

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

3.1.1 Analisis Kebutuhan

Kebutuhan Penelitian yang digunakan dalam Penelitian ini terdiri dari perangkat keras komputer (Hardware) dan perangkat lunak (Software), yaitu:

1. Perangkat Keras (Hardware) sistem

Perangkat keras komputer yang digunakan untuk membuat aplikasi web antara lain sebagai berikut:

- a. Processor: Intel Core i3
- b. RAM: Minimal 4 GB
- c. Penyimpanan: 20GB
- d. Sistem operasi: Windows 10
- e. Monitor: 1366x768 piksel
- f. Keyboard dan Mouse

2. Perangkat Lunak (Software) sistem

Selain Perangkat Keras, untuk membuat sistem dibutuhkan spesifikasi perangkat lunak yang digunakan sebagai pendukung sistem. Adapun spesifikasi perangkat lunak yang digunakan antara lain:

- a. Bahasa Pemrograman Python
- b. Visual Studio code
- c. Notepad
- d. Browser: Microsoft Edge

3.1.1.1 Kebutuhan Fungsional

1. Sistem dapat membaca file teks dengan format .txt.
2. Sistem mampu melakukan kompresi data menggunakan algoritma RLE.
3. Sistem mampu mendeteksi efisiensi kompresi dan memutuskan untuk menyimpan data asli jika hasil kompresi tidak efisien.

4. Sistem dapat menyimpan hasil kompresi dalam file dengan ekstensi rle.
5. Sistem dapat melakukan dekompresi file rle untuk mengembalikan data ke bentuk aslinya.
6. Sistem dapat menyimpan hasil dekompresi kembali ke file teks dengan format .txt.

3.1.1.2 Kebutuhan Non-Fungsional

1. Sistem harus mampu memproses file teks dengan ukuran hingga 1 MB dalam waktu kurang dari 3 detik.
2. Sistem dirancang menggunakan bahasa Python dan kompatibel dengan berbagai sistem operasi.
3. Antarmuka sistem berbasis teks (command-line) yang sederhana dan mudah digunakan.

3.2 Keterbatasan Sistem

1. Algoritma RLE hanya optimal untuk data dengan banyak beracak karakter.
2. Sistem hanya dapat memproses file teks (.txt) dan file hasil kompresi dengan rar dan txt.

3.3 Desain Sistem Konsep Algoritma Run Length Encoding (RLE)

Run Length Encoding (RLE) adalah algoritma kompresi lossless yang bekerja dengan cara menggantikan urutan karakter berulang dalam data menjadi pasangan karakter dan jumlah beracak.

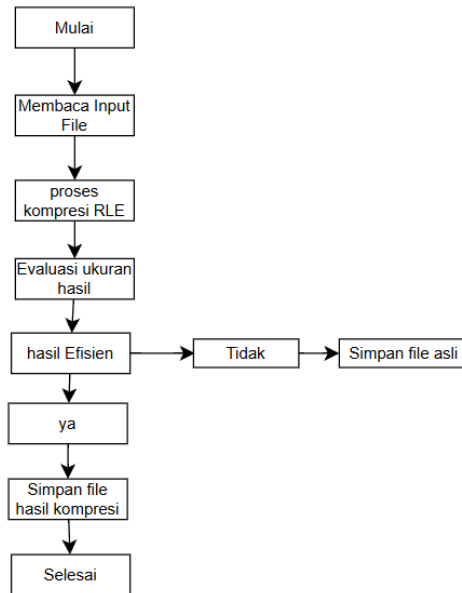
Contoh:

