

**IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI
CAESAR CIPHER UNTUK PROSES ENKRIPSI DAN
DEKRIPSI**

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Matematika
dalam Ilmu Matematika



Oleh : **FADILLAH AL MUSYAFFA**
NIM : 2108046052

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2025

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : FADILLAH AL MUSYAFFA
NIM : 2108046052
Jurusan/Program Studi : MATEMATIKA/ MATEMATIKA

menyatakan bahwa skripsi yang berjudul :

IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI CAESAR CIPHER UNTUK PROSES ENKRIPSI DAN DEKRIPSI

secara keseluruhan adalah hasil penelitian/karya saya sendiri,
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 16 Juni 2025

Pembuat pernyataan,



FADILLAH AL MUSYAFFA

NIM : 2108046052



KEMENTERIAN AGAMA R.I.
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka (Kampus II) Ngaliyan Semarang
Telp. 024-7601295 Fax. 7615387

PENGESAHAN

Naskah skripsi berikut ini :

Judul : **IMPLEMENTASI SANDI KOTAK PADA
KRIPTOGRAFI CAESAR CIPHER UNTUK PROSES
ENKRIPSI DAN DEKRIPSI**

Penulis : **FADILLAH AL MUSYAFFA**

NIM : **2108046052**

Jurusan : **Matematika**

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Matematika.

Semarang, 25 Juni 2025

DEWAN PENGUJI

Penguji I,

Penguji II,

Eva Khoirun Nisa, M.Si
NIP : 1987010220190320010

Agus Wayan Yulianto, M.Sc
NIP : 198907162019031007

Penguji III,

Penguji IV,

Aini Fitriyah, M.Sc
NIP : 198909292019032021

Emy Siswanah, M.Sc
NIP : 198702022011012014

Pembimbing I,

Any Muanalifah, M.Si., Ph.D
NIP : 198201132011012009

NOTA DINAS

Semarang, 16 Juni 2025

Yth. Ketua Program Studi Matematika
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum warahmatullahi wabarakatuh

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI
CAESAR CIPHER UNTUK PROSES ENKRIPSI DAN
DEKRIPSI
Nama : FADILLAH AL MUSYAFFA
NIM : 2108046052
Jurusan : Matematika

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqasyah.

Wassalamu'alaikum warahmatullahi wabarakatuh

Pembimbing I,



Any Muanalifah, M.Si., Ph.D

NIP : 198201132011012009

ABSTRAK

FADILLAH AL MUSYAFFA (2108046052), 2025.
IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI CAESAR
CIPHER UNTUK PROSES ENKRIPSI DAN DEKRIPSI. Jurusan
Matematika, Fakultas Sains dan Teknologi, Universitas Islam
Negeri Walisongo Semarang.

Keamanan informasi merupakan hal penting dalam komunikasi digital, terutama dalam penyampaian pesan rahasia. Salah satu solusi yang digunakan untuk menjaga kerahasiaan pesan adalah dengan kriptografi. Penelitian ini bertujuan untuk mengimplementasikan metode sandi kotak yang dikombinasikan dengan *caesar cipher* dalam bentuk aplikasi berbasis GUI (*Graphical User Interface*) menggunakan MATLAB sebagai alat bantu komputasi. Proses enkripsi dalam sistem ini dilakukan dengan dua tahap, yaitu penyandian huruf menggunakan *caesar cipher* terlebih dahulu, kemudian hasilnya dikonversikan menjadi simbol sandi kotak. Sebaliknya, dekripsi dilakukan dengan membaca simbol sandi kotak menjadi huruf, lalu dilakukan dekripsi *caesar* berdasarkan kunci yang sama. Adapun bentuk matematis proses enkripsi pesan menggunakan sandi kotak dan *caesar cipher* yaitu $C = S(E(P, K))$, proses dekripsi yaitu $P = D(S^{-1}(C), K)$. Hasil penelitian menunjukkan bahwa kombinasi kedua metode tersebut mampu meningkatkan kerahasiaan pesan dengan pendekatan ganda: transformasi teks dan konversi visual. Selain itu, GUI yang dikembangkan memberikan kemudahan bagi pengguna dalam memahami proses kriptografi secara interaktif dan visual. Penelitian ini juga menunjukkan bahwa kriptografi klasik masih relevan untuk tujuan edukatif dan aplikasi dasar dalam pengamanan data.

Kata kunci : Caesar Cipher, Kriptografi, Sandi Kotak.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil 'alamiin. Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat, hidayah, serta karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir (skripsi) yang berjudul **IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI CAESAR CIPHER UNTUK PROSES ENKRIPSI DAN DEKRIPSI** ini dengan lancar. Sholawat serta salam penulis sampaikan kepada junjungan kita, nabi Muhammad SAW yang kita nantikan syafaatnya di *yaumul qiyamah* nanti.

Pada kesempatan ini, penulis ingin mengucapkan banyak terima kasih kepada pihak yang telah mendukung dan membantu hingga terselesaikannya penyusunan tugas akhir (skripsi) ini. Untuk itu penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan kemudahan dalam segala hal sehingga laporan ini dapat terselesaikan.
2. Any Muanalifah, M.Si., Ph.D., selaku Ketua Jurusan Program Matematika dan sebagai dosen pembimbing yang telah membimbing dan mengarahkan penulis dengan ikhlas dan sabar dalam penyusunan skripsi.
3. Agus Wayan Yulianto, M.Sc., selaku dosen wali atau orang tua akademik bagi penulis yang telah sabar mendukung segala proses yang penulis lalui selama perkuliahan.
4. Seluruh Dosen dan Staf Fakultas Sains dan Teknologi yang membantu proses penyelesaian skripsi.

