IMPLEMENTASI KOMBINASI ALGORITMA MYSZKOWSKI TRANSPOSITION DAN VIGENERE CIPHER PADA KEAMANAN UNTUK FILE TEKS

Meylissa[1*], Khairil[2], Juju Jumadi[3]
Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Dehasen Bengkulu[1,2,3]
Meylissa319@gmail.com[1], khairil@unived.ac.id[2], juju.jumadi@unived.ac.id[3]*

ABSTRACT

Security and confidentiality issues are important aspects of data, messages and information. Myszkowski transposition work is done by compiling plaintext into matrix rows. Then the matrix is read per column so that the ciphertext is obtained. The number of columns used in the matrix is determined by the length of the key used. while the Vigenere Cipher works by reading word per character, where if the message sent exceeds the length of the key used, the key will be repeated again until the message sent gets its respective key. By combining the Myszkowski transposition algorithm and the Vigenere Cipher, it produces a method that can provide a better level of security compared to applying each method separately.

The results of the analysis and testing carried out using different decryption keys produced that the ciphertext could not be restored, which shows that this is normal because the method used is symmetric cryptography so that the decryption process can only be carried out using the same key as the key at the time of decryption.

Keywords: Security, Myszkowski transposition, Vigenere Cipher,

ABSTRAK

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu data, pesan dan informasi. Myszkowski transposition bekerja dilakukan dengan cara menyusun *plaintext* ke dalam baris matriks. Kemudian matriks tersebut dibaca perkolom sehingga didapatkan *ciphertext.* Banyak kolom yang digunakan pada matriks ditentukan oleh panjang kunci yang digunakan. sedangkan Vigenere Cipher bekerja dengan membaca kata per karakter, dimana apabila pesan yang dikirim melebihi panjang kunci yang digunakan, maka kunci akan diulang kembali sampai pesan yang dikirim tersebut mendapatkan kunci masing-masing. Dengan mengkombinasikan algoritma Myszkowski transposition dan Vigenere Cipher tersebut menghasilkan sebuah metode yang dapat memberikan tingkat keamanan yang lebih baik dibandingkan dengan penerapan masing – masing metode tersebut secara terpisah.

Hasil dari analisa dan pengujian yang dilakukan dengan menggunakan kunci dekripsi yang berbeda menghasilkan chiperteks tidak dapat dikembalikan yang mana menunjukkan hal yang normal dikarenakan metode yang digunakan merupakan kriptografi simetris sehingga proses dekripsi hanya bisa dilakukan menggunakan kunci yang sama dengan kunci pada saat dekripsi.

Kata kunci : Keamanan, Myszkowski transposition, Vigenere Cipher,

1. Pendahuluan

Perkembangan teknologi dan informasi saat ini telah mengalami kemajuan yang sangat pesat, sehingga dibutuhkan pengamanan data untuk menjaga informasi. Masalah keamanan dan kerahasiaan merupakan aspek penting dari suatu data, pesan dan informasi. kerahasiaan informasi atau data menuntut kemananan untuk *upgrade* (Handoko & Krismawan, 2020 : 43). Maka di kembangkanlah cabang ilmu yang mempelajari tentang penyandian informasi atau data yang dikenal dengan istilah kriptografi (Ilaga & Sari, 2020 : 21). Kriptografi telah dikenal dan dipakai cukup lama sejak kurang lebih tahun 1990 sebelum masehi pada prasasti-prasasti kuburan (Puspita & Wayahdi, 2021 : 11).

Kriptografi merupakan salah satu teknik yang dapat digunakan untuk mengamankan suatu informasi. Kriptografi memiliki dua tahap yang umum dilakukan adalah tahap enkripsi dan dekripsi. Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi *ciphertext*, sedangkan dekripsi adalah proses yang dilakukan untuk mengubah pesan tersandi menjadi pesan yang dapat dibaca dan dimengerti (Ridho et al., 2022 : 29). Saat ini algoritma kriptografi yang digunakan adalah optimasi algoritma-algoritma sebelumnya khususnya untuk mewujudkan prinsip-prinsip teknik kriptografi, yaitu *diffusion* (mengaburkan) dan *confusion* (membingungkan) (Lombu et al., 2020 : 39).

Penggunaan kriptografi klasik tidak banyak lagi di implementasikan dikarenakan sangat mudah untuk dipecahkan. Seperti algoritma *vigenere cipher* yang tergolong sederhana, kemungkinan pesan yang akan kita enkripsi masih dapat diketahui oleh orang lain (Fauzi, 2020 : 41). Sedangkan Algoritma Myszkowski Transposition adalah sebuah transposisi *cipher*. *Dimana* Algoritma ini didasarkan pada penggunaan kunci transposisi dengan menulis teks secara horizontal (baris) dan membacanya secara vertikal (kolom) (Bhowmick et al., 2021 : 68)

Oleh karena itu kombinasi kriptografi klasik dapat menjadi salah satu alternatif yang dapat meningkatkan keamanan terhadap data dan informasi. Pada penelitian ini penulis menggunakan kombinasi algoritma Myszkowski Transposition dan Vigenere Cipher dalam mengamankan data file teks.

