

Implementasi Algoritma *Caesar Cipher* untuk Pengamanan Pesan Menggunakan *Java NetBeans*

Alfinatul Hasanah^{1*}, Muhlis Tahir², Farah Nisa' Salsabila³, Eva Nabila Aprilia⁴, Ani Sofiyah⁵, Putri Septiyawati⁶, Desta Chilyani⁷

^{1,2,3,4,5,6,7}Program Studi Pendidikan Informatika, Universitas Trunojoyo Madura, Indonesia

¹190631100050@student.trunojoyo.ac.id, ²muhlis.tahir@trunojoyo.ac.id,

³190631100040@student.trunojoyo.ac.id, ⁴190631100051@student.trunojoyo.ac.id,

⁵190631100053@student.trunojoyo.ac.id, ⁶190631100055@student.trunojoyo.ac.id,

⁷190631100059@student.trunojoyo.ac.id



Histori Artikel:

Diajukan: 11 Mei 2023

Disetujui: 14 Mei 2023

Dipublikasi: 15 Mei 2023

Kata Kunci:

Kriptografi; *Caesar Cipher*;

Keamanan Pesan; *Java*;

NetBeans

Digital Transformation

Technology (Digitech) is an

Creative Commons License This

work is licensed under a

Creative Commons Attribution-

NonCommercial 4.0

International (CC BY-NC 4.0).

Abstrak

Keamanan dan kerahasiaan suatu pesan merupakan hal yang penting. Pemilik pesan tentunya tidak mengharapkan data yang dimiliki dilihat oleh pihak yang tidak memiliki kepentingan. Sehingga untuk menjaga kerahasiaan pesan diperlukan pengamanan pesan yang disebut algoritma kriptografi. Salah satu teknik kriptografi yang sering dipakai adalah *Caesar Cipher*. Pada penelitian ini mengembangkan suatu sistem pengamanan pesan. Sistem ini menerapkan algoritma *Caesar Cipher* dan dalam prosesnya menggunakan program *Java NetBeans* untuk memperoleh *ciphertext*. Metode yang diterapkan dalam penelitian pengembangan ini adalah model *waterfall* mencakup tahap analisis kebutuhan, desain, implementasi dan pengujian. Kode program ditulis dalam bahasa *Java* menggunakan *software NetBeans IDE 8.2*. Penelitian ini menghasilkan proses enkripsi dan dekripsi menggunakan *Caesar Cipher* dengan *key* yang sama berupa abjad a-z yang akan menghasilkan *ciphertext* yang sulit dipahami sehingga menjadikan pesan menjadi lebih aman, baik pesan yang dikirim atau diterima. Sehingga diperoleh kesimpulan bahwa pengamanan pesan dengan algoritma *Caesar Cipher* telah terbukti dapat diimplementasikan pada *Java NetBeans*.

PENDAHULUAN

Perkembangan teknologi modern sangatlah pesat. Terdapat banyak teknologi yang sudah berkembang dan memberikan manfaat bagi penggunanya. Penggunaan teknologi dapat memengaruhi pekerjaan seseorang yang sebelumnya dikerjakan secara manual kini tergantikan dengan adanya teknologi komputer. Setiap pengguna komputer pasti memiliki suatu privasi ataupun hal rahasia yang harus dilindungi. Oleh karena itu, maka diperlukan suatu sistem keamanan berupa keamanan data yang berfungsi untuk melindungi setiap data yang dimiliki oleh setiap pengguna.

Keamanan data termasuk aspek yang penting dari sebuah sistem informasi. Namun, saat ini masalah keamanan data sering diabaikan. Padahal, sangatlah penting untuk melindungi dan mengamankan semua data atau informasi rahasia (Alhamdi & Siahaan, 2021:70). Cara untuk mengamankan dan meminimalisir kebocoran data bisa menggunakan kriptografi. Kriptografi dapat menyajikan kata atau karakter yang ditampilkan secara samar. Salah satu metode kriptografi yang bisa dipakai untuk proses enkripsi dan dekripsi adalah metode *Caesar Cipher* (Bancin et al., 2023:80).