Input: AAABBBCCCC

Output: A3B3C4

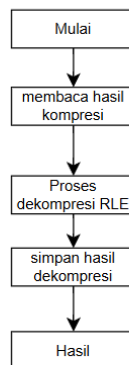
3.3.1 Diagram Alur Sistem

a. Diagram Proses Kompresi



Gambar 3.1. Diagram Proses kompresi

b. Diagram Proses Dekompresi



Gambar 3.2. Diagram Proses Dekompresi

3.4 Komponen Sistem

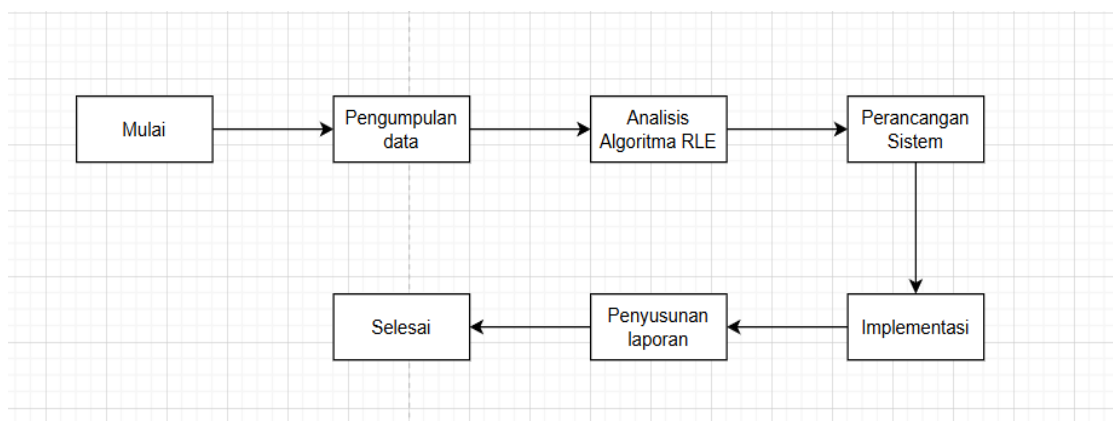
1. Input: File teks yang akan dikompresi (.txt) atau file terkompresi (.rle).
2. Proses Kompresi: Implementasi algoritma RLE untuk mengompresi data.
3. Evaluasi Efisiensi: Mengevaluasi apakah ukuran file hasil kompresi lebih kecil dibandingkan file asli.
4. Proses Dekompresi: Mengembalikan file terkompresi ke bentuk aslinya.
5. Output: File hasil kompresi (.rle) atau hasil dekomposisi (.txt).

3.5 Alur Penelitian

Alur penelitian dirancang melalui Langkah-langkah berikut:

1. Mengumpulkan data file teks dengan pola acak.
2. Menganalisis algoritma RLE untuk memahami cara kerja kompresi data.
3. Mendesain sistem berbasis Python yang mampu melakukan kompresi dan dekomposisi.
4. Melakukan pengujian sistem terhadap berbagai skenario dan ukuran file.
5. Menyusun laporan akhir hasil penelitian.

Diagram alur penelitian:



Gambar 3.3. Alur Penelitian

3.6 Sumber Data

Penelitian ini menggunakan data yang berasal dari sumber berikut:

3.6.1 Dataset File Teks Publik

Sumber data dari internet Test Data Hub yang memiliki jenis data txt di https://testdatahub.com/generate_files

3.7 Perancangan Sistem

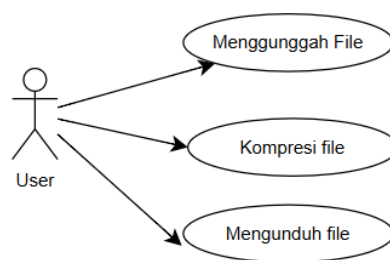
3.7.1 Arsitektur Sistem

Arsitektur sistem dirancang dengan tiga lapisan utama:

1. Input Layer: Menerima file teks dari pengguna.
2. Processing Layer: Melakukan kompresi atau dekompresi menggunakan algoritma RLE.
3. Output Layer: Menyediakan hasil kompresi untuk diunduh atau ditampilkan.

3.8 Diagram UML

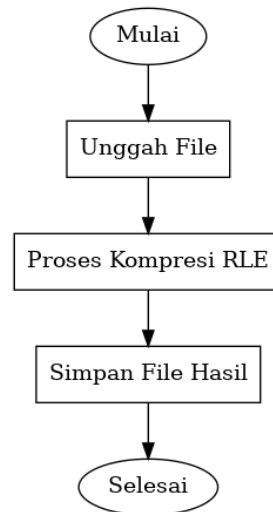
Menggambarkan interaksi antara pengguna dan sistem untuk fitur unggah file, proses kompresi, dan unduh hasil Pada gambar 3.4



Gambar 3.4. Diagram UML

3.8.1 Diagram Activity

Menjelaskan alur aktivitas sistem, mulai dari unggah file hingga proses kompresi dan unduh hasil. Pada gambar 3.5