Berdasarkan uraian latar belakang diatas, maka penulis tertarik untuk melakukan penelitian yang diberi judul **" Implementasi** Kombinasi **Algoritma Myszkowski Transposition dan Vigenere Cipher Pada Keamanan Untuk File Teks"**

2. Tinjauan Pustaka dan Pengembangan Hipotesis

Kriptografi

Kriptografi telah dikenal dan dipakai cukup lama sejak kurang lebih tahun 1900 sebelum masehi pada prasasti-prasasti kuburan. Kriptografi sendiri berasal dari kata "*Crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan. Jadi, dapat dikatakan kriptografi adalah tulisan yang tersembunyi. Dengan adanya tulisan yang tersembunyi ini, orang-orang yang tidak mengetahui bagaimana tulisan tersebut disembunyikan tidak akan mengetahui bagaimana cara membaca maupun menerjemahkan tulisan tersebut (Puspita & Wayahdi, 2021 : 34).

Criptography berasal dari bahasa Yunani. Menurut bahasanya, istilah tersebut terdiri dari kata *kripto* dan *graphia*. *Kripto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan) (Fauzah & Iqbal, 2021 : 61). Menurut terminologinya, kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan, ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi dari pesan tersebut kemungkinan dapat disadap oleh pihak lain. Untuk menjaga keamanan pesan, maka pesan tersebut dapat *discreamble* / diacak atau diubah menjadi kode yang tidak dapat dimengerti oleh orang lain (Azlin et al., 2020 : 38). Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu himpunan yang berisi elemen plainteks dan himpunan yang berisi elemen chipteks. Enkripsi dan dekripsi merupakan fungsi transformasi antara dua himpunan tersebut.

a. Kriptografi`Klasik

Kriptografi klasik merupakan kriptografi yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang. Kriptografi ini melakukan pengacakan huruf pada kata terang (*plaintext*). Kriptografi ini hanya melakukan pengacakan pada huruf A-Z, dan sangatlah tidak disarankan untuk mengamankan informasi-informasi penting karena dapat dipecahkan dalam waktu singkat. Biarpun telah ditinggalkan, kriptografi klasik tetap dapat ditemui disetiap pelajaran kriptografi sebagai pengantar kriptografi modern (Pardede, Manurung, & Filina, 2020 : 51)

b. Kriptografi Modern

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern (Manaor & Pardede, 2020 : 49)

Algoritma *Myszkowski Transposition*

Myszkowski transposition adalah salah satu cipher transposisi. Algoritma ini ditulis oleh Emile Victor Theodore Myszkowski dalam bukunya yang berjudul *Cryptographie Indechiffrable* pada tahun 1902. Buku ini ditulis berdasarkan penemuannya dan diklaim aman dari orang yang tidak berhak membaca pesan tersebut. Hal ini didasarkan pada penggunaan kunci transposisi dengan menulis teks secara horizontal (baris) dan membacanya secara vertikal (kolom) (Bhowmick et al., 2021 : 68).

Proses enkripsi dilakukan dengan cara menyusun *plaintext* ke dalam baris matriks. Kemudian matriks tersebut dibaca perkolom sehingga didapatkan *ciphertext.* Banyak kolom yang digunakan pada matriks ditentukan oleh panjang kunci yang digunakan. Kunci berupa urutan angka unik ataupun terdapat angka yang sama dengan posisi acak, dimulai dari 1. *Plaintext* dibaca sesuai urutan angka pada kunci dengan ketentuan pada angka yang unik dibaca perkolom, sedangkan pada angka yang sama dibaca dari kiri ke kanan perkolomnya. Sedangkan untuk proses dekripsi merupakan kebalikan dari proses enkripsinya. Émile Victor Théodore Myszkowski di tahun 1902 memperkenalkan variasi dari metode penyandian transposisi kolom, yang dibedakan dalam pendefinisian dan permutasian kata kunci-nya. Dalam metode penyandian transposisi kolom.

Algoritma Vigenere Cipher

Vigenère cipher adalah salah satu algoritma kriptografi klasik yang diperkenalkan pada abad 16 atau kira-kira pada tahun 1986. Algoritma kriptografi ini dipublikasikan oleh seorang diplomat dan juga kriptologis yang berasal dari Prancis, yaitu Blaise de Vigenère, namun sebenarnya algoritma ini telah digambarkan sebelumnya pada buku La Cifra del Sig. Giovan Batista Belaso, sebuah buku yang ditulis oleh Giovan Batista Belaso, pada tahun 1553 (Irawan, 2021 : 71)

Karakter yang digunakan dalam Vigenere Cipher yaitu A, B, C, ..., Z dan dikonversi kedalam angka 0, 1, 2, ..., 25. Proses enkripsi dilakukan dengan menulis kunci berulang kali sesuai dengan panjang karakter pada pesan (Mendrofa, dkk, 2020 : 61).

Pada *Vigenère cipher* kunci yang digunakan berbentuk deretan huruf. Kunci yang berbetuk deretan kata tersebut akan memungkinkan setiap huruf *plainteks* untuk dienkripsi dengan kunci yang berbeda. Jika panjang kunci yang digunakan lebih pendek dari panjang plainteks maka kunci akan diulang sampai panjang kunci sama dengan panjang *plainteks*. Algoritma ini akan meminimalkan kemungkinan dipecahkannya *cipherteks* jika satu huruf plainteks diketahui. rumus dari enkripsi dan dekripsi dengan *vigenere chiper* adalah :

Enkripsi : $E(p_i) = (p_i + key)\ mod\ 126$

Dekripsi : $D(c_i) = (c_i - key)\ mod\ 126$

Tinjauan Bahasa Pemrograman PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah *server*. Data yang dikirim oleh *user client* akan diolah dan disimpan pada database *web server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP, file harus di upload kedalam server. Upload adalah proses mentransfer data atau file dari komputer client ke dalam *web server* (Mubarak 2021 : 3).

