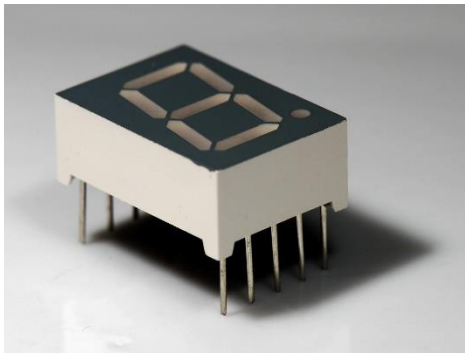
Penerapan Kode Huffman untuk Tampilan Waktu pada Layar 7 Segmen

Richard Christian - 13523024[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*richsukandar1@gmail.com*, *13523024@std.stei.itb.ac.id*

***Abstract*—Kode Huffman adalah salah satu metode penyimpanan data yang dapat digunakan untuk melakukan kompresi *lossless*, khususnya pada data dengan frekuensi yang berbeda. Penelitian ini menjelaskan tentang penerapan kode huffman pada layar 7 segmen, dimulai dari pembahasan tentang layar 7 segmen dan bagaimana data diinterpretasi oleh modul layar. Penerapan Kode Huffman lalu diaplikasikan pada data 24 jam, dimana hasilnya lalu dibandingkan dengan panjang data pada penerapan 7 kode murni dan metode lain yang umum digunakan pada layar 7 seegmen, yaitu BCD. Hasil dari perbandingan ini menunjukan bahwa dalam konteks 24 jam, Kode Huffman berhasil untuk mengurangi panjang data yang ditetapkan. Akan tetapi, pengkodean Huffman hanya dapat diaplikasikan bila data memiliki distribusi yang tertentu, dan membutuhkan pemrosesan yang lebih berat dibandingkan metode lain. Walaupun begitu, Kode Huffman tetap dapat menjadi alternatif yang menarik untuk penerapan ini.**

***Keywords*—Kode Huffman, Layar 7 Segmen, Kompresi Data, Representasi Angka.**

I. PENDAHULUAN



Gambar 1.1 Contoh modul layar 7 segmen
Sumber: https://microcontrollerslab.com/7-segment-display-pinout-working-examples-applications/ diakses pada 6 Januari 2025

Layar 7 segmen adalah layar yang sering digunakan untuk berbagai alat elektronik, dan dapat digunakan untuk menampilkan berbagai informasi numerik. Layar 7 segmen seringkali digunakan di alat-alat sederhana, seperti pada jam, kalkulator, thermometer digital, dan perangkat lainnya. Layar 7 segmen seringkali digunakan untuk menampilkan angka, dan untuk beberapa aplikasi, angka-angka tertentu mendapatkan frekuensi penggunaan yang jauh lebih tinggi, seperti pada jam dimana angka 0, 1, dan 2 lebih sering muncul dibandingkan yang lain.

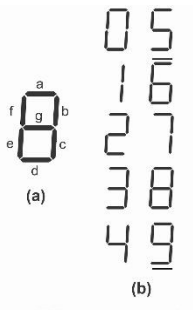
Representasi angka pada layar 7 segmen biasanya menggunakan metode yang disebut Binary-Coded Decimal (BCD). Metode ini menyederhanakan konversi dari data base 10 menjadi format digital base 2, dan menyederhanakan penggunaan bit yang diperlukan untuk menunjukkan angka dalam layar 7 segmen menjadi 4 bit saja. Akan tetapi, metode BCD bukan merupakan metode yang efisien, karena tidak memperhatikan pembobotan pada angka sesuai frekuensi penggunaan dan tidak menggunakan bit yang tersedia secara optimal.

Teknik kompresi dapat membantu menghemat penggunaan data, seperti pada penggunaan layar 7 segmen pada jam. Salah satu metode yang dapat digunakan untuk mencapai hasil ini adalah Kode Huffman, sebuah metode yang dapat digunakan untuk merepresentasikan data tanpa kehilangan resolusi, dengan catatan bahwa frekuensi penggunaan data terdistribusi secara tidak merata.

