatu alat (*tools*) yang sering digunakan untuk memvisualisasikan atau menjelaskan suatu tampilan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek [4]. Pada bahasa pemrograman yang berorientasi kepada penggunaan UML, memiliki beberapa diagram, antara lain yaitu:

a. Use case diagram

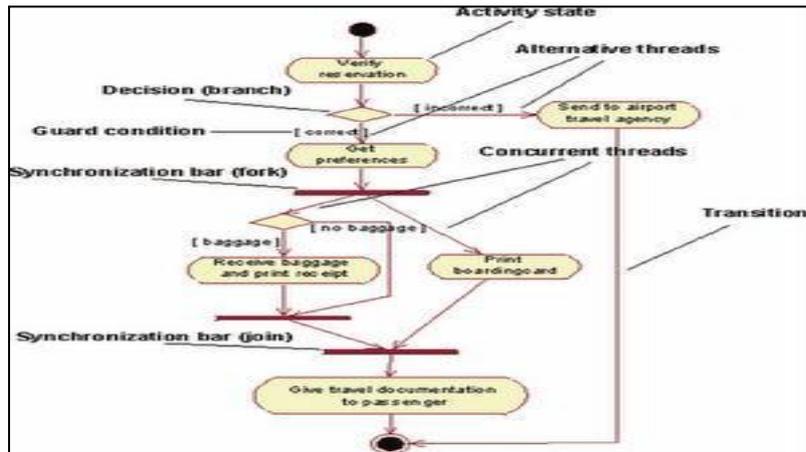
Merupakan suatu diagram yang menggambarkan *actor (user)*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.



Gambar 5. Contoh use case diagram UML

b. Activity diagram

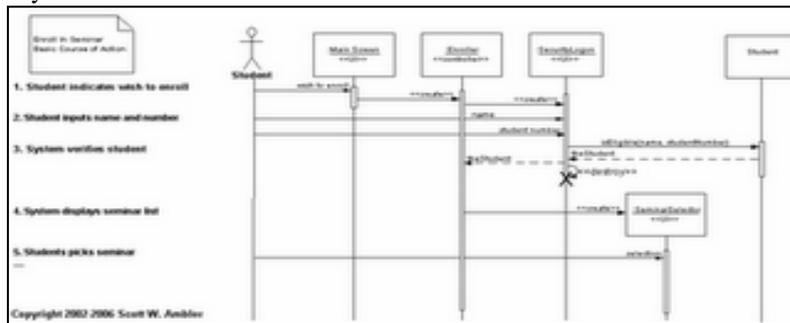
Menggambarkan suatu aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk semua aktivitas yang dilakukan dalam sistem.



Gambar 6. Contoh activity diagram UML

c. Sequence diagram

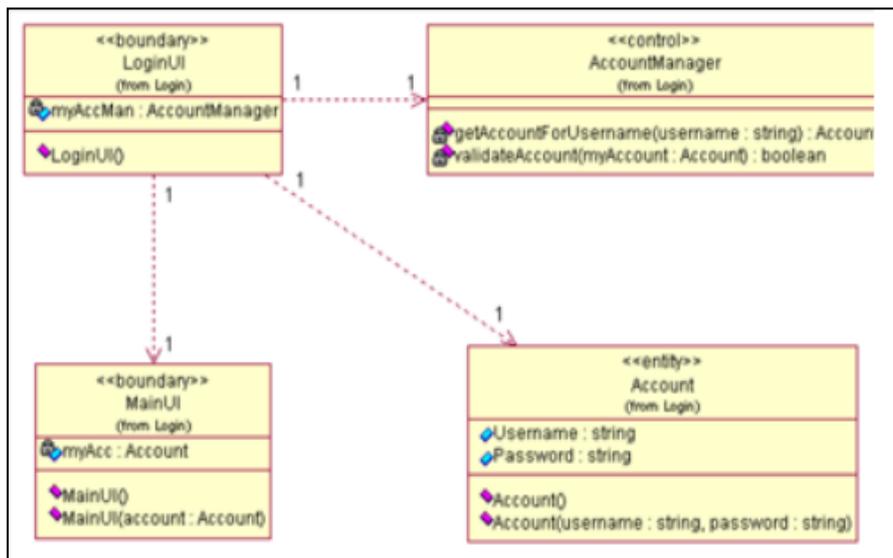
Sequence diagram adalah diagram yang menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagramnya.



Gambar 7. Contoh sequence diagram UML.

d. Class diagram,

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan suatu struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.



