TALENTA Conference Series: Science & Technology

PAPER – OPEN ACCESS

Analisis Dan Implementasi Algoritma Kriptografi Playfair Chiper Dan Algoritma Kompresi Run Length Encoding Dalam Pengamanan Dan Kompresi Data Teks

Author : Ananda Dwi Putri

DOI : 10.32734/st.v1i1.191

Electronic ISSN : 2654-7085

Print ISSN : 2654-7077

Volume 1 Issue 1 – 2018 TALENTA Conference Series: Science & Technology (ST)

SS Conference Series 01 (2018), Page 069–077

TALENTA Conference Series

Available online at https://talentaconfseries.usu.ac.id

Analisis dan Implementasi Algoritma Kriptografi *Playfair Chiper* dan Algoritma Kompresi *Run Length Encoding* Dalam Pengamanan dan Kompresi Data Teks

Ananda Dwi Putri[a], Dian Rachmawati[a], Herriyance[a]

Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Medan-20155

ananda_dwi_putri@students.usu.ac.id, dee230783@gmail.com, herriyance@usu.ac.id

Abstrak

Komunikasi dan bertukar informasi secara jarak jauh sudah sangat mudah dan praktis. Kemudahan ini menuntut peningkatan keamanan terhadap kerahasiaan data yang dikirim. Kriptografi adalah salah satu cara yang digunakan untuk menjaga kerahasian dari sebuah pesan, dimana pesan disamarkan menjadi sandi. Selain keamanan data yang perlu diperhatikan juga adalah kecepatan dalam pengiriman data tersebut. Kecepatan pengiriman ini tergantung dari ukuran informasi tersebut. Kompresi adalah proses pengubahan sekumpulan data menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Dalam penelitian ini penulis mengkombinasi algoritma kriptografi *Playfair Cipher* dengan algoritma kompresi *Run Length Encoding*, serta menganalisa kedua algoritma menggunakan kompleksitas algoritma. waktu eksekusi pesan dengan 16 karakter adalah 0.54423 sekon, waktu eksekusi pesan dengan 78 karakter adalah 1.14617 sekon, dan waktu eksekusi pesan dengan 189 karakter adalah 1.51715 sekon. Hasil pengujian proses kompresi *string Homogen* dengan kompresi rasio rata-rata sebesar 33.83% dan *sring Heterogen* dengan kompresi rasio rata-rata sebesar 15,54%. Dapat disimpulkan jumlah karakter pada pesan berbanding lurus terhadap waktu.

Kata Kunci: Kriptografi; *PlayfairCipher*; Kompresi; *Run Length Encoding*

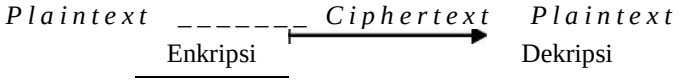
1. Pendahuluan

Dewasa ini, komunikasi dan bertukar informasi secara jarak jauh sudah sangat mudah dan praktis. Kemudahan ini menuntut peningkatan sekuritas (keamanan) terhadap kerahasiaan data yang dikirim. Ada beberapa cara dan teknik yang digunakan untuk menjaga keamanan kerahasiaan dari sebuah pesan yang dikirim. Salah satunya adalah kriptografi, di mana pesan disamarkan menjadi pesan tersandi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan. Pesan asli biasanya disebut juga sebagai plaintext. Sedangkan pesan yang sudah diamankan disebut ciphertext[5]. Algoritma Playfair Cipher ditemukan oleh Sir Charles Wheastone pada tahun 1854, namun dipromosikan oleh Baron Lyon Playfair. Algoritma ini merupakan salah satu dari kriptografi klasik yang proses enkripsinya menggunakan pemrosesan dalam bentuk blok-blok yang sangat besar. Metode ini merupakan salah satu cara untuk mengatasi kelemahan metode kriptografi klasik lainnya yang mudah tertebak karena terdapat korespondensi satu-satu antara plainteks dengan cipherteks[3].