5. Mamahku tersayang, Omah Maryamah yang selalu memberikan semangat serta mendoakan di sepertiga malamnya untuk kesejahteraan anaknya.
6. Ayahku terhebat, Ardi Suardi yang selalu memberikan doa, kasih sayang, memberikan dukungan semangat dan materil, serta mendukung penuh penulis dalam berbagai hal selama ini.
7. Kakak Penulis, Syifa Khairillah, S.Pd., yang selalu mendoakan dan juga senantiasa memberikan dukungan baik secara moril maupun materil saat selama masa perkuliahan penulis.
8. Adek Penulis, Alya Sayla Salsabila dan Azkayra Adreena Kaesa Almahyra. Terima kasih atas doa, dan dukungan kepada penulis.
9. Keluarga Karonsih Pride: Pak Kadar, Anam, Okky, Said, Fajri, Rudi, Fadil botak, Ghoza, Bagus, Abi, Amar, Krida, Ical, dan Apif yang telah banyak membantu penulis selama masa perkuliahan, telah memberikan dukungan kepada penulis, menjadi tempat berbagi canda dan tawa serta suka dan duka. Terima kasih karena telah menjadi bagian perjalanan hidup penulis selama masa perkuliahan.
10. Orang-orangan Benten: Anam, Razan, Rudi, Hani, Sope, Dela, Caca, Rahma, dan Faten yang telah menjadi teman-teman pertama penulis selama masa perkuliahan, membantu penulis selama perkuliahan, menjadi tempat berbagi cerita. Meskipun sudah tidak lengkap kalian selalu menjadi orang-orang yang hebat bagi penulis dimanapun kalian berada penulis selalu mendoakan yang terbaik untuk kalian.

11. Teman-teman Chindo Jawa: Anam, Apif, Jafar, Adnan, Wiwit, Nabila, Salsa, Hanna, Frischa, Desti dan Alfina yang telah memberi warna kehidupan bagi penulis selama perkuliahan.
12. Teman-teman jurusan Matematika angkatan 2021 terutama Matematika B yang telah berjuang bersama melewati masa-masa perkuliahan dan menjadikan kehidupan kampus lebih berwarna.

Semoga bantuan yang tulus dari berbagai pihak, mendapatkan imbalan yang setimpal dari Allah SWT. Penulis berharap semoga skripsi ini dapat bermanfaat bagi penulis sendiri khususnya dan juga bagi segenap pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh

Semarang, 10 Mei 2025

Penulis,

Fadillah Al Musyaffa

NIM : 2108046052

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	ii
PENGESAHAN	iv
NOTA PEMBIMBING I	iv
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Tujuan Penelitian	3
C. Tinjauan Pustaka	3
D. Metodologi Penelitian	5
E. Sistematika Penulisan	6
BAB II LANDASAN PUSTAKA	8
A. Kriptografi	8
B. <i>Caesar Cipher</i>	14
1. Proses Enkripsi	15
2. Proses Dekripsi	17
C. Sandi Kotak	20
1. Proses Enkripsi	23
2. Proses Dekripsi	24
D. Hasil Penelitian yang Relevan	27
BAB III HASIL PENELITIAN DAN PEMBAHASAN	29
A. IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI <i>CAESAR CIPHER</i> UNTUK PROSES ENKRIPSI DAN DEKRIPSI	29
1. Proses Enkripsi	31
2. Proses Dekripsi	34
B. SIMULASI APLIKASI	38

1.	Desain Aplikasi	38
2.	Langkah Pembuatan Aplikasi dan <i>Script</i> Aplikasi	40
3.	Pengujian Fungsional Aplikasi Implementasi Sandi Kotak pada <i>Caesar</i> <i>Cipher</i>	47
4.	Analisis Hasil	52
BAB IV SIMPULAN DAN SARAN		53
A.	Simpulan	53
B.	Saran	54
DAFTAR PUSTAKA		55
Lampiran-lampiran		58

DAFTAR TABEL

Tabel	Judul	Halaman
Tabel 2.1	Mengkonversi huruf ke dalam <i>integer</i>	15
Tabel 3.3	Komponen & Fungsi GUI MATLAB	39
Tabel 3.4	Tabel Unit Aplikasi dan Rancangan Proses	51

DAFTAR GAMBAR

Gambar	Judul	Halaman
Gambar 2.1	Proses Enkripsi dan Dekripsi Pesan	13
Gambar 2.2	Proses Enkripsi dan Dekripsi pada <i>Caesar cipher</i>	20
Gambar 2.3	Sandi Kotak 1	21
Gambar 2.4	Sandi Kotak 2	21
Gambar 2.5	Sandi Kotak 3	22
Gambar 2.6	Proses Enkripsi dan Dekripsi pada Sandi Kotak	26
Gambar 3.1	Flowchart Sandi Kotak pada Kriptografi <i>Caesar Cipher</i>	30
Gambar 3.2	Desain Antarmuka <i>Interface</i> Aplikasi	39
Gambar 3.3	Tampilan pada GUI MATLAB	40
Gambar 3.4	Tampilan <i>property inspector</i> pada GUI MATLAB	40
Gambar 3.5	Tampilan judul yang ditampilkan pada GUI MATLAB	41
Gambar 3.6	Tampilan <i>Static text</i> GUI MATLAB	41
Gambar 3.7	Tampilan <i>Edit text</i> GUI MATLAB	42
Gambar 3.8	Tampilan <i>Pushbutton</i> GUI MATLAB	42
Gambar 3.9	Tampilan <i>Axes</i> GUI MATLAB	43
Gambar 3.10	Tampilan akhir GUI MATLAB sebelum dijalankan	43
Gambar 3.11	Tampilan Pengujian pada Teks Input	47
Gambar 3.12	Tampilan Pengujian pada <i>Kunci Caesar</i>	48

Gambar 3.13	Tampilan Pengujian <i>Button</i> 'Enkripsi'	48
Gambar 3.14	Tampilan Pengujian <i>Button</i> 'Dekripsi'	49
Gambar 3.15	Tampilan Pengujian <i>Button</i> 'Refresh'	49
Gambar 3.16	Tampilan Pengujian <i>Button</i> 'Tutup'	50

DAFTAR LAMPIRAN

	Halaman
Lampiran 1	<i>SCRIPT</i> GUI MATLAB Inisialisasi GUI (Opening dan Output) 58
Lampiran 2	<i>SCRIPT</i> GUI MATLAB Callback Input dan Kunci 59
Lampiran 3	<i>SCRIPT</i> GUI MATLAB Tombol Enkripsi dan Dekripsi 60
Lampiran 4	<i>SCRIPT</i> GUI MATLAB Fungsi <i>Caesar Cipher</i> 61
Lampiran 5	<i>SCRIPT</i> GUI MATLAB Fungsi Display Sandi Kotak 62
Lampiran 6	<i>SCRIPT</i> GUI MATLAB Tombol Refresh dan Tutup 64
Lampiran 7	Hasil Uji Aplikasi 65
Lampiran 8	Daftar Riwayat Hidup 66

