JINTEKS (Jurnal Informatika Teknologi dan Sains) ISSN 2686-3359 (Online)
Vol. 4 No. 3, Agustus 2022, hlm. 179 – 184

# MODIFIKASI ALGORITMA KRIPTOGRAFI CAESAR CIPHER MENJADI ALGORITMA KRIPTOGRAFI ASIMETRIS DENGAN METODE AGILE

**Nabila Oper[1], Sabri Balafif[2], To'o Fathonah Al-Khaliq.Z[3]**
[1]Teknik Informatika, Institut Teknologi dan Bisanis Stikom Ambon
[2,3]Informatika, Universitas Teknologi Sumbawa
*email*: nabila.093@gmail.com

**Abstrak:** Dalam mengirim pesan perlu diperhatikan mengenai keamanan isi pesan tersebut, maka dari itu, di butuhkan sebuah metode untuk mengamankan informasi dengan aman. Algoritma kriptografi merupakan sebuah metode yang bukan hanya melindungi tapi juga mengubah informasi yang ada sehingga meskipun informasi tersebut diketahui pihak yang tidak bertanggung jawab informasi ini belum tentu bisa dimengeri oleh pihak yang tidak bertanggung jawab tersebut. Pada penelitian ini menggunakan metode Agile yang bertujuan mempermudah pemahaman dalam penerapan algoritma kriptografi caesar cipher asimetris. Dalam Melakukan pengujian algoritma kriptografi asimetris, penelitian ini menggunakan bahasa pemrograman python untuk melakukan implementasian. Hasil akhir dari penelitian ini adalah kriptografi asimetris bisa melakukan enkripsi dan dekripsi sama dengan kriptografi caesar cipher simetris meskipun algoritmanya berubah yang awalnya simetris menjadi asimetris.

**Kata Kunci :** Algoritma, Kriptografi, Caesar Cipher, Pesan, Python.

***Abstract:*** *ASending messages needs to be considered regarding the security of the content of the message, therefore, a method is needed to secure information safely. Cryptographic algorithms are a method that not only protects but also changes existing information so that even if the information is known to irresponsible parties this information cannot necessarily be investigated by the irresponsible party. This study uses the Agile method which aims to facilitate understanding in the application of asymmetric caesar cipher cryptographic algorithms. In testing asymmetric cryptographic algorithms, this study used the python programming language to carry out implementations. The end result of this study is that asymmetric cryptography can perform the same encryption and decryption as symmetric caesar cryptography even though the algorithm changes initially symmetrical to asymmetrical.*

***Keywords*** **:** *Algorithm, Cryptography, Caesar Cipher, Message, Python*.

## PENDAHULUAN

Pada zaman ini dimana dalam mengirim pesan sudah bukan lagi hal tabu dalam menggunakan media online sebagai media perantara untuk saling bertukar pesan baik itu secara berkelompok maupun secara individual, baik itu pesan umum atau pesan pribadi.

Dalam hal ini perlu dipertimbangkan keamanan dari isi pesan perlu dirahasiakan agar pesan yang dimaksudkan di terima oleh penerima pesan dan bisa tersampaikan dengan aman tanpa perlu di ketahui oleh orang lain yang tidak berkepentingan (Yusfrizal, 2019). Agar supaya isi pesan yang seharusnya hanya boleh dibaca oleh penerima pesan dan pengirim pesan tetap bisa terahasiakan. Hal ini perlu diperhatikan agar pihak-pihak yang berketerkaitan dengan isi pesan tidak dirugikan oleh karena isi pesan tersebut bisa juga diketahui oleh pihak yang bukan penerima pesan dan pengirim pesan.

