

BAB II

LANDASAN TEORI

2.1 Kompresi Data

Kompresi data merupakan salah satu cabang ilmu komputer yang diturunkan dari teori informasi. Teori informasi berkaitan dengan berbagai metode informasi, termasuk penyimpanan dan pemrosesan pesan. Teori informasi juga mempelajari redundansi (informasi tidak berguna) dalam pesan. Semakin banyak redundansi yang ada, semakin besar pesannya. Keinginan untuk mengurangi redundansi pada akhirnya menjadi alasan berkembangnya kompresi data (Hondro, 2024): Teknik kompresi data dapat dibagi menjadi 2 kategori besar, yaitu :

1. *Kompresi lossless* adalah kompresi yang mempertahankan kesetiaan data tetapi tidak membiarkan informasi apa pun diubah atau hilang, meskipun hanya beberapa bit dalam data, selama proses kompresi. Oleh karena itu, teknik kompresi ini bersifat lossless. Hasil kompresi dapat diubah kembali ke format asli. Teknik ini sangat cocok untuk berkas teks, gambar medis, gambar satelit, dll., di mana hilangnya beberapa detail data dapat berakibat fatal. Contoh algoritma lossless adalah pengkodean aritmatika dan pengkodean Huffman. (Arfida dan kawan-kawan, 2024)
2. *Kompresi lossy* terjadi ketika kata-kata informasi atau bit dalam data berubah atau hilang dan hasil kompresi tidak dapat dikembalikan ke bentuk aslinya (tidak dapat diubah). Akan tetapi, informasi utama data tetap dipertahankan melalui hasil kompresi. Kompresi ini cocok untuk berkas audio, gambar, atau video. Secara umum, bahasa ini menghasilkan hasil kompresi berkualitas rendah, tetapi rasio kompresi biasanya tinggi. *Contoh algoritma kompresi lossy meliputi kompresi fraktal, kompresi wavelet, dan pengkodean Wyner-Ziv (WZC).*

2.2 Python

Python adalah bahasa pemrograman bahasa yang menawarkan berbagai perpustakaan untuk kompresi data dan pengeditan file multimedia. (Raharjo et al., 2019) Python memiliki perpustakaan seperti *PIL (Python Imaging Library)* yang

mendukung pemrosesan gambar, serta modul seperti *zlib* yang digunakan untuk kompresi data umum. Dalam konteks kompresi RLE, Python memberikan fleksibilitas untuk mengembangkan algoritma dan mengubah proses kompresi sesuai kebutuhan, yang penting untuk penelitian ini (Hasan et al., 2024).

Python memudahkan penanganan data multimedia melalui berbagai perpustakaan yang memungkinkan analisis data, kompresi, dan visualisasi hasilnya. Menerapkan algoritma RLE dengan Python membantu mengoptimalkan data multimedia agar dapat disimpan lebih efisien. Menurut Arifin dan Susanto (2023), Python menawarkan pengembangan kode yang mudah dan kinerja yang baik, menjadikannya pilihan ideal untuk pemrosesan data berbasis kompresi.

2.3 Aplikasi berbasis Web

Aplikasi berbasis web merupakan aplikasi yang dijalankan dengan menggunakan teknologi dari internet sehingga dapat dijalankan secara portable. Aplikasi-aplikasi ini umumnya tidak memerlukan banyak program tambahan untuk diinstal di sisi klien agar aplikasi dapat berjalan. (Rosandy et al., 2022) Aplikasi web juga merupakan perangkat lunak yang diberi kode dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti Python.(Prayoga & Suryaningrum, 2020).

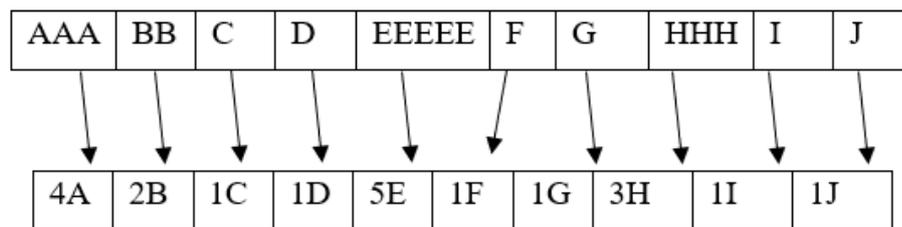
2.4 Algoritma Run Length Encoding (RLE)

Algoritma encoding run-length (RLE) menggunakan karakter berulang secara berurutan pada data dengan mengkodekannya dengan string yang terdiri dari jumlah pengulangan karakter yang terjadi, diikuti oleh karakter berulang. Oleh karena itu, banyaknya karakter berulang dalam data sangat penting untuk keberhasilan kompresi RLE.