Gambar 8. Contoh notasi class diagram UML

3. Metodologi

Metodologi yang dilakukan dalam penelitian ini terdiri dari beberapa tahapan atau cara, yang dapat diuraikan sebagai berikut:

3.1 Melakukan studi literatur

Studi literatur adalah tahapan dimana dilakukan kegiatan pengumpulan data yang terkait dengan kompresi dan dekompresi citra digital. Semu data-data yang dikumpulkan meliputi data-data yang bersumber dari sejumlah atau beberapa jurnal Nasional, buku cetak, dan juga bersumberkan kepada penelitian yang sudah dilakukan sebelumnya.

3.2 Analisa data

Analisa data adalah tahapan berikutnya setelah melakukan studi literatur, dimana tahap ini dilakukan analisa tentang data-data yang telah diperoleh. Dalam hal ini khususnya yang berhubungan dengan kompresi dan dekompresi untuk citra digital. Kemudian juga melakukan analisa terhadap metode Huffman mengenai langkah atau tahapan-tahapan untuk melakukan kompresi atau pemampatan terhadap citra (gambar) yang dalam hal ini citra yang digunakan adalah citra yang ber ekstension BMP (Bitmap). Selanjutnya hasil dari analisa yang sudah selesai akan dijadikan pedoman untuk melakukan rancang bangun sistem aplikasinya.

3.3 Perancangan aplikasi

Perancangan Aplikasi adalah tahapan dimana dibuatnya rancangan sistem dengan berdasarkan hasil analisis data. Perancangan atau rancang bangun aplikasi kompresi dan dekompresi citra digital ini menggunakan pemodelan UML (*Unified Modeling Language*), yang terdiri dari *use case diagram*, *activity diagram*, *sequent diagram* dan *class diagram*. Selain itu juga dibangun rancangan *output input*, rancangan *interface* sebagai antar muka antara output dengan input.

3.4 Pembuatan program

Pembuatan program dilakukan dengan mengimplementasikan hasil dari perancangan yang telah dibuat dengan tool modeling UML yang standar yang ada. Pada tahapan pembuatan program menggunakan bahasa pemrograman, yaitu bahasa pemrograman *Embarcadero Delphi 2010*. Delphi 2010 diperkenalkan oleh Embarcadero Technologies untuk meneruskan kesuksesan versi-versi Delphi sebelumnya. Delphi 2010 merupakan lingkungan pemrograman yang berjalan disistem operasi Microsoft Windows dengan tampilan yang menarik dan mudah digunakan. Dukungan penuh terhadap fitur perangkat tablet seperti keyboard virtual dan layar sentuh menjadi pedoman membangun aplikasi kompleks seperti aplikasi multimedia yang dapat menjalankan hampir semua format multimedia. Oleh karena itu ahasa pemrograman Delphi 2010 sebagai perangkat untuk membangun aplikasi kompresi dan dekompresi pada citra digital mengingat bahwa citra digital merupakan bagian dari multimedia

3.5 Uji coba dan evaluasi

Tahapan ini adalah tahapan terakhir, dimana akan dilakukan uji coba terhadap aplikasi (program) yang sudah selesai dengan menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi yang disesuaikan dengan kebutuhan sistem. Pengujian juga dilakukan dengan teknik *black box* untuk keakuratan data citranya. Kemudian juga dievaluasi terhadap aplikasinya untuk mengetahui kinerja sistem serta kemungkinan terjadinya kesalahan atau ketidak sesuaian agar dapat diperbaiki dan disempurnakan.

4. Hasil dan pembahasan

Secara umum representasi citra digital akan membutuhkan ruang penyimpanan yang cukup besar. Semakin besar kapasitas dari citra digital tersebut maka akan semakin besar pula ruang penyimpanan yang dibutuhkan. Disisi lain, kebanyakan citra digital mengandung duplikasi data. Duplikasi data pada citra digital dapat berarti dua hal, pertama besar kemungkinan suatu piksel dengan piksel disebelahnya

memiliki intensitas yang sama, sehingga setiap piksel akan memboroskan ruang penyimpanan. Sedangkan yang kedua, citra digital banyak mengandung bagian (*region*) yang sama, sehingga bagian yang sama ini tidak perlu dikodekan berulang kali karena akan mengalami redundansi data.

4.1 Rancang bangun aplikasi

Pembuatan aplikasi kompresi dan dekompresi citra digital menggunakan metode Huffman, akan menyesuaikan dengan langkah-langkah algoritmanya. Berikut penjelasan untuk simulasi metode Huffman dalam mengkompres citra digital dengan ukuran 3 x 4.