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman scripting untuk membuat halaman web dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman web, PHP sebenarnya juga dapat digunakan membuat aplikasi *command line* dan GUI. Cara kerja PHP adalah dengan menyelipkannya diantara kode HTML (*Hypertext Markup Language*) (Wahyuni and Irawan 2021 : 19).

Dari pendapat di atas dapat disimpulkan bahwa web merupakan layanan yang dapat oleh pemakai komputer terhubung ke internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (link) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser.*

Kelebihan bahasa pemrograman PHP yaitu :

a. Bahasa pemoraman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunanya.
b. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lightpd, nginx, hingga Xitami dengan konfigurasi lebih mudah.
c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu pengembangan.

d. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak
e. PHP adalah bahasa open source yang dapat digunakan dibeberapa mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem

UML (*Unified Modeling Language)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. (Suendri, 2020 : 78)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Haviluddin, 2021 : 90).

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen komponen yang diperlukan dalam sistem *software* Adapun tujuan dari UML adalah :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya

Unified Modeling Language (UML) biasa digunakan untuk (Alfina & Harahap, 2021 : 42)

1. Menggambarkan batasan sitem dan fungsi -fungsi sistem secara umum, dibuat dengan *use case* dan *actor*
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction* diagram
3. Menggambarkan representasi struktur statik sebuah sistem dalam betuk *class diagram*
4. Membuat model behavior "yang menggambarkan kebiasaan atau sifat sebuah sistem" dengan state *transition diagram*
5. Menyatakan arsitektur implementasi fisik menggunakan c*omponent and development*
6. Menyampaikan atau memperluas *fungsionality* dengan *stereo types*

***Activity* Diagram**

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem,yang dimulai dari pandangan business level hingga ke operational level. Pada dasarnya, *activity* diagram merupakan variasi dari statechart diagram. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.

***Sequence* Diagram**

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek – obyek yang melakukan suatu tugas atau aksi tertentu. Obyek – obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

***Use Case* Diagram**

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yangditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana" (*Romi,* 2013). Sebuah *use case*merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan

sebuahpekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya.

Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Flowchart

Flowchart dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus. Manfaat yang akan diperoleh bila menggunakan flowchart dalam pemecahan masalah komputasi: Terbiasa berfikir secara sistematis dan terstruktur, mudah mengecek dan menemukan bagian-bagian prosedur yang tidak valid dan bertele-tele. Prosedur akan mudah dikembangkan (Nuraini, 2021 : 73)

3. Metode Penelitian

Dalam melakukan penelitian ini, penulis menggunakan metode terapan (*applied research*). Penelitian terapan atau *applied research* dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau kelompok maupun untuk keperluan industri atau politik dan bukan untuk wawasan keilmuan semata.

Dalam melaksanakan penelitian terapan ini terdapat 5(lima) langkah, diantaranya :

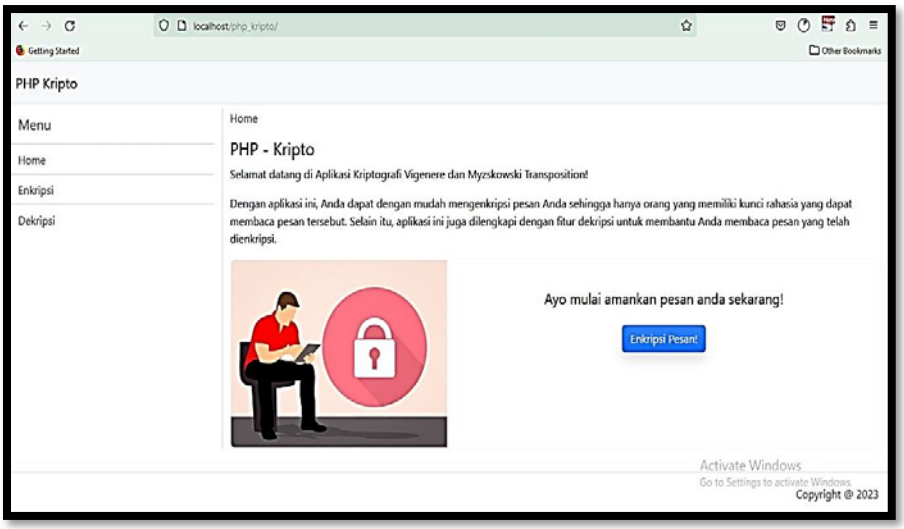
a. Melakukan sesuatu yang sedang diperlukan, dipelajari, diukur, dan diperiksa kelemahannya.
b. Mencari satu dari kelemahan-kelemahan yang diperoleh dipilih untuk penelitian.
c. Mencari dan memberikan solusi dalam melakukan pemecahan masalah
d. Kemudian dilakukan modifikasi sehingga penyelesaian dapat dilakukan untuk diterapkan.

Pemecahan dipertahankan dan menempatkannya dalam suatu kesatuan sehingga jadi bagian permanen dalam satu sistem.