Saat diaplikasikan pada representasi jam dalam layar 7 segmen, kode Huffman dapat menjadi alternatif yang menarik untuk mengurangi penggunaan memori. Penelitian ini akan menganalisis hasil aplikasi kode Huffman pada tampilan layar 7 segmen 24 jam, dan membandingkannya dengan metode-metode lain seperti BCD.

II. DASAR TEORI

A. Layar 7 Segmen



Gambar 2.1 Contoh label layar 7 segmen dan representasi angka
Sumber: https://www.electroniclinic.com/seven-segment-display-truth-table-and-circuit-diagram/ diakses pada 6 Januari 2025

Layar 7 segmen adalah layar yang terdiri dari 7 buah segmen yang disusun membentuk angka 8, sehingga dapat digunakan untuk menampilkan angka dari 0 hingga 9 dengan menyalakan dan mematikan segmen sesuai kebutuhan. Pada layar 7 segmen, setiap segmen biasanya dilabeli dengan huruf a hingga g. Sebagai contoh, untuk menampilkan angka 8, semua segmen dinyalakan, sedangkan untuk menampilkan angka 1, hanya segmen b dan c yang perlu untuk dinyalakan. Sistem ini memungkinkan perangkat digital untuk dapat menampilkan angka secara efisien dengan kompleksitas yang rendah, dan cocok untuk penggunaan dalam perangkat yang sederhana.

Pada perangkat digital, informasi direpresentasikan dalam format base 2, dengan digit yang tersedia hanya 0 dan 1. Pada penggunaan layar 7 digit, saat data berisi angka 1, maka segmen tersebut akan menyala, sedangkan bila berisi angka 0 maka segment tersebut akan mati. Berikut adalah tabel representasi bit untuk angka 0 – 9 segmen.

A	B	C	D	E	F	G	Angka
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	1	1	1	9

Gambar 2.2 Representasi bit pada angka layar 7 segmen
Sumber: dokumentasi pribadi

B. Binary-Coded Decimal (BCD)

BCD adalah metode yang paling umum digunakan untuk merepresentasikan angka dalam layar 7 segmen. Ini dikarenakan BCD memberikan kemudahan karena sesuai dengan cara merepresentasikan angka desimal dalam format base 2. Bentuk paling umum BCD adalah BCD 8-4-2-1, dengan bobot bit pertama 8, bit kedua 4, bit ketiga 2, dan bit keempat 1.

Metode ini mempermudah penerapan angka dalam layar 7 segmen, sekaligus mengurangi kebutuhan memori karena mengurangi data yang awalnya 7 bit menjadi 4 bit saja. Sebagai contoh, untuk menampilkan digit 8, maka BCD akan berisi 1000, dengan bit pertama merepresentasikan 8. Untuk menampilkan digit 5, maka BCD akan berisi 0101, dengan bit kedua merepresentasikan 4 dan bit pertama merepresentasikan 1, sehingga berjumlah 5. Kode 4 bit ini lalu akan diinterpretasikan oleh serangkaian sirkuit logika sehingga dapat ditranslasikan menjadi 7 bit dan menyalakan segmen sesuai dengan angka masukan.

BCD				Angka
8	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	0	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Gambar 2.3 Representasi BCD pada angka layar 7 segmen
Sumber: dokumentasi pribadi

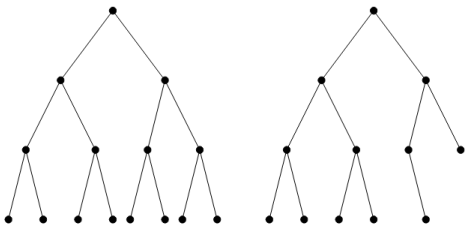
C. Pohon

Pohon adalah salah satu teori dalam Matematika Diskrit, yang digunakan untuk menampilkan suatu data dalam bentuk hierarki. Pohon adalah salah satu bentuk dari Graf, tetapi dengan perbedaan bahwa bentuk pohon adalah graf tak terhubung.

