

PENERAPAN ALGORITMA PREFIX CODE DALAM FILE DATABASE MySQL

Hikmah Abdul Aziz Samudra¹, Berto Nadeak², Rian Syahputra³
^{1,2,3}Universitas Budi Darma, Medan
Email: ¹azissamudra8@gmail.com, ²nadeak.baru@gmail.com,
³ryansyah93@gmail.com

ABSTRAK

Di era saat ini, kemajuan teknologi telah mengakibatkan kemajuan yang pesat, khususnya dalam media penyimpanan yang menghasilkan data informasi yang krusial. Akan tetapi, ruang yang tersedia untuk menyimpan data terbatas, dan tidak semua orang memiliki akses internet berkecepatan tinggi. Oleh karena itu, mentransfer data dalam jumlah besar melalui internet merupakan tantangan tersendiri. Kecepatan dalam transfer data sangat penting untuk pertukaran informasi, khususnya data tekstual. Kompresi data digunakan untuk mengurangi ukuran data, yang melibatkan pengkodean informasi menggunakan bit atau unit pembawa lainnya yang tidak mewakili data yang tidak dikodekan oleh sistem pengkodean tertentu. *Database MySQL* banyak digunakan untuk manajemen data, tetapi memiliki keterbatasan dalam kapasitas ukuran berkas dan memerlukan akses internet yang cepat untuk menyimpan sejumlah besar data. Penerapan algoritma *prefix code* untuk kompresi data dalam berkas *database MySQL* dapat mengurangi ukuran berkas tanpa kehilangan informasi, sehingga memudahkan pemanfaatan ruang penyimpanan. *Algoritma prefix code* melibatkan pengkodean setiap karakter pesan atau data dengan kode yang lebih pendek sambil mempertahankan konten informasi dengan menggunakan rancangan aplikasi dengan menggunakan *Microsoft visual studio 2008*. Penelitian sebelumnya telah menunjukkan efektivitas algoritma *prefix code* dalam kompresi data, yang mengarah pada peningkatan kinerja dan kemudahan penerapan. Penelitian di masa mendatang dapat mengeksplorasi lebih jauh aspek metode kompresi data dan implikasi praktisnya. Hasil dari implementasi kompresi data dengan menggunakan algoritma *prefix code* menggunakan kode C1 file *MySQL* sebesar 72% rasio pada file tersebut.

Kata Kunci: *kompresi file , prefix code, MySQL*

ABSTRACT

In today's era, technological advancements have led to rapid progress, particularly in storage media that produce crucial data and information. However, the available space for data storage is limited, and not everyone has access to high-speed internet. Therefore, transferring large amounts of data over the internet presents its own challenges. The speed of data transfer is essential for the exchange of information, especially textual data. Data compression is used to reduce the size of data by encoding information using bits or other carrier units that do not represent unencoded data within a specific coding system. MySQL databases are widely used for data management, but they face limitations in file size capacity and require fast internet access to store large amounts of data. The application of the prefix code algorithm for compressing data in MySQL database files can reduce

file size without losing information, thus optimizing storage space utilization. The prefix code algorithm encodes each character of a message or data with a shorter code while preserving the information content, implemented through application design using Microsoft Visual Studio 2008. Previous research has shown the effectiveness of the prefix code algorithm in data compression, leading to improved performance and ease of application. Future research could further explore the aspects of data compression methods and their practical implications. The results of the data compression implementation using the prefix code algorithm on a MySQL file with C1 code achieved a 72% compression ratio.

Keywords: file compression, prefix code, MySQL

A. Pendahuluan

Pada masa Sekarang ini kemajuan teknologi mengalami kemajuan yang sangat pesat. Salah satunya adalah media penyimpanan yang menghasilkan data informasi sangatlah penting. Namun, ruang yang tersedia untuk menyimpan data saat ini terbatas, dan tidak semua orang memiliki akses Internet berkecepatan tinggi. Oleh karena itu, sangat sulit untuk mentransfer data dalam jumlah besar melalui jaringan Internet. Kecepatan pengiriman akan menjadi bagian penting dalam pertukaran informasi saat ini dan masa depan seperti data tekstual (Saragih & Utomo, 2020), maka dari itu untuk memperkecil masa data menggunakan kompresi data, Kompresi berarti menggunakan atau memperkecil ukuran. Kompresi data melibatkan pengkodean informasi menggunakan bit atau unit pembawa informasi lainnya yang tidak mewakili

data yang tidak dikodekan oleh sistem pengkodean tertentu (Azhar, 2023);(Heri, 2024). Tujuan kompresi adalah meminimalkan jumlah bit yang dibutuhkan untuk merepresentasikan data (Yanti, 2021).

