

Implementasi Algoritma *Run Length Encoding (RLE)* Pada Kompres Data Teks menggunakan *Python*

Implementation of Run Length Encoding (RLE) Algorithm on Text Data Compress using Python

¹Hairil Kurniadi Siradjuddin*, ²Amal Khairan, ³Muhammad Ridha Albaar, ⁴Saiful Do Abdullah
^{1,2,3,4}Teknik Informatika, Universitas Khairun, Ternate, Indonesia
*e-mail: hairil.kurniadi@unkhair.ac.id

(received: 22 May 2024, revised: 10 June 2024, accepted: 20 July 2024)

Abstrak

Kompresi data teks dilakukan guna membuat ukuran file tersebut lebih kecil. Algoritma merupakan urutan langkah-langkah yang bertujuan untuk menyelesaikan suatu persoalan. Dalam penelitian Kompres data teks ini menggunakan Metode kompres data *Run Length Encoding (RLE)*, dapat dimanfaatkan untuk memangkas data agar meminimalkan penggunaan ruang penyimpanan agar bisa dimanfaatkan lebih baik lagi. Sebagaimana fungsi dari kompres data sendiri untuk memangkas file maka penggunaannya sangat bermanfaat bagi teknologi masa depan. Pemrograman yang diterapkan yaitu aplikasi *python*, menerapkan konsep pemrograman terstruktur. Pemrograman terstruktur merupakan konsep atau paradigma pemrograman yang menyelesaikan permasalahan secara struktural, dengan tanpa melihat objek atau pembagian tetapi harus terstruktur, Hasil dari penelitian ini adalah dengan aplikasi *python* mampu memangkas data teks sehingga meminimalkan penggunaan ruang penyimpanan agar bisa dimanfaatkan lebih baik lagi.

Kata kunci: data kompres teks; RLE, aplikasi *python*

Abstract

Text data compression is done to make the file size smaller. Algorithm is a sequence of steps that aims to solve a problem. In this text data compress research using the Run Length Encoding (RLE) data compress method, it can be used to trim data to minimize the use of storage space so that it can be utilized better. As the function of the data compress itself to trim the file, its use is very beneficial for future technology. The programming applied is a python application, applying the concept of structured programming. Structured programming is a programming concept or paradigm that solves problems structurally, without looking at objects or divisions but must be structured, The result of this research is that the python application is able to trim text data so as to minimize the use of storage space so that it can be utilized better.

Keywords: text compress data, RLE, python application

1 Pendahuluan

Dalam era di mana pertukaran dan penyimpanan data menjadi semakin penting, efisiensi dalam penggunaan ruang penyimpanan dan kecepatan pengiriman data menjadi fokus utama. Salah satu cara untuk mencapai efisiensi tersebut adalah melalui teknik kompresi data. Kompresi data adalah proses pengurangan ukuran file atau data dengan tujuan menghemat ruang penyimpanan dan mempercepat proses pengiriman. Masalah baru muncul ketika berbicara terkait data. Data yang akan diolah diterima atau dikirim dalam berbagai macam bentuk, termasuk data yang kecil dan cepat diproses maupun data yang berkapasitas besar dan sulit diolah. Solusi yang paling mendekati permasalahan ini tidak lain adalah kompresi data, atau memangkas ukuran data, sehingga bisa diterima dan diolah dengan efektif.

Dengan memahami dasar-dasar kompresi data teks, kita dapat mengimplementasikan solusi yang dapat meningkatkan efisiensi penggunaan data dalam berbagai konteks aplikasi. Algoritma Run Length Encoding merupakan salah satu bentuk skema metode terbaik kompresi data lossless yang paling

<http://sistemasi.ftik.unisi.ac.id>

sederhana dan mudah dipahami. Metode ini digunakan untuk kompresi data yang berkarakter berulang dan bernilai sama lalu disimpan sebagai perhitungan data tunggal tetapi bukan sebagai proses asli. Metode ini sering digunakan untuk kompresi data teks dan audio [1]. *Run Length Encoding* (RLE) algoritma kompres data yang mungkin bisa dijadikan alternatif pilihan. Seperti yang diketahui, data dapat diterima dalam bentuk apa saja, melingkupi satu kesatuan sistem multimedia, Multimedia sendiri disajikan dalam bentuk berbagai media misal teks, suara, gambar dan video. Bukti bahwa pengaruh multimedia sudah sampai pada tahap penting. sekarang banyak kita mengetahui berbagai macam informasi dalam media sosial bahkan mencakup berbagai unsur media informasi tadi, belum lagi melalui teks pada koran dan majalah, juga gambar dan suara pada iklan dan lain-lain. Kompresi memiliki arti memperkecil ukuran atau memampatkan. Kompresi merupakan suatu proses pengkodean informasi menggunakan bit atau unit pembawa informasi yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem *encoding* tertentu, sehingga data yang dikompresi lebih kecil sehingga dapat mempersingkat waktu dalam transmisi data tersebut [2]. *Python* adalah bahasa pemrograman yang diinterpretasikan tingkat tinggi, yang dikenal luas karena mudah dipelajari, namun tetap mampu memanfaatkan kekuatan tingkat sistem bahasa pemrograman tingkat sistem bila diperlukan. Selain manfaat dari bahasa itu sendiri, komunitas untuk tools dan library yang tersedia membuat *Python* sangat menarik [3].