Kriptografi adalah ilmu yg digunakan buat mengganti suatu data utau pesan menjadi tidak dapat dimengerti, kemudian dengan sedemikian rupa. dapat diubah ulang sebagai data atau informasi yg dapat dimengerti (Afand & Nurhayati, 2020). pada ilmu kriptografi ada dua konsep utama. yaitu enkripsi dan dekripsi. Enkripsi artinya proses membarui data atau informasi atau plaintext sebagai ciphertext sehingga tidak bisa dimengerti sang pihak ketiga. Sedangkan dekripsi ialah proses mengubah data atau info ciphertext yg sudah dienkrips kedalam bentuk semula atau bisa disebut plaintext (Septian Widiyanto, Govindo Adnan, Moh. Fatkuroji, Dwi Wahyu Handoyo, 2021).

## TINJAUAN PUSTAKA

### 1. Pesan

Pengertian pesan adalah sesuatu yang disampaikan pengirim kepada penerima. Pesan dapat disampaikan dengan cara tatap muka atau melalui media komunikasi. Isinya bisa berupa ilmu pengetahuan, hiburan, informasi, nasihat atau propaganda (Bahari et al., 2022)

### 2. Kriptografi

Menurut bahasa Kriptografi berasal istilah berasal *crypto* yang memiliki arti misteri dan *graphy* yang memiliki arti tulisan. Jadi kriptografi bisa maknai goresan pena yang bersifat tersembunyi. Menurut istilah bisa diartikan sebagai pelajaran rumus dalam matematika yang fokus pada keamanan. Kriptografi sendiri ada 2 jenis yaitu simetri dan asimetri. (Septian Widiyanto, Govindo Adnan, Moh. Fatkuroji, Dwi Wahyu Handoyo, 2021)

 ISSN 2686-3359 (Online)

### 3. Enkripsi

Pesan yang masih bisa dibaca oleh pihak manapun(*Plaintext)* akan diubah menggunakan proses Enkripsi sehingga nantinya hasil dari Enkripsi tersebut akan menjadi pesan yang hanya bisa dibaca oleh pihak tertentu saja*(Ciphertext).* Merupakan proses mengubah *plaintext* menjadi ciphertext.(Rachman, 2018)

### 4. Dekripsi

Berbanding terbalik dengan proses Enkripsi, proses Dekripsi adalah proses dimana pesan yang sebelumnya tidak bisa diibaca oleh pihak manapun(*Ciphertext)* akan diubah kembali menjadi pesan yang bisa dibaca oleh pihak manapun (*Plaintext).* Merupakan proses mengubah *ciphertext* menjadi plaintext.(Rachman, 2018)

### 5. Plaintexs

*Plaintext* atau bisa diartikan sebagai teks biasa merupakan sebutan untuk pesan yang bisa dimengerti oleh siapapun yang membaca isi pesan tersebut. menurut ahli *plaintext* Merupakan pesan asli sebelum diubah menjadi pesan rahasia. (Rachman, 2018)

### 6. Ciphertexs

*Ciphertext* atau bisa diartikan sebagai teks sandi merupakan sebutan untuk pesan yang hanya bisa dimengerti oleh segelintir orang yang mengetahui Key dari pesan tersebut. menurut ahli *Ciphertext* Merupakan pesan sandi atau pesan rahasia yang sulit diterjemahkan (Rachman, 2018)

### 7. Key

*Key* atau bisa diartikan sebagai kunci merupakan sebutan untuk suatu variable yang wajib ada untuk mengubah *Ciphertext* menjadi *plaintext*. Menurut ahli *Key* Merupakan kunci rahasia yang digunakan untuk mengubah pesan asli menjadi pesan rahasia.(Rachman, 2018)

### 8. Algoritma

Merupakan langkah eksklusif yang terstruktur. Sifat pertama algoritma harus benar. sifat kedua yang wajib diperhatikan adalah kita harus mengetahui celah dibalik algoritma. Dan ketiga harus efisiensi dalam prosedur pemecahannya. (Situmorang, 2018)

### 9. Sistem

Sistem ialah kumpulan / class asal sub sistem / bagian / komponen apapun baik phisik ataupun non phisik yang saling berhubungan satu sama lain serta bekerja sama secara harmonis untuk mencapai satu tujuan eksklusif. (Nilfaidah, 2021)