Caesar Cipher merupakan salah satu metode algoritma yang dipakai untuk sistem kriptografi simetri. Metode ini sering dipakai sebelum adanya sistem kriptografi kunci publik (Hidayah et al., 2023:8564). *Caesar Cipher* termasuk jenis metode *cipher* substitusi yang membuat *cipher* melalui proses pergantian satu karakter yang diubah dengan karakter lain di beberapa jumlah digit sebelah kanan atau kirinya, sesuai arah pergeserannya (Putri et al., 2019:87). *Caesar Cipher* memiliki kelebihan untuk menyamarkan suatu pesan yang bisa tersandi, karena hanya pengirim dan penerima saja yang bisa mengetahui sandinya. Selain itu, *Caesar Cipher* juga memiliki kelemahan yaitu tidak dapat melakukan enkripsi ataupun dekripsi terhadap pesan yang tersusun dari beberapa kalimat dan juga rumus yang ditemukan (Hidayah et al., 2023:8564).

Pada penelitian terdahulu yang dilakukan oleh Bancin et al. (2023) yang berjudul "*Implementation of Cryptography with the Caesar Cipher Method to Secure Data Files in Java NetBeans*" didapatkan kesimpulan bahwa pengamanan data menggunakan metode *Caesar Cipher* terbukti bisa dilakukan pada *Java NetBeans* dan file yang bisa diproses enkripsi adalah file dokumen berbentuk teks berekstensi .txt. Selain itu, penelitian yang

dilakukan oleh Febrianingsih & Hafiz (2019) yang berjudul “Implementasi Kriptografi berbasis *Caesar Cipher* untuk Keamanan Data” diperoleh kesimpulan bahwa perhitungan matematis *Caesar Cipher* bisa dilakukan dengan sistem substitusi satu (1), dua (2), tiga (3), empat (4), lima (5) dan seterusnya serta aplikasi enkripsi menggunakan *Caesar Cipher* yang dapat mengacak file sehingga tidak bisa dibaca membuat data menjadi lebih aman.

Berdasarkan permasalahan dan rujukan dari penelitian terdahulu, maka peneliti mengambil penelitian yang berjudul “Implementasi Algoritma *Caesar Cipher* untuk Pengamanan Pesan Menggunakan *Java NetBeans*”. Penelitian tersebut diharapkan dapat membantu pengguna dalam menjaga dan mengamankan pesan yang berisi data atau informasi rahasia menggunakan *Java Netbeans*.

STUDI LITERATUR

Cryptography atau biasa dikenal dengan kriptografi merupakan bahasa Yunani yang terbentuk dari kata “*kryptos*” yang berarti tersembunyi serta “*graphien*” yang berarti tulisan. Jika diartikan secara harfiah, kriptografi berarti tulisan yang tersembunyi atau tulisan rahasia. Kriptografi adalah cara untuk melindungi pesan agar tetap aman dari pihak ketiga (Murtiyasa dalam Hayaty, 2020).

Caesar Cipher termasuk teknik enkripsi yang banyak digunakan karena teknik yang sederhana. Jenis dari *Caesar Cipher* yaitu substitusi (setiap huruf plainteksnya diubah ke huruf lain yang juga alfabet). Pada proses enkripsi, *Caesar Cipher* dapat direpresentasikan dengan module 26 yang sebelumnya setiap huruf ditransformasikan ke dalam angka, yaitu: A = 0, B = 1, C = 2, ..., Z = 25 (Putri et al., 2019:88).

Rumus *Caesar Cipher* secara matematis adalah sebagai berikut:

$$\text{Enkripsi: } C = E(P) = (P + K) \bmod 26 \quad (1)$$

$$\text{Dekripsi: } P = D(C) = (C - K) \bmod 26 \quad (2)$$

Keterangan:

C = Ciphertext

P = Plaintext

K = Key (Kunci rahasia)

E(P) = Enkripsi

D(C) = Dekripsi

Keamanan atau kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Sehingga informasi data hanya dapat diakses oleh pemilik data atau pengguna yang ditunjuk oleh pemilik data (Hariady et al., 2016:1). Pesan (*message*) yaitu sebuah informasi atau yang dapat dibaca dan dipahami maknanya. Pesan juga bisa disebut plainteks. Pesan dapat berupa data atau informasi yang dikirimkan melalui komunikasi. Pesan dapat disimpan berupa teks, gambar, suara ataupun video. Agar pihak lain tidak memahami pesan tersebut, pesan tersebut harus disimpan dalam bentuk yang berbeda sehingga sulit untuk dipahami. Format pesan yang tersandi disebut cipherteks. Cipherteks harus dapat diubah kembali menjadi plainteks semula agar pesan yang diterima dapat terbaca (Munir, 2006).

