

Aplikasi Kode Huffman untuk Kompresi Data Partitur Musik Digital

Fazel Ginanda - 13521098
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13521098@std.stei.itb.ac.id

Abstrak—Partitur musik merupakan salah satu komponen penting dalam pelaksanaan aktivitas bermusik. Sebagian besar karya musik didokumentasikan dalam bentuk partitur musik. Saat ini, sudah banyak tersedia berbagai partitur digital yang dapat diakses melalui perangkat elektronik. Kebutuhan penyimpanan data yang terus meningkat menyebabkan keperluan untuk meminimalisasi penyimpanan data di memori. Salah satu metode kompresi data adalah dengan menggunakan kode Huffman.

Keywords—Kode Huffman, Partitur, Not Balok, Pohon.

I. PENDAHULUAN

Musik adalah bagian dari kebudayaan dan peradaban manusia. Musik juga merepresentasikan dinamika masyarakat pada suatu zaman. Secara praktis, musik menjadi bagian dari gaya hidup masyarakat. Hal ini dikarenakan musik menjadi hiburan bagi seluruh orang.

Sifat musik yang universal dan inklusif berasal dari sumber musik itu sendiri. Pada dasarnya, setiap musik yang dibuat oleh manusia merupakan ekspresi dari pembuat musik atau mengekspresikan lingkungan di sekitar pembuat musik itu.

Dalam kegiatan bermusik, setiap pemain musik memerlukan partitur, yaitu dokumentasi not balok dari suatu lagu. Budaya baca-tulis ini sudah lama diterapkan dan dipertahankan oleh berbagai peradaban yang mengembangkan musik. Metode ini dipilih agar suatu karya musik dapat dipelajari dan dimainkan tanpa terikat oleh masa hidup dari pemain musik itu sendiri. Selain itu, budaya baca-tulis dalam permainan musik juga ditujukan agar setiap karya dapat terdokumentasi dengan baik. Pendokumentasian ini memudahkan komposer dalam memperbaiki karya-karya yang sudah ada sebelumnya, maupun menjadi pelajaran dalam penulisan karya yang akan datang. Dengan demikian, budaya baca-tulis melalui partitur musik telah mempertahankan aspek keberlanjutan dalam suatu karya musik.

Saat ini, partitur musik dapat diakses melalui berbagai perangkat elektronik, seperti gawai, *laptop*, dan sebagainya. Banyak pemain musik yang lebih memilih untuk menggunakan gawai dalam mengakses partitur musik karena ringkas, praktis, dan mudah menambah koleksi dari partiture yang dimiliki.

Seiring bertambahnya data yang disimpan, maka kebutuhan penyimpanan pada ruang memori juga meningkat. Oleh karena itu, penulis mencari suatu alternatif dalam kompresi data partitur

musik digital menggunakan kode Huffman.

II. LANDASAN TEORI

A. Graf

Graf adalah struktur data yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf dinyatakan sebagai pasangan himpunan simpul dan sisi. Notasi dari suatu graf adalah sebagai berikut.

$$G = (V, E)$$

Pada notasi tersebut G menyatakan graf. V adalah himpunan tidak kosong dari simpul-simpul (*vertices*). Sementara itu, E adalah himpunan sisi (*edges*) yang menghubungkan sepasang simpul.

Berdasarkan keberadaan gelang atau sisi ganda pada suatu graf, graf dibedakan menjadi dua jenis, yaitu graf sederhana dan graf ganda. Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda. Graf tak sederhana adalah graf yang mengandung sisi ganda atau gelang.

Graf tak sederhana dibedakan menjadi graf ganda dan graf semu. Graf ganda adalah graf yang mengandung sisi ganda. Sementara itu, graf semu adalah graf yang mengandung sisi gelang.

Berdasarkan orientasi arah pada graf, graf dibedakan menjadi dua jenis, yaitu, graf tak berarah dan graf berarah. Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah.