4. Hasil dan Pembahasan

Pada aplikasi implementasi Kombinasi Algoritma Myszkowski Transposition dan Vigenere Cipher pada keamanan untuk file teks terdapat beberapa *interface* atau antarmuka yang di desain untuk mempermudah *user* atau pemakai dalam menggunakan atau menjalankan aplikasi ini. Adapun *interface* atau antarmuka tampilan beranda (Halaman Utama) aplikasi

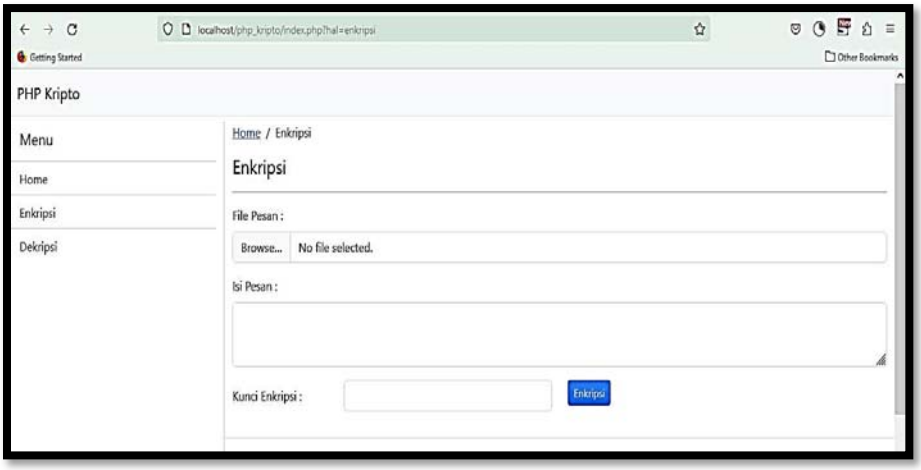
Untuk mengoperasikan atau menjalankan aplikasi dilakukan dengan cara mengetikan *localhost/php_kripto*. Halaman menu utama merupakan halaman pertama kali muncul ketika aplikasi berhasil dijalankan. Pada halaman menu utama terdapat beberapa menu Home. Enkripsi dan Dekripsi sehingga tampilan aplikasi terlihat seperti gambar berikut :



Gambar 4.2 Beranda Aplikasi

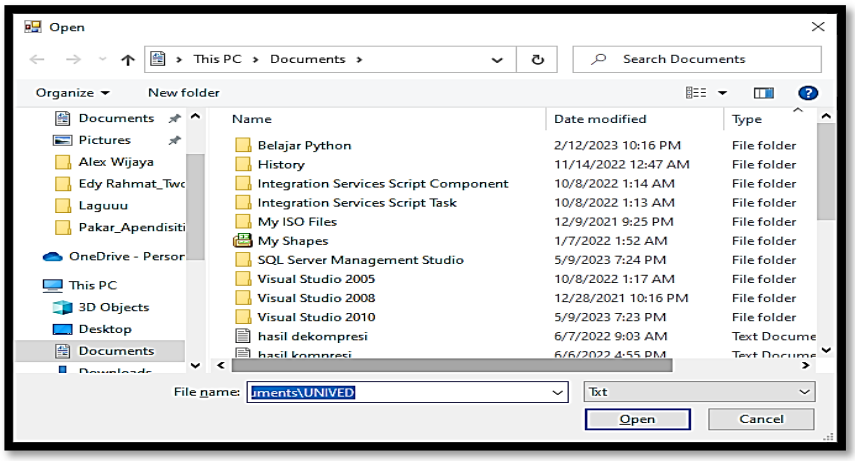
1. Tampilan Halaman Form Enkripsi

Form ini digunakan untuk melakukan enkripsi terhadap pesan plainteks yang di masukkan oleh pengguna. Pengguna pertama sekali membuka berkas plainteks yang akan di enkripsi atau mengetikkan langsung pesan plainteks yang akan di enkripsi pada kolom plainteks.



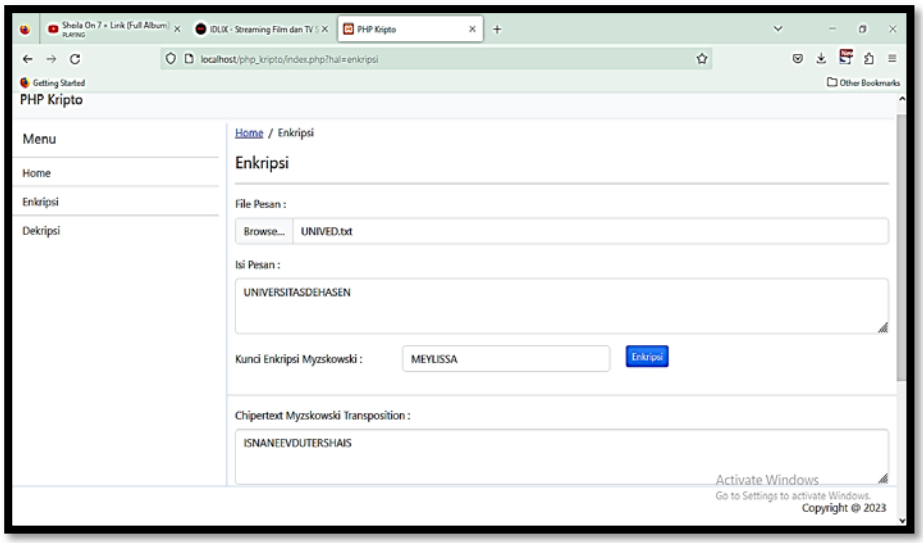
Gambar 4.3 Form Enkripsi

File plainteks dapat dibuka dengan menggunakan tombol "*browse*" yang kemudian akan menampilkan dialog untuk memilih *file* plainteks yang akan di – enkripsi. Tampilan dialog untuk memilih *file* plainteks.