Java merupakan bahasa pemrograman yang bisa diakses dan dioperasikan melalui komputer maupun ponsel. Bahasa pemrograman ini dikembangkan oleh *James Gosling* ketika bergabung di *Sun Microsystems* yang saat ini dimiliki oleh *Oracle* yang rilis pada tahun 1995. Bahasa pemrograman *Java* mewarisi banyak sintaks yang ada pada bahasa pemrograman C dan bahasa pemrograman C++. Namun, dalam bahasa pemrograman *Java* sintaks model objek dibuat lebih sederhana. Bahasa pemrograman *Java* bersifat generik atau non-spesifik (*general purpose*) yang dirancang khusus didesain untuk menggunakan dependensi implementasi yang minimal. Saat ini bahasa pemrograman *Java* menjadi bahasa pemrograman yang paling populer digunakan dalam pengembangan berbagai program aplikasi (Maya, 2015).

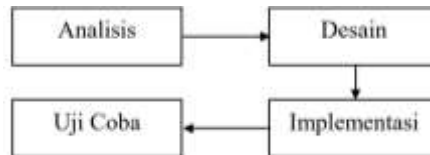
NetBeans adalah aplikasi *Integrated Development Environment* (IDE) berbasis bahasa *Java* yang berjalan *Swing*. *Swing* adalah teknologi pengembangan aplikasi *desktop* dan dapat berjalan di berbagai *platform* seperti *Mac OS*, *Windows*, *Solaris*, dan *Linux*. *NetBeans* adalah *software* untuk membangun aplikasi berbasis *Java*. Perangkat lunak ini dimanfaatkan sebagai alat untuk menjalankan program dan memudahkan pengguna menemukan kesalahan yang ada pada program (Kurniawan, 2011). Menurut Halimatussadiyah (2021:712) *NetBeans* merupakan salah satu perangkat lunak *Java* yang paling populer dan menyediakan *wizard* dan *template* untuk memfasilitasi pembuatan aplikasi dengan cepat dan mudah. Jadi, dapat disimpulkan bahwa *NetBeans* merupakan aplikasi *Integrated Development Environment* (IDE) berbasis *Java* yang berjalan di *Swing* serta menyediakan *wizard* dan *template* yang memudahkan pengembang membuat aplikasi dengan cepat.

METODE

Penelitian ini merupakan penelitian pengembangan (*Research & Development*) berupa sistem yang mengaplikasikan algoritma *Caesar Cipher* untuk pengamanan pesan dengan *Java NetBeans*. Selama melakukan penelitian, prosedur atau proses harus diikuti untuk menyelesaikan tahapan analisis sistem, perancangan,

penerapan, dan evaluasi (Cahyani et al., 2019:109). Proses tersebut dikenal dengan istilah *System Development Life Cycle* (SDLC) yang artinya siklus hidup pengembangan sistem. Sedangkan model pengembangan yang dipakai untuk menerapkan algoritma yang dipakai pada penelitian ini yaitu *Software Development Life Cycle* (SDLC) dengan model *waterfall*.

Model air terjun atau biasa disebut dengan model *waterfall* menyediakan tahapan yang berurutan (Yusup et al., 2020:437). Menurut Pressman (dalam Mulyanto, 2009:243) model *waterfall* memiliki empat tahapan, antara lain: (1) Analisis, (2) Desain, (3) Implementasi, dan (4) Uji Coba. Model *waterfall* secara sistematis dapat diilustrasikan sebagai berikut:



Gambar 1 Tahap-Tahap SDLC Model *Waterfall* (Roger Pressman)

1. Analisis

Pada tahapan awal ini berupa analisis kebutuhan. Peneliti melakukan pengumpulan data terkait alur kerja sistem pengamanan pesan, kebutuhan *software* yang diperlukan dan studi pustaka tentang algoritma *Caesar Cipher*.

