1. Pendahuluan

1.1 Latar belakang

Kompresi merupakan proses untuk memampatkan atau mengecilkan ukuran data. Dengan data yang telah dikompresi akan memudahkan proses penyimpanannya dalam media dengan *space* yang sangat terbatas dan mempersingkat waktu pengiriman data. Berdasarkan *output* nya kompresi dapat dibedakan menjadi dua, yaitu kompresi *lossless* dan *lossy*. Perbedaan antara keduanya adalah, kompresi *lossless* memungkinkan untuk mengembalikan data hasil kompresi tepat sama dengan data sebelum kompresi, sebaliknya kompresi *lossy* tidak dapat mengembalikan data hasil kompresi tepat sama dengan data sebelum kompresi.

Untuk mengamankan data yang sangat penting dan harus dirahasiakan maka dilakukan proses enkripsi atau pengacakan data. Data multimedia adalah data *hypermedia* yang dapat diamankan lewat proses enkripsi. *Ciphertext* enkripsi multimedia yang berukuran besar memerlukan *space* penyimpanan yang besar dan waktu pengiriman akan lebih lama, sehingga diperlukan kompresi untuk mengatasi masalah tersebut. Mengingat data yang akan dikompresi bernilai sangat penting maka algoritma kompresi yang diterapkan harus menjamin keakuratan data hasil dekompresi tepat sama dengan data sebelum kompresi, untuk itu sangat tepat bila digunakan metode kompresi *lossless*.

Algoritma kompresi Lempel Ziv Welch (LZW) adalah salah satu contoh algoritma kompresi *lossless* yang menggunakan teknik *dictionary* dalam proses kompresinya. Algoritma LZW bersifat *array based dictionary* yang mana setiap karakter *string* digantikan oleh kode tabel yang dibuat setiap ada *string* yang masuk. Ukuran tabel *dictionary* pada algoritma LZW asli adalah 4096 sampel (2^{12}) dimana 256 sampel pertama digunakan untuk tabel karakter tunggal (*extended ASCII*) dan sisanya digunakan untuk pasangan karakter atau *string* dalam data *input*.

Algoritma kompresi Huffman adalah contoh algoritma kompresi *lossless* lainnya yang menggunakan teknik statistik dalam proses kompresinya. Algoritma Huffman melakukan dua kali pembacaan pada file yang akan dikompresi yaitu, pertama untuk menghitung frekuensi kemunculan karakter dalam pembentukkan Pohon Huffman dan kedua, untuk mengkodekan simbol dalam Kode Huffman.

Algoritma LZW yang bersifat *array based dictionary* memiliki kelemahan saat memasukkan *string* dan pencarian *string* dalam *dictionary* yang bersifat *brute force* untuk mengatasi masalah tersebut kita dapat menggunakan *binary search tree based dictionary*.

Dalam pembuatan tugas akhir, penulis akan membandingkan kinerja Algoritma LZW setelah diterapkannya *binary search tree based dictionary* terhadap kinerja Algoritma Huffman dalam mengkompresi file *ciphertext* enkripsi Rijndael. Hipotesis awal dari tugas akhir ini adalah dengan menerapkan *binary search tree based dictionary* pada Algoritma LZW akan meningkatkan kemampuan rasio kompresi, waktu kompresi dan waktu dekompresinya.

1.2 Perumusan masalah

Berdasarkan latar belakang permasalahan diatas, maka dapat dirumuskan masalah pokok sebagai berikut :

1. Bagaimana menerapkan *binary search tree based dictionary* pada Algoritma LZW.
2. Bagaimana analisis perbandingan algoritma LZW *binary search tree based dictionary* terhadap algoritma Huffman untuk menentukan mana yang paling baik kinerjanya dengan meneliti rasio kompresi, waktu kompresi dan waktu dekompresi.

1.3 Tujuan

Tujuan yang ingin dicapai adalah :

1. Menerapkan *binary search tree based dictionary* pada Algoritma LZW.
2. Menganalisis perbandingan rasio kompresi, waktu kompresi dan waktu dekompresi Algoritma LZW *binary search tree based dictionary* terhadap Algoritma Huffman.

1.4 Batasan masalah

Dalam implementasi tugas akhir ini, akan dilakukan pembatasan pada beberapa hal sebagai berikut :

1. Tugas akhir ini tidak membahas analisis kriptografi.
2. Algoritma kriptografi yang digunakan adalah Rijndael AES-128.
3. File uji yang digunakan berupa file gambar dalam format raw.

1.5 Metodologi Penyelesaian Masalah

Pengerjaan tugas akhir ini menggunakan metodologi :

1. **Studi Literatur**
   Studi literatur akan dilakukan pada seluruh proses pengerjaan tugas akhir.

2. **Pengumpulan Data**
   Pada tahap ini, dilakukan pengumpulan file uji.

3. **Analisis dan Perancangan Sistem**
   Analisis terhadap keseluruhan sistem meliputi perancangan sistem, kebutuhan perangkat keras dan lunak untuk pembangunan sistem.

4. **Implementasi Sistem**
   Implementasi fungsionalitas enkripsi-dekripsi Rijndael, kompresi-dekompresi dengan algoritma LZW binary search tree based dictionary dan algoritma Huffman.

5. **Pengujian Sistem**
   Sistem akan diuji coba untuk mengenkripsi file uji kemudian mengkompresi *ciphertext*, mendekompresinya lalu mendekripsi file tersebut dan terakhir melakukan analisis terhadap hasil pengujian.

6. **Penyusunan Laporan**
   Pada tahap ini, bertujuan untuk mendokumentasikan setiap tahap pembuatan tugas akhir mulai dari landasan teori sampai dengan kesimpulan dari penelitian yang dilakukan.