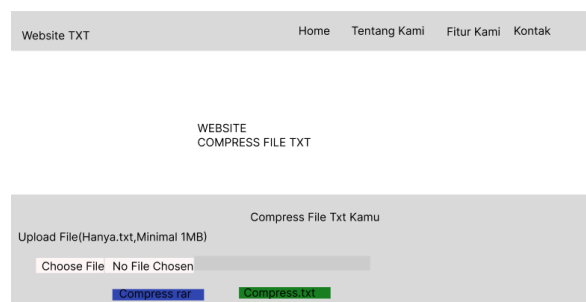


Gambar 3.5. Diagram Activity

3.9 Desain Antarmuka Pengguna (UI)

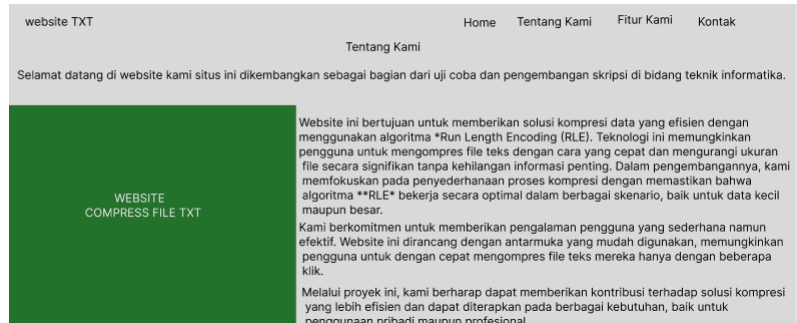
Antarmuka pengguna dirancang menggunakan Figma, dengan fitur:

1. Halaman Home



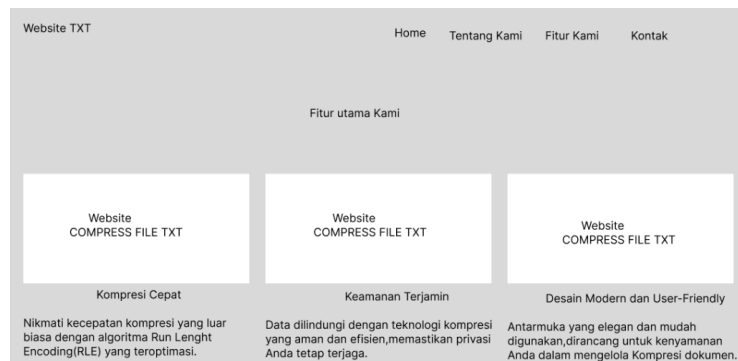
Gambar 3.6. Tampilan Halaman Home

2. Halaman untuk menampilkan Tentang Kami



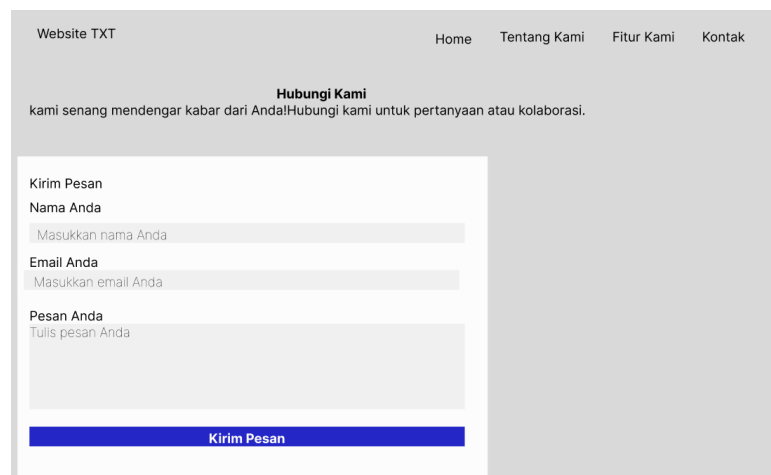
Gambar 3.7. Tampilan Halaman Tentang Kami