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Aljabar adalah cabang dari matematika yang berperan penting dalam mempelajari struktur dan operasi melalui simbol-simbol abstrak. Konsep aljabar tidak hanya bersifat teoretis, tetapi juga menjadi dasar penting dalam berbagai penerapan modern, seperti dalam pemrograman, kecerdasan buatan, dan sistem kriptografi yang mendorong inovasi dalam dunia teknologi (Siregar, 2025). Salah satu bentuk nyata penerapan konsep matematika dalam kehidupan modern adalah pada sistem pengamanan informasi, terutama dalam bentuk kriptografi.

Perkembangan teknologi komunikasi digital yang pesat telah membawa tantangan baru dalam aspek keamanan informasi. Pertukaran data secara digital sehari-hari adalah suatu hal yang biasa, misalnya pengiriman pesan-pesan penting yang mengandung informasi rahasia. Jika tidak diamankan dengan baik, pesan-pesan ini sangat berisiko disalahgunakan oleh pihak yang tidak memiliki wewenang (Nasution dkk., 2018). Oleh karena itu, kriptografi menjadi sangat penting untuk menjaga kerahasiaan data dengan cara mengenkripsi informasi. Dengan demikian, hanya pihak yang memiliki kunci tertentu yang bisa mengakses isi pesan (Prayitno & Nurdin, 2017).

Secara umum, kriptografi diklasifikasikan ke dalam dua kategori utama, yaitu kriptografi kunci simetris dan kriptografi kunci asimetris. Pada sistem kriptografi simetris, proses enkripsi dan dekripsi menggunakan kunci yang sama, sehingga menjaga

kerahasiaan kunci tersebut menjadi hal yang sangat penting. Sebaliknya, kriptografi asimetris menggunakan dua kunci yang berbeda yaitu kunci publik untuk melakukan enkripsi dan kunci privat untuk mendekripsi pesan (Ariyus, 2008; Martolia dkk., 2018). Dalam praktik kriptografi simetris, terdapat beberapa teknik penyandian yang digunakan, antara lain adalah sandi kotak dan *caesar cipher*.

Sandi kotak sendiri dikenal luas dalam dunia kepramukaan sebagai alat latihan komunikasi rahasia. Metode ini menggunakan kotak atau tabel simbol untuk mengganti setiap karakter dalam pesan berdasarkan pola tertentu. Dalam proses enkripsi, karakter teks diganti menjadi simbol sesuai tabel yang telah ditentukan, sedangkan pada proses dekripsi simbol-simbol tersebut diubah kembali ke huruf asli (Abdelmgeid dkk., 2017). Pesan yang disandikan dengan metode ini sangat sulit dikenali oleh pihak luar, kecuali mereka memiliki pengetahuan tentang pola atau bahasa sandi kotak (Martolia dkk., 2018).

Caesar Cipher merupakan contoh kriptografi klasik dengan kunci simetris, dimana proses penyandian dilakukan dengan menggeser huruf-huruf dalam alfabet sebanyak nilai kunci (k) tertentu secara modular untuk mengenkripsi dan dekripsi data (Gurning, 2014). Namun, kelemahan *caesar cipher* adalah sifatnya yang sederhana dan mudah dikenali oleh kriptואנליס (Purnomo & Sembiring, 2022).

Menurut Wrixon (2005), suatu sandi akan sangat efektif apabila konstruksinya tidak diketahui oleh pihak ketiga, meskipun metode yang digunakan sebenarnya sederhana. Oleh karena itu, untuk meningkatkan keamanan pesan, *caesar cipher* dapat dikombinasikan dengan metode penyandian visual seperti sandi

kotak. Sandi kotak menyandikan huruf menjadi bentuk simbol geometris yang tidak langsung dikenali sebagai alfabet, sehingga menambah lapisan penyamaran dalam proses enkripsi.

Penelitian sebelumnya oleh Darmayanti dkk (2018) menunjukkan bahwa *caesar cipher* dapat dimodifikasi menjadi sandi *morse* sebagai bentuk konversi visual dari hasil enkripsi. Namun, belum ditemukan penelitian yang secara khusus mengimplementasikan *caesar cipher* yang dikombinasikan dengan sandi kotak untuk membentuk sistem kriptografi dua lapis.

Oleh karena itu, penelitian kali ini bertujuan untuk mengimplementasikan sandi kotak pada *caesar cipher* untuk meningkatkan keamanan informasi. *Caesar cipher* sebagai metode penyandian berbasis pergeseran huruf dengan sandi kotak sebagai penyamaran visual *ciphertext*, serta mengimplementasikannya dalam bentuk *Graphical User Interface* (GUI) menggunakan perangkat lunak MATLAB.

B. Tujuan Penelitian

Mengimplementasikan sandi kotak pada kriptografi *caesar cipher* untuk proses enkripsi dan dekripsi, dengan menggunakan MATLAB.

C. Tinjauan Pustaka

1. Abdelmgeid dkk. (2017) yang mengembangkan metode *steganografi* citra menggunakan *Pigpen Cipher*. Meskipun konteksnya berbeda karena fokus pada penyembunyian pesan dalam gambar, pendekatan simbolik melalui sandi kotak yang digunakan memiliki kesamaan dengan

penelitian ini dalam hal pemanfaatan bentuk simbol untuk menyamarkan pesan.