Tabel 1. Citra dengan ukuran 3 x 4

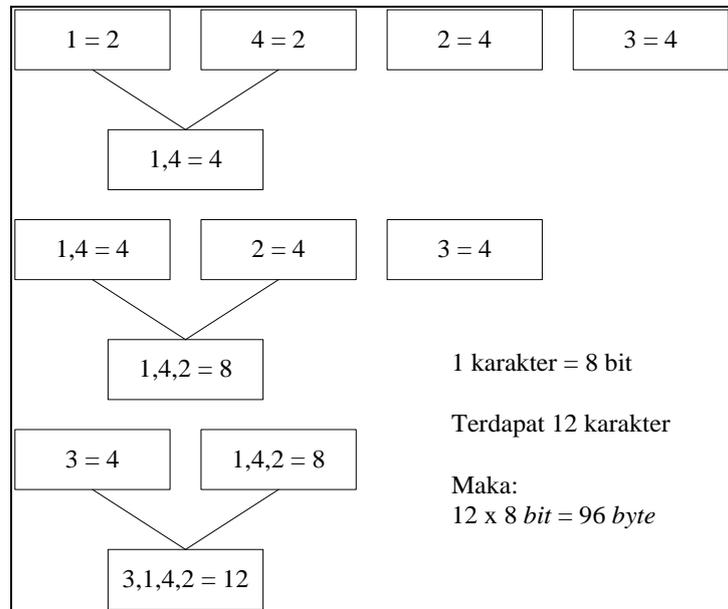
3	2	1	4
2	1	2	3
2	3	4	3

Dari citra dengan ukuran 3 x 4 tersebut, dapat diperoleh frekuensi kemunculan nilai pada setiap piksel seperti yang ditampilkan pada tabel 2, sebagai berikut ini:

Tabel 2. Frekuensi kemunculan nilai piksel

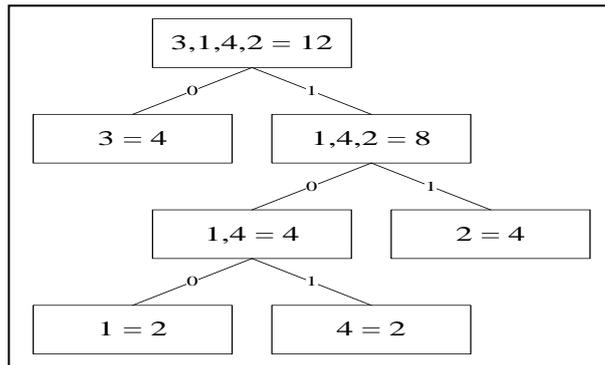
nilai	Frekuensi kemunculan
1	2
2	4
3	4
4	2

Setelah frekuensi kemunculan nilai pada setiap piksel di dapat, selanjutnya buat pohon Huffman untuk mendapatkan kode Huffman untuk menggantikan nilai piksel agar menghemat penggunaan ruang penyimpanan. Langkah-langkahnya adalah sebagai berikut:



Gambar 9. Pohon huffman sebagai kode huffman

Selanjutnya lakukan penelusuran setiap nilai piksel yang terdapat dalam pohon Huffman. Setiap penelusuran ditandai dengan nilai biner 0 untuk cabang sebelah kiri dan 1 untuk cabang sebelah kanan. Langkah-langkah yang dilakukan yaitu sebagai berikut:



Gambar 10. Penelusuran nilai piksel pohon huffman

Langkah berikutnya adalah simpan kode Huffman berdasarkan penelusuran nilai piksel pada citra digital, yaitu sebagai berikut:

Tabel 3. Menyimpan kode huffman

nilai	Frekuensi kemunculan	Kode Huffman
1	2	1 0 0
2	4	1 1
3	4	0
4	2	1 0 1

Langkah terakhir yang harus dilakukan yaitu menempatkan kode Huffman kedalam citra digital untuk menggantikan nilai-nilai piksel yang sebelumnya, seperti yang dapat dilihat pada tabel berikut:

Tabel 4. Memasukan kode huffman dalam nilai digital

0	1 1	1 0 0	1 0 1
1 1	1 0 0	1 1	0
1 1	0	1 0 1	0

Dari perolehan kode Huffman di atas maka penggunaan ruang penyimpanan dapat dihitung sebagai berikut:

$$(2 * 3) + (4 * 2) + (4 * 1) + (2 * 3) = 6 + 8 + 4 + 6 = 24 \text{ byte}$$

Dengan demikian dapat diperoleh rasio hasil kompresi citra yaitu:

$$\text{Rasio} = (1 - [\frac{\text{ukuran file kompresi}}{\text{ukuran citra asli}}]) \times 100 \%$$

$$\text{Rasio} = (1 - [\frac{24}{96}]) \times 100 \%$$

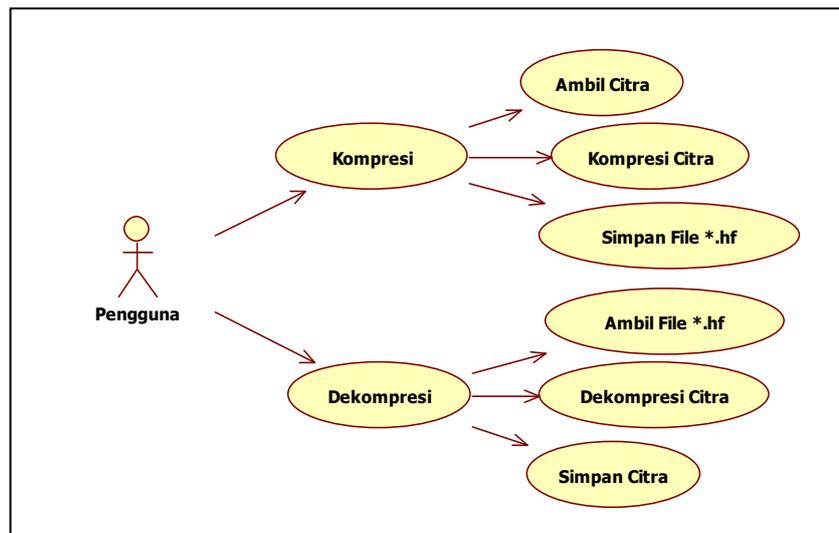
$$\text{Rasio} = (1 - 0.25) \times 100\% = 0.75 \times 100\% = 75 \%$$

Dari ukuran citra 96 byte, setelah di kompresi di peroleh hasil dengan ukuran 24 byte. Dengan demikian rasio dari proses kompresi diatas yaitu sebanyak 75%.

4.2 Pemodelan UML

Pemodelan UML digunakan untuk menjelaskan rancangan yang berhubungan dengan sistem aplikasi yang akan dibangun. Pemodelan UML dengan menggunakan 3 (tiga) diagram, yaitu *use case diagram*, *activity diagram*, *sequent diagram*. Lebih jelasnya rancangan dengan pemodela UML dapat dijelaskan sebagai berikut :

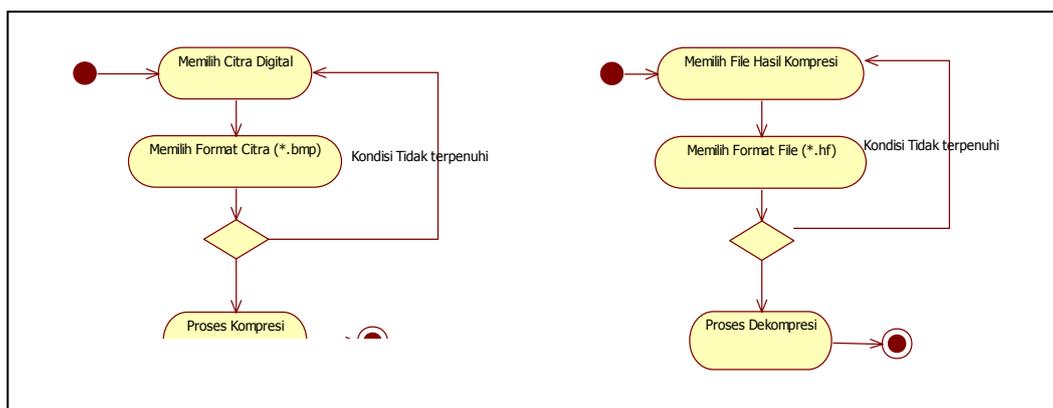
a. *Use case diagram*, pada sistem ini digunakan untuk menggambarkan proses kompresi dan dekompresi terjadi. Dalam use case ini menjelaskan bagaimana hubungan pengguna (user) berinteraksi dengan sistem untuk melakukan kompresi atau dekompresi. Lebih jelasnya pemodelan dengan use case diagram, dapat dilihat pada gambar berikut ini:



Gambar 11. Use case diagram aplikasi kompresi dan dekompresi

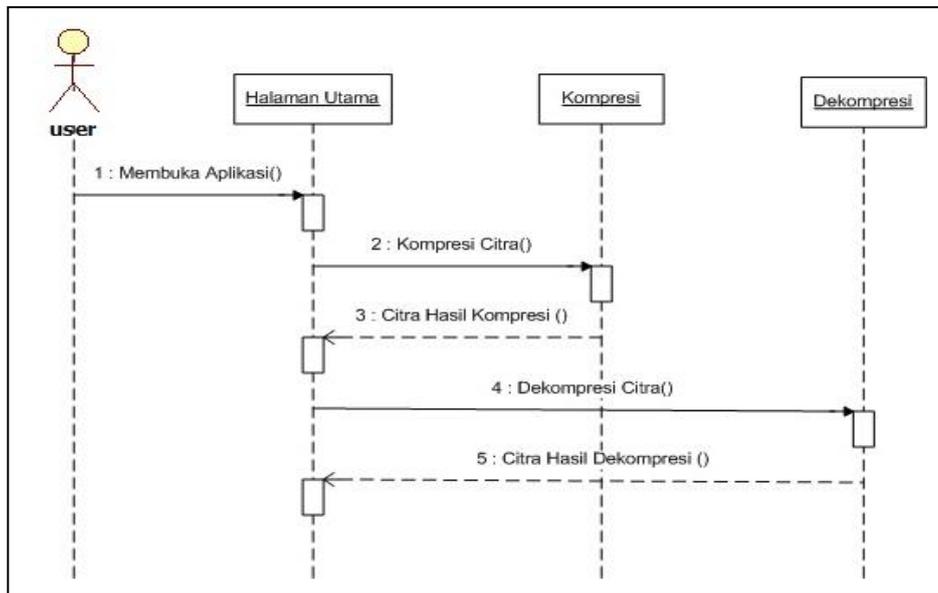
Pada gambar use case diagram diatas terdapat 2 buah use case utama yaitu use case Kompresi dan use case Dekompresi. Pada use case Kompresi terdapat 3 use case dan use case Dekompresi terdapat 3 use case. Pada proses Kompresi, terlebih dahulu pengguna harus menentukan citra yang akan dikompresi. Selanjutnya citra tersebut dikompresi. Adapun tujuan dari kompresi tersebut adalah meminimalisir penggunaan ruang penyimpanan. Dan selanjutnya citra hasil kompresi disimpan. Sedangkan pada proses Dekompresi, mengambil file yang telah terkompresi dengan ekstensi (*.hf). Selanjutnya file tersebut didekompresi. Tujuan dilakukannya proses dekompresi adalah mengembalikan citra yang terkompresi kedalam format (*.hf). kembali ke bentuk semula. Dan selanjutnya citra hasil dekompresi tersebut disimpan.

b. Activity diagram, untuk pemodelan ini menjelaskan mengenai kegiatan atau aktifitas aa saja yang dilakukan. Dalam hal ini diagram activity dibagi menjadi 2 (dua) yaitu activity kompresi dan activity dekompresi, yang dapat dilihat sebagai berikut :



Gambar 12. Activity diagram kompresi dan dekompresi aplikasi citra digital

c. Sequence diagram, merupakan diagram yang memodelkan suatu tahapan atau langkah tentang penggunaan aplikasi oleh pengguna (user). Melalui sequence diagram akan terlihat langkah apa saja yang harus dilakukan untuk melakukan kompresi dan dekompresi citra digital. Dalam sequence diagram juga menjelaskan keterhubungan antara antar muka (interface) yang ada di sistem. Pemodelan sequence diagram rancang bangun aplikasi kompresi dan dekomprese, dapat dilihat berikut ini :