Dua buah dalam suatu graf disebut bertetangga jika dua simpul tersebut dihubungkan oleh sisi. Pada graf tak berarah, ketetanggaan hanya ditinjau berdasarkan keberadaan sisi yang menghubungkan dua buah simpul. Akan tetapi, pada graf berarah, ketetanggaan ditinjau berdasarkan arah dari sisi yang menghubungkan kedua simpul. Misalkan terdapat simpul A dan simpul B dan sebuah sisi yang mengarah dari simpul A ke simpul B . Dengan demikian, dikatakan bahwa simpul A bertetangga dengan simpul B . Akan tetapi, simpul B tidak bertetangga dengan simpul A .

Kebersisian dari suatu simpul dan sisi ditentukan berdasarkan sisi yang berkoresponden dengan simpul-simpul. Untuk sembarang sisi $e = (v_j, v_k)$, dikatakan e bersisian dengan simpul v_j atau v_k .

Simpul memiliki besaran derajat. Derajat didefinisikan

sebagai jumlah sisi yang bersisian dengan simpul tersebut. Pada graf berarah, derajat simpul dibedakan menjadi derajat masuk dan derajat keluar. Jumlah derajat seluruh simpul pada graf selalu genap. Hal ini ditegaskan dalam lemma jabat tangan. Lemma ini menyatakan bahwa jumlah derajat seluruh simpul dalam suatu graf sama dengan dua kali jumlah sisi pada graf tersebut. Lemma jabat tangan juga mengimplikasikan simpul berderajat ganjil dalam suatu graf selalu berjumlah genap. Hal ini disebabkan simpul berderajat ganjil yang berjumlah ganjil menyebabkan jumlah derajat seluruh simpul menjadi ganjil. Ini bertentangan dengan lemma jabat tangan.

Graf yang himpunan sisinya tidak kosong memiliki lintasan. Lintasan adalah barisan berselang-seling antara simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$. Lintasan memiliki besaran panjang, Panjang dari suatu lintasan adalah jumlah sisi di dalam lintasan tersebut.

Terdapat terminologi lain dari graf yang berhubungan dengan lintasan yaitu sirkuit. Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Sirkuit disebut juga sebagai siklus. Sirkuit memiliki besaran panjang yang didefinisikan sebagai jumlah sisi dalam sirkuit tersebut.

Keterhubungan dalam suatu graf ditentukan berdasarkan hubungan antara simpul-simpul pada graf tersebut. Dua buah simpul disebut terhubung jika terdapat lintasan antara kedua simpul tersebut. Jika tidak terdapat lintasan antara kedua simpul tersebut, maka kedua simpul dikatakan tidak terhubung.

Graf terhubung adalah graf yang mengandung lintasan untuk setiap pasang simpul yang terdapat pada graf tersebut. Jika suatu graf mengandung pasangan simpul yang tidak memiliki lintasan, maka graf tersebut graf tak terhubung.

Graf berarah dikatakan terhubung jika graf tidak berarahnya terhubung. Graf tidak berarah dari graf berarah diperoleh dengan menghilangkan orientasi arah pada setiap sisi yang ada pada graf. Selain itu, pada graf berarah, terdapat istilah terhubung kuat dan terhubung lemah. Dua simpul dikatakan terhubung kuat jika terdapat lintasan berarah antara kedua simpul tersebut. Jika dua buah simpul pada graf berarah tidak terhubung kuat dan terhubung pada graf tidak berarahnya, maka dua buah simpul tersebut dikatakan terhubung lemah.

Graf berarah secara keseluruhan juga dapat ditentukan keterhubungannya. Jika setiap pasang simpul pada graf berarah terhubung kuat, maka graf berarah tersebut disebut graf terhubung kuat. Akan tetapi, jika terdapat sekurang-kurangnya satu pasang simpul yang tidak terhubung kuat, graf berarah tersebut disebut graf terhubung lemah.