Sebuah pohon terdiri dari berbagai komponen yaitu:

- Simpul : Elemen yang terkandung dalam pohon, terbagi menjadi 2 jenis yaitu simpul daun, yaitu simpul yang tidak memiliki anak dan simpul internal, yaitu simpul yang memiliki seminimalnya satu anak.
- Akar : Simpul utama pada pohon
- Induk : Simpul induk memiliki satu atau lebih anak.
- Anak : Simpul anak hanya memiliki satu induk.
- Cabang : Hubungan antara suatu simpul dengan anak simpul
- Tingkat : Kedalaman simpul pada sebuah pohon.
- Tinggi : Jumlah tingkat dalam pohon.

Pada penerapan Kode Huffman, pohon yang digunakan adalah pohon biner, yaitu pohon dimana setiap simpul memiliki maksimum 2 buah anak. Pohon biner sesuai dengan kebutuhan kode Huffman karena bersifat biner dan secara alami sesuai dengan kebutuhan perangkat digital. Teori pohon menjadi dasar teori yang penting dan menjadi fondasi dari dibentuknya Kode Huffman.



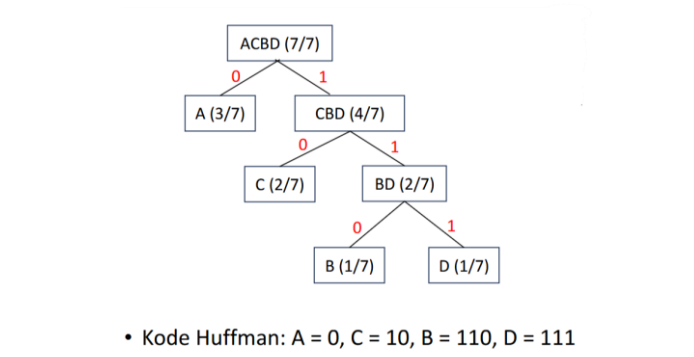
Gambar 2.4 Contoh Pohon Biner
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/matdis24-25.htm diakses pada 6 Januari 2025

D. Kode Huffman

Kode Huffman adalah salah satu metode pengkodean untuk menyimpan suatu data secara lebih efisien. Kode Huffman merupakan metode kompresi yang lossless, sehingga pada metode kompresi Huffman tidak ada data yang hilang. Kode Huffman membutuhkan variasi frekuensi dari setiap data agar dapat diaplikasikan secara efektif. Kode Huffman akan mengalokasikan kode yang lebih pendek untuk data yang lebih sering muncul, sedangkan kode yang lebih panjang digunakan untuk data yang lebih jarang muncul.

Kode Huffman menggunakan pohon biner untuk dapat merumuskan kode yang diperlukan. Setiap data pada kode Huffman lalu diberikan sebuah label frekuensi agar pembentukan pohon biner dapat menyesuaikan dengan frekuensi data. Berikut adalah proses pembuatan pohon Huffman:

- Setiap simbol diurutkan sesuai frekuensi kemunculannya, dari frekuensi terkecil dan membesar.
- 2 frekuensi terendah akan dijadikan simpul anak, dan digabungkan menjadi simpul orangtua baru, dengan bobot simpul orangtua sama dengan bobot simpul anaknya.
- Proses diulangi hingga setiap simbol sudah habis digunakan
- Cabang yang bergerak ke kiri akan dilabeli 0 dan cabang yang bergerak ke kanan dilabeli 1
- Kode Huffman didapatkan dengan membaca berurutan dari pohon teratas hingga simpul data.



Gambar 2.5 Contoh Aplikasi Kode Huffman
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/matdis24-25.htm diakses pada 6 Januari 2025

Kode Huffman dapat bekerja karena setiap kode Huffman memiliki bentuk yang unik, sehingga setiap data akan dapat dibedakan walaupun panjangnya bervariasi. Pemberian kode sesuai dengan frekuensi dan binary tree memastikan bahwa panjang Kode Huffman menyesuaikan dengan frekuensi kode.