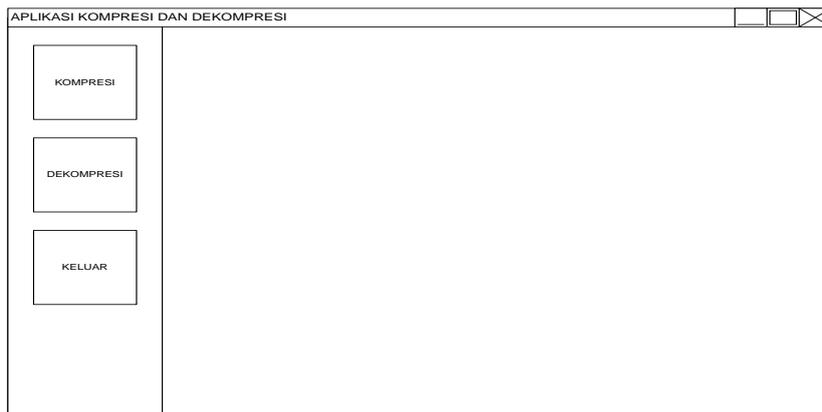


Gambar 13. *Sequence diagram rancang bangun applikasi kompresi dan dekompresi*

Pada gambar diatas merupakan penggunaan aplikasi oleh pengguna (*user*). Diawali dengan membuka aplikasi yang sudah selesai pada halaman *interface* utama, seorang pengguna dapat melakukan kegiatan untuk proses kompresi citra digital dan juga proses dekompresi citra digital. Setelah melakukan proses kompresi citra dan dekompresi citra digitalnya, maka pengguna (*user*) akan mendapatkan citra hasil kompresi dan citra hasil dekompresi.

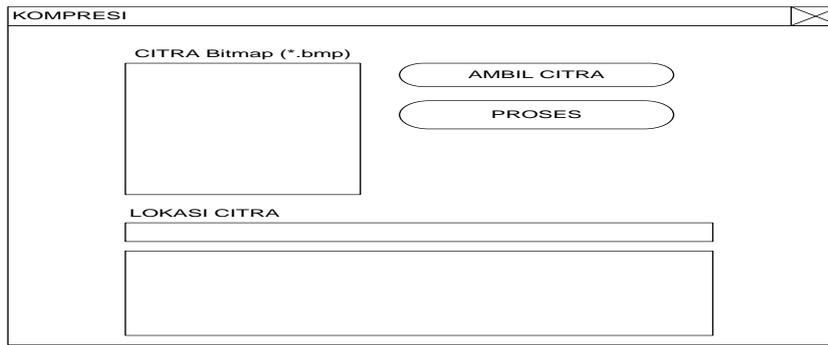
4.3 Perancangan interface aplikasi

Perancangan *interface* disebut juga dengan antar muka, merupakan perancangan yang dijadikan pedoman untuk melakukan interaksi dengan sistem. Adanya inteface akan memudahkan pengguna (*user*) untuk berinteraksi (menggunakan) sistemnya. Perancangan interface aplikasi kompresi dan dekompresi citra digital, dapat digambarkan sebagai berikut :



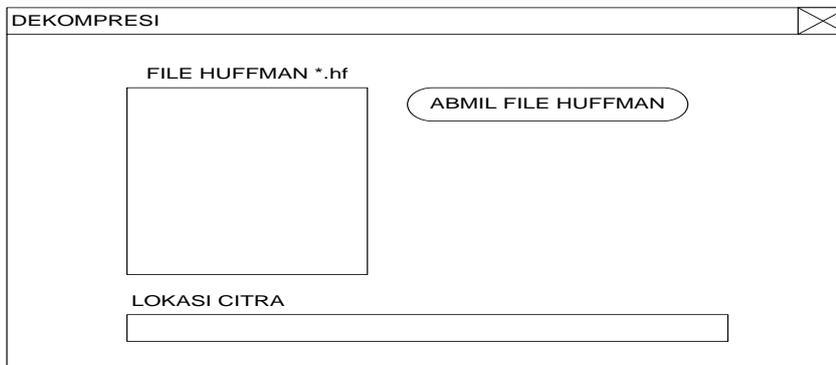
Gambar 14. *Perancangan antar muka tampilan menu utama applikasi*

Dalam perancangan antar muka ini terdapat tiga tombol sebagai pilihan utama, yaitu kompresi, dekompresi citra digital dan pilihn tombol keluar untuk kegiatan yang sudah selesai dilakukan. Selanjutn aplikasi akan menampilkan hasilnya sesuai dengan pilihn yang sudah ditentukan oleh pengguna (*user*). Untuk mendukung antar muka tampilan pada menu utama, maka jika tombol kompresi yang dipilih, maka akan tampil interface untuk kompresi citra digital, dimana tampilannya dapat digambarkan sebagai berikut :