2. Desain

Tahap desain dilakukan saat semua yang dibutuhkan dari tahap analisis terpenuhi. Pada tahap ini peneliti mendesain rancangan alur sistem sistem yang bertujuan untuk mendefinisikan konsep dan desain sistem pengamanan pesan.

3. Implementasi

Pada tahap ini dilakukan implementasi terhadap desain yang telah dirancang pada *software* yang sudah disiapkan.

4. Pengujian

Tahap terakhir yaitu menguji sistem pengamanan pesan yang bertujuan untuk mengetahui tingkat keberhasilan dan kesempurnaan sistem yang sudah dibuat.

HASIL

1. Analisis

Pada tahap ini peneliti melakukan pengumpulan data terkait alur kerja sistem pengamanan pesan, kebutuhan *software* yang diperlukan dan studi pustaka tentang algoritma *Caesar Cipher*. Alur kerja sistem pengamanan pesan peneliti dapat dari membaca sumber acuan dari jurnal, buku dan sumber referensi lainnya. Studi pustaka tentang algoritma *Caesar Cipher* dipakai untuk keperluan desain sistem pengamanan pesan. Selanjutnya terkait kebutuhan *software* yang digunakan untuk membangun sistem pengamanan pesan adalah *software NetBeans IDE 8.2* dengan menerapkan bahasa pemrograman *Java*.

2. Desain

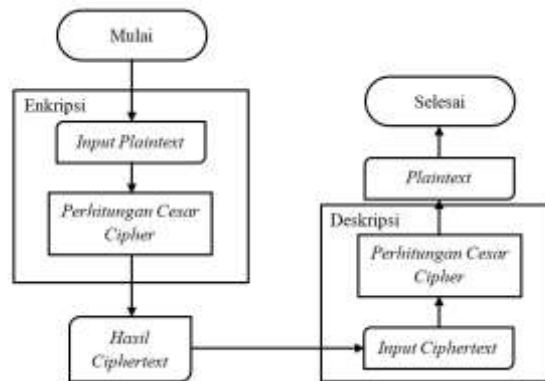
Pada tahap ini peneliti mendesain alur sistem sistem berupa *flowchart* sistem pengamanan pesan dan *flowchart* algoritma *Caesar Cipher* yang diterapkan dalam sistem. Alur sistem pengamanan pesan ditunjukkan pada Gambar 2 berikut.



Gambar 2 *Flowchart* Sistem Pengamanan Pesan

Gambar 2 menunjukkan *flowchart* alur sistem pengamanan pesan. Langkah pertama dimulai dengan menginputkan pesan asli (*plaintext*). Langkah kedua menginputkan kunci (*key*) berupa kata atau *text*. Langkah ketiga akan diproses menggunakan enkripsi (*ciphertext*) yaitu pesan dijadikan desimal + *key*, kemudian hasil penjumlahan dijadikan bilangan ASCII. Langkah keempat akan ditampilkan pesan hasil enkripsi data.

Penerapan algoritma *Caesar Cipher* dalam penelitian ini diproses oleh *Java NetBeans* untuk memperoleh *ciphertext*. Proses ini menggunakan diagram alur seperti ditunjukkan pada Gambar 3 berikut.



Gambar 3 *Flowchart* Algoritma *Caesar Cipher*

Gambar 3 merupakan diagram alur (*flowchart*) dari algoritma *Caesar Cipher*. Algoritma *Caesar Cipher* adalah salah satu metode dalam proses enkripsi dan dekripsi data (Bancin et al., 2023). Diawali dengan memasukkan data yang akan diproses yang masih tersedia dalam bentuk *plaintext*. *Plaintext* adalah sebuah informasi atau pesan yang akan dikirimkan dalam bentuk aslinya (Bancin et al., 2023). Pesan *plaintext* selanjutnya akan melalui proses enkripsi. Enkripsi adalah sebuah pesan atau informasi yang dipakai untuk mengubah pesan yang asli menjadi *ciphertext* (Bancin et al., 2023). *Ciphertext* adalah sebuah pesan atau informasi yang telah dienkripsi dan tidak dapat dibaca karena merupakan hasil enkripsi data (Bancin et al., 2023). Agar pesan dapat terbuka dan dapat terbaca, maka diperlukan proses dekripsi. Dekripsi adalah proses mengubah data terenkripsi kembali menjadi data asli (*plaintext*). *Key* atau kunci adalah sebuah angka atau bilangan yang bersifat rahasia dan dimanfaatkan dalam proses enkripsi dan dekripsi data (Bancin et al., 2023).