2. Selain itu, Darmayanti dkk. (2018) dalam penelitiannya menerapkan *Caesar Cipher* yang dimodifikasi menjadi bentuk sandi *Morse*. Pendekatan dua lapis ini serupa dengan penelitian ini yang juga menggunakan dua metode kriptografi: *caesar cipher* sebagai enkripsi teks dan sandi kotak sebagai bentuk substitusi simbolik. Penelitian tersebut menunjukkan bahwa kombinasi cipher klasik dengan metode simbol dapat meningkatkan keamanan pesan secara signifikan.
3. Purnomo dan Sembiring (2022) menyumbangkan ide tentang modifikasi *caesar cipher* pada karakter *ASCII*, menunjukkan bahwa algoritma klasik ini masih bisa dikembangkan agar sesuai dengan karakter modern, tidak hanya alfabet. Penelitian ini mendasari keyakinan bahwa *caesar cipher* tetap relevan jika dipadukan dengan metode lain, seperti dalam penelitian ini yang menggabungkannya dengan sandi kotak.
4. Penelitian oleh Pratiwi dkk. (2022) juga membahas perancangan sistem keamanan data menggunakan *caesar cipher* Menggunakan perangkat lunak Visual Basic 2010. Mereka menekankan bahwa *caesar cipher* dapat menjadi metode dasar yang efektif untuk penyandian pesan jika diterapkan dengan benar dan disertai mekanisme antarmuka yang baik. Hal ini mendukung desain penelitian ini yang menggunakan GUI MATLAB sebagai media enkripsi-dekripsi yang ramah pengguna.

D. Metodologi Penelitian

Penelitian ini dapat dibagi menjadi beberapa tahap sebagai berikut :

1. Mengkaji Materi Mengenai Kriptografi, *Caesar Cipher* dan Sandi Kotak

Pada tahap ini dilakukan kajian literatur yang mendalam mengenai konsep dasar kriptografi, khususnya algoritma *caesar cipher* sebagai metode pergeseran huruf dan sandi kotak sebagai metode substitusi simbol. Kajian ini menjadi dasar teoretis dalam membangun sistem enkripsi dan dekripsi pada tahap-tahap selanjutnya.

2. Melakukan Enkripsi dan Dekripsi dengan Metode *Caesar Cipher*

Setelah memahami konsep dasar, dilakukan proses enkripsi terhadap *plaintext* menggunakan metode *caesar cipher*. Proses ini menghasilkan *ciphertext* 1, yaitu teks hasil penyandian huruf berdasarkan pergeseran kunci numerik tertentu. *Ciphertext* 1 kemudian diproses lagi lebih lanjut pada tahap selanjutnya.

3. Melakukan Enkripsi dan Dekripsi dengan Metode Sandi Kotak

Ciphertext 1 yang diperoleh dari proses *caesar cipher* kemudian diubah menjadi bentuk simbol visual menggunakan metode sandi kotak. Setiap huruf dari *ciphertext* 1 akan disubstitusikan menjadi simbol sesuai dengan tabel sandi kotak, menghasilkan *ciphertext* akhir dalam bentuk simbol geometris. Tahapan dekripsi dilakukan

secara berbalik, yaitu mengubah simbol ke huruf, lalu menerapkan dekripsi *caesar cipher* untuk mengembalikan ke *plaintext*.

4. Simulasi dan Implementasi Program dalam GUI MATLAB

Pada tahap ini, dilakukan pembuatan program berbasis *Graphical User Interface* (GUI) menggunakan MATLAB sebagai alat bantu komputasi. Aplikasi diuji dengan memasukkan input *plaintext* dan kunci, kemudian diamati apakah hasil enkripsi dan dekripsi sesuai dengan teori yang telah dikaji. Simulasi ini juga digunakan untuk mengidentifikasi kesalahan atau ketidaksesuaian pada proses transformasi teks dan simbol.

E. Sistematika Penulisan

1. Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan penelitian, tinjauan pustaka, metodologi penelitian, dan sistematika penulisan.

2. Bab II Landasan Pustaka

Bab ini memuat teori-teori dasar yang mendukung penelitian, seperti konsep kriptografi, *Caesar Cipher*, Sandi kotak. Selain itu, juga dijelaskan penelitian-penelitian terdahulu yang relevan sebagai pembanding dan penguat landasan pustaka.

3. Bab III Hasil Penelitian dan Pembahasan

Bab ini menyajikan hasil implementasi sistem yang telah dibuat menggunakan GUI MATLAB. Disertakan

pula penjelasan proses enkripsi dan dekripsi dengan implementasi sandi kotak dan *caesar cipher*, tampilan antarmuka aplikasi, dan analisis hasil uji coba aplikasi.

4. Bab IV Simpulan dan Saran

Bab terakhir ini memuat rangkuman simpulan dari hasil penelitian yang telah dilaksanakan, serta sejumlah saran yang dapat dijadikan acuan untuk pengembangan lebih lanjut dalam penelitian selanjutnya.

BAB II

LANDASAN PUSTAKA

A. Kriptografi

Istilah kriptografi berasal dari bahasa Yunani, yang merupakan gabungan dari dua kata, yakni "*kripto*" yang berarti menyembunyikan atau bersifat rahasia, dan "*graphia*" yang berarti tulisan. Dengan demikian, kriptografi dapat diartikan sebagai suatu disiplin ilmu yang berfokus pada penerapan teknik-teknik matematika untuk menjamin keamanan informasi. Aspek-aspek penting yang dilindungi melalui kriptografi meliputi kerahasiaan data, keaslian sumber informasi, integritas konten, serta proses otentikasi dalam pertukaran data (Ariyus, 2006).

Kriptografi merupakan cabang ilmu yang berperan penting dalam menjaga kerahasiaan dan keamanan informasi melalui penerapan metode enkripsi dan dekripsi. Dalam praktiknya, kriptografi mengubah data asli yang dapat dibaca (*plaintext*) menjadi bentuk yang tidak dapat dimengerti (*ciphertext*) melalui proses enkripsi. Transformasi ini memastikan bahwa hanya pihak yang memiliki kunci dekripsi yang tepat yang dapat mengakses dan mengembalikan informasi tersebut ke bentuk aslinya (Prayitno & Nurdin, 2017). Dengan demikian, kriptografi menjadi hal yang penting dalam perlindungan data sensitif dari akses yang tidak memiliki wewenang.