Bentuk penelitian dengan menggunakan algoritma *Run Length Encoding* (RLE) ini sebagai metode atau algoritma yang penulis ambil untuk menyelesaikan fokus masalah kompresi data teks. Perlu diperhatikan fokus masalah masih pada link memangkas teks saja. Namun, perkembangan kompres data sudah pada tahap gambar dan video dengan masing-masing algoritmanya yang unik. Terdapat beberapa penelitian terdahulu yang berkaitan dengan kompresi data khususnya memanfaatkan algoritma *run length encoding*, diantaranya adalah *ECG Data Compression Using Modified Run Length Encoding of Wavelet Coefficients for Holter Monitoring*, yang memanfaatkan algoritma *Run Length Encoding* untuk kompresi data EKG sehingga meminimalkan kebutuhan penyimpanan data dan biaya transmisi [4]. *Run Length Encoding* (RLE) dipilih karena kesederhanaan algoritma yang diterapkan. Penulis berusaha mengembangkan kompleksitas metode ini untuk menyelesaikan masalah mulai dari teks yang di implementasi dalam program *python* akan dijadikan khas dari penelitian ini agar bisa diteliti dan bermanfaat bagi ilmu kompres data kedepan.

2 Tinjauan Literatur

Menurut penelitian sebelumnya Ulfa Lu'luilmaknun & Nilza Humaira Salsabila [5] yang berjudul "Penggunaan Metode *Run Length Encoding* Untuk Kompresi Data)" Metode RLE merupakan salah satu jenis *lossless* yang paling sederhana dari skema kompresi data dan didasarkan pada prinsip sederhana data *encoding*. Metode RLE sangat cocok untuk mengompresi data yang berisi karakter-karakter berulang misalnya citra grafis sederhana. Data yang dikompresi adalah 28 citra RGB (*Red, Green, Blue*) dan 28 citra *grayscale* masing-masing dalam format jpg, png, bmp, dan tiff. Data citra dikompresi dengan program *encoder* dan *decoder* menggunakan algoritma RLE pada aplikasi matlab. Metode RLE dikatakan efektif mengompresi data citra jika rasio kompresi kurang dari 100% dikarenakan memiliki banyak perulangan warna pada piksel-pikselya. Penelitian yang dilakukan oleh Lucky Adi Guna [6] yang berjudul "Implementasi Prosedur Dan Fungsi Dalam Bahasa Pemrograman *Python*", implementasi prosedur dan fungsi dalam bahasa pemrograman *Python*, penjelasan tentang konsep dasar, sintaksis, dan manfaat penggunaan prosedur dan fungsi dalam pengembangan perangkat lunak. contoh-contoh implementasi yang relevan, yang memberikan pemahaman yang lebih baik tentang penggunaan prosedur dan fungsi dalam konteks nyata. Penelitian berikutnya dilakukan oleh Darno Willfrid Midukta Simamora, Garuda Ginting, dan Yasir Hasan [7] yang berjudul "*Implementasi Algoritma Run Length Encoding* Pada Kompresi File Mp3" Pengiriman file MP3 seringkali terhambat karena besarnya ukuran file yang akan dikirim. Salah satu cara untuk mengatasi hal tersebut adalah dengan memampatkan (kompresi) file tersebut sebelum dikirim. Penulis menggunakan algoritma *Run Length Encoding* yang bersifat *lossless* dalam pemampatan file MP3. Masukan dalam sistem ini adalah file audio MP3. Pada sistem ini terdapat tahap kompresi dan dekompresi. Tahap kompresi bertujuan untuk memampatkan ukuran file MP3, sedangkan tahap dekompresi bertujuan untuk mengembalikan ukuran file MP3 ke ukuran semula. Algoritma yang dapat digunakan untuk proses kompresi salah satunya adalah *Run Length Encoding* (RLE). RLE menjadi salah satu metode memampatkan data, dengan menggunakan perhitungan matematika terhadap sebuah bilangan, yang dibuat berdasarkan total *hexa* yang terdapat pada file MP3