Selection and peer-review under responsibility of Seminar Ilmiah Nasional Dies Natalis USU-64

2. Tinjauan Pustaka

Secara etimologi, kata *Cryptography* berasal dari bahasa Yunani yang terdiri dari dua kata yaitu *kryptos* yang berarti rahasia dan *graphein* yang berarti tulisan[4].Pada kriptografi, pesan asli disebut *plaintext* dan pesan yang disamarkan disebut *ciphertext.* Proses menyamarkan *plaintext* menjadi *ciphertext* disebut enkripsi. Sedangkan proses untuk mengubah *ciphertext* kembali menjadi *plaintext* disebut dekripsi. Adapun skema dari kriptografi dapat dilihat pada Gambar 1.



Gambar 1: Skema Proses Kriptografi

Kompresi adalah sebuah proses mengubah ukuran data menjadi lebih kecil dari ukarannya yang semula sehingga dapat lebih menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data.

2.1. Playfair Cipher Dengan Teknik Pemutaran Kunci

Proses enkripsi dengan menggunakan algoritma *Playfair Cipher*, dilakukan dengan mengenkripsi dua huruf atau pasangan-pasangan huruf. Kunci yang di gunakan dalam proses enkripsi harus disepakati oleh pengirim dan penerima pesan terlebih dahulu, agar pesan dapat di deskripsi oleh penerima pesan. Kunci tersebut disusun pertama kali dalam sebuah bujur sangkar yang memiliki ukuran 5x5, setelah itu sisa dari elemen-elemen bujur sangkar yang masih kosong akan diisi dengan seluruh alfabet (A-Z). Teknik pemutaran kunci dua arah merupakan suatu cara untuk mengganti matriks agar berbeda dari kunci matriks untuk mengenkripsi *bigram* sebelumnya. Teknik pemutaran kunci dua arah dilakukan dengan memutar 4 huruf disekitar huruf pertama dengan menempatkan huruf pertama yang dienkripsi searah jarum jam, kemudian memutar 4 huruf disekitar huruf kedua berlawanan arah jarum jam[1].

Sebagai contoh, misalkan kata kunci yang disetujui oleh pengirim dan penerima pesan adalah ILKOM. Contoh matriks kunci *Playfair Cipher* dengan penulisan kunci dalam baris dapat dilihat pada Tabel 1.

Tabel 1: Contoh matriks kunci

I	**L**	**K**	**O**	**M**
A	**B**	**C**	**D**	**E**
F	**G**	**H**	**N**	**P**
Q	**R**	**S**	**T**	**U**
V	**W**	**X**	**Y**	**Z**

Contoh *Plaintext*: ANANDA

Contoh pada matriks kunci "ILKOM" pada Tabel 1, *bigram* "AN" terlebih dahulu dienkripsi, hasilnya yaitu "DF" baru matriks kunci diputar. Pada matrik kunci, huruf yang berada disekitar 'A‘ dapat dilihat pada Tabel 2. Kemudian matrik 4 huruf tersebut diputar searah jarum jam, sehingga hasilnya dapat dilihat pada Tabel 3. Selanjutnya matrik kunci, huruf yang berada disekitar 'N‘ dapat dilihat pada Tabel 4. Kemudian matrik 4 huruf tersebut diputar berlawanan jarum jam, sehingga hasilnya dapat dilihat pada Tabel 5.

Tabel 2: Matriks 4 huruf' A sebelum diputar

I	L	K	O	M
A	B	C	D	E
F	G	H	N	P
Q	R	S	T	U
V	W	X	Y	Z

Table 3: Matriks 4 huruf ' A setelah diputar

I	L	K	O	M
F	A	C	D	E
G	B	H	N	P
Q	R	S	T	U
V	W	X	Y	Z

Table 4: Matriks 4 huruf ' N sebelum diputar

I	L	K	O	M
F	A	C	D	E
G	B	H	N	P
Q	R	S	T	U
V	W	X	Y	Z

Table 5: Matriks 4 huruf 'N setelah diputar

I	L	K	O	M
F	A	C	D	E
G	B	H	P	U
Q	R	S	N	T
V	W	X	Y	Z

Kunci Matriks pada Tabel 5 digunakan untuk mengenkripsi *bigram* kedua pada *plaintext*. Proses ini terus berulang sampai *plaintext* habis dienkripsi. Dengan menggunakan teknik pemutaran kunci dua arah, maka *ciphertext* yang dihasilkan dari plaintext "ANANDA" adalah:

Ciphertext yang dihasilkan: DFDRED

2.2. Run Length Encoding

Algoritma *Run Length Encoding* mengurangi ukuran karakter string yang berulang.Algoritma ini memanfaatkan karakter yang berulang secara berurutan pada sebuah data dengan mengkodekannya dengan sebuah string yang terdiri dari jumlah karakter yang berulang dan diikuti dengan karakter itu sendiri.Sehingga banyak tidaknya karakter yang berulang pada sebuah data menjadi penentu keberhasilan kompresi algoritma *RLE*.