Pada saat ini penggunaan *database* sudah banyak ditemukan terkhususnya untuk manajemen data dalam sistem contohnya *Database MySQL, Database MySQL* atau *Structured Query Language* adalah jenis server *Database* yang sangat terkenal. *MySQL* adalah jenis *RDBMS (Relational Database Management System)*. *MySQL* mendukung bahasa pemrograman *PH*, bahasa query terstruktur. Hal ini karena ketika menggunakan *MySQL*, ada beberapa aturan yang distandarisi oleh asosiasi yang disebut ANSI (*American National Standards Institute*). *MySQL* adalah server *RDBMS (Relational Database Management System)*. *RDBMS*

adalah program yang memungkinkan pengguna database membuat, mengelola, dan menggunakan data dalam model relasional. Oleh karena itu, ada hubungan antar tabel dalam *database* dari satu tabel ke tabel lainnya. *Database MySQL* adalah data yang sering digunakan dalam perusahaan untuk mempermudah manajemen data, namun memiliki kelemahan dalam kapasitas ukuran file yang terbatas dan memerlukan kecepatan akses internet yang dimiliki pengguna untuk banyak menyimpan data yang mengakibatkan sedikitnya ruang penyimpanan memori yang digunakan, hal ini menjadi salah satu masalah untuk melakukan kompresi data untuk memperkecil ukuran file tanpa kehilangan informasi agar mempermudah penggunaan database untuk mendapatkan penyimpanan ruang yang lebih dalam penggunaan *Database* (Nasution, 2024);(Maulana, 2016).

Untuk memperkecil ukuran *Database* dengan menggunakan kompresi data di perlukan metode yang tepat untuk memperkecil kapasitas file data pada ruang penyimpanan dengan menggunakan jenis metode Kompresi *prefix code*

merupakan jenis kompresi tanpa kehilangan (*lossless compression*). *Prefix code* adalah metode pengkodean di mana tidak ada kode yang menjadi awalan (*Prefix*) dari kode lainnya, sehingga setiap kode dapat diidentifikasi secara unik dalam aliran data yang dikompresi (Alfarizi & Aripin, 2020). Artinya tidak ada satupun kata kode yang berfungsi sebagai awalan (segmen pertama) dari kata kode lain dalam sistem untuk mengurangi redundansi yang ada terutama dalam kompresi *data base My SQL*. Berdasarkan penelitian sebelumnya berjudul "*Decoding Prefix Codes*" yang diterbitkan dalam jurnal oleh *Mike Liddell* dan *Alistair Moffat*, kode awalan dengan redundansi minimal dikompresi secara komersial oleh *David Huffman* lebih dari 50 tahun yang lalu dan perkembangan sejak penemuannya (Liddell, M., & Moffat, 2006). Melalui percobaan, mereka mengevaluasi strategi pengkodean yang berbeda dengan pengulangan yang rendah dan menentukan keuntungan relatif dari pembuatan kode kecil khusus untuk tujuan decoding. Selain itu, penelitian yang dilakukan oleh Renato Pajarola dalam jurnal bertajuk "*Fast Prefix*

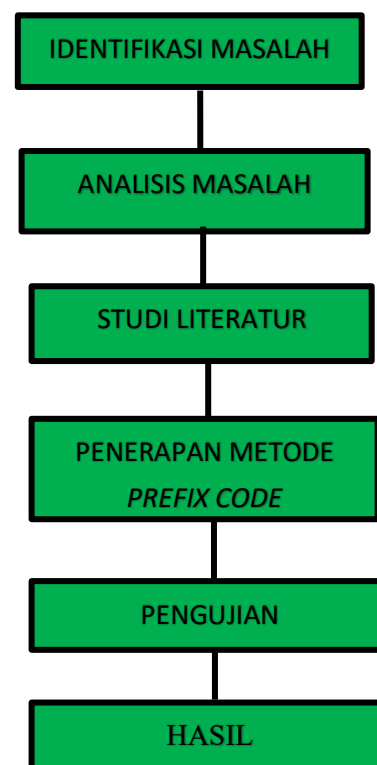
Code Processing" menyimpulkan bahwa dalam karyanya, ia menjelaskan penggunaan mesin keadaan terbatas untuk memproses kode awalan dengan cepat. Hal ini meningkatkan kinerja decoding, meningkatkan praktik, dan mudah diterapkan. (Saragih & Utomo, 2020)(Seregar, 2023)(Hermiati et al., 2021).