Gambar 15. Perancangan antar muka aplikasi kompresi citra

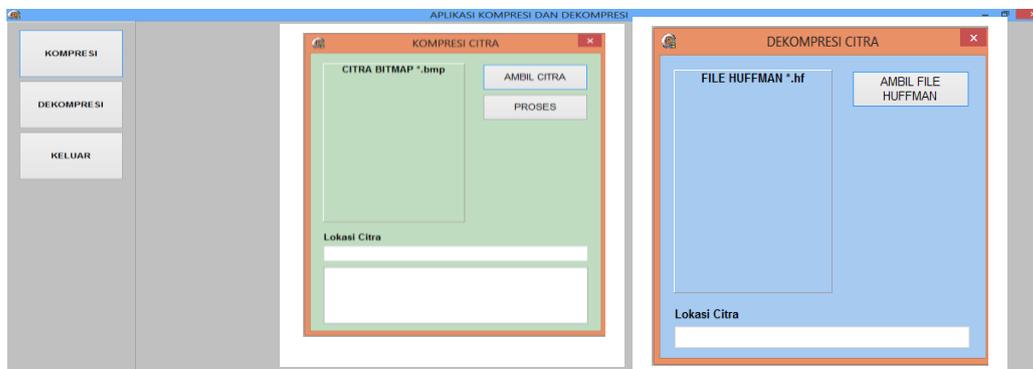
Dari perancangan antar muka ini terdapat pilihan untuk memasukkan citra dan sekaligus melakukan proses kompresi citra digitalnya. Dari proses kompresi yang dilakukan maka hasilnya akan bisa dilihat. Untuk mengembalikan citra digital ke bentuk awal atau aslinya maka perlu adanya antar muka dari proses dekompresi citranya, yang digambarkan berikut ini :



Gambar 16. Perancangan antar muka aplikasi dekompresi citra

4.4 Uji coba aplikasi

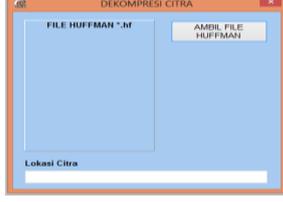
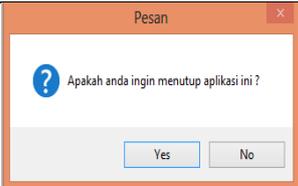
Aplikasi yang sudah selesai selanjutnya dilakukan uji coba atau implementasi, hal ini perlu dilakukan untuk memastikan bahwa hasil rancangan yang dibuat sesuai dengan hasil keluaran seperti yang diinginkan. Dalam melakukan uji coba aplikasi kompresi dan dekompresi citra digital, memerlukan hardware dan software. Hardware yang digunakan adalah *Personal Computer* dengan spesifikasi sebagai berikut, prosesor, minimal Intel Core i3, Kapasitas harddisk dengan 500 GB, memiliki RAM (*Random Access Memory*) 2 GB, sedangkan software yang diperlukan antara lain, Sistem Operasi berbasis Windows 8 32-bit, bahasa pemrograman Delphi 2010.



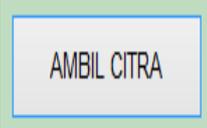
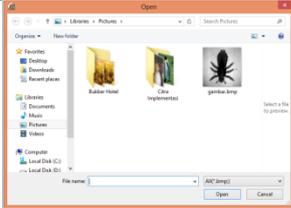
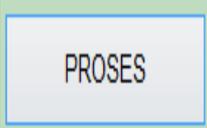
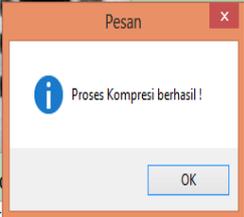
Gambar 17. Hasil uji coba antar muka utama proses kompresi dan dekompresi citra

Uji coba untuk aplikasi kompresi dan dekompresi juga menggunakan teknik *black box*. Pada teknik pengujian *black box* merupakan teknik yang memfokuskan pada fungsional dari aplikasi yang dibuat, seperti fungsi setiap form dan tombol-tombol yang terdapat didalamnya. Hasil dari pengujian dengan teknik *black box* dapat jelaskan dengan tabel berikut ini :

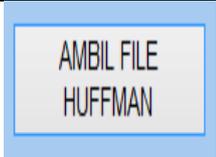
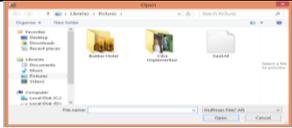
Tabel 5. Pengujian *black box* pada antar muka menu utama

No	Pengujian sistem aplikasi	Hasil yang diharapkan	Respon sistem aplikasi	Kesimpulan
1	User mengklik tombol <i>kompresi</i> 	Aplikasi akan menampilkan form kompresi.		<i>valid</i>
2	User mengklik tombol <i>dekompresi</i> 	Aplikasi akan menampilkan form dekompresi		<i>valid</i>
3	User mengklik tombol <i>keluar</i> 	Keluar dari aplikasi		<i>valid</i>