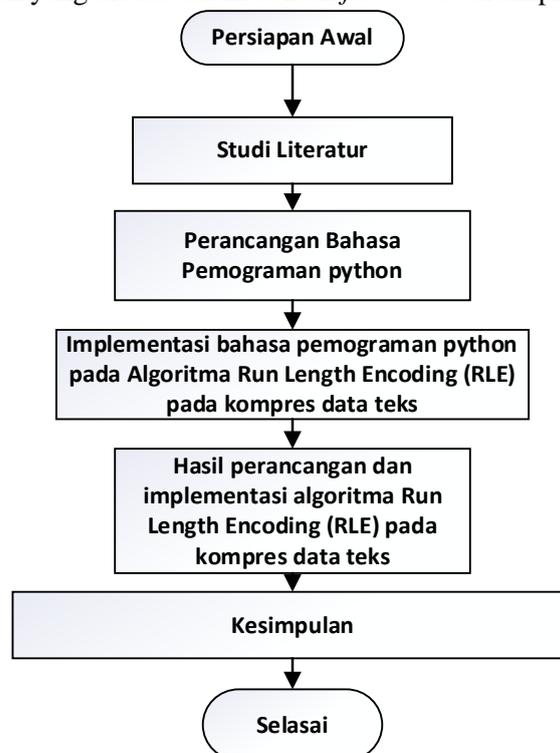
<http://sistemasi.ftik.unisi.ac.id>

tersebut. Selanjutnya penelitian dilakukan oleh Nur Adillah Siregar, Yusuf Ramadhan Nasution dan Armansyah [8] yang berjudul “Perbandingan Algoritma *Run Length Encoding* (RLE) dan Algoritma *Variable Length Binary Encoding* (VLBE) dalam Mengkompresi File Video Untuk Menghemat Penyimpanan” Pada penelitian ini menggunakan algoritma *Run Length Encoding* (RLE) dan algoritma *Variable Length Binary Encoding* (VLBE). Format file video yang dalam penelitian ini adalah .avi dan .mp4 dan menggunakan Microsoft Visual Studio dengan bahasa pemrograman C# berbasis desktop. Hasil dari penelitian ini adalah file yang video yang telah dikompresi akan memperlihatkan performanya dan parameter perbandingan berdasarkan *Ratio of Compression* (RC), *Compression Ratio* (CR), dan *Redudancy* (RD) diantara algoritma *Run Length Encoding* (RLE) dan algoritma *Variable Length Binary Encoding* (VLBE). Penelitian yang dilakukan oleh Sylvia Rahma dan Aditya Prapanca [9] yang berjudul Analisis “Kompresi dan Dekompresi Data Teks dan Audio”, Algoritma *Run Length Encoding* (RLE) merupakan kompresi data yang paling sederhana dan paling signifikan untuk digunakan apa saja, algoritma ini bekerja pada pengulangan karakter dalam string yang dimasukkan. Tujuan dari analisis penelitian ini yakni untuk mengukur kinerja rasio dan kecepatan kompresi dekompresi data pada format .docx, .txt, .pdf .mp3, dan .wav. Hasil pengujian menunjukkan bahwa setiap ukuran data asli yang telah melalui proses kompresi menghasilkan angka yang berbeda-beda tergantung pada isi string karakter di dalamnya, semakin banyak string karakter berulang maka menghasilkan ukuran data yang lebih kecil. Begitu juga sebaliknya dengan proses dekompresi, dapat dikatakan berhasil jika ukuran file dan isi file dapat kembali seperti semula dan durasi yang dibutuhkan untuk melakukan kompresi dekompresi pun juga tergantung pada besar ukuran data asli.

Berdasarkan beberapa penelitian yang telah membahas mengenai metode *Run Length Encoding* (RLE) dan aplikasi *python* maka disusunlah penelitian terkait penggunaan kompres data teks, berbeda dengan penelitian sebelumnya, pada penelitian ini di membuat perancangan pemograman data teks menggunakan aplikasi *python* pada kompres teks dengan menggunakan konsep algoritma *Run Length Encoding* (RLE).

3. Metode Penelitian

Skema langkah penelitian adalah awal dari tahap perencanaan sistem dan aplikasi yang akan dibuat. Skema perancangan ini dibuat berdasarkan alur langkah pengerjaan penelitian. Berikut ini alur langkah pengerjaan penelitian yang dibuat dalam bentuk *flow chart* ditampilkan dalam Gambar 1.



Gambar 1. *Flow chart* penelitian

Flowchart penelitian yang terlihat pada gambar 1 menjelaskan *flow chart* dijelaskan secara detailnya adalah sebagai berikut:[10]