Solusi dari permasalahan ini adalah dengan mengompres file data yang tersimpan pada database *MySQL*, proses yang terlibat dalam kompresi ini menggunakan metode *prefix code* yang bertujuan untuk menghemat penyimpanan dan waktu yang diperlukan untuk mentransfer data, kompresi data dilakukan dengan mengkodekan setiap karakter pesan atau data dengan kode yang lebih pendek tetapi tidak menghilangkan makna atau informasi yang terdapat dalam file tersebut, dengan demikian untuk mencerminkan focus pada rumusan masalah tersebut penggunaan algoritma *prefix code* dalam data kompresi untuk menyertai aspek tujuan tersebut, penulis memutuskan mengangkat judul "**Penerapan Algoritma *Prefix code* dalam File Database *MySQL***".

B. Metode Penelitian

Kerangka Penelitian

Kerangka penelitian adalah sebuah langkah awal untuk melakukan tahapan tahapan yang akan dilakukan dalam mengidentifikasi hal-hal yang akan dibutuhkan dalam pemecahan masalah (Sarumaha et al., 2023), maka dari itu adapun kerangka penelitian yang dirancang sebagai berikut:



Gambar 1 Kerangka penelitian Berdasarkan kerangka penelitian yang disusun, langkah-langkah yang diambil untuk mencapai tujuan utama penelitian ini dimulai

dengan proses identifikasi masalah, di mana permasalahan yang diangkat adalah kompresi data file SQL menggunakan algoritma Prefix Code. Analisis masalah dilakukan dengan mempertimbangkan penerapan algoritma tersebut yang bersifat reversibel, sehingga memungkinkan dekompresi tanpa kehilangan data. Penulis juga melakukan studi literatur dengan membaca berbagai penelitian terkait kompresi data teks dan algoritma Prefix Code sebagai pedoman. Selanjutnya, algoritma Prefix Code diterapkan dalam proses kompresi lossless, yang menghasilkan data kompresi yang identik dengan data asli. Pengujian sistem dilakukan untuk memastikan fungsionalitasnya, dan hasil akhirnya menunjukkan bahwa penerapan algoritma ini pada PHP My Admin berjalan sesuai tujuan awal, yaitu tanpa kehilangan data asli.

Sample Data

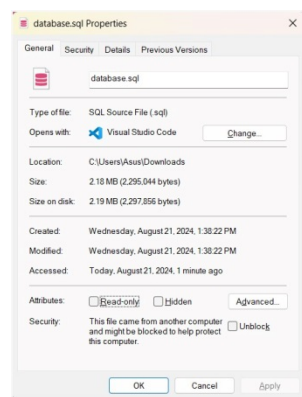
Dalam penelitian ini sampel data ini dijadikan sebagai objek untuk melakukan kompresi dalam penelitian ini, maka dari itu diperlukan sample data yang akan menjadi objek utamanya, dalam mengkompresi data dibutuhkan *file SQL source file* yang digunakan sebagai format, dengan

menggunakan *algoritma prefix*, dengan menggunakan kompresi ini dapat mempermudah pengiriman, penyimpanan data tersebut, adapun sample *file SQL source* yang akan digunakan sebagai berikut:

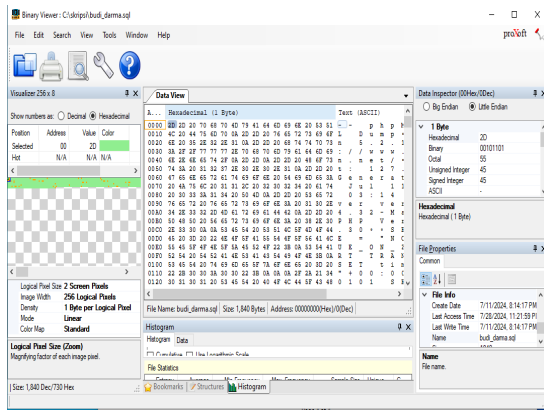
Tabel 1 Sample Data

Jenis file	<i>SQL source file</i>
Nama file	budi_darma.sql
Ukuran file	2,18 MB

Dari sample file SQL source File pada table dapat menggunakan aplikasi binary viewer untuk mencari nilai bilangan biner serta mencari nilai bilangan heksadesimal maupun frekuensinya. Adapun langkah mencari bilangan heksadesimal sebagai berikut:



Gambar 2 Informasi *File SQL*



Gambar 3 file Database MySQL

Dari sampel data diatas di ambil menggunakan binary viewer untuk mencari bilangan heksadesimal, adapun tabel dari bilang yang akan dikompresi sebagai berikut:

Tabel 2 Tabel Binary

001011	001011	001000	011100
01	01	00	00
011010	011100	010011	011110

Tabel 3 Tabel frekuensi sebelum kompresi

No	Hexadecimal	Binner	Bit	Frekuensi	Frek * Bit
1	77	01110111	8	3	24
2	2F	00101111	8	2	16
3	6E	01101110	8	1	8
4	65	01100101	8	1	8
5	74	01110100	8	1	8
Total Bit					64

Proses selanjutnya adalah yang di dapat dari tabel diatas. melakukan kompresi nilai Berikut merupakan proses kompresi hexadecimal dengan nilai kode C1 pada sampel data diatas.

Tabel 4 Tabel frekuensi sesudah kompresi

No	Hexadecimal	Codeword	Bit	Frekuensi	Frek * Bit
1	77	11	2	3	6
2	2F	01	2	2	4

00	00	01	01
010000	011001	011011	011010
01	00	01	01
011011	001000	010100	010100
10	00	11	01

C. Hasil Penelitian dan Pembahasan Penetapan Data Kriteria

Penerapan Algoritma Prefix Code

Pada algoritma *prefix code* tahapan pertama yang dilakukan adalah dengan melakukan penyusunan frekuensi pada sampel data yang digunakan. Berikut merupakan tahapan penerapan kompresi menggunakan Algoritma *Prefix Code*:

Penyusunan statistic frekuensi.

3	6E	100	3	1	3
4	65	101	3	1	3
5	74	00	2	1	2
Total Bit					18

Selanjutnya adalah menggabungkan seluruh string bit yang dihasilkan berdasarkan *codeword*. Berikut string bit yang dihasilkan : "011111111001010001" Dengan total bit sebanyak 18.

Berdasarkan perhitungan rasio kompresi menggunakan algoritma prefix code, ukuran awal data yang sebesar 512 bit berhasil dikompresi menjadi 144 bit. Rasio kompresi yang diperoleh adalah 72%, menunjukkan bahwa ukuran file berkurang secara signifikan setelah proses kompresi. Pada tahap dekompresi, data berhasil dikembalikan ke bentuk semula dengan memanfaatkan kamus kode yang telah dibuat. Proses dekompresi mengikuti urutan string bit yang dihasilkan dari proses kompresi, dan data asli berupa "2F 77 77 77 6E 65 74 2F" berhasil dipulihkan secara sempurna. Hal ini membuktikan bahwa algoritma prefix code tidak hanya efektif dalam mengurangi ukuran file, tetapi juga mampu mengembalikan data kompresi tanpa kehilangan informasi.

Percancangan Model dan Perancangan Program

Perancangan dan pemodelan merupakan tahap merancang dan memodelkan program berdasarkan hasil Analisa dengan tujuan agar dapat mempermudah dalam membangun sistem dan mudah diapahami oleh pengguna.

Use Case Diagram

Use case diagram adalah sebuah gambaran sistem yang akan dirancang dengan penjelasan fungsi ideal yang mudah antara pengguna dan sistem. *Use case diagram* dalam penelitian ini menggambarkan intraksi antara *user* dan sistem dalam proses kompresi *File database MySQL* dengan *algoritma prefix code*, dengan menggunakan rancangan dan pembuatan intraksi yang simple dan mudah dimengerti.

Perancangan Program

Rancangan program merupakan representasi hasil rancangan model yang telah dibuat sebelumnya dengan menggunakan *Microsoft visual Basic 2008*, berikut adalah hasil dari perancangan program yang dibuat tersebut sebagai berikut:



Gambar 4 Form Menu Utama menggunakan *Microsoft Visual Basic2008*

Ket:

pada menu utama terdapat dua tombol yaitu tombol kompresi yang fungsinya untuk melakukan proses kompresi dan keluar untuk keluar dari aplikasi.



Gambar 5 Form menu kompresi menggunakan *Microsoft visual Basic 2008*

Ket :

Form menu kompresi merupakan form yang melakukan proses kompresi Dimana user memasukkan file yang akan dikompresi.

