**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Perkembangan teknologi informasi yang pesat telah menjadi peran yang penting untuk pertukaran informasi yang cepat. Kecepatan pengiriman informasi dalam bentuk perpaduan teks, suara dan gambar 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 pemampatan (kompresi) data teks, suara dan citra sebelum ditransmisikan dan kemudian penerima akan merekonstruksinya kembali menjadi data aslinya (dekompresi).

Di dalam dunia digital, audio dapat disimpan dalam beberapa format sesuai kebutuhan. Seperti format *MP3* yang biasa kita temukan sehari-hari, mempunyai keunggulan ukuran *file* yang kecil. Namun bukan berarti *file* dengan ukuran besar tidak lagi diminati. Seperti *file* *audio* berformat *\*.wav,* walaupun ukuran *file* ini jauh lebih besar dari ukuran *file* *audio* berformat *\*.mp3*, namun *file* *audio* dengan format *\*.wav* ini hingga sekarang masih digunakan. Alasan menggunakan format *\*.wav* karena *file* *audio* dengan format tersebut memiliki keunggulan kualitas terbaik karena tanpa kompresi.

*File* *wave* sangat dibutuhkan bagi orang yang bergelut dibidang musik editor. Karena *file* ini yang memiliki kualitas terbaik, sehingga lebih maksimal dalam hal hasil pengolahan. Karena *file* ini tidak dikompresi maka ukuran *file* ini lebih besar dibandingkan dengan yang sudah dikompresi, sehingga menimbulkan masalah dalam hal memori atau tempat penyimpanan *file* ini. Ukuran *file* ini biasanya 10MB untuk setiap menit perekaman, jadi jika kita merekam *audio* dengan durasi 4 menit, maka kita harus menyiapakan memori sebesar 40MB. Dengan ukuran seperti ini, akan menjadi masalah yang besar bagi orang yang bergelut di bidang musik.

Tidak menutup kemungkinan, suatu saat kita akan bertukar data *audio* dengan kualitas maksimal seperti *file* *wave*. Masalah selanjutnya adalah jika kita melakukan pertukaran data dengan ingin mempertahankan kualitas data, hal ini sangat mengganggu mengingat ukuran *file* *wave* ini yang cukup besar. Oleh sebab itu, dibutuhkan suatu cara agar kita bisa mempertahankan kualitas data namun memperkecil ukuran data tersebut.

Algoritma A*daptive* Huffman *Code* merupakan suatu algoritma kompresi yang bersifat *lossless*. Jadi diharapkan dengan menerapkan algoritma ini dalam hal kompresi *audio*, dapat memperkecil ukuran data *audio*.

Pada algoritma *Adaptive Huffman Code*, semakin seragam data yang diproses, maka akan memperkecil nilai rasio kompresi data. Untuk meningkatkan keseragam data yang diproses pada Algoritma *Adaptive Huffman Code* kita dapat menggunakan teknik ambang batas seragam. Diharapkan dengan menggabungkan Algoritma *Adaptive Huffman Code* dan teknik ambang batas seragam ini, dapat memperkecil nilai rasio kompresi data.

* 1. **Rumusan Masalah**

Sesuai dengan latar belakang di atas, yang menjadi masalah dalam skripsi ini adalah :

1. Bagaimana mengimplementasikan salah satu algoritma kompresi, yaitu algoritma *Adaptive Huffman Code* dan menerapkan metode ambang batas seragam untuk meningkatkan rasio kompresi.
2. Bagaimana membangun aplikasi yang dapat melakukan kompresi dan dekompresi menggunakan algoritma *Adaptive Huffman Code*.
3. Seberapa besar rasio kompresi.
4. Seberapa besar *error* data hasil dekompresi *MSE* (*Mean Squer Error*).
5. Apakah *file* hasil kompresi dapat kembali seperti semula atau tidak.
   1. **Batasan Masalah**

Adapun batasan masalah dalam penelitian ini adalah :

1. Pada penelitian ini, membuat sebuah aplikasi *desktop* untuk melakukan kompresi *audio* dan berjalan pada sistem operasi *windows*.
2. Pada penelitian ini, *file* *audio* yang diujikan adalah *file* *audio* berekstensi *\*.wave* dengan ukuran *bit* per sample 8 *bit*.
3. Pada penelitian ini hanya dibahas mengenai bagaimana proses kompresi dan dekompresi *file* *audio* *(.wav)* serta berapa besar *error* data hasil dekompresi (*Mean Squer Error*).
   1. **Tujuan Penelitian**

Tujuan dari penulis melakukan penelitian ini adalah :

1. Menerapkan algoritma *Adaptive Huffman Code* (*AHC*) untuk kompresi *file* *audio* (*.wav*).
2. Membangun sebuah aplikasi *dekstop* yang dapat melakukan proses kompresi dan dekompresi.
3. Mengetahui seberapa besar rasio kompresi *file* *audio* (*.wav*) dan berapa besar *error* data hasil dekompresi (*Mean Squer Error*).
4. Mengetahui *file* hasil kompresi bisa dibalikkan (dekompresi) menjadi *file* aslinya.
   1. **Manfaat Penelitian**

Penyusunan Tugas akhir ini diharapkan dapat bermanfaat untuk

1. Untuk menambah ilmu pengetahuan mengenai konsep kompresi dan dekompresi data menggunakan struktur data algoritma *Adaptive Huffman Code* (*AHC*) yang dikombinasikan dengan metode ambang batas seragam untuk meningkatkan rasio kompresi.
2. Dengan menerapkan algoritma *Adaptive Huffman Code* dalam kompresi *file* *audio* (.*wav*) diharapkan mampu mengurangi ukuran *file* (*.wav*) yang cukup besar dan dapat dikembalikan lagi ke bentuk *file* aslinya.
3. Mengetahui rasio kompresi algoritma *Adaptive Huffman Code* (*AHC*) dalam penghematan*.*