Menurut Ariyus (2008), terdapat beberapa istilah atau terminologi penting dalam kriptografi sebagai berikut:

1) Plaintext dan Ciphertext

Plaintext merujuk pada data atau informasi yang disusun dalam format yang dapat dibaca dan dipahami oleh manusia atau sistem tanpa proses khusus. Agar informasi tersebut tidak dapat diakses oleh pihak yang tidak berwenang, maka dilakukan proses konversi terhadap *plaintext* menjadi bentuk yang tidak dapat dimengerti tanpa kunci tertentu. Hasil dari proses ini dikenal sebagai *ciphertext*, yakni bentuk terenkripsi dari data yang bertujuan melindungi isi pesan dari penyalahgunaan selama proses penyimpanan atau pengiriman.

2) Peserta Komunikasi

Komunikasi data merupakan proses pertukaran pesan antara minimal dua entitas. Entitas pertama bertindak sebagai pengirim pesan, sementara entitas lainnya berperan sebagai penerima. Kedua entitas ini bisa berupa manusia, perangkat mesin seperti komputer, atau lainnya.

3) Enkripsi dan Dekripsi

Proses untuk mengubah pesan asli (*plaintext*) menjadi bentuk *ciphertext* atau pesan yang tidak dapat dibaca tanpa kunci khusus dikenal dengan istilah enkripsi. Sebaliknya, proses untuk mengembalikan pesan dari bentuk (*ciphertext*) kembali ke bentuk pesan aslinya (*plaintext*) disebut dekripsi. Kedua proses ini, enkripsi maupun dekripsi, dapat diterapkan baik pada data yang sedang dikirim maupun yang sedang disimpan. Enkripsi terhadap data yang sedang

dalam perjalanan, atau dikenal sebagai *encryption of data in motion*, merujuk pada perlindungan data yang sedang ditransmisikan melalui jaringan komunikasi. Sementara itu, enkripsi terhadap data yang disimpan, atau *encryption of data at-rest*, mengacu pada pengamanan informasi yang berada dalam media penyimpanan seperti *hard disk*, *server*, atau perangkat digital lainnya.

4) **Kriptanalisis dan Kriptologi**

Perkembangan kriptografi tidak dapat dipisahkan dari keberadaan ilmu yang menjadi tandingannya, yakni kriptanalisis. Kriptanalisis merupakan cabang ilmu sekaligus seni dalam menguraikan *ciphertext* kembali menjadi *plaintext* tanpa bantuan kunci yang tepat. Individu yang menguasai dan menjalankan aktivitas ini dikenal sebagai kriptanalisis. Di sisi lain, seseorang yang mendalami dan mengaplikasikan teknik-teknik kriptografi untuk mengubah pesan asli (*plaintext*) menjadi bentuk (*ciphertext*) dengan menggunakan kunci tertentu disebut sebagai kriptografer. Dengan demikian, kriptologi adalah bidang studi yang lebih luas yang mencakup dua ranah utama tersebut, yakni kriptografi dan kriptanalisis, yang saling berkaitan dalam konteks pengamanan dan pembongkaran informasi.

Menurut Ariyus (2008), kriptografi diklasifikasikan menjadi dua cabang utama berdasarkan jenis kunci yang digunakan dalam proses enkripsi dan dekripsi algoritma. Kedua cabang tersebut adalah:

1) **Kunci Simetris (Algoritma Simetris)**

Kunci simetris, yang juga dikenal sebagai kriptografi kunci privat atau kriptografi kunci rahasia, merupakan metode enkripsi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. Dalam sistem ini, baik pengirim maupun penerima pesan harus memiliki dan menggunakan kunci yang serupa untuk menyandikan serta menguraikan isi pesan. Oleh karena itu, keamanan komunikasi bergantung sepenuhnya pada kerahasiaan kunci tersebut. Sebelum pertukaran pesan berlangsung, kedua pihak harus terlebih dahulu menyepakati dan menjaga kunci secara aman agar tidak jatuh ke tangan pihak yang tidak memiliki wewenang. Contoh algoritma kunci simetris meliputi *Caesar Cipher*, *Vigenere Cipher*, dan *Hill Cipher*, yang dijelaskan sebagai berikut:

- (a) *Caesar Cipher* adalah algoritma enkripsi sederhana yang menggunakan teknik substitusi monoalfabetik. Setiap huruf dalam teks asli (*plaintext*) digeser sejumlah posisi tertentu dalam alfabet berdasarkan kunci.
- (b) *Vigenere Cipher* merupakan algoritma enkripsi polialfabetik yang menggunakan kata kunci untuk menentukan geseran huruf pada setiap posisi teks. Geseran dilakukan berdasarkan tabel *Vigenere*, dengan kata kunci diulang untuk menyesuaikan panjang teks.
- (c) *Hill Cipher* adalah algoritma yang memanfaatkan operasi matriks untuk mengenkripsi dan mendekripsi teks. Teks diubah menjadi vektor angka, lalu dikalikan

dengan matriks kunci persegi (misalnya, 2×2 atau 3×3) dalam aritmatika modulo 26. Proses dekripsi menggunakan matriks invers dari kunci tersebut.

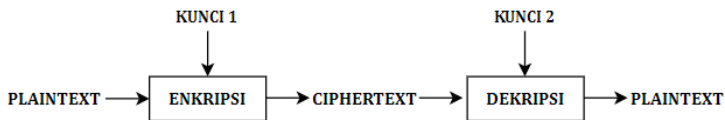
2) Kunci Asimetris (Algoritma Asimetris)

Kunci asimetris, yang juga dikenal sebagai kriptografi kunci publik, merupakan pendekatan kriptografi yang menggunakan sepasang kunci berbeda untuk proses enkripsi dan dekripsi. Berbeda dengan sistem kunci simetris yang hanya menggunakan satu kunci yang sama, metode ini memanfaatkan kunci publik untuk enkripsi pesan dan kunci privat untuk dekripsinya. Kunci publik dapat disebarluaskan secara bebas tanpa mengorbankan keamanan, karena hanya kunci privat yang dapat membuka pesan yang telah dienkripsi. Dalam praktiknya, pengirim cukup mengetahui kunci publik milik penerima untuk menyandikan pesan, sementara hanya penerima yang memiliki kunci privat yang dapat mengakses informasi aslinya. Contoh algoritma kunci asimetris meliputi *RSA*, *Diffie-Hellman*, dan *ElGamal*, yang dijelaskan sebagai berikut:

- (a) *RSA* adalah algoritma yang memanfaatkan sifat matematis bilangan prima besar. Kunci publik dihasilkan dari perkalian dua bilangan prima, sedangkan kunci privat didasarkan pada invers modular.
- (b) *Diffie-Hellman* adalah algoritma pertukaran kunci berdasarkan masalah logaritma diskret, yang memungkinkan dua pihak menghasilkan kunci simetris bersama melalui saluran yang tidak aman. Algoritma

ini memungkinkan negosiasi kunci yang aman, seperti dalam protokol komunikasi TLS/SSL.