### 10.Caesar Cipher

Dalam penggunaannya algoritma kriptografi caesar cipher mudah untuk pahami. Dalam pengerjaannya algoritma kriptografi ini hanya melakukan pergeseran urutan karakter sebanyak nilai yang ada. (Angriani , 2019)

### 11.Agile methodology

*Agile software development methods* atau metodologi *Agile* merupakan metodologi pengembangan perangkat lunak aplikasi yang berfokus pada pengembangan yang bersifat iteratif, yang mana persyaratan dan perberkembang segala sesuatu harus berdasarkan kolaborasi antar tim yang terorganisir (Mahendra & Eby Yanto, 2018)

## METODE PENELITIAN

Dalam penelitian ini peneliti menggunakan metode penelitian *Agile* karena metode *Agile* selain mudah beradaptasi dalam perkembangan zaman juga mudah dirombak tanpa harus merubah seluruh sistem yang ada.

Berikut adalah tahap-tahap pemodelan Agile berdasarkan referensi :

1) Analisa Kebutuhan

Di tahap ini peneliti melakukan analisa terhadap pengguna guna mengetahui perangkat lunak seperti apa yang diharapkan.

2) Desain

Tahap kedua yaitu tahap desain, yaitu melakukan perancangan sistem, mulai dari desain arsitektur sistem, sampai desain user interface. Dalam penelitian ini peneliti menggunakan alat bantu yaitu *Unified Modeling Language (UML)*.

Alasan mengapa peneliti menggunakan *UML* tidak lain karena *UML* alat bantu desain yang memiliki banyak fitur dan kegunaan.

*3) Code Generation*

Tahap ketiga yaitu tahap ini penulisan kode atau *source cod*e menggunakan bahasa Python. Dan *compiler*nya adalah IDLE Python itu sendiri.

*4) Testing*

Tahap keempat yaitu tahap uji coba yang mana tahap ini dilakukan untuk mengurangi dan memperbaiki hasil *output* setelah proses penulisan kode.

*5) Support*

Tahap terakhir adalah *support* merupakan tahap terakhir sebelum hasil akhir didapatkan, tahap ini hanyalah tahap sederhana untuk mengakhiri penelitian dan memberikan hasil berupa testimoni program yang sudah ditahap akhir.

 ISSN 2686-3359 (Online)

## HASIL DAN PEMBAHASAN

1. Analisa Kebutuhan
   a. Analisa Kebutuhan Algoritma Kriptografi

   Pada tahap pertama peneliti melakukan analisa dasar dalam menjelaskan mengenai algoritma kriptografi Caesar Cipher. Selanjutnya peneliti melakukan uji coba kembali untuk mengubah algoritma kriptografi Caesar Cipher menjadi kriptografi modern yang semula kriptografi Caesar Cipher ini adalah kriptografi klasik.

   Perombakan algoritma kriptografi Caesar Cipher yang peneliti lakukan hanya pada urutan alfabet saja. Yang mana urutan alfabet yang seharusnya sesuai dengan kaidah yaitu berurutan dari A sampai Z, akan peneliti acak seperti berikut ini.

   Plaintext = zugdqfchtjkvmasperoiblwxyn
   ciphertext = dqfchtjkvmasperoiblwxynzug

   Adapun untuk pergesaran atau *key* masih menggunakan *key* 3 langkah pergeseran. Selanjutnya penulis akan melakukan dekripsi sebuah teks untuk membuktikan bahwa akan terjadi perubahan pada hasil dekripsi, sebagai berikut :

   Plaintext = universitas Teknologi Sumbawa
   Ciphertext = qgwsibrwverViaglylfwRqpxene

   b. Analisa Kebutuhan Sistem

   Adapun analisa kebutuhan fungsional sebagai berikut:

   1) Menerima *input plaintext*

      Sistem menerima *input plaintext* dari pengguna secara manual. Sistem hanya dapat membaca data berupa teks

   2) Menerima *input ciphertext*

      Sistem menerima *input ciphertext* dari pengguna secara manual. Sistem hanya dapat membaca data berupa teks