3. Implementasi

Pada tahap ini, desain yang telah dirancang seperti pada Gambar 2 diimplementasikan. Peneliti menerjemahkan desain sistem ke dalam kode-kode program untuk dimasukkan ke dalam aplikasi pembuat sistem yaitu *software NetBeans IDE 8.2* dengan memakai bahasa pemrograman *Java*.

a. Implementasi *code program* proses enkripsi dan dekripsi algoritma *Caesar Cipher* dengan bahasa pemrograman *Java* pada *software NetBeans IDE 8.2*.

```

196 private void btnEnkripsiActionPerformed(java.awt.event.ActionEvent evt) {
197     String key = txtKey.getText();
198     String plaintext = txtPlaintext.getText();
199     String out = "";
200     for (int i = 0; i < plaintext.length(); i++) {
201         int index = plaintext.charAt(i);
202         char c = (char) (index + key.charAt(0));
203         out = out + String.valueOf(c);
204     }
205     txtCiphertext.setText(out);
206     txtPlaintext.setText("");
207     txtKey.setText("");
208 }
  
```

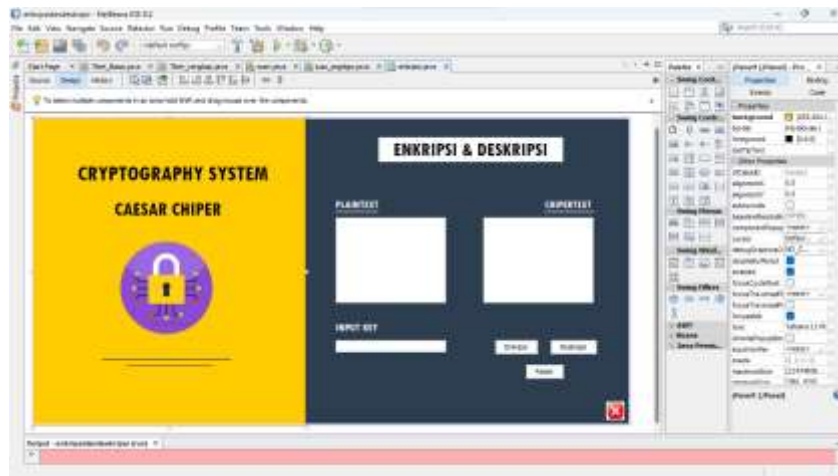
Gambar 4 *Code Program* Proses Enkripsi Algoritma *Caesar Cipher*

```

210 private void btnDekripsiActionPerformed(java.awt.event.ActionEvent evt) {
211     String ciphertext = txtCiphertext.getText();
212     char k = (char) (ciphertext.charAt(0));
213     String key = "", out = "";
214     int j = Integer.parseInt(String.valueOf(k));
215     for (int i = 1; i < ciphertext.length(); i++) {
216         int index = ciphertext.charAt(i);
217         char c = (char) (index - j);
218         key = key + String.valueOf(c);
219     }
220     if (key.equals(ciphertext)) {
221         for (int i = j; i < ciphertext.length(); i++) {
222             int index = ciphertext.charAt(i);
223             char c = (char) (index - j);
224             out = out + String.valueOf(c);
225         }
226         txtPlaintext.setText(out);
227     } else {
228         JOptionPane.showMessageDialog(null, "Key tidak sesuai dengan data input!");
229     }
230 }
  