Contoh :

String : "AAABBBBBCCDDDCCCCC"

Kompresi *RLE* : "#3A#4B#2C#3D#5C"

3. Analisis Dan Perancangan

3.1. AnalisisMasalah

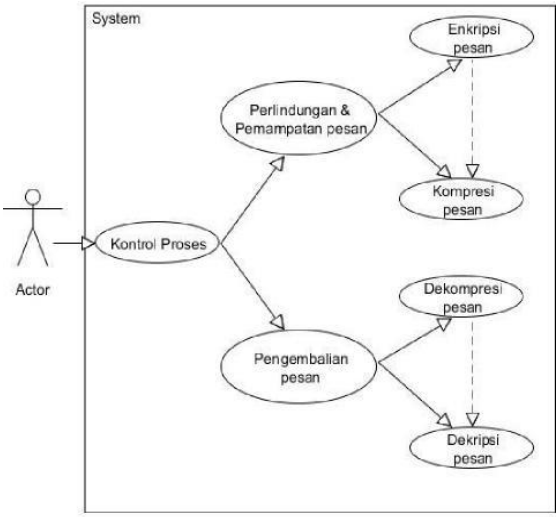
Untuk dapat menghindari ancaman dalam proses pengiriman dan penerimaan data rahasia, diperlukan sebuah algoritma enkripsi untuk penyandian data agar data menjadi sulit dimengerti. Namun, agar penerima yang dituju dapat membaca data yang dikirim, diperlukan algoritma dekripsi untuk mengembalikan data asli seperti sebelum penyandian. Selanjutnya untuk menghindari lamanya waktu pengiriman data dalam ukuran yang besar dibutuhkan algoritma kompresi dan dekompresi data

3.2. Perancangan

- Use Case Diagram

Use case adalah rangkaian uraian sekelompok yang saling terkait dan membentuk sebuah sistem secara teratur yang dilakukan oleh *actor*. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan dengan kolaborasi[7].

Adapun *use case* dari sistem pada penelitian ini dapat dilihat pada Gambar 2.