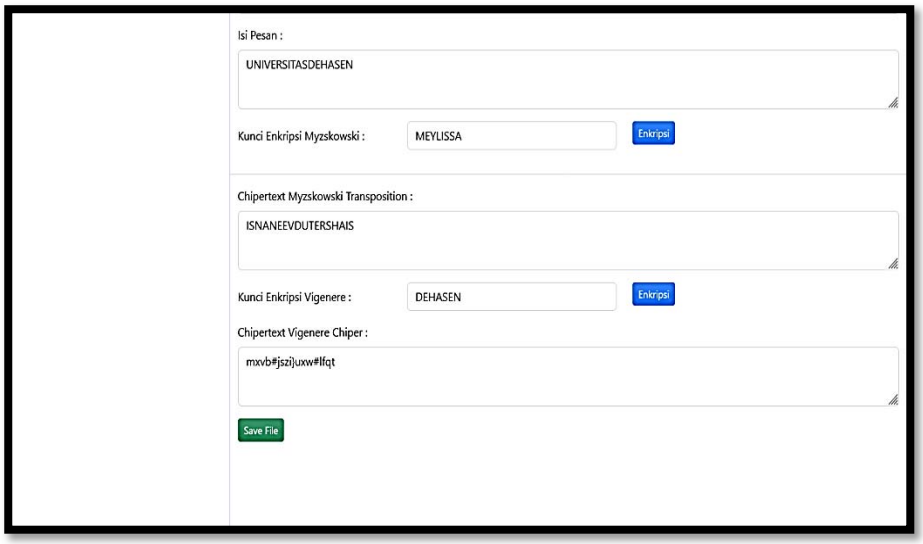
Gambar 4.4 Dialog Memilih Berkas Plainteks

Setelah plainteks dibuka selanjutnya pengguna dapat melakukan proses enkripsi dengan menerapkan algoritma Myszkowski Transposition dengan memasukkan kunci dan melakukan proses enkripsi dengan menekan tombol "Enkripsi" sehingga proses enkripsi akan dilakukan dan akan menampilkan chiperteks.



Gambar 4.5 Proses Enkripsi Myszkowski Transposition

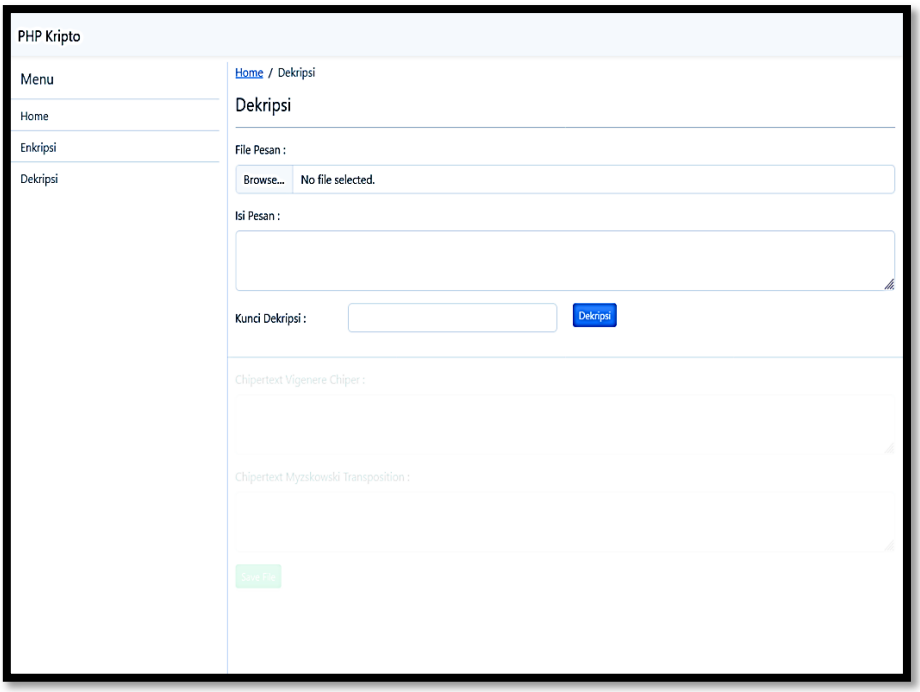
Setelah memperoleh hasil dari enkripsi algoritma Myszkowski Transposition, selanjutnya ciphertext dari algoritma Myszkowski Transposition dilakukan dengan enkripsi dengan algoritma Vigenere Cipher. Adapun hasilnya.