1. Studi Literatur: pada tahap ini merupakan kegiatan untuk mencari dan mempelajari literatur-literatur yang mendukung penelitian. Literatur yang dipelajari berkaitan dengan Perancangan Algoritma *Run Length Encoding* (RLE) terhadap Metode Kompres Data Teks menggunakan Aplikasi *Python*. Literatur diperoleh dari berbagai sumber antara lain dari jurnal ilmiah tentang penelitian yang terkait, hasil penelitian, dan *browsing* di internet. [11],[12],[13].
2. Perancangan Bahasa pemrograman *python* Pada tahap ini, dilakukan perancangan atau perencanaan implementasi menggunakan bahasa pemrograman *Python*. Ini mencakup pemilihan algoritma, struktur data, dan desain umum dari implementasi yang akan dilakukan.[14], [15].
3. Implementasi Bahasa pemrograman *Python* pada Algoritma *Run Length Encoding* (RLE) pada kompres data teks Tahap ini merupakan pelaksanaan dari rencana yang telah dirancang sebelumnya. Algoritma *Run Length Encoding* (RLE) digunakan untuk mengompresi data teks menggunakan bahasa pemrograman *Python*. [16]
4. Hasil Perancangan dan Implementasi Algoritma *Run Length Encoding* (RLE) pada kompres data teks Pada tahap ini, hasil dari implementasi RLE dievaluasi dan didokumentasikan. Ini termasuk analisis performa, efektivitas kompresi, dan pemecahan masalah yang mungkin timbul selama proses implementasi [17].
5. Kesimpulan :Tahap terakhir dari proyek atau penelitian ini adalah menyimpulkan temuan dan hasil dari keseluruhan proses. Kesimpulan ini mencakup ringkasan dari tujuan proyek, hasil yang dicapai, implikasi dari temuan tersebut, dan saran untuk penelitian atau pengembangan lebih lanjut.

4. Hasil dan Pembahasan

Pada bagian ini menjelaskan proses analisis data yang didapatkan pada saat penelitian dengan analisis sebagai berikut:

4.1 Implementasi Program

Perhitungan manual Algoritma *Run-Length Encoding* (RLE) dapat di terapkan dalam program dalam penelitian ini menggunakan Bahasa pemrograman *python*. *Python* mendukung program untuk dijalankan sebab termasuk kategori bahasa tingkat tinggi yang juga mempermudah manusia dalam analisa logika perhitungannya.

4.2 Perangkat lunak yang digunakan

Dalam membuat Aplikasi Kompresi Data Menggunakan Metode Algoritma *Run-Length Encodin* (RLE) aplikasi *python* diperlukan beberapa perangkat lunak (*software*) untuk membantu pengerjaan aplikasi. Perangkat lunak yang digunakan didalam pembuatan aplikasi dapat dilihat pada Tabel 1.

Tabel 1. Tabel kebutuhan perangkat lunak (*software*)

No.	Perangkat Lunak	Keterangan
1.	<i>Windows 7 Home Premium</i>	Sistem Operasi
2.	<i>Visual Code Studio</i>	Perangkat lunak untuk <i>design interface</i> dan <i>code editor</i>
3.	<i>Python 3.12 (64-Bit)</i>	Perangkat lunak untuk proses <i>debugging</i> program dari <i>code editor</i>
4.	<i>Library streamlit</i>	Untuk implementasi user interfacenya ke websait

4.3 Perangkat keras yang digunakan

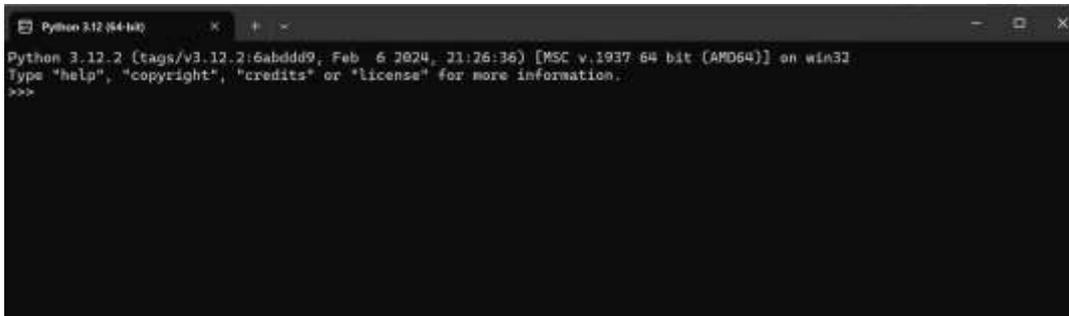
Dalam membuat Aplikasi Kompresi Data Menggunakan Metode Algoritma *Run-Length Encodin* (RLE) Pada aplikasi *Python* diperlukan beberapa perangkat keras (*hardware*) untuk membantu pengerjaan aplikasi. Perangkat keras yang digunakan didalam pembuatan aplikasi dapat dilihat pada Tabel 2.

Tabel 2. Tabel kebutuhan perangkat keras (*hardware*)

No.	Perangkat Keras	Keterangan
1.	Laptop	Lenovo ideapad slim 3 (N6087LMS)
2.	Processor	Intel® Core™ i3-10110U CPU @ 2.10GHz 2.59 GHz
3.	Memory	4.00 GB (3,84 GB usable)
4.	<i>System type</i>	64-bit operating system, x64-based processor

4.4 Tampilan Awal Aplikasi *Python*

Python dapat digunakan membaca dan memodifikasi file, *Python* dapat digunakan untuk menangani data besar dan melakukan matematika yang kompleks ditampilkan dalam Gambar 2.