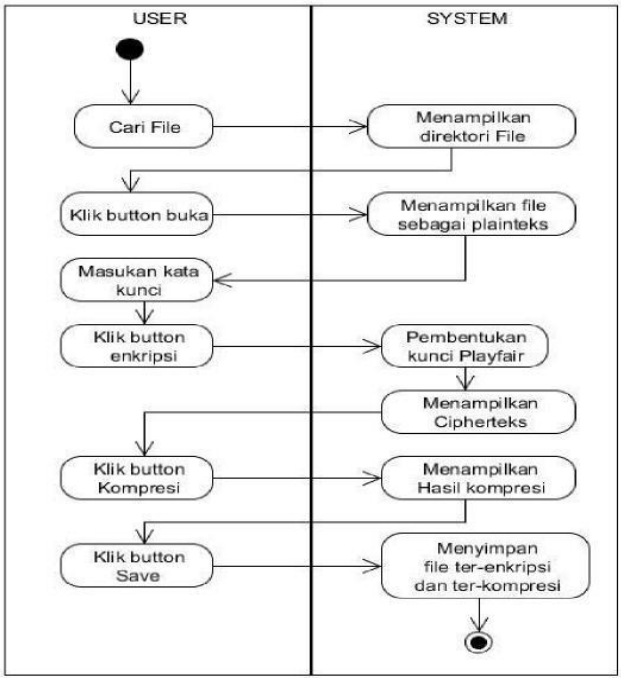


Gambar 2: *Use Case Diagram* Sistem

- Activity Diagram

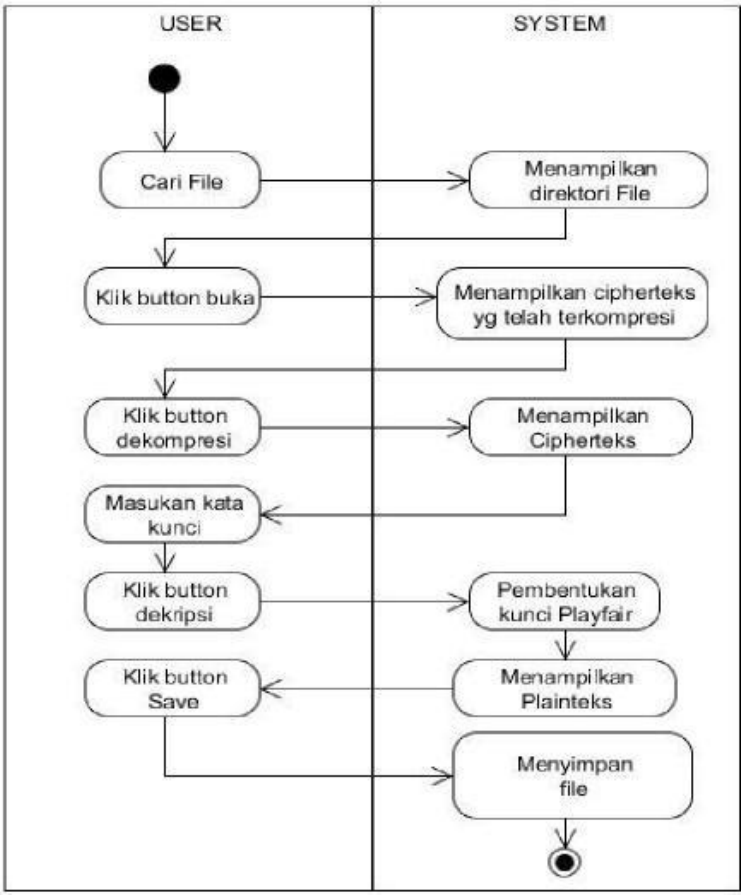
Activity diagram juga merupakan bagian dari pemodelan sistem, menampilkan gambaran berbagai alur aktivitas dalam sistem yang sedang dirancang. *Activity diagram* lebih menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum[7].

Adapun activity diagram untuk enkripsi-kompresi dapat dilihat pada Gambar 3



Gambar 3: Activity Diagram Enkripsi-Kompresi

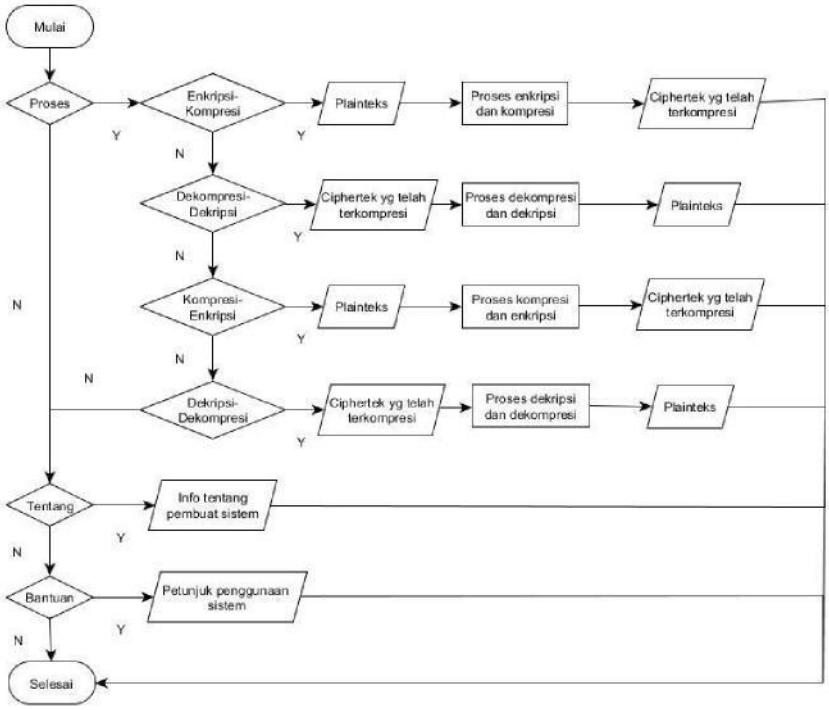
Adapun activity diagram untuk dekompresi-dekripsi dapat dilihat pada Gambar 4.