- (c) *ElGamal* adalah algoritma yang juga berbasis masalah logaritma diskret. Enkripsi dengan *ElGamal* menghasilkan *ciphertext* yang berbeda untuk setiap eksekusi meskipun teks asli sama.



Gambar 2.1. Proses Enkripsi dan Dekripsi Pesan

Jika kunci pertama identik atau sama dengan kunci kedua, maka sistem tersebut dikategorikan sebagai kriptografi kunci simetris. Namun, jika kunci pertama berbeda dari kunci kedua, maka sistem tersebut termasuk dalam kriptografi kunci asimetris (Munir, 2010).

Dalam penelitian ini, algoritma kunci simetris yaitu *caesar cipher* dipilih sebagai lapisan pertama enkripsi sebelum penerapan sandi kotak 1. Pemilihan ini didasarkan pada kesederhanaan algoritma yang hanya menggeser huruf dalam alfabet berdasarkan kunci tertentu, sehingga mudah diimplementasikan dan mendukung tujuan edukatif (Ariyus, 2008). Fokus penggunaan *caesar cipher* adalah untuk merepresentasikan prinsip kriptografi kunci simetris dan mendukung visualisasi proses enkripsi-dekripsi melalui antarmuka grafis (GUI) berbasis MATLAB.

B. Caesar Cipher

Caesar cipher merupakan salah satu algoritma yang tergolong dalam kriptografi kunci simetris dan telah digunakan sejak lama, bahkan sebelum ditemukannya sistem kriptografi kunci publik. Metode penyandian ini dinamakan kode kaisar atau bisa disebut *caesar cipher*, setelah digunakan oleh seorang jendral romawi kuno yaitu Julius Caesar untuk berkomunikasi dengan para bawahannya agar tidak mudah dibaca oleh musuh (Ariyus, 2008).

Caesar cipher, yang juga dikenal dengan sebutan sandi geser, adalah salah satu teknik enkripsi paling sederhana dan populer. Metode ini bekerja dengan cara mengganti setiap huruf dalam teks asli (*plaintext*) dengan huruf lain yang berada pada posisi tertentu yang telah digeser beberapa langkah dalam urutan alfabet. Pada *caesar cipher*, setiap huruf digantikan oleh huruf yang berada beberapa posisi setelahnya dalam urutan alfabet yang sama, sesuai dengan kunci yang ditetapkan (Seftyanto dkk., 2012).

Aritmetika modular memainkan peranan yang penting dalam operasi pada bilangan bulat, khususnya pada *caesar cipher*. Operator yang digunakan pada aritmetika modular adalah mod yang berfungsi untuk menghasilkan sisa dari suatu pembagian, yang memungkinkan perhitungan dalam rentang terbatas. Misalnya 23 dibagi 5 memberikan hasil yaitu 4 dan sisa yaitu 3, sehingga dapat ditulis $23 \bmod 5 = 3$ (Munir, 2010).

Definisi 2.1 Misalkan a adalah bilangan bulat dan m adalah bilangan bulat positif > 0 . Operasi $a \bmod m$ (dibaca " a modulo m ") menghasilkan sisa dari pembagian a oleh m . Dengan demikian, $a \bmod m = r$ sehingga $a = mq + r$, dimana $0 \leq r < m$, dengan r adalah sisa pembagian dan q adalah hasil bagi (Munir, 2010).

Dalam algoritma *caesar cipher*, setiap huruf alfabet dikonversi dalam bentuk *integer* seperti berikut.

Tabel 2.1. Mengkonversi huruf ke dalam *integer*

A=0	B=1	C=2	D=3	E=4	F=5	G=6
H=7	I=8	J=9	K=10	L=11	M=12	N=13
O=14	P=15	Q=16	R=17	S=18	T=19	U=20
V=21	W=22	X=23	Y=24	Z=25		

Huruf-huruf ini digeser sebanyak kunci K menggunakan operasi modulo 26 untuk menjaga hasil dalam rentang alfabet.

Proses dalam algoritma *caesar cipher* diklasifikasikan menjadi dua tahapan utama berdasarkan fungsinya dalam pengamanan pesan. diantaranya sebagai berikut:

1. Proses Enkripsi

Algoritma 2.1

Input:

1. *Plaintext* harus berupa alfabet A-Z
2. Kunci rahasia K harus berupa bilangan bulat

Output: Ciphertext

Langkah-langkah:

1. Konversi Setiap huruf *plaintext* ke dalam bentuk *integer* sesuai dengan Tabel 2.1.
2. Setiap *integer* tersebut, digeser sejauh kunci K posisi ke kanan dalam alfabet untuk menghasilkan *ciphertext*.

3. Rumus Enkripsi:

$$C = E(P) = (P + K) \bmod 26 \quad (2.1)$$

4. Di sini, C adalah karakter *ciphertext*, $E(P)$ adalah proses enkripsi *plaintext*, P adalah karakter *plaintext*, dan K adalah kunci rahasia berupa bilangan bulat.

Contoh 2.1

Plaintext : AKU INGIN MEMBICARAKAN SESUATU

Kunci : 10

Langkah Pertama : Konversi setiap huruf pada *plaintext* ke dalam bentuk *integer* sesuai Tabel 2.1

Langkah Kedua : Lakukan proses enkripsi terhadap *plaintext* yang telah dikonversi ke dalam bentuk *integer*, dengan menggunakan persamaan (2.1).