Tabel 2.1. Format Kode

M	N	s
---	---	---

Deskripsi gambar di atas adalah: m adalah tanda byte, n adalah jumlah string yang diulang, dan s adalah karakter yang berulang. Misalnya, jika string: “AAAABBCDEEEEEFGHHHIJ” dikompresi menggunakan algoritma RLE, hasilnya akan menjadi “^4A2^B^1C^1D^5E^1F^1G^3H^1I^1J”. Karena bahasa kode yang dihasilkan RLE minimal 3 byte untuk setiap string karakter, maka jumlah pengulangan karakter harus lebih dari 3 (tiga) kali agar pengkodean dapat dilakukan. Hal penting yang perlu diingat tentang tanda m adalah sebaiknya dipilih sebagai karakter yang jarang digunakan dalam data (seperti #, ^, | atau ~). Algoritma decoding di RLE cukup sederhana untuk mengembalikan karakter original berdasarkan jumlah kompresi yang digunakan untuk setiap karakter. Decoding dilakukan dengan langkah-langkah berikut:



Gambar 2.1. Proses decoding

Misalnya, hasil kompresi adalah “^4A2^B^1C^1D^5E^1F^1G^3H^1I^1J”, dan hasil dekompresi adalah “AAAABBCDEEEEEFGHHHIJ”. Algoritma RLE optimal biasanya digunakan untuk file yang cenderung mengandung karakter homogen. Oleh karena itu, jika algoritma digunakan secara umum, pengelompokan atau transformasi karakter atau simbol serupa diperlukan. RLE memiliki beberapa keuntungan, seperti:

1. Sederhana dan mudah diimplementasikan: Algoritma ini cukup mudah untuk dipahami dan diimplementasikan.
2. Efisien untuk Data Berulang: RLE sangat efektif untuk data yang berisi urutan berulang yang panjang seperti: Gambar dengan area warna dominan serupa atau teks dengan kata-kata yang berulang.

Kompresi lossless: Data yang dikompresi dengan RLE dapat dikembalikan ke bentuk aslinya tanpa kehilangan informasi. RLE juga memiliki kekurangan, yaitu:

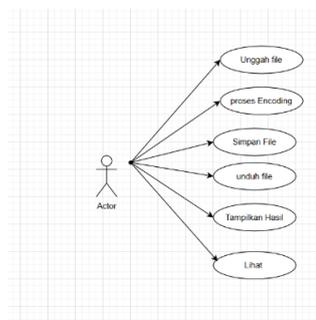
1. Tidak efektif untuk data acak: Jika data tidak memiliki banyak pola berulang, ukuran file yang dihasilkan bahkan mungkin lebih besar dari file aslinya.
2. Kurang efektif untuk data yang kompleks: Untuk file yang lebih kompleks, seperti: Misalnya file teks yang mengandung banyak karakter, RLE tidak memberikan hasil kompresi yang signifikan.

2.5 UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah metodologi pemodelan visual yang dikembangkan khusus untuk desain sistem berbasis objek. UML juga diakui sebagai bahasa standar untuk desain perangkat lunak. Menurut Sonata (2021), UML memiliki beberapa tujuan dan fitur penting, antara lain:

1. Memberikan representasi visual kepada pengembang terlepas dari latar belakang bahasa pemrograman mereka.
2. Mengintegrasikan informasi penting saat membuat model.
3. Menampilkan pemodelan visual ekspresif dalam pengembangan sistem.
4. Membantu pemodelan sistem berorientasi objek.
5. Berfungsi sebagai cetak biru yang menjelaskan desain kode proyek.

Lima karakteristik di atas menunjukkan bahwa UML memainkan peran penting dalam proses desain proyek. Berikut adalah beberapa contoh diagram UML yang umum digunakan dalam pengembangan proyek:

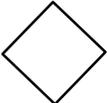
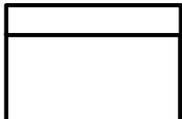


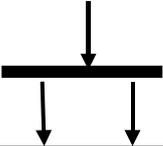
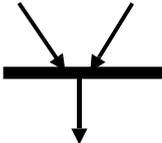
Gambar 2.2. Diagram UML

2.5.1 Diagram Activity

Salah satu jenis diagram UML yang digunakan untuk memodelkan alur kerja atau aktivitas dalam proses bisnis adalah diagram aktivitas. Diagram ini menunjukkan perilaku dinamis sistem dan menunjukkan bagaimana objek berinteraksi dalam berbagai aktivitas. Diagram aktivitas juga memiliki ikon unik, seperti yang ditunjukkan pada Tabel 2. 2 di bawah ini.