```

Gambar 5 *Code Program* Proses Dekripsi Algoritma *Caesar Cipher*

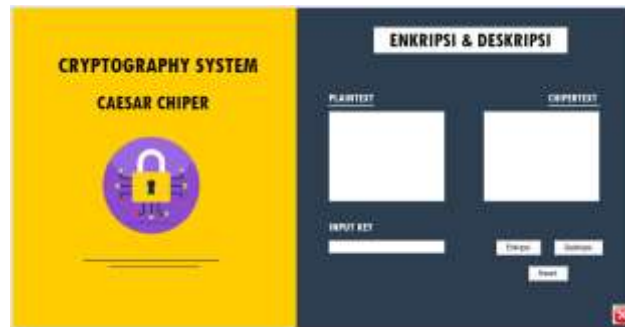
- b. Hasil *running* sistem pengamanan pesan dengan algoritma *Caesar Cipher* pada *software NetBeans IDE 8.2*.



Gambar 6 Hasil *Running* Sistem Pengamanan Pesan pada *Software NetBeans IDE 8.2*

4. Pengujian

Pada tahap ini peneliti menguji sistem pengamanan pesan yang sudah dibuat. Pengujian terbagi menjadi dua bagian yaitu untuk menguji enkripsi dan dekripsi pada algoritma kriptografi *Caesar Cipher*. Hasil sistem pengamanan pesan yang peneliti buat menggunakan bahasa pemrograman *Java NetBeans* yang memiliki beberapa *button* untuk memudahkan *user*. Tampilan sistem pengamanan pesan ditunjukkan pada Gambar 7 hingga Gambar 11.



Gambar 7 Tampilan Awal Sistem Pengamanan Pesan

Gambar 4 menunjukkan halaman awal sistem pengamanan pesan yang terdapat menu enkripsi, dekripsi, *form input plaintext*, *form output ciphertext*, *form input key*, *button reset*, dan keluar.

a. Pengujian Proses Enkripsi



Gambar 8 Tampilan *Input Plaintext* dan *Key* saat Proses Enkripsi

Gambar 9 Tampilan Hasil *Output Ciphertext* saat Proses Enkripsi

Pada halaman ini, *user* dapat menginputkan *plaintext* dan *key* sesuai dengan keinginan. *Key* yang diinputkan pada program ini yaitu berupa abjad. Pada contoh diatas *plaintext* yang diinputkan yaitu: “KEAMANANJARINGANKOMPUTER” dengan *key* “CAESAR”. Sehingga *ciphertext* yang diperoleh dari proses enkripsi yaitu: “7DBFTBSLFBNOBOKBSJOHBOLPNQVUFS”.

b. Pengujian Proses Dekripsi

Gambar 10 Tampilan *Input Key* saat Proses DekripsiGambar 11 Tampilan *Output Plaintext* saat Proses Dekripsi

Pada halaman ini, *user* dapat melakukan proses dekripsi dari *ciphertext* yang diperoleh sehingga *user* dapat membaca pesan yang telah diterima. Pada proses ini *user* harus menginputkan *key* yang sesuai dengan yang digunakan pada proses enkripsi. Saat *user* menginputkan *key* yang salah maka sistem akan menampilkan pesan “kunci tidak cocok, silahkan coba lagi!”. Pada implementasi sistem ini *user* melakukan proses dekripsi dengan menginputkan *ciphertext* yang diperoleh dari proses enkripsi sebelumnya yaitu: “7DBFTBSLFBNOBOKBSJOHBOLPNQVUFS” dengan *key* yang sesuai yaitu “CAESAR”. Sehingga diperoleh *plaintext* sebagai hasil dari proses dekripsi: “KEAMANANJARINGANKOMPUTER”.

PEMBAHASAN

Berdasarkan hasil pengujian enkripsi dan dekripsi pada sistem pengamanan pesan yang telah dilakukan, pesan yang digunakan pada *plaintext* berupa karakter huruf a-z saja. Dimana proses enkripsi pada pesan dilakukan menggunakan metode kriptografi *Caesar Cipher* sehingga menghasilkan *ciphertext*. *Ciphertext* tersebut kemudian didekripsi menjadi *plaintext* agar dapat dibaca kembali. Dalam proses enkripsi dan dekripsi tentunya diperlukan

key atau kunci rahasia yang dimanfaatkan untuk membuka dan melihat isi pesan yang asli. Tabel substitusi perubahan metode *Caesar Cipher* sebagai berikut:

Tabel 1
Substitusi *Caesar Cipher*

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

Perhitungan Matematis Proses Enkripsi

Langkah enkripsi yang diterapkan pada sistem ini yaitu memasukkan *key* dan *plaintext* sesuai dengan keinginan. Kemudian geser 1 huruf kebelakang pada *key* dan *plaintext* yang telah diinputkan. Selanjutnya jumlahkan banyaknya *key* yang telah diinputkan sebagai huruf pertama pada *ciphertext* sebagai hasil enkripsi.