3. Halaman untuk menampilkan Fitur kami



Gambar 3.8. Tampilan Halaman Fitur Kami

4. Halaman Untuk menampilkan Kontak



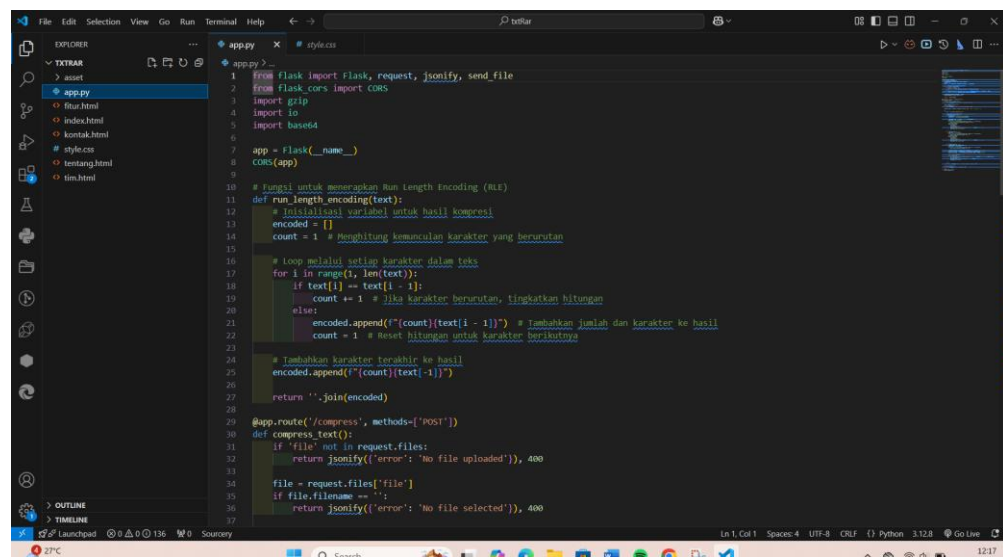
Gambar 3.9. Tampilan halaman Kontak

3.9.1 Implementasi Sistem

Pada pembahasan hasil pembuatan aplikasi web ini dijelaskan dalam bentuk tampilan program yang telah dijalankan (*running*). Adapun Implementasi dilakukan dengan:

1. Bahasa dan Alat Pengembangan

- **Bahasa Pemrograman: Python**



Gambar 3.10. Bahasa pemrograman python

- **Framework: Flask**

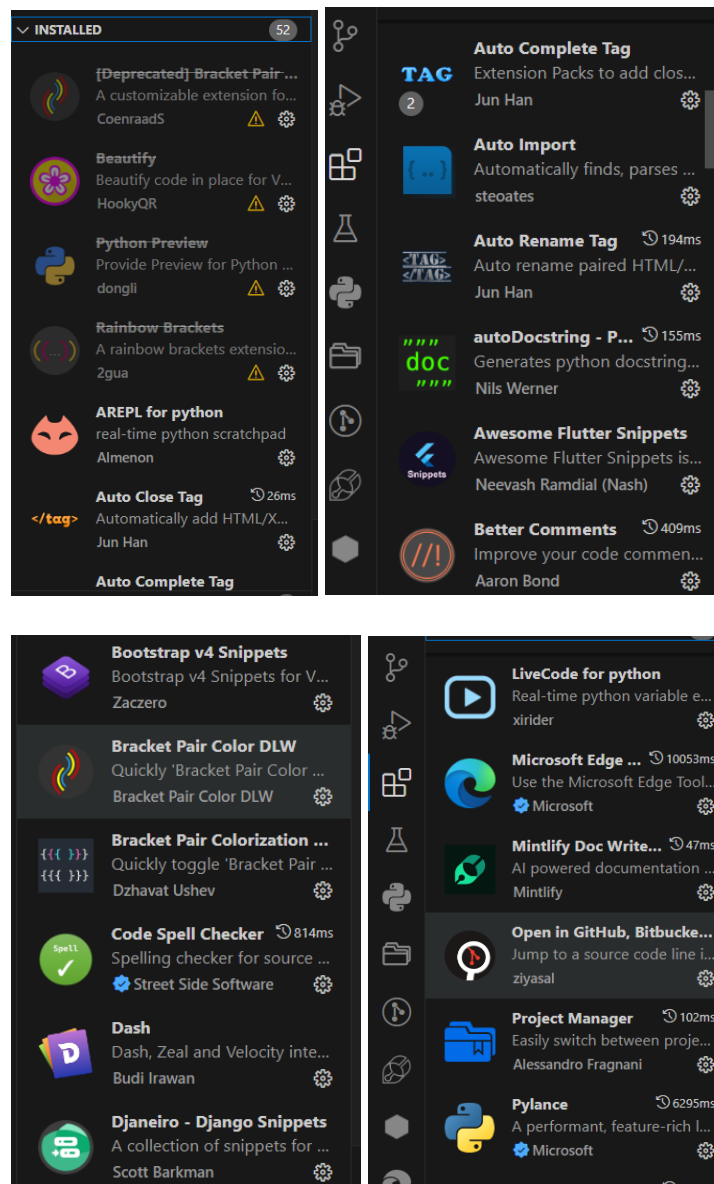
```

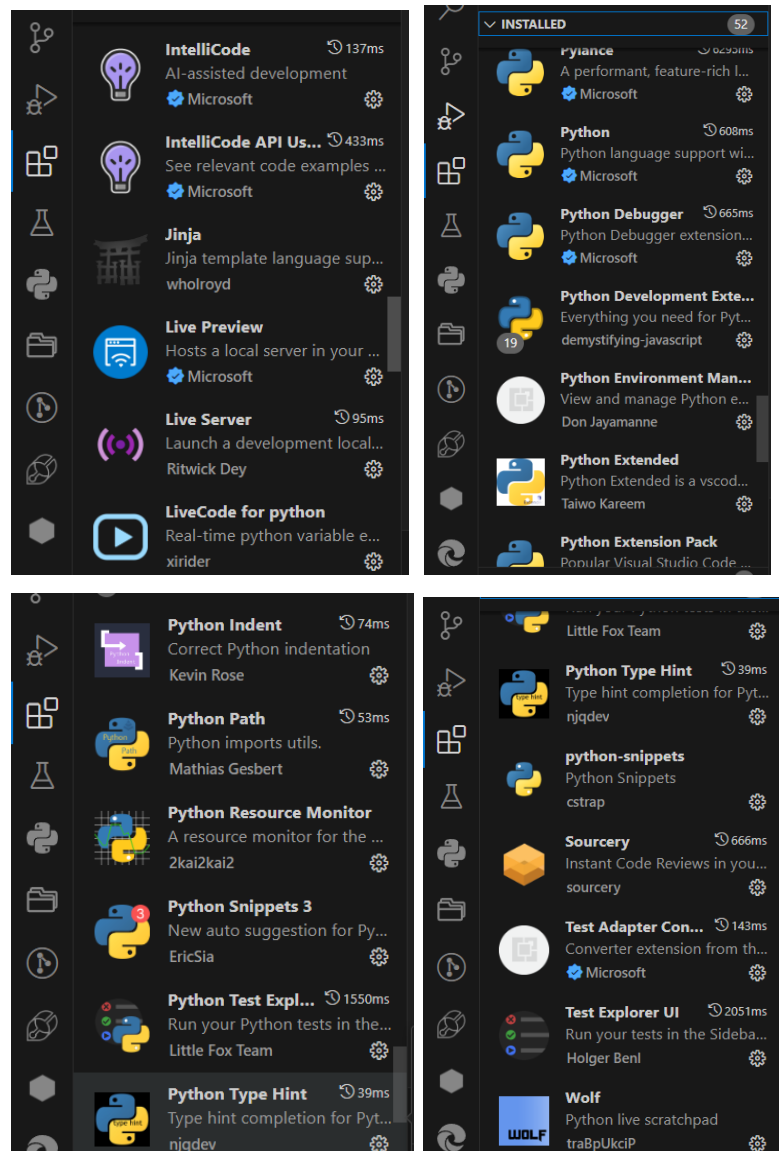
from flask import Flask, request, jsonify, send_file
from flask_cors import CORS