Gambar 4.6 Proses Enkripsi *Vigenere Cipher*

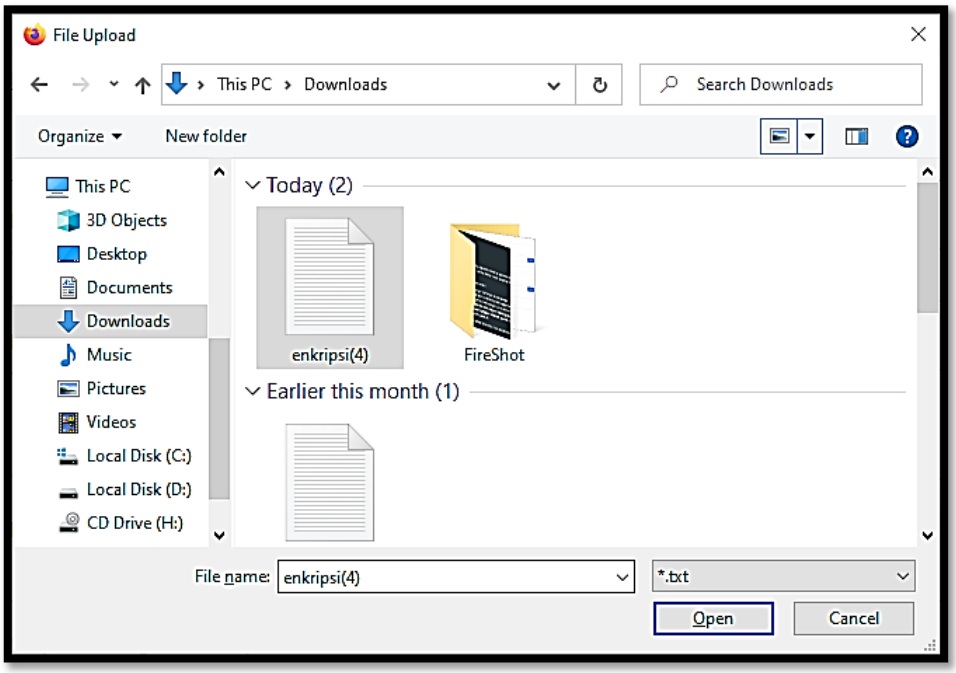
2. Form Dekripsi

Form ini digunakan untuk melakukan dekripsi terhadap pesan chiperteks yang di masukkan oleh pengguna. Pengguna pertama sekali membuka berkas chiperteks yang akan di dekripsi. Adapun tampilan dari *form dekripsi*.



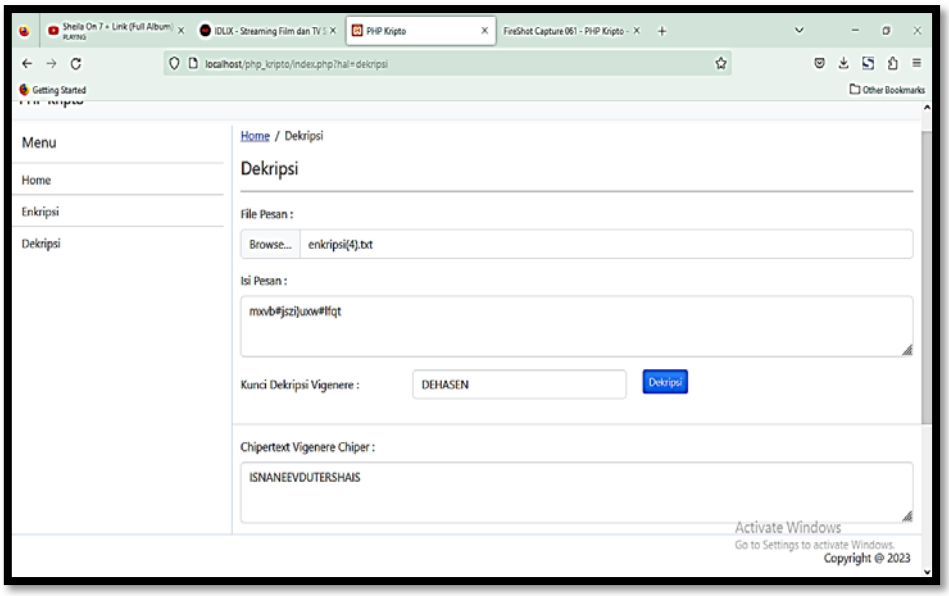
Gambar 4.7 Halaman (*Form*) Dekripsi

Berkas chiperteks dapat dibuka dengan menggunakan tombol "*Browse*" yang kemudian akan menampilkan dialog untuk memilih *file* chiperteks yang akan di – dekripsi. Tampilan dialog untuk memilih *file* chiperteks.



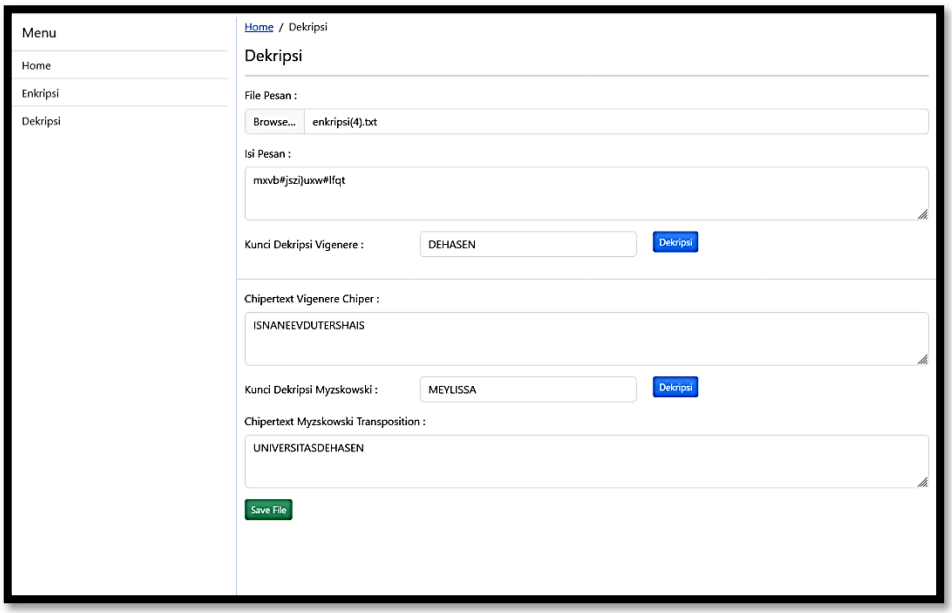
Gambar 4.8 Dialog Memilih Berkas Chiperteks

Setelah plainteks dibuka selanjutnya pengguna dapat melakukan proses dekripsi dengan memasukkan kunci yang telah dibangkitkan sebelumnya dan melakukan proses dekripsi dengan menekan tombol "Dekripsi" sehingga proses dekripsi akan dilakukan dan akan menampilkan plainteks.