   3) Menerima input *key*

      Sistem menerima input *key* atau kunci dari pengguna secara manual. Sistem hanya dapat membaca data berupa numerik

   4) Mengenkripsikan *plaintext*

      Sistem dapat mengenkripsi teks sesuai dengan algoritma kriptografi caesar chiper yang sudah di modifikasi kemudian menampilkan hasil berupa *ciphertext*

   5) Mendekripsikan *ciphertext*

      Sistem dapat mengembalikan pesan yang sudah menjadi *ciphertext* menjadi pesan asli atau *plaintext*

   Adapun analisa kebutuhan non fungsional sebagai berikut:

   1) Performa

      Sistem yang dibangun dapat menampilkan hasil dari fungsi kriptografi yang dilakukan yaitu enkripsi dan dekripsi dengan hasil performa yang baik.

   2) Kompatibel

      Sistem yang dibangun dapat digunakan pada setiap komputer bersistem operasi windows selama bahasa pemrograman python bisa berjalan lancar .

   3) User friendly

      Sistem yang dibangun mudah dimengerti dan digunakan oleh user (pengguna).

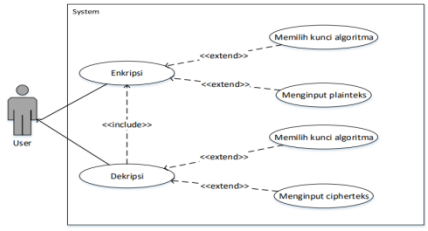
   c. Analisa Kebutuhan Perangkat

   Pada tahap ini peneliti hanya melakukan istalasi perangkat lunak berupa software penerjemah bahasa pemrograman yaitu Python beserta PIP dari packagesnya

2. Desain

   Desain merupakan tahapan yang dilakukan setelah tahap analisa keputusan. Adapun desain yang dilakukan adalah pembuatan diagram *Unified Modeling Language* (UML) dan *Flowchart.*

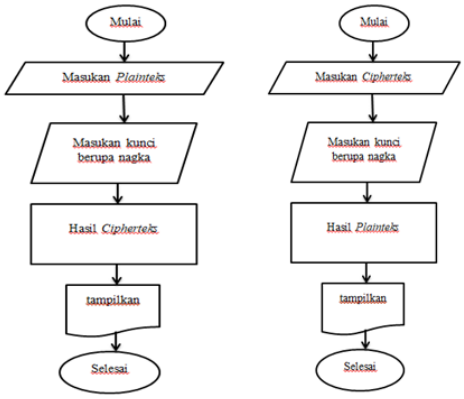
   Diagram UML (*use case*) itu sendiri peneliti gunakan untuk menggambarkan bagaimana perancangan sistem yang berjalan pada penelitian ini. dan *Flowchart* adalah suatu proses yang menggambarkan bagaimana algoritma didalam aplikasi berfungsi untuk menjalankan perintah yang dimasukan ke dalam aplikasi.




**Gambar 1. *Use case diagram***

Gambar 1 menjelaskan bahwa user memiliki peran untuk melakukan enkripsi dan dekripsi pesan teks. untuk melakukan enkripsi pesan teks, pengguna terlebih dahulu memilih *key* algoritma yang akan digunakan. *key* digunakan untuk mengenkripsi *plaintext* menjadi *ciphertext*. Lalu pengguna menginput *plaintext*. Begitu pula dengan deskripsi jika pengguna memiliki *ciphertext* dan *key* yang sama maka hasil deskripsi dari *ciphertext* akan menghasilkan *plaintext* yang diinginkan.