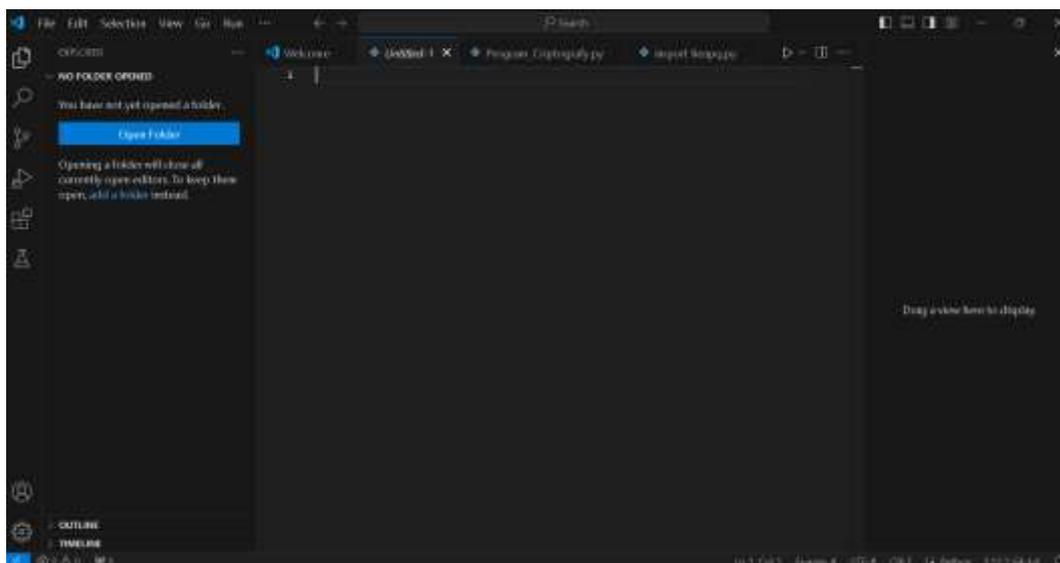


Gambar 2. Tampilan aplikasi *python*

Pada gambar 2 menunjukkan tampilan awal mengaktifkan *Visual studio code python* yang dapat digunakan untuk pembuatan prototipe dengan cepat, atau untuk pengembangan perangkat lunak siap produksi.

4.5 Tampilan *Visual Studio Code*

Visual Studio Code menggunakan *opensource* NET perkakas untuk memberikan dukungan untuk ASP.NET C # kode, membangun alat pengembang Omnisharp NET dan *compiler Roslyn* ditampilkan dalam Gambar 3.

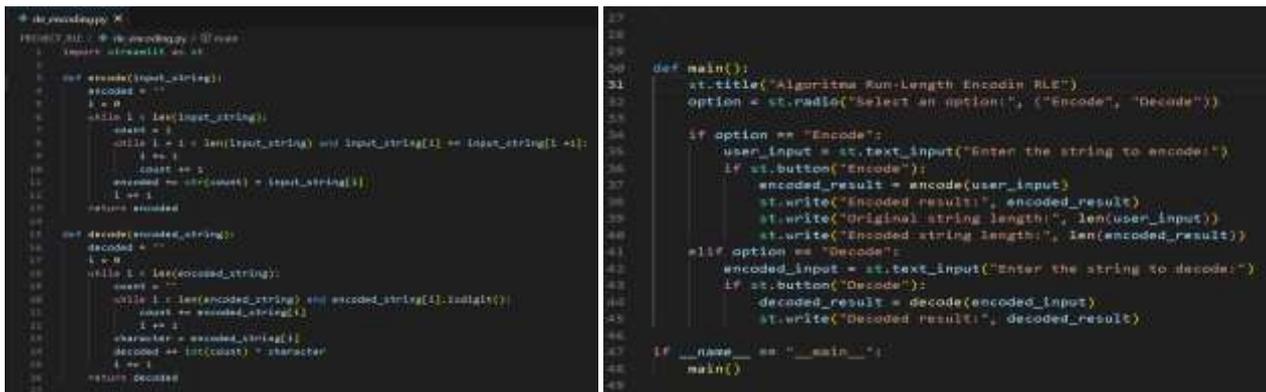


Gambar 3. Tampilan *visual studio code*

Gambar 3 menunjukan pemrograman tampilan *visual studio code* dengan antarmuka yang mudah untuk bekerja dengannya, karena didasarkan pada gaya *explorer* umum, dengan *panel* di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari file yang telah dibuka. dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan *intellisense* dan *autocomplete* bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js.