Upagraf dari suatu graf adalah graf yang himpunan sisinya merupakan himpunan bagian dari himpunan sisi dari graf asalnya dan himpunan sisinya merupakan himpunan bagian dari himpunan sisi dari graf asalnya. Upagraf memiliki komplemen terhadap graf asalnya. Komplemen dari upagraf didefinisikan sebagai graf yang himpunan sisinya adalah selisih dari himpunan sisi pada graf asal dengan himpunan sisi pada upagraf. Sementara itu, anggota himpunan simpul dari komplemen upagraf terdiri dari simpul-simpul yang bersisian dengan himpunan sisi komplemen upagraf tersebut.

Suatu graf memiliki komponen. Komponen graf adalah jumlah maksimum upagraf terhubung dalam graf. Pada graf berarah, terdapat istilah komponen terhubung kuat. Komponen

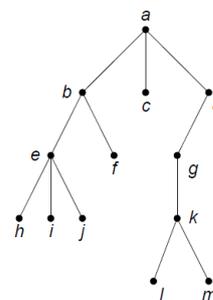
terhubung kuat pada graf berarah adalah jumlah maksimum upagraf yang terhubung kuat pada graf berarah tersebut.

B. Pohon

Pohon adalah graf tak berarah terhubung yang tidak mengandung sirkuit. Berdasarkan teorema, pohon memiliki sifat-sifat yang harus dipenuhi. Pertama, setiap pasang simpul di dalam pohon terhubung dengan lintasan tunggal. Kedua, pohon tidak mengandung sirkuit. Ketiga, penambahan satu sisi pada pohon hanya menyebabkan pembentukan satu buah sirkuit. Keempat, pohon terhubung dan semua sisinya adalah jembatan.

Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Tanda panah yang selalu dituliskan dalam penggambaran graf berarah boleh dihilangkan dalam penggambaran pohon.

Graf berarah yang memiliki lebih dari dua simpul dapat menghasilkan pohon berakar yang berbeda. Perbedaan itu disebabkan oleh pemilihan akar yang berbeda. Hal itu menegaskan bahwa pemilihan akar berperan penting dalam pembentukan sebuah pohon.



Gambar 1 Pohon Berakar

(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

Pada gambar 1, simpul a adalah orangtua dari simpul b,c, dan d. Sementara itu, simpul b,c, dan d adalah anak dari simpul a.

Sama seperti graf, pohon juga memiliki istilah lintasan. Lintasan pada pohon didefinisikan sebagai barisan berselang-seling antara simpul-simpul dan sisi-sisi. Pada gambar 1, lintasan dari a ke j adalah a, b, e, j. Panjang dari lintasan tersebut adalah tiga.

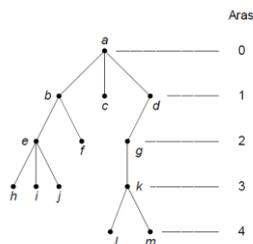
Sebuah simpul memiliki saudara kandung jika orangtuanya memiliki anak selain simpul tersebut. Jika orangtua dari sebuah simpul tidak memiliki anak yang lain, maka simpul tersebut tidak memiliki saudara kandung. Pada graf 1, simpul f adalah saudara kandung simpul e. Alasannya adalah simpul f dan e memiliki orangtua yang sama, yaitu simpul b. Akan tetapi, simpul g bukan saudara kandung simpul f, karena orangtua mereka berbeda.

Setiap simpul pada pohon memiliki besaran derajat. Derajat dari sebuah simpul dihitung berdasarkan jumlah anak yang dimiliki oleh simpul tersebut. Dengan kata lain, derajat suatu simpul pada pohon sama dengan derajat keluar dari suatu simpul pada graf berarah.

Pada gambar 1, derajat simpul a, b, c, d secara berturut-turut adalah tiga, dua, nol, dan satu. Selain itu, pada pohon juga

dikenal istilah derajat pohon. Derajat dari suatu pohon adalah derajat maksimum dari semua simpul.