 ISSN 2686-3359 (Online)




**Gambar 2. *Flowchart* enkripsi dan dekripsi**

Pada gambar 2 menjelaskan algoritma yang berjalan didalam aplikasi yaitu dimulai dari memasukan plainteks yang berupa huruf alfabet kemudian ke tahap memasukan kunci berupa angka lalu tahap selanjutnya menghasilkan *ciphertext* yang sesuai dengan *plaintext* yang dimasukan dan tahap terakhir yaitu menampilkan output yang berupa *ciphertext*. adapun untuk proses dekripsi hampir sama dengan proses *flowchart* enkripsi hanya saja pada tahap dekripsi, *ciphertext* akan di ubah kembali menjadi *plaintext* dengan algoritma yang sama.

3. Code Generation

Pada tahap ini peneliti melakukan proses penulisan kode program yang dilakukan pada compiler python yaitu IDLE Python. Pada penelitian ini peneliti membuat 2 program python yang saling berkaitan agar tahap testing bisa berjalan. Berikut kode program yang peneliti tulis ke dalam pemrograman python:

**Tabel 1. Caesar.py**

| | |
|---|---|
| class Caesar:<br>def __init__(self,plainText,key):<br>self.plainText = str.lower(plainText)<br>self.key = key | Mengubah semua huruf kapital yang diinput akan menghasilkan output huruf kecil |
| def encrypt(self):<br>abjad = "zugdqfchtjkvmasperoiblwxyn"<br>chiperText = ""<br>for words in self.plainText:<br><br>wordsIndex = (abjad.find(words)+self.key)%len(abjad)<br><br>chiperText= chiperText+abjad[wordsIndex]<br>return chiperText | Melakukan enkripsi |
| def decrypt(self):<br>abjad = "zugdqfchtjkvmasperoiblwxyn"<br>chiperText = ""<br>for words in self.plainText:<br><br>wordsIndex = (abjad.find(words)-self.key)%len(abjad)<br><br>chiperText= chiperText+abjad[wordsIndex]<br>return chiperText | Melakukan |

**Tabel 2. Main.py**

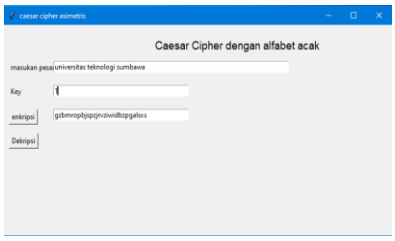
| | |
|---|---|
| from Caesar import Caesar<br>from tkinter import *<br>app = Tk()<br>app.title("caesar cipher asimetris") | Mengimport fungsi Caesar dan fungsi library Tkinter |
| TitleLabel = Label(app, text="Caesar Cipher dengan alfabet acak", font=('bold', 14))<br>TitleLabel.place(x=270, y=20)<br><br>plainLabel = Label(app, text="masukan pesan").place(x=10, y=60)<br>plain = Entry(app, width=70, borderwidth=1)<br>plain.place(x=90, y=60)<br>keyCodeLabel = Label(app, text="Key").place(x=10, y=100)<br>keyCode = Entry(app, width=40, borderwidth=1)<br>keyCode.place(x=90, y=100)<br>hasil = Entry(app, width=40, borderwidth=1)<br>hasil.place(x=90, y=140) | Menambah fitur untuk menambahkan label dan menambahkan textfield |
| def ProcessEncrypt(text, key):<br>chiper = Caesar(text, key)<br>return chiper.encrypt()<br>def ProcessDecrypt(text, key):<br>chiper = Caesar(text, key)<br>return chiper.decrypt() | Fungsi enkripsi dan fungsi dekripsi |
| def EncryptButton():<br>text = plain.get()<br>key = int(keyCode.get())<br>chiper = ProcessEncrypt(text, key)<br>hasil.delete(0, END)<br>hasil.insert(0, chiper)<br>def DecryptButton():<br>text = plain.get()<br>key = int(keyCode.get())<br>chiper = ProcessDecrypt(text, key)<br>hasil.delete(0, END)<br>hasil.insert(0, chiper) | Menambahkan fungsi enkripsi button dan dekripsi button |
| processButtonEncrypt = | Menambahkan |