4.6 Aplikasi *python* kompresi teks pada Algoritma *Run-Length Encodin (RLE)*

Perhitungan manual *Run Length Encoding (RLE)* dapat diterapkan dalam program dalam penelitian ini menggunakan bahasa pemrograman *python* ditampilkan dalam Gambar 4.



```
def encode(input_string):
    encoded = ""
    i = 0
    while i < len(input_string):
        count = 1
        while i + 1 < len(input_string) and input_string[i] == input_string[i + 1]:
            count += 1
            i += 1
        encoded += str(count) + input_string[i]
        i += 1
    return encoded

def decode(encoded_string):
    decoded = ""
    i = 0
    while i < len(encoded_string):
        count = ""
        while i < len(encoded_string) and encoded_string[i].isdigit():
            count += encoded_string[i]
            i += 1
        character = encoded_string[i]
        decoded += int(count) * character
        i += 1
    return decoded

def main():
    st.title("Algoritma Run-Length Encodin RLE")
    option = st.radio("Select an option:", ("Encode", "Decode"))

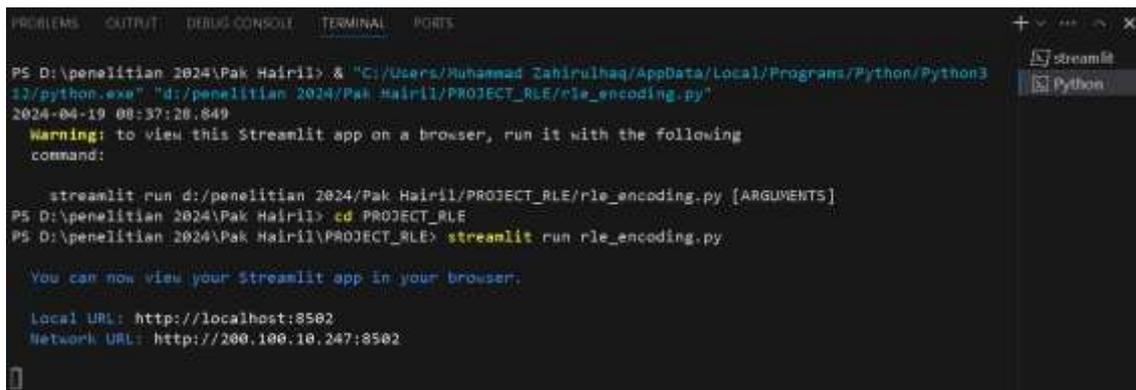
    if option == "Encode":
        user_input = st.text_input("Enter the string to encode:")
        if st.button("Encode"):
            encoded_result = encode(user_input)
            st.write("Encoded result:", encoded_result)
            st.write("Original string length:", len(user_input))
            st.write("Encoded string length:", len(encoded_result))
    elif option == "Decode":
        encoded_input = st.text_input("Enter the string to decode:")
        if st.button("Decode"):
            decoded_result = decode(encoded_input)
            st.write("Decoded result:", decoded_result)

if __name__ == "__main__":
    main()
```

Gambar 4. Tampilan pemrograman python algoritma *run-length encoding* (RLE)

Gambar 4 menunjukkan *source code* perancangan tampilan pemrograman python dengan menggunakan algoritma *run length encoding* (RLE). Mulai dari import *streamlit*, *while*, *return*, *encode* dan *decoded* pada teks yang di kompres.

Setelah berhasil pembuatan perancangan pemrograman teks *encode* dan *decode* selanjutnya program di jalankan dengan menekan enter maka tampilan pemrograman yang dihasilkan adalah file data program dan *source code* yang tersimpan yaitu *rle_encoding.py* dan tanggal dan tahun ditampilkan dalam Gambar 5.



```
PS D:\penelitian 2024\Pak Hairil> "C:/Users/Muhammad Zahirulhaq/AppData/Local/Programs/Python/Python313/python.exe" "d:/penelitian 2024/Pak Hairil/PROJECT_RLE/rle_encoding.py"
2024-04-19 08:37:28.849
Warning: to view this Streamlit app on a browser, run it with the following command:
    streamlit run d:/penelitian 2024/Pak Hairil/PROJECT_RLE/rle_encoding.py [ARGUMENTS]
PS D:\penelitian 2024\Pak Hairil> cd PROJECT_RLE
PS D:\penelitian 2024\Pak Hairil\PROJECT_RLE> streamlit run rle_encoding.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8502
Network URL: http://200.100.10.247:8502
```