E. Rasio dan Persentase Penghematan

Kompresi adalah konsep yang penting dalam proses optimasi penggunaan memori dengan mengurangi ukuran data yang diperlukan tanpa mengurangi fungsionalitas data yang dikompresi. Dalam kompresi, dikenal beberapa alat pengukur keefektivan suatu metode kompresi, diantaranya rasio kompresi dan persentase penghematan.

$$Rasio\ Kompresi = \frac{Rata{-}rata\ bit\ per\ digit\ kompresi}{Rata{-}rata\ bit\ per\ digit\ original}$$

Gambar 2.6 Perhitungan Rasio Kompresi
Sumber: Dokumentasi Pribadi

Rasio kompresi menghitung seberapa kecil hasil kompresi dibandingkan dengan data originalnya. Rasio Kompresi dihitung dengan membagi ukuran data kompresi dengan data yang original. Dalam konteks ini, digunakan rata-rata bit yang diperlukan per digit untuk mewakili ukuran data. Rasio kompresi yang lebih rendah berarti suatu algoritma kompresi lebih efektif dalam menyederhanakan data.

$$Penghematan\ (\%) = \left(1 - \frac{Rata{-}rata\ bit\ per\ digit\ kompresi}{Rata{-}rata\ bit\ per\ digit\ original}\right) \times 100$$

Gambar 2.7 Perhitungan Persentase Penghematan
Sumber: Dokumentasi Pribadi

Persentase penghematan, berbeda dengan rasio kompresi biasanya disajikan dalam bentuk persentase. Persentase penghematan menghitung seberapa besar pengurangan ukuran data dibandingkan dengan data originalnya. Persentase penghematan menghitung pengurangan ukuran data dengan mengurangi 1 dengan ukuran data yang sudah dikompresi dan dibagi dengan ukuran data original, kemudian dikalikan dengan 100 untuk mengubah dalam bentuk persentase.

III. IMPLEMENTASI

A. Dataset

Data yang digunakan untuk pengujian ini adalah data angka yang muncul dalam sebuah jam digital 24 jam, sehingga angka yang dapat muncul adalah angka 0 hingga 9. Struktur 24 jam yang akan digunakan adalah dalam bentuk HH:MM, dengan HH merepresentasikan jam yang dapat berisi 00 hingga 23, dan MM merepresentasikan menit yang dapat berisi 00 hingga 59.

Dari bentuk data, maka terlihat bahwa akan terdapat sebuah perbedaan frekuensi pada data, dikarenakan beberapa angka muncul lebih sering dibandingkan angka lain, seperti angka 0, 1, dan 2 yang dapat muncul pada H pertama pada HH. Angka lain, seperti 6 akan mendapatkan frekuensi yang lebih kecil, karena angka 6 tidak dapat muncul pada H pertama pada HH dan M pertama pada MM. Hasil keseluruhan frekuensi sebanyak 5760 kemunculan, sesuai dengan jumlah kemungkinan 4 digit * 24 jam * 60 menit. Berikut adalah tabel frekuensi kemunculan setiap angka :

Angka	Frekuensi
0	1164
1	1164
2	804
3	564
4	504
5	504
6	264
7	264
8	264
9	264
Total	5760

Gambar 3.1 Tabel kemunculan angka 24 jam
Sumber: dokumentasi pribadi

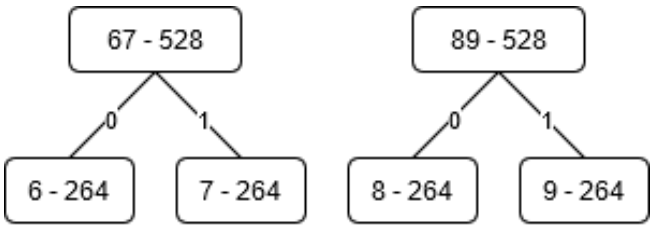
B. Pembentukan Kode Huffman

Kode Huffman kemudian akan dibentuk berdasarkan tabel frekuensi dari dataset yang sudah didapatkan. Proses pemrosesan data akan dimulai dengan mengurutkan angka sesuai frekuensi kemunculan. Berikut data tabel yang sudah diurutkan berdasarkan kemunculan:

Angka	Frekuensi
6	264
7	264
8	264
9	264
4	504
5	504
3	564
2	804
1	1164
0	1164
Total	5760

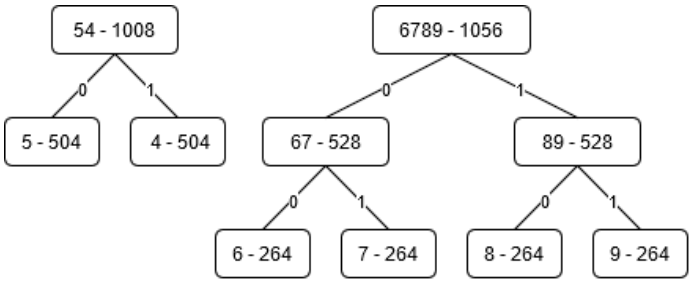
Gambar 3.1 Tabel kemunculan angka 24 jam terurut menaik
Sumber: dokumentasi pribadi

Pemrosesan lalu dilanjutkan dengan menghubungkan data dengan frekuensi terendah, yaitu 6, 7, 8, dan 9.



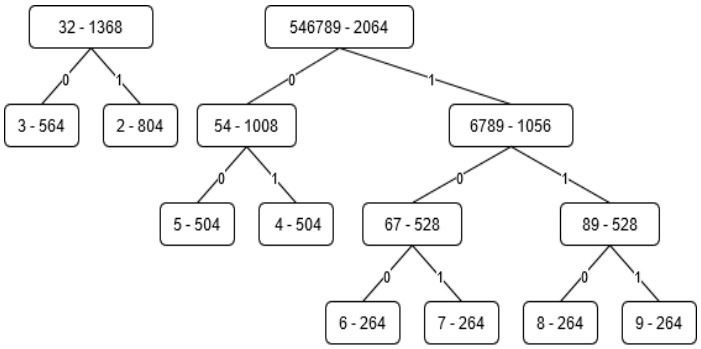
Gambar 3.2 Pembuatan Pohon Huffman 1
Sumber: dokumentasi pribadi

5 dan 4 kemudian dihubungkan juga, dilanjutkan dengan menghubungkan 67 dengan 89 karena mempunyai frekuensi yang masih lebih rendah daripada 3.



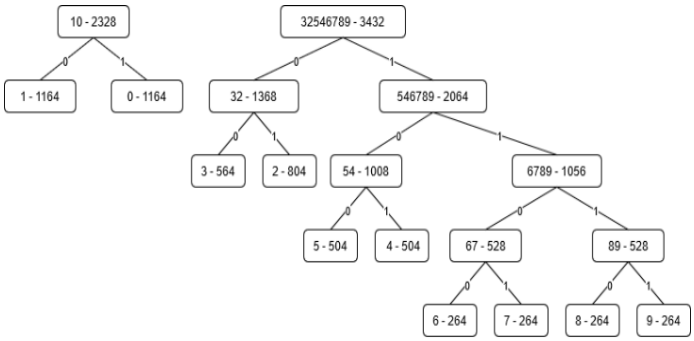
Gambar 3.3 Pembuatan Pohon Huffman 2
Sumber: dokumentasi pribadi

Proses dilanjutkan dengan menghubungkan bagian 3 dengan 2, dan 54 dihubungkan juga dengan 6789.