Gambar 4.9 Proses Hasil Dekripsi *Vigenere Cipher*

Setelah diperoleh ciphertext dari *Vigenere Cipher,* selanjutnya dilakukan dekripsi dengan algoritma Myszkowski Transposition dengan kata kunci saya sama pada saat enkrispi. Adapun tampilan hasil deksripsi dengan algoritma Myszkowski Transposition.



Gambar 4.10 Proses Hasil Dekripsi Myszkowski Transposition

Plainteks hasil dekripsi kemudian dapat disimpan menjadi file menggunakan tombol "Simpan Hasil" sehingga dapat di gunakan lebih lanjut oleh pengguna.

Hasil Pengujian

Pengujian yang dilakukan pada aplikasi ini adalah dengan menggunakan teknik *black box*, teknik *black box* ini merupakan teknik pengujian yang berfokus pada keluaran hasil dari respon, atau secara simpel untuk mengetahui apakah ada *error* atau ada fungsi yang tidak berjalan sesuai dengan harapan. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Berikut tabel pengujian *black box*

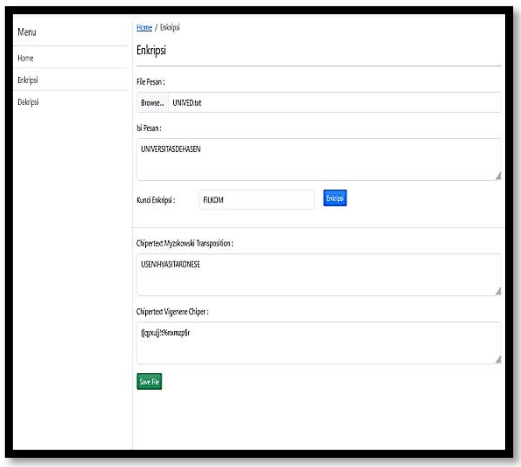
Tabel 4.1 Pengujian *Black Box*

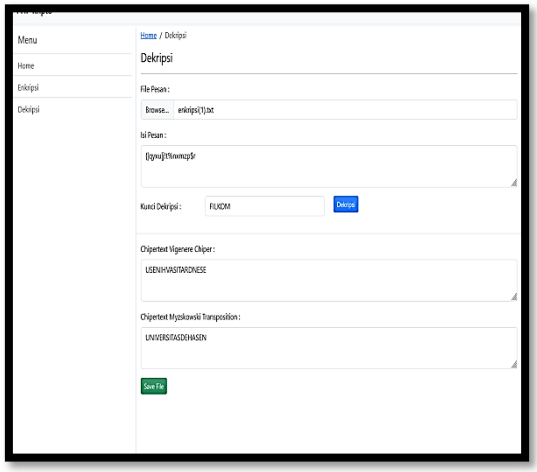
Jenis Uji	Keterangan Uji	Jenis Pengujian
Open	Mencari file.txt	*Black Box*
Enkripsi	Proses Enkripsi	*Black Box*
Dekripsi	Proses Deksipsi	*Black Box*

Tabel 4.2 Kasus dan Hasil Uji File .txt

Kasus dan Hasil Uji File. txt			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan file.txt	File dapat diproses	File berhasil diproses	[x] diterima [ ] ditolak
Enkripsi	File berhasil di enkripsi	File berhasil berubah sesuai dengan kunci yang digunakan	[x] diterima [ ] ditolak
Dekripsi	File berhasil di dekripsi	File berhasil kembali menjadi plainteks dengan menggunakan kunci sama pada saat enkripsi	[x] diterima [ ] ditolak

Tabel 4.3 Kasus dan Hasil Uji txt .docx

Kasus dan Hasil Uji File.txt			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan file.docx	File dapat diproses	File berhasil diproses	[x] diterima [ ] ditolak
Enkripsi	File berhasil di enkripsi	file berhasil berubah sesuai dengan kunci yang digunakan 	[x] diterima [ ] ditolak
Dekripsi	File berhasil di	File berhasil kembali menjadi plainteks dengan menggunakan kunci sama pada saat	[x] diterima [ ] ditolak

	dekripsi	enkripsi 	

5. Kesimpulan

Berdasarkan pembahasan dan pengujian program yang dilakukan, maka dapat di tarik kesimpulan sebagai berikut :