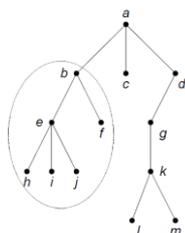
Simpul yang berderajat nol atau simpul yang tidak mempunyai anak disebut daun. Pada gambar 1, simpul h, i, j, f, c, l, dan m adalah daun. Di sisi lain, simpul yang memiliki anak disebut simpul dalam. Pada gambar 1, simpul b, d, e, g, k adalah simpul dalam.



Gambar 2 Aras Pohon
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

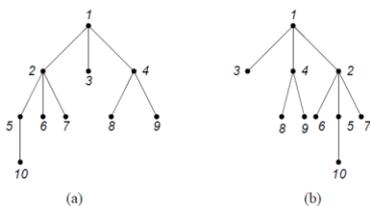
Penghitungan aras pada pohon dimulai dari akar. Akar pohon berada pada aras 0. Selain itu, pada pohon, terdapat istilah tinggi atau kedalaman. Tinggi atau kedalaman didefinisikan sebagai aras maksimum dari suatu pohon. Pohon pada gambar 2 mempunyai tinggi 4.



Gambar 3 Upapohon
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

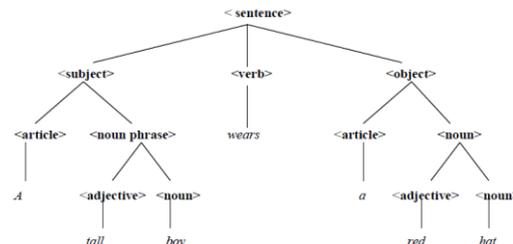
Pohon juga dapat didefinisikan secara rekursif. Elemen dari suatu pohon adalah akar dan elemen lain yang dibagi-bagi menjadi beberapa upahimpunan dan masing-masing upahimpunan tersebut adalah pohon. Elemen yang disebutkan terakhir itu disebut juga sebagai upapohon. Pada gambar 3, simpul a adalah akar yang mempunyai anak, yaitu simpul b. Simpul b disebut upapohon dari pohon dengan simpul a sebagai akar.



Gambar 4 Pohon Terurut
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

Pohon terurut adalah pohon berakar yang urutan anak-anaknya penting. Pada gambar 4, pohon (a) dan pohon (b) adalah dua pohon terurut yang berbeda. Walaupun pohon (a) dan pohon (b) memiliki pasangan orangtua dan anak yang sama, urutan dari anak-anak berbeda.

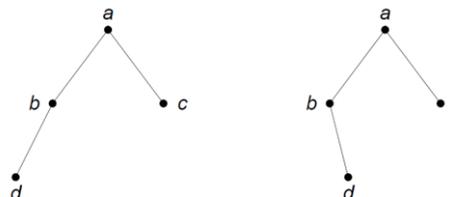


Gambar 5 Pohon n-ary
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

Pohon n-ary adalah pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak. Pada gambar 5, simpul cabang yang memiliki anak paling banyak adalah simpul akar dengan anak sebanyak tiga buah. Oleh karena itu, pohon tersebut disebut dengan pohon 3-ary.

Pohon biner adalah pohon n-ary dengan n bernilai 2. Pohon ini adalah pohon yang paling penting karena penerapannya sangat banyak. Setiap simpul pada pohon biner mempunyai paling banyak dua buah anak. Setiap simpul akar pada pohon biner mempunyai anak kiri dan anak kanan. Oleh karena ada perbedaan urutan anak, maka pohon biner adalah pohon terurut.



Gambar 6 Dua Pohon Biner yang Berbeda
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>, diakses pada 12 Desember 2022)

Pada gambar 6, terdapat dua buah pohon biner. Kedua pohon biner berbeda karena terdapat perbedaan urutan anak pada kedua pohon itu. Pada pohon di sebelah kiri, d merupakan anak kiri dari b. Akan tetapi, pada pohon di sebelah kanan, d merupakan anak kanan dari b. Dengan demikian, kedua pohon dinyatakan berbeda.