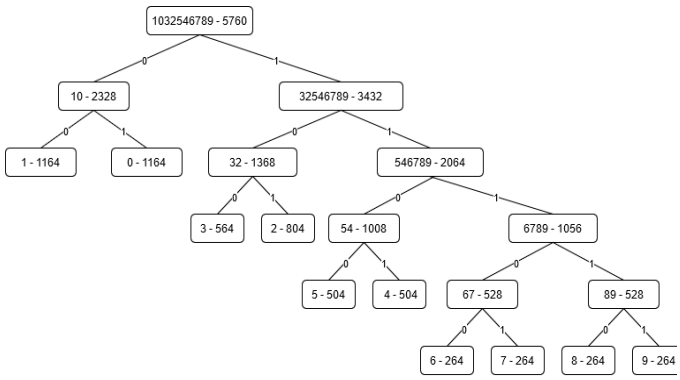
Gambar 3.4 Pembuatan Pohon Huffman 3
Sumber: dokumentasi pribadi

Proses yang sama dilakukan untuk menghubungkan 1 dengan 0, dan 32 dengan 546789.



Gambar 3.5 Pembuatan Pohon Huffman 4
Sumber: dokumentasi pribadi

Karena sudah tidak ada lagi data pada tabel yang belum dimasukkan, maka tahap terakhir adalah penggabungan dari node 10 dengan node 32546789, sehingga terbentuk Pohon Huffman seperti berikut :



Gambar 3.6 Pembuatan Pohon Huffman 5
Sumber: dokumentasi pribadi

Dengan membaca hasil dari Pohon Huffman yang sudah dibuat, maka Kode Huffman untuk setiap data dapat ditentukan dengan membaca pohon dari atas. Berikut adalah hasil kode Huffman dari pohon:

Angka	Frekuensi
0	01
1	00
2	101
3	100
4	1101
5	1100
6	11100
7	11111
8	11110
9	11101

Gambar 3.7 Hasil pemrosesan Kode Huffman
Sumber: dokumentasi pribadi

IV. HASIL DAN ANALISIS

A. Perhitungan ukuran kode Huffman.

Untuk mengukur keefektifan dari kode Huffman yang dihasilkan, maka ukuran data yang diperlukan untuk menunjukkan waktu 24 jam akan dihitung. Perhitungan panjang data akan dilakukan dengan mengalikan panjang bit dari setiap data dengan frekuensi kemunculan data. Berikut adalah hasil perhitungan ukuran yang dibutuhkan oleh kode Huffman untuk menyimpan data 24 jam :

Angka	Frekuensi	Panjang (Bit)	Ukuran Data
0	1164	2	2328
1	1164	2	2328
2	804	3	2412
3	564	3	1692
4	504	4	2016
5	504	4	2016
6	264	5	1320
7	264	5	1320
8	264	5	1320
9	264	5	1320
Total	57		18072

Gambar 4.1 Panjang data dengan Kode Huffman
Sumber: dokumentasi pribadi

B. Membandingkan hasil

Sebagai pembanding, efisiensi dari kode Huffman akan dibandingkan dengan rata-rata bit per digit menggunakan representasi bit murni dan BCD. Perhitungan rata-rata bit per digit untuk representasi bit murni dan BCD lebih sederhana, karena representasi setiap angka memiliki panjang bit yang sama. Karena ukuran bit murni selalu 7, maka rata-rata bit per digit murni adalah 7. Pada metode BCD, ukuran bit juga selalu sama sehingga rata-rata bit per digit BCD adalah 4.

Ukuran data total dari hasil kode Huffman lalu diubah juga menjadi rata-rata bit per digit Huffman, sesuai dengan proses:

$$Rata - rata\ bit\ per\ digit\ (Huffman) = \frac{18072}{5760} = 3.1375$$

Gambar 4.2 Perhitungan bit per digit Huffman
Sumber: dokumentasi pribadi

Perhitungan rasio kompresi dan persentase penghematan kode lalu dilakukan pada BCD dan Huffman, dengan menggunakan bit murni sebagai benchmark.

$$Rasio\ Kompresi\ BCD = \frac{4}{7} = 0.57$$

$$Rasio\ Kompresi\ Huffman = \frac{7}{3.1375} = 0.45$$

$$Penghematan\ BCD\ (\%) = \left(1 - \frac{4}{7}\right) \times 100 = 42.85\%$$

$$Penghematan\ Huffman\ (\%) = \left(1 - \frac{3.1375}{7}\right) \times 100 = 55.17\%$$