Gambar 4: Activity Diagram Dekompresi-Dekripsi

- Flowchart Sistem.

Flowchart merupakan suatu bagan yang menggambarkan urutan suatu proses secara rinci menggunakan simbol-simbol tertentu dan menggambarkan hubungan antara satu proses dengan proses lainnya dengan menggunakan tanda panah. *Flowchart* sistem dapat dilihat pada Gambar 5.



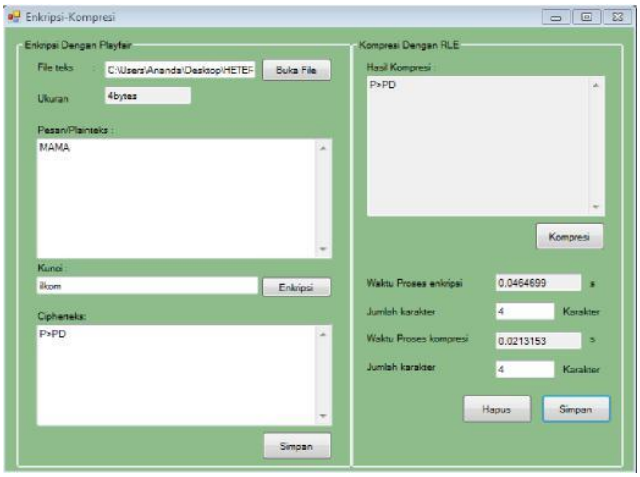
Gambar 5: *Flowchart* Sistem

4. Implementasi Dan Pengujian

4.1. Implementasi

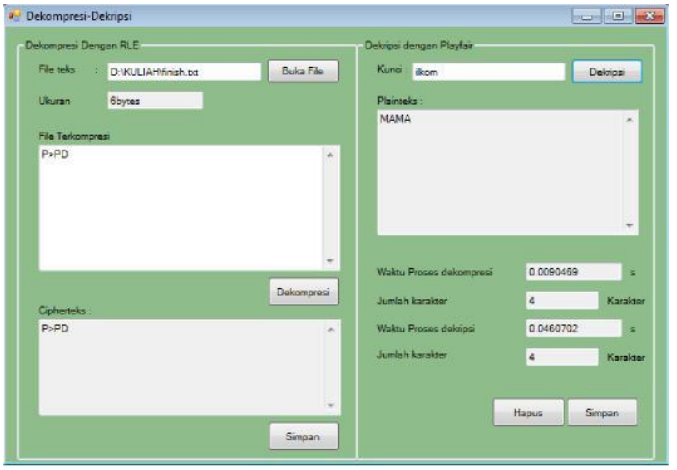
Setelah sistem dianalisis dan dirancang, tahap selanjutnya adalah mengimplementasikan sistem tersebut kedalam bahasa pemrograman. Sistem ini dibangun dengan menggunakan Microsoft Visual Basic 2010. Pada sistem ini terdapat empat halaman, yaitu halaman Beranda, halaman Proses, halaman Tentang, dan halaman Bantuan.

Berikut merupakan tampilan dari proses enkripsi-kompresi yang dapat dilihat pada Gambar 6.



Gambar 6: Tampilan Hasil Enkripsi-Kompresi

Berikut merupakan tampilan dari proses Dekompresi-Dekripsi yang dapat dilihat pada Gambar 7.



Gambar 7: Tampilan Hasil Dekompresi-Dekripsi

4.2. Pengujian

Pada pengujian tugas akhir ini parameter yang dihitung adalah waktu dari kedua algoritma melakukan proses enkripsi-dekripsi dan kompresi-dekompresi serta ukuran file setelah proses kompresi dan dekompresi. Digunakan pula notasi big-O untuk menganalisa kompleksitas algoritma.

