

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Disamping peningkatan teknologi saat ini, data mempunyai peran yang sangat penting. Data tidak hanya diperkenalkan dalam bentuk teks, namun informasi dan data juga dapat berupa gambar, suara, dan video yang biasa disebut multimedia. Ekspansi dalam penggunaan data telah menciptakan beberapa masalah dalam penyimpanan data dan secara tidak langsung telah memperluas minat akan penyimpanan data. Semakin besar ukuran *file*, semakin banyak pula ruang ekstra yang dibutuhkan dalam penyimpanan (Lamsah and Utomo 2020). Saat ini ada beberapa jenis format dalam *file*, salah satunya adalah format teks.

Teks adalah kumpulan *string* atau karakter-karakter yang menyusun satu kesatuan. Salah satu media yang biasa kita jumpai dalam penggunaan teks yakni media analog atau kertas yang biasa dipakai dalam penulisan suatu teks atau karakter-karakter. Sampai sekarang, ada cukup banyak metode/algorithm yang digunakan dalam kompresi/pemampatan data. Sehingga dalam hal ini kita perlu mempelajari tentang teknik pengompresan data teks tersebut. Kompresi (pemampatan) dilakukan untuk memperkecil kapasitas data dari data aslinya (Yonathan, Nasution, and Priyanto 2021). Kompresi data teks dapat dilakukan dengan berbagai algoritma, seperti metode algoritma *Huffman* dan Algoritma *Unary Code*. *Unary Code* umumnya direpresentasikan dalam sebuah string dari  $n$  dan bit 1 diikuti dengan satu bit 0 yang mengakhiri yang didefinisikan sebagai  $n-1$  bit 1 diikuti satu bit 0 atau sebaliknya sebagai alternatif dapat juga secara ekuivalen dimulai dari  $n$  bit 0 diikuti dengan bit 1 yang mengakhiri yang didefinisikan sebagai  $n-1$  bit 0 diikuti dengan satu bit 1. Pada *Unary Code* tidak terdapat pembagian frekuensi simbol-simbol yang ada pada *string* (Michael 2020).

Sementara *Algoritma Huffman* menggunakan standar pengkodean seperti kode morse, artinya setiap karakter (simbol) dikodekan hanya dengan serangkaian beberapa bit, di mana karakter yang biasa muncul dikodekan dengan kumpulan bit pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang (Kharisma Mahesa 2018).

Pada penelitian kali ini, maka penulis ingin mengembangkan penelitian sebelumnya dengan mengompresi *file* data berupa *file* teks dengan Judul penelitian “Penerapan Algoritma *Huffman* Dan *Unary Codes* Untuk Kompresi File Teks”. Hasil yang diharapkan adalah mampu menganalisa algoritma mana paling efisien dalam melakukan kompresi file teks, dalam hal kecepatan dan ukuran file hasil kompres.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, maka permasalahan dalam penelitian ini adalah bagaimana menerapkan algoritma *Huffman* dan *Unary Codes* pada kompresi file teks.

## **1.3 Batasan Masalah**

Dengan rumusan masalah yang telah disebutkan diatas, Penulis membatasi permasalahan yang akan diteliti yakni sebagai berikut:

1. Bahasa pemrograman yang akan digunakan adalah *Java*.
2. Format yang digunakan untuk melakukan pengujian adalah format *.txt* dengan 10 file teks sebagai data sampel.
3. Perancangan aplikasi ini dilakukan pada aplikasi Netbeans versi 8.2.

## **1.4 Tujuan Dan Manfaat**

Adapun tujuan dan manfaat dari penelitian ini, diantaranya:

### **1.4.1 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk menghasilkan sebuah aplikasi untuk kompresi file teks.

1. Menganalisis kompresi file text dengan algoritma *Unary Code* dan algoritma *Huffman*.
2. Membandingkan kecepatan proses kompresi *file* teks antara algoritma *Huffman* dan algoritma *Unary Code*.
3. Membandingkan ukuran file hasil proses kompresi *file* teks antara algoritma *Huffman* dan Algoritma *Unary Code*.

### 1.4.2 Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah untuk mengetahui kinerja dari algoritma *Unary Codes* dan algoritma *Huffman* dalam pengkompresian *file* teks.

### 1.5 Keterbaruan

(Fitri Angelina 2018) *Unary Coding* atau disebut dengan *thermometer code* adalah sebuah *encoding entropy* yang menggunakan angka dimulai  $n$  dengan  $n$  berikutnya dimulai oleh 0 (apabila angka awal dianggap sebagai bilangan bulat *non negative integer*) atau dengan  $n-1$  diikuti oleh 0 (apabila angka pertama dianggap sebagai angka positif). Misalnya 111110 atau 11110 adalah bilangan biner dari 5. Sering menggunakan  $n$  atau  $n-1$ , 0 diikuti oleh 1. 1 dan 0 bergantian tanpa kehilangan nilai biasanya. *Unary coding* adalah keduanya sebuah prefix kode dan sinkronisasi kode sendiri.

(Rahmat 2020) Algoritma *unary coding* adalah algoritma yang dapat mengkompresi citra *ultrasonografi*. Format citra yang paling baik untuk mengkopresi citra *ultrasonografi* adalah format *bitmap* (.bmp) yang hasilnya jauh lebih baik dari ekstensi lainnya. Hasil kompresi bisa terlihat melalui rasio yang diperoleh, apabila rasio negatif maka hasil kompresinya buruk serta apabila hasilnya positif maka hasil kompresinya bagus. Dalam mengolah data citra untuk mengkompresi, *software* Matlab sangat mempermudah proses perancangan aplikasi.

(Mhd. Rajani 2017) telah berhasil dalam pembuatan aplikasi kompresi teks menggunakan *Unary Coding*. Aplikasi yang dibuat berhasil mengkompresi *file* dengan membaca secara teliti setiap karakter yang ada, sehingga proses kompresi teks ini berjalan dengan efektif dengan hasil yang terlihat dalam data dimana teks menjadi berkapasitas kecil.

(Meisiria Sitorus 2020) telah menyelesaikan penelitian dengan kesimpulan bahwa mengkompresi SMS dapat dibuat menggunakan algoritma *Unary Coding*, Pengkompresian SMS dapat berjalan baik dengan adanya *Unary Coding*. (Siahaan 2016) Pada algoritma ini teks yang ingin dikompresi mulai dibaca terlebih dahulu, kemudian dilakukan pencarian teks yang sama agar tidak terjadi perulangan huruf dan tentunya dapat menghemat teks yang ada.

Irfaan Hanif (2019) sudah menyelesaikan penelitian dengan hasil pengujiannya adalah, sekitar 70% teks yang dapat terkompres oleh algoritma *Huffman* dibandingkan dengan menggunakan kode ASCII, serta berkisar 25% apabila dibandingkan dengan kita menggunakan 3-bit kode.

(Chulkamdi, Pramono, and Yudaningtyas 2015) berhasil membuat aplikasi kompresi teks dengan algoritma *Huffman*. Apabila kapasitas *file* yang digunakan 16 bit dan 96 bit maka tingkat kompresi dapat mencapai 75% dan setiap jenis memiliki frekuensi munculnya karakter yang hampir sama. Apabila isi teks sangat sedikit maka kompresi teks ini kurang berhasil dikarenakan karakter yang muncul tidak terjadi.