Dalam beberapa aplikasi, seperti pengelolaan indeks dalam file system dan database system digunakan pohon seimbang. Pohon seimbang adalah pohon biner dengan perbedaan ketinggian antara upapohon kiri dengan upapohon kanan paling banyak satu. Selain itu, upapohon kiri dan upapohon kanan dari pohon seimbang juga merupakan pohon seimbang. Pohon

seimbang yang setiap simpulnya memiliki paling banyak dua buah anak disebut pohon biner seimbang.

C. Pengkodean Karakter UTF-8

Pengkodean karakter adalah proses memberikan nilai angka ke karakter grafis, khususnya karakter dari bahasa alami agar bisa disimpan, ditransmisikan, dan diubah menggunakan komputer digital. Nilai numerik yang membentuk pengkodean karakter disebut titik kode.

UTF-8 memiliki kemampuan untuk mengkodekan sebanyak 1.112.064 karakter valid pada *Unicode* menggunakan satu byte dengan susunan bit yang sama seperti ASCII. Oleh karena itu, UTF-8 memiliki kompatibilitas yang memadai dengan ASCII.

D. Kode Huffman

Kode Huffman adalah kode prefix yang digunakan untuk kompresi data. Algoritma pengkodean Huffman dikembangkan oleh David A. Huffman pada tahun 1952.

Algoritma pengkodean Huffman menerima himpunan simbol dan bobotnya. Dalam hal ini, bobot dimaknai sebagai frekuensi kemunculan suatu simbol. Algoritma ini mencari kode biner dengan panjang minimum. Kode biner dengan panjang minimum ini dapat direpresentasikan sebagai pohon dengan bobot lintasan minimum dari akar.

Mekanisme algoritma pengkodean Huffman dimulai dengan memilih dua buah simbol dengan peluang paling kecil. Kedua simbol itu digabungkan menjadi simpul orangtua dengan peluang sebesar jumlah dari peluang kedua anaknya. Langkah tersebut diulangi terus-menerus hingga seluruh simbol habis. Pada pohon hasil pembentukan kode Huffman itu, sisi kiri dilabeli dengan 0. Sementara itu, sisi kanan dilabeli dengan 1. Kode Huffman untuk sebuah simbol sama dengan deretan label pada lintasan dari akar ke simpul daun dari simbol tersebut.

E. Not Balok

Partitur musik terdiri dari suatu komponen penting yang membangun keseluruhan partiture, yaitu not balok. Not balok adalah representasi nada dalam bentuk tulisan. Not balok berada dalam garis paranada yang didahului oleh kunci atau nada dasar. Terdapat tiga jenis nada dasar, yaitu G, F, dan C.



Gambar 7 Not Balok pada Kunci G

(Sumber: <https://www.musictheoryacademy.com/wp-content/uploads/2020/06/Treble-Clef-Notes.jpg>)

III. PEMBAHASAN

A. Pengkodean UTF-8 untuk Simbol Musik

Dalam aturan pengkodean UTF-8, simbol musik berada pada rentang 1D100 hingga 1D1FF. Ini berarti untuk setiap simbol, diperlukan 3 byte atau 24 bit.

	1D10	1D11	1D12	1D13	1D14	1D15	1D16	1D17	1D18	1D19	1D1A	1D1B	1D1C	1D1D	1D1E	1D1F
0	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
1	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
2	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
3	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
4	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
5	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
6	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
7	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
8	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
9	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
A	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
B	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
C	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
D	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
E	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒
F	🎵	🎶	🎷	🎸	🎹	🎺	🎻	🎼	🎽	🎾	🎿	🏎	🏏	🏐	🏑	🏒

Gambar 8 Kode UTF-8 untuk Simbol Musik
(Sumber: <https://www.unicode.org/charts/PDF/U1D100.pdf>)

B. Kode Huffman untuk Simbol Musik

Penerapan algoritma Huffman dalam mengkodekan setiap simbol dalam partiture musik dapat dilakukan dengan cara yang sama dengan mengkodekan karakter. Hal yang menjadi pembeda adalah setiap not balok yang dibaca harus dibedakan terlebih dahulu berdasarkan posisi oktafnya dan keberadaan tanda kromatik pada not tersebut. Hal itu dapat diselesaikan dengan membuat bit tambahan sebagai penanda pada setiap pengkodean not balok.