Hasil Pengujian

Pada hasil pengujian yang telah dilakukan berdasarkan program yang telah dirancang menunjukkan bahwa program berjalan sesuai dengan inputan yang diberikan. Pengujian dilakukan pada 3 file *database* dapat dilihat dibawah ini.

Tabel.5 Hasil pengujian

No	Nama File	Ukuran File Awal	Ukuran File Kompresi
1	Database.sql	2.18mb	371 Kb
2	Dbperpus.sql	1,67 mb	161 Kb
3	Dbmahasiswa.sql	3mb	1.24 mb

D. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan tentang penerapan kompresi file database MySQL dengan menggunakan algoritma prefix code, dapat ditarik beberapa kesimpulan yang signifikan mengenai efektivitas dan keandalannya.

Algoritma prefix code terbukti mampu mengkompresi file database MySQL secara efisien serta melakukan proses dekompresi tanpa kehilangan data, sehingga menjaga integritas informasi dalam file. Proses kompresi ini berhasil mengurangi ukuran file database MySQL secara substansial,

yaitu dari ukuran asli 2,18 MB menjadi hanya 371 KB. Hal ini memberikan dampak positif dalam mengoptimalkan penggunaan ruang penyimpanan, yang sangat penting dalam manajemen database, terutama ketika berhadapan dengan skala data yang besar dan terbatasnya kapasitas penyimpanan. Selain itu, algoritma ini juga memungkinkan penerapan yang praktis melalui perancangan dan pembangunan perangkat lunak kompresi dengan menggunakan *Microsoft Visual Basic 2008*. Platform ini memberikan kemudahan bagi penulis dalam mengimplementasikan algoritma kompresi, sehingga proses kompresi dan dekompresi dapat dilakukan dengan lebih efisien dan mudah dioperasikan. Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma prefix code adalah solusi yang efektif dalam mengurangi ukuran file database tanpa mengorbankan integritas data, serta dapat diterapkan dalam lingkungan pengembangan perangkat lunak yang umum digunakan, seperti *Microsoft Visual Basic*, untuk mendukung efisiensi operasional dalam pengelolaan data yang lebih baik.

DAFTAR PUSTAKA

- Alfarizi, M., & Aripin, S. (2020). Penarapan Algoritma Prefix Code Dalam Kompresi File Video. *KOMIK (Konferensi Nasional ...)*, 4(1), 249–252.
- Azhar, A. (2023). Kompresi Record Database MySQL Dengan Menerapkan Algoritma Goldbach Code G1. *KOMIK (Konferensi Nasional Teknologi ...)*, 6(November), 279–290. <https://doi.org/10.30865/komik.v6i1.5711>
- Heri, A. (2024). Perbandingan Algoritma Elias Gamma Code Dan Puncturead Elias Code Dalam Mengkompresi File Audio MP3. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, 7(1), 108–115.
- Hermiati, R., Asnawati, A., & Kanedi, I. (2021). Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql. *Jurnal Media Infotama*, 17(1), 54–66. <https://doi.org/10.37676/jmi.v17i1.1317>
- Liddell, M., & Moffat, A. (2006). *Decoding prefix codes. Software: Practice and Experience*, 36(15).
- Maulana, H. (2016). ANALISIS DAN PERANCANGAN SISTEM REPLIKASI DATABASE MYSQL DENGAN MENGGUNAKAN VMWARE PADA SISTEM. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 1(1), 32–37.
- Nasution, S. D. (2024). *Pengamanan Perintah Koneksi ke Database*

MySQL Menggunakan Algoritma Caesar Cipher dan Algoritma Stout Codes. 5(1), 9–16.
<https://doi.org/10.47065/bit.v5i1.1149>

Saragih, S. R., & Utomo, D. P. (2020). Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks. *KOMIK (Konferensi Nasional ...)*, 4(1), 249–252.

Sarumaha, K. Y., Syahrizal, M., & Siagian, E. R. (2023). Analisis Perbandingan Algoritma Elias Delta Code Dengan Algoritma Prefix Code Dalam Mengkompresi Data Teks. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, 6(1), 460–469.
<https://doi.org/10.30865/komik.v6i1.5698>

Seregar, N. (2023). Penerapan Algoritma Lzy Untuk Mengkompresi Record Database Mysql. *Jurnal Ilmu Komputer, Teknologi Dan Informasi*, 1(2), 43–50.

Yanti, M. (2021). Aplikasi Kompresi Citra Dengan Menerapkan Algoritma SPIHT (Set Partitioning In Hierarchical Trees). *Journal Global Tecnology Computer*, 1(1), 20–28.