1. Pengamanan data teks dengan menerapkan algoritma kriptografi *Myszkowski Transposition* dan *Vigenere Cipher* dalam pengamanan dokumen dapat dikombinasikan dengan baik. Proses enkripsi dimulai terlebih dahulu menggunakan metode *Myszkowski Transposition* yang kemudian hasil enkripsi tersebut di enkripsi lagi menggunakan metode *Vigenere Cipher* sehingga seperti proses enkripsi beruntun. Proses dekripsi merupakan proses kebalikan dari proses dekripsi sehingga plainteks dapat diperoleh kembali. Implementasi kedua metode tersebut menghasilkan sebuah metode yang dapat memberikan tingkat keamanan yang lebih baik dibandingkan dengan penerapan masing – masing metode tersebut secara terpisah.
2. Dari hasil pengujian yang telah dilakukan dapat dilihat bahwa aplikasi yang dikembangkan dapat bekerja sesuai dengan yang diharapkan. Pengujian normal menghasilkan chiperteks dan plainteks yang sesuai.
3. Pengujian enkripsi dan dekripsi menggunakan kunci yang berbeda untuk melihat fungsi aplikasi jika diberikan kunci yang tidak sama pada saat proses enkripsi dengan dekripsi. Dari pengujian yang dilakukan chiperteks tidak dapat dikembalikan yang mana menunjukkan hal yang normal dikarenakan metode yang digunakan merupakan kriptografi simetris sehingga proses dekripsi hanya bisa dilakukan menggunakan kunci yang sama dengan kunci pada saat dekripsi
4. Dari hasil pembangunan aplikasi perangkat lunak serta pengujian yang telah dilakukan pada perangkat lunak tersebut maka aplikasi yang dibangun pada penelitian ini dapat digunakan oleh pengguna umum untuk mengamankan pesan.

REFERENSI

1. Alfina, O., & Harahap, F. (2021). Pemodelan uml sistem pendukung keputusan dalam penentuan kelas siswa siswa tunagrahita. *Methomika: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 143-150
2. Azlin, Musadat, F., & Nur, J. (2020). *Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma Base64*. *7*(2), 1–5.
3. Bhowmick, A., Anand Vardhan Lal, & Ranjan, N. (2021). Enhanced 6x6 Playfair Cipher using Double Myszkowski Transposition. *International Journal of Engineering Research And*, *V4*(07), 1100–1104. https://doi.org/10.17577/ijertv4is070849
4. Darmawan , M., & Windarto , W. (2020). Implementasi Algoritma Kriptografi Vigener Cipher Dan Affine Cipher Untuk Mengamankan Pesan Pada Aplikasi Chatting Berbasis Android. *SKANIKA (Sistem Komputer dan Teknik Informatika)*, 24-32

5. Fauzah, R., & Iqbal. (2021). Aplikasi Kriptografi Dalam Mengamankan Pesan Teks Dengan Metode Algoritma Rc4 Berbasis Android. *Jurnal Tika*, *6*(01), 69–73. https://doi.org/10.51179/tika.v6i01.416
6. Fauzi, M. A. (2020). Perancangan Aplikasi Keamanan Pesan Teks dengan menggunakan Algoritma Triple Transposition Vigenere Cipher. *MEANS (Media Informasi Analisa Dan Sistem)*, *4*(1), 27–32. https://doi.org/10.54367/means.v4i1.315
7. Handoko, B. L., & Krismawan, A. D. (2020). Aplikasi Super Enkripsi Kriptografi Menggunakan Kombinasi Transposisi Kolom Dan Vigenere Cipher. *Semnas Lppm*, 534–539.
8. Haviluddin. (2021). Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Informatika Mulawarman*, 18-29
9. Hidayatullah, P. (2022). *Visual Basic.Net Membuat Aplikasi Database Kreatif dan Program Kreatif*. Bandung: Informatika
10. Ilaga, K. R., & Sari, C. A. (2020). Analysis of Secure Image Crypto-Stegano Based on Electronic Code Book and Least Significant Bit. *Journal of Applied Intelligent System*, *3*(1), 28–38.
11. Irawan, M. (2021). Implementasi kriptografi vigenere cipher dengan php. *Jurnal teknologi informasi (JurTI)*, 11-21
12. Lombu, D., Tarihoran, S. D., & Gulo, I. (2020). Kombinasi Mode Cipher Block Chaining Dengan Algoritma Triangle Chain Cipher Pada Penyandian Login Website. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *2*(1), 1. https://doi.org/10.30645/j-sakti.v2i1.51
13. Mendrofa. (2020). Metode Algoritma Vigenere Chipher. Jurnal Informatika, 50-61
14. Nuraini, R. (2021). DESAIN ALGORITHMA OPERASI PERKALIAN MATRIKS MENGGUNAKAN METODE FLOWCHART. *JURNAL TEKNIK KOMPUTER*, 144 -151
15. Pardede, A., Manurung, H., & Filina, D. (2021). Algoritma Vigenere Cipher Dan Hill Cipher Dalam Aplikasi Keamanan Data Pada File Dokumen. *Jurnal Teknik Informatika Kaputama (JTIK)*, 26-33.
16. Puspita, khairani, & Wayahdi, M. R. (2021). Analisis Kombinasi Metode Caesar Cipher , Vernam Cipher , Dan Hill Cipher Dalam Proses Kriptografi. *Seminar Nasional Teknologi Informasi Dan Multimedia 2015*, *Februari*, 43–48.
17. R.H Sianipar. (2022). *Visual Basic.Net Untuk Programmer*. Yogyakarta: Andi Offset
18. Ridho, A., Mutia, C., & Sinaga, A. P. (2022). Analisis Enkripsi dan Dekripsi Cipher Teks Menggunakan Kombinasi Gronsfeld Cipher Dengan Reverse Cipher. *Jurnal Teknik Informatika Kaputama (JTIK)*, *6*(1). https://jurnal.kaputama.ac.id/index.php/JTIK/article/view/689
19. Suendri. (2020). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, 1-9
20. Ziliwu, K. B., & Maslan, A. (2022). *Jurnal Comasie*. *02*, 117–126.