 ISSN 2686-3359 (Online)

| | |
|---|---|
| Button(app, text="enkripsi", command=EncryptButton)<br>processButtonEncrypt.place(x=10, y=140)<br>processButtonDecrypt = Button(app, text="Dekripsi", command=DecryptButton)<br>processButtonDecrypt.place(x=10, y=180) | fungsi button dan parameter didalam button untuk menghubungkan fungsi button nantinya |

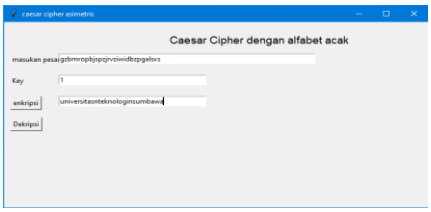
4. Testing

Setelah penulisan kode menggunakan bahasa pemrograman Python selanjutnya akan dilakukan tahap *testing* untuk melakukan uji coba pada *code* yang sudah ditulis. Pada tahap testing ini dibagi menjadi dua bagian yaitu enkripsi dan dekripsi




Gambar 3. Melakukan enkripsi

Pada gambar 3 diperlihatkan bahwa proses enkripsi teks dengan *key* satu dapat berjalan dengan lancar dan semua *plaintext* dapat di ubah menjadi *ciphertext* tanpa ada kendala.




**Gambar 4. Melakukan dekripsi**

Pada gambar 4 pula diperlihatkan bahwa proses dekripsi teks dengan *key* satu dapat berjalan dengan lancar pula dan semua *ciphertext* dapat di ubah kembali menjadi *plaintext* kembali tanpa ada kendala.

5. Support

1. Perbandingan Caesar Cipher Simteris dan Caesar Cipher Asimetris

Setelah di teliti dengan seksama Algoritma Caesar Cipher Asimetris dapat mempengaruhi hasil dari enkripsi maupun dekripsi algoritma Caesar Cipher Simetris. Ini dikarenakan algoritma Caesar Cipher Simetris hanya mempunyai publik key saja sedangkan algoritma Caesar Cipher Asimetris mempunyai dua key, yaitu publik key dan privat key.

Privat key itu sendiri pada penelitian ini diterapkan pada urutan alfabet algoritma Caesar Cipher Asimetris. Urutan alfabet akan diacak sesuai keinginan pembuat algoritma dan tidak akan sebarluaskan Berbeda dengan algoritma Caesar Cipher Simetris yang hanya memiliki publik key saja dan key-nya diketahui oleh orang-orang awam

6. Waktu Proses

Pada penelitian ini tentunya diperlukan pertimbangan waktu yang dibutuhkan untuk melakukan enkripsi dan dekripsi baik itu dengan algoritma Caesar Cipher Simetris atau Algoritma Caesar Cipher Asimetris. Oleh karena itu peneliti akan melakukan perbandingan waktu antara kedua algoritma diatas dalam bentuk tabel. Adapun untuk satuan waktu yang peneliti gunakan sebagai acuan adalah second.

Tabel 3. Waktu Proses

| panjang teks | Simetris | | Asimetris | |
|---|---|---|---|---|
| | Enkripsi | Dekripsi | Enkripsi | Dekripsi |
| 10 | 0.41 | 0.41 | 0.46 | 0.46 |
| 20 | 0.50 | 0.50 | 0.49 | 0.49 |
| 30 | 0.42 | 0.42 | 0.65 | 0.65 |
| ratarata | 0.44 | 0.44 | 0.53 | 0.53 |

Perlu digaris bawahi pula bahwa perangkat keras dan perangkat lunak berpengaruh besar pada saat menjalankan program oleh karena itu akan ada perbedaan jumlah waktu dan kinerja program yang tidak sama tergantung perangkat keras dan perangkat lunak yang digunakan.