Tabel 2.2. Symbol Diagram Activity

Symbol	Definition
	"Start" adalah status awal sistem, dan simbol ini digunakan dalam diagram untuk memulai aktivitas proses.
	Serangkaian tindakan atau langkah-langkah dalam suatu proses disebut sebagai aktivitas. Setiap aktivitas diberi nama dan ditampilkan dalam bentuk persegi panjang dengan sudut melengkung. Aktivitas awal dimulai dan aktivitas akhir diakhiri.
	Percabang, yang sering disebut sebagai keputusan, biasanya digunakan ketika terdapat lebih dari satu pilihan aktivitas. Ini menggambarkan situasi atau opsi yang harus diambil dalam suatu proses.
	Penggabungan: Simbol ini umumnya digunakan untuk mengintegrasikan dua atau lebih aktivitas menjadi satu kesatuan
	Ketika sebuah diagram aktivitas selesai, simbol Status Akhir digunakan untuk menunjukkan bahwa proses tersebut telah berakhir.
	<i>Swimlane</i> memisahkan perusahaan yang saling berhubungan satu sama lain, dengan masing-masing memiliki tugas untuk melakukan apa yang mereka lakukan.

	Aliran aktivitas dipisahkan dengan fork menjadi beberapa jalur yang berjalan secara paralel.
	Sementara itu, join berfungsi untuk menggabungkan beberapa jalur paralel menjadi satu kesatuan.

2.6 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang relevan dengan topik kompresi data menggunakan algoritma RLE.