Tabel 6. Pengujian *black box* pada antar muka proses kompresi

No	Pengujian sistem aplikasi	Hasil yang diharapkan	Respon sistem aplikasi	Kesimpulan
1	User mengklik tombol <i>ambil citra</i> 	Aplikasi akan membuka kotak dialog untuk memilih citra dengan format Bitmap (*.bmp)		<i>valid</i>
2	User mengklik tombol <i>proses</i> 	Aplikasi akan mengkompres citra dan menampilkan pesan <i>proses kompresi berhasil</i>		<i>valid</i>

Tabel 7. Pengujian *black box* pada antar muka dekompresi

No	Pengujian sistem aplikasi	Hasil yang diharapkan	Respon sistem aplikasi	Kesimpulan
1	User mengklik tombol <i>ambil file Huffman</i>	Aplikasi akan membuka kotak dialog untuk	Menampilkan kotak dialog membuka file Huffman	<i>valid</i>
		memilih file dengan format Huffman (*.hf) dan secara langsung melakukan proses dekompresi pada file tersebut	 Menampilkan pesan setelah proses dekompresi dilakukan 	<i>valid</i>

Tabel 8. Pengujian aplikasi kompresi dan dekompresi citra digital

No	Citra Digital	Nama dan Size Citra	Size Setelah di Kompresi	Rasio Kompresi
1		penguin.bmp 86,91 kb	81,9 kb	5,77 %
2		fellac.bmp 81,05 kb	74,02 kb	8,67 %
3		cat_mirror.bmp 98,78 kb	97,06 kb	1,74 %

Berdasarkan tahapan - tahapan yang telah dilakukan uji coba dengan teknik *black box* seperti urian pada masing-masing tabel di atas, maka diperoleh kesimpulan bahwa citra digital yang digunakan sebagai bahan uji coba (*implementasi*) adalah citra dengan format Bitmap (*.bmp) dengan kedalaman warna 24 bit. Sedangkan file hasil dari kompresi adalah file dengan format Huffman File (*.hf). Proses kompresi dilakukan dengan menggunakan metode Huffman. Pada hasil uji coba, dan telah dilakukan evaluasi terhadap citra digital program aplikasi ternyata mampu mengkompresi citra digital dan dapat mengembalikan citra yang terkompresi kedalam bentuk semula, dengan kualitas citra atau gambar yang sama dengan citra asalnya (citra semula)..

5. Kesimpulan dan Saran

Berdasarkan hasil uji coba dan evaluasi (*implementasi*) yang sudah dilakukan pada aplikasi kompresi dan dekompresi pada citra digital dengan menggunakan metode Huffman, baik menggunakan aplikasi program maupun dengan menggunakan teknik *Black Box*, maka dapat diambil beberapa kesimpulan yang dapat dijelaskan berikut ini.

5.1 Kesimpulan

Kesimpulan dari uji coba dan evaluasi yang dilakukan terhadap aplikasi program maupun dengan teknik *Black Box*, antara lain yaitu:

1. Aplikasi ini dibuat untuk mengkompresi citra digital dan dapat mengembalikan citra yang terkompresi kedalam bentuk semula.
2. Citra yang digunakan dalam proses implementasi adalah citra digital dengan format Bitmap (*.bmp) dengan kedalaman warna 24 bit.

5.2 Saran

Aplikasi Kompresi dan Dekompresi sangat dibutuhkan untuk mengkompresi citra digital agar penggunaan ruang penyimpanan menjadi lebih sedikit. Oleh sebab itu penulis memberikan beberapa saran diantaranya:

1. Bahan implementasi dapat diganti dengan file lain berupa audio maupun video.
2. Apabila tetap ingin menggunakan citra digital, maka gunakan citra digital dengan format lain seperti JPEG (*.jpeg), JPG (*.jpg), PNG (*.png) dan yang lainnya.

6. Daftar pustaka

- [1] A. Satyapratama, M. Yunus, P. Studi, and T. Informatika, "KOMPRESI FILE GAMBAR BMP DAN PNG," *Ejurnal Stimata*, vol. 6, no. 2, pp. 69–81, 2015.
- [2] M. Roslin and A. Neta, "Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan JAVA," *Semant. 2013*, vol. 2013, no. November, pp. 224–230, 2013.
- [3] H. Purwanto, "Penerapan algoritma huffman pada kompresi file wave," *Tech. J.*, vol. 2, no. 2, pp. 40–59, 2010.
- [4] Haviluddin, "Memahami Penggunaan UML (Unified Modelling Language)," *Memahami Pengguna. UML (Unified Model. Lang.*, vol. 6, no. 1, pp. 1–15, 2011.