## KESIMPULAN DAN SARAN

1. Kesimpulan

Berdasarkan analisis, penulisan dan pengujian dari penelitian Modifikasi Algoritma Kriptografi Caesar Cipher Menjadi Algoritma Kriptografi Asimetris Menggunakan Bahasa Pemrograman Python, maka diperoleh beberapa kesimpulan:

a. Implementasi algoritma kriptografi Caesar cipher berbasis Python berhasil dilakukan. Sistem yang dihasilkan berjalan sesuai dengan algoritma yang digunakan. Plainteks yang diacak dapat dikembalikan ke bentuk semula.
b. Hasil enkripsi dan dekripsi Algoritma Caesar Cipher Asimetris menghasilkan hasil yang berbeda dengan Algoritma Caesar Cipher Simetris

2. Saran

Berikut ini beberapa saran yang dapat dipertimbangkan untuk memperbaiki penelitian ini dan untuk penelitian kedepannya, yaitu:

a. Untuk pengembangan sistem selanjutnya dapat menggunakan kombinasi algoritma kriptografi lainnya.

 ISSN 2686-3359 (Online)

b. Sistem ini menerima inputan langsung dari keyboard, sehingga penelitian selanjutnya dapat menggunakan inputan berupa file seperti .txt dan sebagainya
c. Sistem ini bekerja hanya pada teks, sehingga diharapkan untuk kedepannya penelitian ini dapat diimplementasikan pada gambar, video dan sebagainya
d. Diharapkan pada penelitian selanjutnya dapat mengembangkan sistem yang mampu menginput teks maupun key secara leluasa
e. Diharapkan untuk penelitian selanjutnya bisa membuat aplikasi berbasis dekstop, andoid, maupun web

## DAFTAR PUSTAKA

[1] Afand, M. I., & Nurhayati. (2020). Implementasi Algoritma Vigenere Cipher Dan Atbash Cipher Untuk Keamanan Teks Pada Aplikasi Catatan Berbasis Android.*30*. *IT Journal*, *8*(1), 2252–2746.
[2] Angriani, H., Saharaeni, Y., Informasi, S., & Cipher, C. (2019). IMPLEMENTASI ALGORITMA CAESAR CIPHER PADA KEAMANAN DATA SISTEM E-VOTING PEMILIHAN. 123–126.
[3] Bahari, M. F., Studi, G., Informatika, T., Ilmu, P., Dan, K., Informasi, T., & Darma, U. B. (2022). Analisa Dan Implementasi Keamanan Pesan Chatting Menggunakan Algoritma Challenge Response. 1(2), 49–53.
[4] Mahendra, I., & Eby Yanto, D. T. (2018). Sistem Informasi Pengajuan Kredit Berbasis Web Menggunakan Agile Development Methods Pada Bank Bri Unit Kolonel Sugiono. *Jurnal Teknologi Dan Open Source*, *1*(2), 13–24. https://doi.org/10.36378/jtos.v1i2.20.
[5] Nilfaidah, N., Miru, A. S., & Lamada, M. (2021). Pengembangan Sistem Absensi Mahasiswa Realtime Menggunakan PHP, MYSQL, SMS Gateway, dan Framework Codeigniter. *Eprints*, *3*, 1–6.
[6] Rachman, T.. (2018). METODE ANALITIS ENKRIPSI DAN DEKRIPSI DENGAN PENERAPAN ALGORITMA KRIPTOGRAFI KLASIK KE DALAM CIPHER Jurnal Elektro dan Telkomunikasi. Jurnal Elektro Dan Telkomunikasi, 26–34.
[7] Septian Widiyanto, Govindo Adnan, Moh. Fatkuroji, Dwi Wahyu Handoyo, M. A. H. (2021). Pengamanan Pesan Text dengan menggunakan Kriptografi Klasik Metode Shift Chipper dan Metode Substitution Chipper. *Riau Journal of Computer Science*, *7*(01), 9–17.
[8] Situmorang, B. H., Sinurat, S., & Tampubolon, K. (2018). Implementasi Algoritma Atbash Untuk Menyandikan Pesan Teks Berbasis Android. *Jurnal Pelita Informatika*, *7*(2), 157–161.
[9] Yusfrizal, Y. (2019). Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android. Jurnal Teknik Informatika Kaputama (JTIK), 3(2), 29–37.