P	<i>Integer</i>	Enkripsi	Hasil	C
A	0	$(0 + 10) = 10 \bmod 26 = 10$	10	K
K	10	$(10 + 10) = 20 \bmod 26 = 20$	20	U
U	20	$(20 + 10) = 30 \bmod 26 = 4$	4	E
I	8	$(8 + 10) = 18 \bmod 26 = 18$	18	S
N	13	$(13 + 10) = 23 \bmod 26 = 23$	23	X
G	6	$(6 + 10) = 16 \bmod 26 = 16$	16	Q
I	8	$(8 + 10) = 18 \bmod 26 = 18$	18	S
N	13	$(13 + 10) = 23 \bmod 26 = 23$	23	X
M	12	$(12 + 10) = 22 \bmod 26 = 22$	22	W
E	4	$(4 + 10) = 14 \bmod 26 = 14$	14	O

M	12	$(12 + 10) = 22 \mod 26 = 22$	22	W
B	1	$(1 + 10) = 11 \mod 26 = 11$	11	L
I	8	$(8 + 10) = 18 \mod 26 = 18$	18	S
C	2	$(2 + 10) = 12 \mod 26 = 12$	12	M
A	0	$(0 + 10) = 10 \mod 26 = 10$	10	K
R	17	$(17 + 10) = 27 \mod 26 = 1$	1	B
A	0	$(0 + 10) = 10 \mod 26 = 10$	10	K
K	10	$(10 + 10) = 20 \mod 26 = 20$	20	U
A	0	$(0 + 10) = 10 \mod 26 = 10$	10	K
N	13	$(13 + 10) = 23 \mod 26 = 23$	23	X
S	18	$(18 + 10) = 28 \mod 26 = 2$	2	C
E	4	$(4 + 10) = 14 \mod 26 = 14$	14	O
S	18	$(18 + 10) = 28 \mod 26 = 2$	2	C
U	20	$(20 + 10) = 30 \mod 26 = 4$	4	E
A	0	$(0 + 10) = 10 \mod 26 = 10$	10	K
T	19	$(19 + 10) = 29 \mod 26 = 3$	3	D
U	20	$(20 + 10) = 30 \mod 26 = 4$	4	E

Jadi, *Ciphertext*nya yaitu KUE SXQ SX WOWLSMKBKUKX
COCEKDE

2. Proses Dekripsi

Algoritma 2.2

Input:

1. *Ciphertext* harus berupa alfabet A–Z
2. Kunci K yang sama seperti pada proses enkripsi

Output: Plaintext

Langkah-langkah:

1. Konversi Setiap huruf *ciphertext* menjadi bentuk *integer* sesuai dengan Tabel 2.1.
2. Setiap *integer* tersebut, digeser sejauh kunci K posisi ke kiri dalam alfabet untuk menghasilkan *plaintext*.
3. Rumus Dekripsi:

$$P = D(C) = (C - K) \bmod 26 \quad (2.2)$$

4. Di sini, P adalah karakter *plaintext*, $D(C)$ adalah proses dekripsi *ciphertext*, C adalah karakter *ciphertext*, dan K adalah kunci rahasia berupa bilangan bulat.

Contoh 2.2

Misalnya, *ciphertext* yang akan didekripsi yaitu hasil enkripsi dari contoh sebelumnya menggunakan jumlah kunci yang serupa.

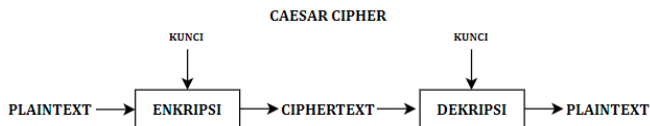
Langkah Pertama : Konversi setiap huruf pada *ciphertext* ke dalam bentuk *integer* sesuai Tabel 2.1

Langkah Kedua : Lakukan proses dekripsi terhadap *ciphertext* yang telah dikonversi ke dalam bentuk *integer*, dengan menggunakan persamaan (2.2).

C	<i>Integer</i>	Dekripsi	Hasil	P
K	10	$(10 - 10) = 0 \bmod 26 = 0$	0	A
U	20	$(20 - 10) = 10 \bmod 26 = 10$	10	K
E	4	$(4 - 10) = -6 \bmod 26 = 20$	20	U

S	18	$(18 - 10) = 8 \pmod{26} = 8$	8	I
X	23	$(23 - 10) = 13 \pmod{26} = 13$	13	N
Q	16	$(16 - 10) = 6 \pmod{26} = 6$	6	G
S	18	$(18 - 10) = 8 \pmod{26} = 8$	8	I
X	23	$(23 - 10) = 13 \pmod{26} = 13$	13	N
W	22	$(22 - 10) = 12 \pmod{26} = 12$	12	M
O	14	$(14 - 10) = 4 \pmod{26} = 4$	4	E
W	22	$(22 - 10) = 12 \pmod{26} = 12$	12	M
L	11	$(11 - 10) = 1 \pmod{26} = 1$	1	B
S	18	$(18 - 10) = 8 \pmod{26} = 8$	8	I
M	12	$(12 - 10) = 2 \pmod{26} = 2$	2	C
K	10	$(10 - 10) = 0 \pmod{26} = 0$	0	A
B	1	$(1 - 10) = -9 \pmod{26} = 17$	17	R
K	10	$(10 - 10) = 0 \pmod{26} = 0$	0	A
U	20	$(20 - 10) = 10 \pmod{26} = 10$	10	K
K	10	$(10 - 10) = 0 \pmod{26} = 0$	0	A
X	23	$(23 - 10) = 13 \pmod{26} = 13$	13	N
C	2	$(2 - 10) = -8 \pmod{26} = 18$	18	S
O	14	$(14 - 10) = 4 \pmod{26} = 4$	4	E
C	2	$(2 - 10) = -8 \pmod{26} = 18$	18	S
E	4	$(4 - 10) = -6 \pmod{26} = 20$	20	U
K	10	$(10 - 10) = 0 \pmod{26} = 0$	0	A
D	3	$(3 - 10) = -7 \pmod{26} = 19$	19	T
E	4	$(4 - 10) = -6 \pmod{26} = 20$	20	U

Jadi, *Plaintext* yang ingin disampaikan yaitu AKU INGIN
MEMBICARAKAN SESUATU



Gambar 2.2. Proses Enkripsi dan Dekripsi pada *Caesar cipher*

Caesar cipher merupakan salah satu bentuk kriptografi klasik yang masih menggunakan metode penyandian sederhana. Karena kesederhanaannya, hasil enkripsi yang dihasilkan menjadi rentan terhadap berbagai upaya pemecahan atau serangan dari pihak yang tidak berwenang, sehingga perlu adanya pengembangan atau modifikasi terhadap metode penyandian *caesar cipher* guna meningkatkan tingkat keamanannya dan mencegah agar tidak mudah dipecahkan oleh kriptanalisis (Azhar dkk., 2024).