- Pengujian Algoritma terhadap Waktu

Pada pengujian kali ini akan dilakukan dengan menggunakan 3 *plaintext* dengan jumlah karakter yang berbeda. Pada masing-masing *plaintext* akan dilakukan pengujian sebanyak 3 kali dengan kunci yang berbeda terhadap waktu proses enkripsi dan kompresinya. Pengujian terhadap waktu proses untuk plainteks sebanyak 16 karakter dapat dilihat pada Table 6.

Tabel 6: Waktu Proses untuk Plainteks 16 Karakter

Plainteks	**Cipherteks**	**HasilKompresi**	**Waktu Proses**	**Rata-rata waktu**
AnandaDwi Putri	Bjetec!CykoNquvk	Bjetec!CykoNquvk	0,1153196	0.1814109
	Fs#xibeCxg#Nzpym	Fs#xibeCxg#Nzpym	0.3534246	
	@an' #&CygnSyppl	@an' #&CygnSy2pl	0.0754886	

Pengujian terhadap waktu proses untuk plainteks sebanyak 78 karakter dapat dilihat pada Table 7.

Tabel 7: Waktu Proses untuk Plainteks 78 karakter

Plainteks	**Cipherteks**	**HasilKompresi**	**Waktu Proses**	**Rata-rata waktu**
Kompresi lossless run length encoding dankripto grafisime trisplayfa ir cipher	P#istcpoi!m}ymnokiv}wl*zhn nr%}hn&akaj.rcj2mq$dd{r!ga ~5({$rdvs(;qob)zh"<tl{qke Lnnlqcjg%jrioi)ov$}st+x$mhg q,+jkyqmqz0mkz/!uqzgtyakjt7 bqo,q$qb9jt#ghkt&:a\|hn6+ Nkjkpgmhdrujpwquoatl +qgaflkdl"^s!}'{a%d- nbtqu}urj$}_%x_tjc|xe#'v"xg. q)ate$v&-	P#istcpoi!m}ymnokiv}wl*zh 2nr%}hn&akaj.rcj¥2mq$2d{r !ga~¥5({$rdvs(;qob)zh2'<tl{q ke L2nlqcjg%jrioi)ov$}st+x$mh gq,+jkyqmqz¥0mkz/!uqzgtya kjt¥7bqo,q$qb¥9jt#ghkt&:a\|h n¥6+ Nkjkpgmhdrujpwquoatl +qgaflkdl"^s!}'{a%d- nbtqu}urj$}_%x_tjc|xe#'v"xg .q)ate$v&-	0,36191 15 0.40922 15 0.37504 35	0.382058 8

Pengujian terhadap waktu proses untuk plainteks sebanyak 189 karakter dapat dilihat pada Tabel 8.

Tabel 8: Waktu Proses untuk Plainteks 189 Karakter.

Plainteks	Cipherteks	Hasil Kompresi	Waktu Prose	Rata – rata waktu
Kompresi adalah sebuah proses mengubah ukuran data menjadi lebih kecil dari ukarannya yang semula sehingga dapat lebih meng hemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data	P#istcpo$bfjoqw"qtv{~f){hk\|sr,&igxhvuzrhypid *gedh0eotd!ge1qunmq8 ipju!0j,{r1epl(lbyp,<qlx w<\|m'i24<f&)&q\|n4?}1 g8w@- rg)(G+(oq%(,3yGrrpiz& 45/Bi+*!li@&zzou8o6/ >&FsD&PjHnviLi'bi/Uy !K&o5tq/[sJ(S"j Lnnlqcjg#bh#s&f#g(%z .ialhif.g&h0wnz%+r(\|$c o.w1q/f1ei dt2jy4n !xi1z#mv2s<fm/z$<i=d x};2o9}f9j7q$};>f4y||nx @FvCfHj@"?.$mB}2fh $2u=%;.:- l\|'dE\|@h2zvF#@f:$'m+v HoNrCxUoQqY4'xRdg md*Sc$ "%+}`oX$`lw3 g	P#istcpo$bfjoqw"qtv{~f){h k\|sr,&igxhvuz- rhypid*gedh¥0eotd!ge¥1qu nmq¥8ipju!¥0j,{r¥1epl(lby p,<qlxw<\|m'i¥2¥4<f&)&q\| n¥4?}¥1g¥8w@- rg)(G+(oq%(,¥3yG2rpiz&¥ 4¥5/Bi+*!li@&2zou¥8o¥6/ >&FsD&PjHnviLi'bi/Uy!K &o¥5tq/[sJ(S"j L2nlqcjg#bh#s&f#g(%z.ial hif.g&h¥0wnz%+r(\|$co.w¥ 1q/f¥1ei dt¥2jy¥4n !xi¥1z #mv¥2s<fm/z$<i=dx};¥2o¥ 9}f¥9j¥7q$};>f¥4y2|nx@F vCfHj@"?.$mB}¥2fh$¥2u =%;.:- l|'dE|@h¥2zvF#@f:$'m+vH oNrCxUoQqY¥4'xRdgmd* Sc$_"%+}`oX$`lw¥3g	0,4802678 0.4543715	0.505719