Gambar 4.3 Perhitungan Rasio Kompresi dan Persentase Penghematan
Sumber: dokumentasi pribadi

Perbandingan dari hasil kompresi dari kode Huffman dengan BCD menunjukkan bahwa terdapat penghematan data pada kode Huffman yang cukup signifikan. Kode Huffman berhasil

mencapai rasio kompresi sebesar 0.45, sedangkan BCD hanya mendapatkan hasil 0.57. Sebanding dengan rasio kompresi, persentase penghematan dari kode Huffman juga mencapai 55,17% dibandingkan BCD yang hanya mencapai 42.85%.

Hasil ini menunjukkan keefektifan dari implementasi kode Huffman untuk kasus ini, dan berasal dari sifat kode Huffman yang menyesuaikan panjang pengkodean data sesuai dengan frekuensi data. BCD, berbeda dengan Huffman mengkode data dengan panjang yang sudah ditentukan, yaitu 4 bit. Ini menyebabkan data dengan frekuensi tinggi tetap mendapatkan alokasi bit yang sama dengan data dengan frekuensi rendah, sehingga menyebabkan pemborosan dari penggunaan bit.

Walaupun kode Huffman menunjukkan hasil yang positif dalam pengujian untuk waktu 24 jam pada layar 7 segmen, perlu diperhatikan bahwa kode Huffman memiliki keterbatasannya sendiri dibandingkan metode lain seperti BCD. Kode Huffman memperlukan pemrosesan yang lebih kompleks pada tahap pengkodean dan pembacaan data. Berbeda dengan kode Huffman, penerapan BCD lebih sederhana untuk diimplementasikan, dan tidak memperlukan data frekuensi. Ini membuat kode Huffman dapat efektif digunakan pada sistem yang lebih memprioritaskan kompresi data dibandingkan kompleksitas program, dan memiliki distribusi frekuensi yang dapat diprediksi.

VI. KESIMPULAN

Penelitian ini membahas tentang penerapan dari kode Huffman untuk optimasi representasi data pada layar 7 segmen, sebuah komponen yang sering digunakan untuk menunjukkan angka secara digital pada alat-alat sederhana. Penelitian ini melakukan penerapan Huffman kepada penerapan waktu 24 jam dengan format HH:MM, dan membandingkannya dengan metode BCD.

Hasil dari penerapan menunjukkan bahwa kode Huffman berhasil melakukan kompresi yang lebih efektif, dengan rasio kompresi sebesar 0.45 dan persentase penghematan sebesar 55.17%, dibandingkan dengan BCD yang hanya mencapai rasio kompresi sebesar 0.57 dan persentase penghematan sebesar 42.85%. Hasil ini menunjukkan efisiensi kode Huffman dengan data yang mempunyai data frekuensi yang tidak merata. Walaupun begitu, kode Huffman mempunyai beberapa kelemahan dibandingkan metode lain, seperti beban pemrosesan yang lebih berat. Penerapan kode Huffman maupun metode lain harus digunakan sesuai dengan kebutuhan sistem, dengan mempertimbangkan kelebihan dan kekurangan dari setiap metode.

REFERENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/matdis24-25.htm diakses pada 6 Januari 2025

[2] "Compression Ratio," ScienceDirect. https://www.sciencedirect.com/topics/computer-science/compression-ratio Diakses pada 6 Januari 2025.

[3] "7 Segment Display Pinout, Working, Examples, Applications," Microcontrollers Lab. https://microcontrollerslab.com/7-segment-display-pinout-working-examples-applications/ Diakses pada 6 Januari 2025.

[4] "Seven Segment Display Truth Table and Circuit Diagram," Electronic Clinic. https://www.electroniclinic.com/seven-segment-display-truth-table-and-circuit-diagram/ Diakses pada 6 Januari 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Januari 2025

Richard Christian 13523024