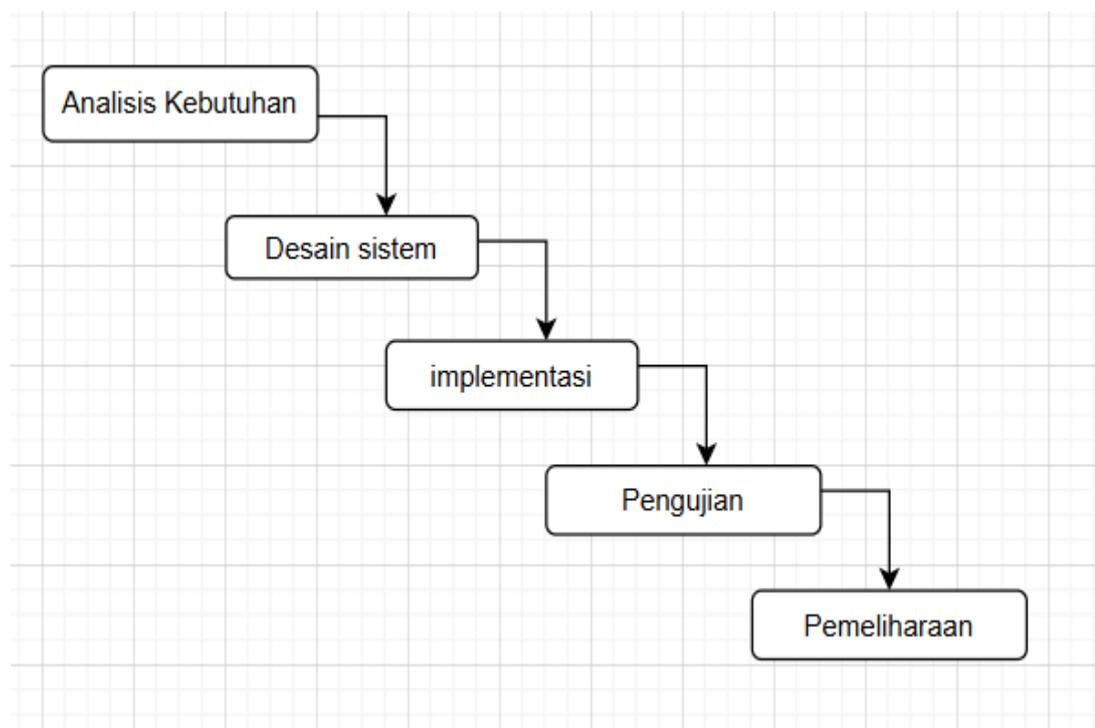
Tabel 2.3. Penelitian Terdahulu

Nama Penelitian	Judul Penelitian	Tahun	Permasalahan	Metode	RLE	Hasil
Darno Willfrid Midukta Simamora, Garuda Ginting, Yasir Hasan	Penerapan Algoritma Run Length Encoding untuk Kompresi File MP3	2020	Apakah RLE bisa digunakan untuk mengompresi file MP3?	Digunakan untuk kompresi audio, tetapi kurang efisien	Digunakan untuk kompresi audio, tetapi kurang efisien	RLE kurang efektif pada MP3 karena kurangnya pola berulang dalam data audio
Sylvia Rahma, Aditya Prapanca	Analisis Kompresi dan Dekompresi Data Teks dan Audio dengan Algoritma Run Length	2021	Seberapa efektif RLE dalam kompresi data teks dan audio	Analisis perbandingan rasio kompresi berbagai format file	Digunakan untuk teks dan audio, tetapi kurang optimal untuk audio.	RLE kurang efektif pada MP3 karena kurangnya pola berulang dalam data audio
Ulfah Lu'luilmaknun, Nurul Hidayah Salsabila	Penggunaan Metode Run Length Encoding untuk Kompresi Data	2022	Bagaimana RLE digunakan dalam kompresi berbagai jenis data?	Studi literatur dan implementasi sederhana.	Digunakan untuk mengompresi data dengan pola berulang	Cocok untuk data dengan pola berulang, tetapi tidak optimal untuk data acak algoritma ini bisa kurang efektif
Nur Adillah Siregar, Yusuf Ramadhan Nasution, Armansyah	Perbandingan Algoritma Run Length Encoding (RLE) dan Algoritma	2023	Metode mana yang lebih baik untuk mengompresi file video?	Perbandingan dua algoritma pada file video	Digunakan dan dibandingkan dengan VLBE	VLBE lebih unggul dalam rasio kompresi, tetapi RLE lebih

Nama Penelitian	Judul Penelitian	Tahun	Permasalahan	Metode	RLE	Hasil
	Variable Length Binary Encoding (VLBE) dalam Mengkompresi File Video Untuk Menghemat Penyimpanan					sederhana dan cepat
Hairil Kurniadi	Implementasi Algoritma Run Length Encoding (RLE) pada Kompresi Data Teks menggunakan Python	2024	Bagaimana mengoptimalkan kompresi data teks menggunakan Python?	Eksperimen dengan dataset teks	Digunakan untuk mengompresi teks berbasis pola berulang	Mampu mengurangi ukuran data teks, sehingga menghemat ruang penyimpanan

2.6.1 Metode waterfall

Dalam penelitian ini digunakan metode perencanaan aplikasi berdasarkan model air terjun. Model air terjun merupakan model klasik yang mendekati pengembangan perangkat lunak secara sistematis dan berurutan, dimana prosesnya bertahap dan harus menunggu hingga fase sebelumnya selesai sebelum memulai fase berikutnya (Tujni dan Hutrianto, 2020).



Gambar 2.3. Diagram Alur Waterfall

2.6.2 Analisis kebutuhan

Analisis kebutuhan sistem ini mencakup persyaratan fungsional yaitu sistem dapat melakukan kompresi dan dekompresi file teks dan gambar menggunakan algoritma RLE, menampilkan ukuran file sebelum dan sesudah kompresi, dan mencatat waktu pemrosesan. Persyaratan non-fungsional mencakup kecepatan dan efisiensi pemrosesan, antarmuka yang ramah pengguna, dan dukungan untuk berbagai jenis file (Khairi & Jambek, 2021).

2.6.3 Desain Sistem

Perancangan sistem berbasis web ini terdiri dari arsitektur client-server. Pengguna mengunggah file di sisi klien dan sistem server memproses file tersebut menggunakan algoritma RLE. Antarmuka pengguna mencakup formulir untuk pengunggahan file dan keluaran hasil kompresi, yang menampilkan informasi tentang ukuran file dan waktu pemrosesan (Mufid et al., 2022).

2.6.4 Implementasi

Implementasinya menggunakan Python untuk algoritma RLE, Flask atau Django sebagai framework web, dan MySQL untuk menyimpan data terkait proses. Langkah implementasinya meliputi pengembangan fitur kompresi dan dekompresi, (Fitria & Apri Triansyah, 2021) integrasi UI dengan backend, dan pengujian untuk memastikan fitur berjalan dengan baik. (Utari, 2020).

2.6.5 Pengujian Black Box

Pengujian black box adalah metode pengujian perangkat lunak yang fokus pada evaluasi fungsionalitas tanpa mempertimbangkan struktur internal atau kode program. Dalam pengujian ini (Mutiara & Murod, 2022). penguji akan memberikan input dan memeriksa output untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang diinginkan (Marselly & Simarmata, 2024).

2.6.6 Pemeliharaan

Pemeliharaan meliputi tindakan perbaikan untuk memperbaiki kesalahan, tindakan adaptif untuk mengadaptasi teknologi, tindakan preventif untuk mencegah masalah, dan tindakan prospektif untuk menambah fitur baru berdasarkan kebutuhan pengguna” (YANWASTIKA, 2022).