Contoh:

Plaintext : KEAMANANJARINGANKOMPUTER

Key : CAESAR

Penyelesaian :

1. Jumlah *key* + 1 = 6+1 = 7.
2. Hasil substitusi *key*:

Tabel 2
Hasil Substitusi *Key* (Enkripsi)

Huruf	Substitusi 1
C	D
A	B
E	F
S	T
A	B
R	S

3. Hasil substitusi *plaintext*:

Tabel 3
Hasil Substitusi *Plaintext*

Huruf	Substitusi 1
K	L
E	F
A	B
M	N
A	B
N	O
A	B
N	O
J	K
A	B
R	S
I	J
N	O
G	H
A	B
N	O
K	L
O	P
M	N
P	Q
U	V
T	U
E	F
R	S

4. Jadi, *ciphertext* yang didapat dari proses enkripsi yaitu: “7DBFTBSLFBNBOBOKBSJOHBOLPNQVUFS”. Untuk aturan penulisan dimulai dari jumlah *key* lalu diikuti dengan hasil substitusi dari *key* dan *plaintext*.

Perhitungan Matematis Proses Dekripsi

Langkah dekripsi yang diterapkan pada sistem ini yaitu menginputkan *ciphertext* serta kunci yang sama digunakan saat proses enkripsi sehingga dapat didekripsi kembali. Pada langkah dekripsi tidak jauh berbeda dengan langkah enkripsi. Langkahnya yaitu dengan membuang jumlah *key* dan hasil pergeseran dari *key* serta menggeser satu huruf kedepan pada *ciphertext*. Contoh:

Ciphertext : 7DBFTBSLFBNBOBOKBSJOHBOLPNQVUFS

Key : CAESAR

Penyelesaian :

1. Membuang jumlah *key* dan hasil pergeseran dari *key* saat proses enkripsi, sehingga didapat: LFBNBOBOKBSJOHBOLPNQVUFS
2. Hasil substitusi *key*:

Tabel 4

Hasil Substitusi <i>Key</i> (Dekripsi)	
Huruf	Substitusi 1
L	K
F	E
B	A
N	M
B	A
O	N
B	A
O	N
K	J
B	A
S	R
J	I
O	N
H	G
B	A
O	N
L	K
P	O
N	M
Q	P
V	U
U	T
F	E
S	R

3. Jadi, *plaintext* yang didapat dari proses dekripsi yaitu: “KEAMANANJARINGANKOMPUTER”

KESIMPULAN

Berdasarkan hasil sistem pengamanan pesan menggunakan metode *Caesar Cipher* untuk mengamankan data file pada *Java NetBeans*, maka dapat disimpulkan bahwa pengamanan pesan dengan algoritma *Caesar Cipher* telah terbukti dapat diimplementasikan pada *Java NetBeans*. Hal ini bertujuan untuk menjaga kerahasiaan pesan dibutuhkan keamanan saat memasukkan teks. Dimana proses enkripsi dan dekripsi menggunakan *Caesar Cipher* dengan *key* yang sama berupa abjad a-z yang menghasilkan *ciphertext* yang sulit dipahami sehingga membuat pesan menjadi lebih aman, baik pesan yang diterima maupun dikirim. Dalam penelitian ini, proses enkripsi-dekripsi menggunakan karakter huruf. Oleh karena itu, penelitian selanjutnya diharapkan dapat mengaktifkan enkripsi-dekripsi menggunakan semua karakter dan simbol, seperti angka, titik, koma, dan lain-lain.