```

Gambar 3.11. framework flask

- Library Pendukung





Gambar 3.12. Library Pendukung

2. Proses Kompresi

Kompresi: Menggunakan algoritma RLE untuk menggantikan pola berulang dalam file teks dengan format terkompresi.

3.9.2 Pengujian Sistem

Pengujian sistem dilakukan dengan metode pengujian kotak hitam (black box test), dan sistemnya adalah sebagai berikut: Validasi input file.

1. Proses kompresi dan dekompresi.

2. Kecepatan waktu eksekusi untuk file dengan ukuran berbeda.

3.9.3 Pengujian Perangkat Lunak

Hasilnya menunjukkan bahwa metodologi pengujian kotak hitam digunakan untuk menguji kinerja aplikasi yang dibuat. Metode ini digunakan saat aplikasi dianggap lengkap. Sebelum aplikasi dirilis dan digunakan secara luas, ini adalah langkah akhir.

3.9.4 Pembahasan Hasil pengujian Software

Ketika aplikasi dianggap selesai, pengujian black box dilakukan untuk mengevaluasi kinerjanya. Sebelum aplikasi dirilis dan digunakan secara luas, ini adalah langkah akhir.

3.9.5 Pengujian Halaman Utama

Penulis halaman ini berharap semua menu dan gambar di halaman utama berfungsi dengan baik dan mudah digunakan. Tabel 3.1 menunjukkan hasil tes ini.

Tabel 3.1. Pengujian Halaman Utama

No	Perangkat	Hasil Pengujian
1	Komputer Acer	Halaman utama memiliki semua menu dan tampilan yang jelas dan berfungsi dengan baik.
2	komputer Asus	Menu berfungsi dengan lancar dan ditampilkan dengan baik.
3	Hp Android vivo y20s	Dengan tampilan halaman utama yang mudah dipahami, menu dapat diakses.

3.9.6 Pengujian sistem file

Tabel 3.2. Pengujian Sistem File

No	user	Tindakan	Hasil Pengujian
1	sistem	Pilihan file yang akan di kompres	Data tersimpan otomatis jenis pilihannya rar dan txt