Nkjkpgmh!da(k#f%ps^o *s#io}qwq)et$f~_*g(ex) k'$,% k2.o|ei6&g*`{`ed5{zlob 5#<lm."|<e;+kc;.|>+p8w ryh|B8ti)oinoBD)HsF|D pgj~)>or{u&noF+<"xy$ xduMtBr"kbH$Bsok-- }bJtS$GtX$KvY'k$J$0(00S'iv$lkjlsXmfh`4%	Nkjkpgmh!da(k#f%ps^o*s #io}qwq)et$f~_*g(ex)k'$, % k¥2.o|ei¥6&g*`{`ed¥5{zlob ¥5#<lm."|<e;+kc;.|>+p¥8wr yh|B¥8ti)oinoBD)HsF|Dpgj ~)>or{u&noF+<"xy$xduM tBr"kbH$Bsok2- }bJtSGtXKvY'kJ¥0(2 ¥0S'iv$lkjlsXmfh`¥4%	0.5825178

5. Kesimpulan

- Berdasarkan grafik hubungan antara waktu proses dengan panjang plainteks diperoleh hasil panjang plainteks berbanding lurus terhadap waktu. Semakin panjang plainteks maka waktu yang dibutuhkan juga akan semakin banyak.
- Hasil pengujian kompresi String dengan karakter yang sama (homogen) berdasarkan variabel Ratio of compression (Rc), Compression ratio (Cr), Redundancy (Rd) dengan rasio kompresi rata-rata sebesar 33,83% dan String dengan karakter berbeda (heterogen) sebesar 15,54%.
- Kombinasi dengan mendahulukan proses kompresi dilanjutkan dengan proses enkripsi teks lebih baik digunakan untuk teks dengan banyak perulangan berurut, karena kombinasi keduanya berhasil terkompresi dengan baik.
- Untuk proses kompresi maupun dekompresi algoritma Run Length Encoding memiliki kompleksitas yang sama yaitu O(n), sedangkan untuk proses enkripsi dan dekripsi algoritma Playfair Cipher kompleksitasnya adalah O(n3).

Referensi

[1] Azhari, Pocut Rizky. 2014. Implementasi Kombinasi Algoritma Kriptografi Modifikasi *playfair cipher* dan Teknik Steganografi *begin of file* pada Pengamanan Pesan Teks. Skripsi. Universitas Sumatera Utara.

[2] Herry dan Yessi. 2000. *Algoritma Run-Length Half Byte dan Huffman untuk Pemampatan File.* Bandung: Institut Teknologi Bndung.

[3] Kurniawan, Yusuf. 2004. *Kriptografikeamanan Internet danJaringanKomunikasi.* Bandung :Informatika Bandung.

[4] Mollin, R.A. 2007. *An Introduction to Cryptography.*2nd Edition. Taylor &Francis Group: LLC. United State of America.

[5] Sadikin, R. 2012. *KriptografiuntukKeamananJaringandanImplementasinyadalamBahasa Java*. Andi Offset: Yogyakarta.

[6] Tanjung, Namira Listya Utami. 2014. Analisis kombinasi algoritma *Knapsack* dan *RLE* pada file teks. Skripsi. Universitas Sumatera Utara.

[7] Whitten, J.L., Bentley, L.D. & Dittman, K.C. 2004. *Metode Desain & Analisis Sistem*. Edisi 6. ANDI: Yogyakarta.