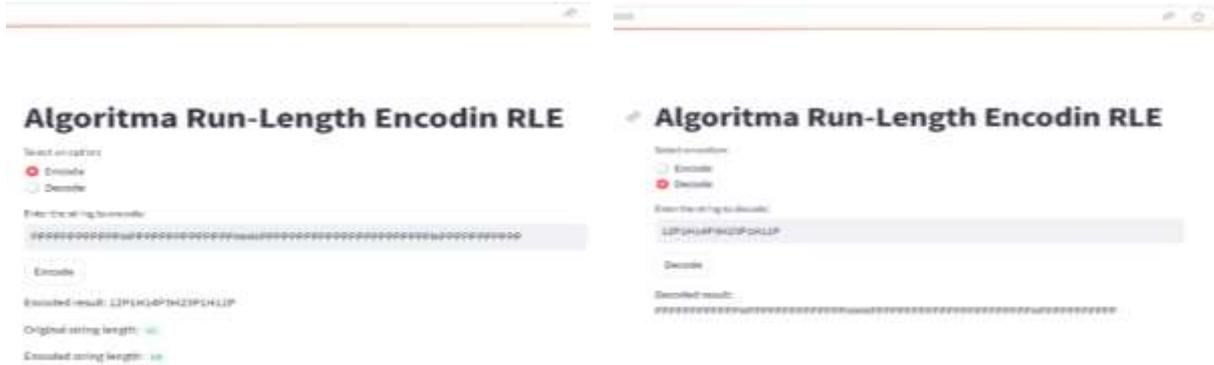
Gambar 5. Tampilan setelah pemrograman dijalankan

Gambar 5 menunjukkan setelah program dijalankan pada warning: *to view streamlit app on a browser, run it eith following command: streamlit run d:* ketik *cd project_RLE* tekan enter selanjutnya mengetk *streamlit run rle_encoding.py* (file yang telah disimpan). Kemudian tekan enter seperti hasil ditampilkan dalam Gambar 6.



Gambar 6. Tampilan hasil python algoritma *run-length encodin* (RLE) pada *website*

Setelah tampilan website terbuka selanjutnya dilakukan pengujian dengan mengetik studi kasus yaitu:
PPPPPPPPPPPPHPPPPPPPPPPPPPPHHPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPHPPPPPPPPPPPP
Maka hasilnya adalah 12P1H14P3H23P1H11P untuk lebih jelasnya dapat dilihat pada gambar 7 dibawah ini adalah sebagai berikut.



Gambar 7. Tampilan hasil python algoritma run-length encodin (RLE) pada website

Gambar 7 menunjukkan hasil pemrograman *python* pada *website* setelah di *running* dengan me

culkan studi kasus pada Encode dan Decode ditampilkan dalam Gambar 8.



Gambar 8. Tampilan hasil python algoritma run-length encodin (RLE) decode pada website

Gambar 8 menunjukkan hasil lanjutan dengan mengetik Ternatemalukuutara kompresinya menjadi: IT1e1r1n1a1t1e1m1a1l1u1k2u1t1a1r1a Original string length: 18 Encoded string length: 34

5. Kesimpulan

Berdasarkan penelitian yang telah dilakukan terhadap perancangan aplikasi *Python* menggunakan *Algoritma Run-Length Encoding* (RLE), dapat diambil beberapa kesimpulan. Hasil perancangan aplikasi menunjukkan bahwa metode kompresi data *Run-Length Encoding* (RLE) efektif dalam memangkas ukuran data teks. Penggunaan RLE berhasil mengurangi penggunaan ruang penyimpanan secara signifikan, yang pada gilirannya meningkatkan efisiensi penyimpanan data. Bahasa pemrograman *Python* terbukti andal dalam pengembangan aplikasi kompresi data. *Python* mempermudah proses perancangan dan implementasi algoritma kompresi, serta memungkinkan pembuatan prototipe dengan cepat dan pengembangan perangkat lunak siap produksi. Implementasi RLE pada aplikasi berbasis *Python* berjalan dengan baik dan sesuai dengan tujuan penelitian. Aplikasi ini mampu mengkompresi data teks dengan efisien, meminimalkan penggunaan ruang penyimpanan, dan tetap menjaga integritas data selama proses kompresi dan dekompresi. Penelitian ini memberikan dasar yang kuat untuk pengembangan lebih lanjut dalam bidang kompresi data menggunakan *Python*. Algoritma RLE yang sederhana namun efektif ini dapat diadaptasi dan dikembangkan lebih lanjut untuk berbagai jenis data selain teks, seperti gambar dan video. Dengan kemampuan kompresi yang ditawarkan, teknologi kompresi data menggunakan RLE memiliki potensi besar untuk dimanfaatkan dalam berbagai aplikasi teknologi masa depan, terutama dalam konteks di mana efisiensi ruang penyimpanan dan kecepatan pengiriman data menjadi sangat

penting. Dengan kesimpulan ini, diharapkan penelitian ini dapat memberikan kontribusi yang berarti bagi pengembangan teknologi kompresi data dan aplikasi berbasis *Python* di masa mendatang.