REFERENSI

- Alhamdi, A. T. F., & Siahaan, R. F. (2021). Penerapan Kriptografi dalam Pengamanan Pesan *Text* Berbasis Android dengan Menggunakan Metode *Rijndael*. *Jurnal Mahajana Informasi*, 6(2), 69-74.
- Angriani, H., & Yeni S. (2019). Implementasi Algoritma *Caesar Cipher* pada Keamanan Data Sistem *E-Voting*

- Pemilihan Ketua Organisasi Kemahasiswaan. *Jurnal Teknologi Informasi dan Komunikasi*, 9(2), 123-126.
- Bancin, H., Panjaitan, M. A., Putri, S., & Nasution, A. B. (2023). *Implementation of Cryptography with the Caesar Cipher Method to Secure Data Files in Java NetBeans*. *JTECS: Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*, 3(1), 79-86.
- Cahyani, L., Arif, M., & Ningsih, F. (2019). Sistem Pendukung Keputusan Pemilihan Mahasiswa Berprestasi Menggunakan Metode *Moora* (Studi Kasus Fakultas Ilmu Pendidikan Universitas Trunojoyo Madura). *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*, 5(2), 108-114.
- Febrianingsih, R., & Hafiz, A. (2019). Implementasi Kriptografi Berbasis *Caesar Cipher* untuk Keamanan Data. *Jurnal Informasi dan Komputer*, 7(2), 81-86.
- Gusmana, R., Haryansyah, H., & Fitria, F. (2022). Implementasi Algoritma *Affine Cipher* dan *Caesar Cipher* dalam Mengamankan Data Teks. *Sebatik*, 26(2), 517-524.
- Halimatusadiah, H. (2021). Rancang Bangun Sistem Informasi Pelayanan *Dry & Wet Laundry* Berbasis *Java NetBeans*. *Jurnal Inovasi Penelitian*, 2(3), 711-716.
- Hariady, M. M., Suyatno, A., & Astuti, I. F. (2016). Keamanan dan Penyisipan Pesan Rahasia pada Gambar Dengan Enkripsi *Blowfish* dan *Steganografi End Of File*. *Jurnal Informatika Mulawarman*, 11(2), 1-10.
- Hayaty, Nurul. (2020). *Buku Ajar: Sistem Keamanan Sebuah Pegangan untuk Mahasiswa Teknik Informatika UMRAH*. Tanjungpinang: Universitas Teknik Maritim Raja Ali Haji.
- Hidayah, V. M., Mulyana, D. I., & Bachtiar, Y. (2023). Algoritma *Caesar Cipher* atau *Vigenere Cipher* pada Pengenkripsian Pesan Teks. *Journal on Education*, 5(3), 8563-8573.
- Kurniawan, H., Mardiani, E., & Rahmansyah, N. (2011). *Aplikasi Penjualan dengan Program Java Netbeans, Xampp, dan iReport*. Jakarta: Elex Media Komputindo.
- Maya. (2015). *Membangun Sistem Informasi dengan Java NetBeans dan MySQL*. Yogyakarta: CV. Andi Offset.
- R. Munir. (2006). *Kriptografi*. Bandung: Informatika.
- Nasution, A. B. (2019). Implementasi Pengamanan Data dengan Menggunakan Algoritma *Caesar Cipher* dan *Transposisi Cipher*. (*JurTI*) *Jurnal Teknologi Informasi*, 3(1), 1-6.
- Natsir, M. (2017). Pengembangan *Prototype* Sistem Kriptografi untuk Enkripsi dan Dekripsi Data *Office* Menggunakan Metode *Blowfish* dengan Bahasa Pemrograman *Java*. *Jurnal Format*, 6(1), 87-105.
- Pressman, Roger S. (2005). *Software Engineering A Practitioner's Approach*. Singapore: McGraw-Hill Education.
- Putri, Y. D., Rosihan, R., & Lutfi, S. (2019). Penerapan Kriptografi *Caesar Cipher* pada Fitur *Chatting* Sistem Informasi *Freelance*. *JIKO (Jurnal Informatika dan Komputer)*, 2(2), 87-94.
- Yusup, I. M., Carudin, C., & Purnamasari, I. (2020). Implementasi Algoritma *Caesar Cipher* dan *Steganografi Least Significant Bit* untuk *File* Dokumen. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(3), 434-441.