Langkah-langkah pengkodean not balok menggunakan kode Huffman secara rinci sebagai berikut. Pertama, baca kunci atau nada dasar dari suatu partitur. Oleh karena hanya terdapat tiga jenis nada dasar, maka informasi ini dapat disimpan pada dua bit pertama. Kedua, baca setiap not balok pada partitur, dimulai dari not pertama sambil menghitung kemunculan setiap not balok tersebut. Jika ditemui tanda ulang, maka simpan posisi tanda ulang tersebut beserta posisi pengulangan lagu. Ketiga, pilih dua not dengan kemunculan paling sedikit. Kedua not itu lalu digabungkan menjadi simpul orangtua dengan peluang sebesar jumlah dari kedua simpul anaknya. Langkah tersebut diulangi terus menerus hingga tidak ada lagi not yang tersisa. Setiap pemberian label pada sisi dari pohon, sisi kiri selalu diberi label 0, sedangkan sisi kanan diberi label 1. Not balok pada pohon hasil penerapan algoritma pengkodean Huffman berada pada simpul daun. Kode Huffman untuk setiap not balok dan simbol-simbol musik lainnya adalah deretan label 0 dan 1 dari akar sampai ke simpul daun tersebut. Dengan demikian, pengkodean partiture musik dapat dilakukan dengan ringkas dan efektif karena jumlah bit untuk setiap not balok bergantung pada frekuensi kemunculannya pada suatu lagu. Ini berbanding terbalik dengan pengkodean UTF-8 yang memerlukan

kebutuhan memori yang lebih besar sebab not balok termasuk simbol yang unik.

IV. SIMPULAN

Kode Huffman merupakan salah satu metode kompresi data teks. Algoritma pengkodean Huffman membuat kebutuhan ruang memori menjadi lebih sedikit karena mengubah representasi bit dari suatu data. Representasi suatu data dengan pengkodean standar belum meminimalkan kebutuhan ruang memori. Dengan memanfaatkan kode Huffman, representasi bit dari setiap simbol dapat dimanipulasi dengan menghitung frekuensinya terlebih dahulu. Dalam makalah ini, telah dibahas penerapan kode Huffman untuk kompresi data partitur musik digital. Hal yang bisa dikembangkan menjadi pembahasan selanjutnya adalah mengimplementasikan gagasan ini secara nyata pada perangkat lunak aplikasi partiture musik digital sehingga kebutuhan ruang memori yang minimum dapat tercapai.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Esa atas berkat dan rahmat yang diberikan, penulis mampu menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada orangtua yang selalu memberikan dukungan dalam pengerjaan tugas ini. Penulis juga mengucapkan terima kasih kepada Ibu Nur Ulfa Maulidevi yang telah memberikan pengajaran dan bimbingan kepada penulis selama satu semester ini, serta kepada Bapak Rinaldi Munir yang telah menyediakan berbagai bahan kuliah matematika diskrit.

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf> (Diakses pada tanggal 12 Desember 2022)
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf> (Diakses pada tanggal 12 Desember 2022)
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf> (Diakses pada tanggal 12 Desember 2022)
- [4] <https://www.unicode.org/charts/PDF/UID100.pdf> (Diakses pada tanggal 12 Desember 2022)
- [5] <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/> (Diakses pada tanggal 12 Desember 2022)

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, 12 Desember 2022



Fazel Ginanda - 13521098