Referensi

- [1] H. C. Rustamaji, M. Mariani, and B. Yuwono, "Aplikasi Kompresi Data Menggunakan Metode Huffman Statik pada Perangkat Mobile Berbasis Android," *Telematika*, vol. 11, no. 1, 2015, doi: 10.31315/telematika.v11i1.311.
- [2] J. Adisantoso, D. Dimas Sulistio, and B. Paruhum Silalahi, "Seminar Nasional Aplikasi Teknologi Informasi 2004 Kompresi Data Menggunakan *Algoritme Huffman*," *Snati*, vol. 1, no. 1, pp. 1–8, 2004.
- [3] E. Prayoga and K. M. Suryaningrum, "Implementasi Algoritma *Huffman* dan *Run Length Encoding* pada Aplikasi Kompresi Berbasis Web," *J. Ilm. Teknol. Infomasi Terap.*, vol. 4, no. 2, pp. 92–101, 2018, doi: 10.33197/jitter.vol4.iss2.2018.154.
- [4] N. L. Sugara, T. W. Purboyo, and A. L. Prasasti, "Implementasi dan Analisis Efektivitas Discrete Wavelet Transform dan *Huffman* Pada Berbagai Jenis Citra Digital," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 6163–6170, 2019.
- [5] U. Lu'luilmaknun and N. H. Salsabila, "Penggunaan Metode *Run Length Encoding* untuk Kompresi Data," *Semin. Mat. Dan Pendidik. Mat. Uny 2017*, no. 1, pp. 273–280, 2017.
- [6] T. Suhesti, "Bahasa Pemrograman Python," *J. Ilmuti Ilmu Teknol. Inf.*, vol. 2, no. 11, pp. 1–6, 2014.
- [7] D. Willfrid, M. Simamora, G. Ginting, and Y. Hasan, "Implementasi Algoritma *Run Length Encoding* Pada Kompresi File Mp3," *Jurikom (Jurnal Ris. Komputer)*, vol. 3, no. 4, pp. 5–9, 2016.
- [8] N. Adillah, Y. R. Nasution, and Armansyah, "Perbandingan Algoritma *Run Length Encoding* (RLE) dan *Algoritma Variable Length Binary Encoding* (VLBE) dalam Mengkompresi File Video Untuk Menghemat Penyimpanan," *G-Tech J. Teknol. Terap.*, vol. 7, no. 4, pp. 1203–1212, 2023, doi: 10.33379/gtech.v7i4.3020.
- [9] S. Rahma and A. Prapanca, "Analisis Kompresi dan Dekompresi Data Teks dan Audio dengan Algoritma *Run Length Encoding* (RLE)," *J. Informatics Comput. Sci.*, vol. 2, no. 04, pp. 313–320, 2021, doi: 10.26740/jinacs.v2n04.p313-320.
- [10] A. I. Basri, W. P. Sumiyar, and V. A. Tisya, "Pemanfaatan *Flowchart* Untuk Memudahkan Dalam Proses Bisnis Kerjasama Daerah Pemerintah Kota Yogyakarta," *ABDIMAS Nusant. J. Pengabd. Kpd. Masy.*, vol. 3, no. September, pp. 34–37, 2022.
- [11] U. Mansyuri, "Kompresi Data Teks Dengan Metode *Run Length Encoding*," *J. Ilm. Sist. Inf.*, vol. 1, no. 2, pp. 102–109, 2021.
- [12] M. R. Ramadhan, "Analisa Perbandingan *Algoritma Run Length Encoding* Dengan *Burrows-Wheeler* Transform Dalam Kompresi File Video," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 322–332, 2023, doi: 10.30865/komik.v6i1.5716.
- [13] Y. Reswan and L. Tri Anggoro, "Implementasi Metode Cepat Kompresi File Document Otomatis Pada Aplikasi Berbasis Web Menggunakan Algoritma Additive Code," *J. Media Inform.*, vol. 20, no. 1, pp. 211–215, 2024.
- [14] M. Rizq Daffa Jodi, "Fakultas Komputer Algoritma dan Struktur data," *Fak. Kompiter*, vol. 1, pp. 1–10, 2020.
- [15] Jaka Naufal Semendawai, Indah Febiola, Bima Pamungkas, and Muhammad Deka Ruliansyah, "Perancangan Aplikasi Otomatisasi Menggunakan Bahasa Pemrograman Python Pada Aktivitas Monitoring Pemakaian Data Harian Kartu *Internet Of Things*," *J. Rekayasa Elektro Sriwij.*, vol. 3, no. 1, pp. 193–198, 2021, doi: 10.36706/jres.v3i1.42.
- [16] M. Romzi and B. Kurniawan, "Implementasi Pemrograman Python Menggunakan Visual Studio Code," *urnal Inform. dan Komput.*, vol. XI, no. 2, pp. 1–9, 2020, [Online]. Available: www.python.org
- [17] C. Try Utari, "Implementasi Algoritma *Run Length Encoding* untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra," *J. TIMES*, vol. V, no. 2, pp. 24–31, 2016.