C. Sandi Kotak

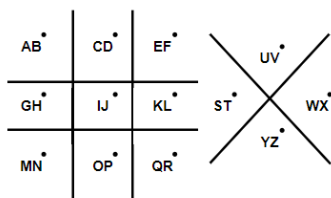
Sandi kotak merupakan salah satu metode kriptografi klasik yang termasuk dalam kategori *cipher* substitusi. Berbeda dengan *cipher* substitusi konvensional yang mengganti huruf dengan huruf lain, sandi kotak mengganti huruf-huruf dalam alfabet dengan simbol-simbol geometris yang diatur dalam pola kotak atau grid (Abdelmgeid dkk., 2017).

Menurut Zubaidi (2018), beberapa jenis sandi kotak diantaranya sebagai berikut:

a. Sandi Kotak 1

Sandi kotak ini disusun dalam kotak yang diatur secara vertikal, horizontal, dan menyilang. Kotak tersebut terbagi menjadi sembilan bagian, masing-masing berisi huruf dari A

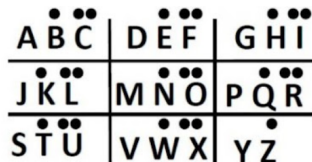
hingga R yang disusun secara vertikal dan horizontal dengan urutan: 1 AB, 2 CD, 3 EF, 4 GH, 5 IJ, 6 KL, 7 MN, 8 OP, dan 9 QR. Selain itu, kombinasi huruf ST, UV, WX, dan YZ ditempatkan pada kotak yang diatur secara menyilang. Setiap kotak vertikal dan horizontal berisi dua huruf, dengan tanda titik digunakan untuk menunjukkan huruf kedua. Misalnya, pada kotak AB, huruf B ditandai dengan titik. Seperti yang terlihat pada Gambar 2.3.



Gambar 2.3. Sandi Kotak 1

b. Sandi Kotak 2

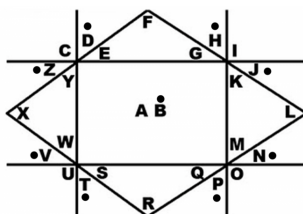
Sandi kotak ini memiliki kotak yang disusun secara vertikal dan horizontal, terdiri dari 9 kolom, di mana setiap kolom umumnya berisi tiga huruf. Namun, pada kolom terakhir terdapat pengecualian, yaitu huruf Y dan Z ditempatkan dalam satu kotak yang sama dan hanya terdiri dari dua huruf. Seperti yang terlihat pada Gambar 2.4.



Gambar 2.4. Sandi Kotak 2

c. Sandi Kotak 3

Merupakan jenis sandi yang menggabungkan bentuk kotak vertikal-horizontal dengan kotak berbentuk persegi empat menyerupai belah ketupat, di mana huruf-huruf alfabet tersebar di dalamnya, dan sebagian dilengkapi dengan titik-titik sebagai penanda. Seperti yang terlihat pada Gambar 2.5.



Gambar 2.5. Sandi Kotak 3

Dalam penelitian ini, digunakan sandi kotak 1 sebagai lapisan kedua setelah enkripsi *caesar cipher*. Pemilihan ini didasarkan pada kesederhanaan struktur dan kesesuaiannya untuk tujuan edukatif dalam pengembangan aplikasi GUI MATLAB. Fokus penelitian tertuju pada Sandi Kotak 1 karena bentuknya yang sederhana memudahkan visualisasi grafis dan implementasi algoritma.

Sandi kotak 1 merupakan salah satu teknik penyandian yang umum digunakan dalam kegiatan pramuka sebagai sarana untuk menyampaikan pesan-pesan yang bersifat rahasia. Teknik ini menggunakan pola geometris yang menyerupai kotak atau grid, setiap huruf ditempatkan dalam segmen tertentu dari grid, dan untuk menulis pesan terenkripsi, huruf-huruf tersebut diganti dengan simbol atau tanda yang sesuai dengan posisi mereka dalam grid. Huruf pertama dalam grid dienkripsi dengan

bentuk garis di sekeliling huruf, sedangkan huruf kedua dalam grid dienkripsi dengan bentuk garis di sekeliling huruf diikuti dengan titik (Martolia dkk., 2018). Sandi kotak tidak hanya berfungsi untuk mengamankan pesan, tetapi juga memberikan unsur tantangan dan keseruan dalam pramuka, serta melatih keterampilan memecahkan kode dan kerja sama tim.

Proses dalam algoritma sandi kotak diklasifikasikan menjadi dua tahapan utama berdasarkan fungsinya dalam pengamanan pesan. diantaranya sebagai berikut:

1. Proses Enkripsi

Algoritma 2.3

Input: Plaintext harus berupa alfabet A-Z

Output: Ciphertext dalam bentuk simbol sandi kotak

Langkah-langkah:

1. Membuat tabel kotak berbentuk grid-grid yang berisi huruf dari A-Z sesuai dengan Gambar 2.3.
2. Setiap huruf *plaintext* akan dikonversi menjadi *ciphertext* berupa simbol atau karakter yang sesuai dengan posisi huruf dalam grid sandi kotak.
3. Huruf pertama pada grid dikonversi menjadi simbol garis kotak tanpa titik, sedangkan huruf kedua dikonversi menjadi simbol yang sama tetapi dengan tambahan titik.
4. Rumus Enkripsi:


$$C = E(P) = S(P) \quad (2.3)$$


5. Di sini, C adalah karakter *ciphertext*, $E(P)$ adalah proses enkripsi *plaintext*, $S(P)$ adalah fungsi substitusi *plaintext* berdasarkan tabel sandi kotak.
6. *Ciphertext* adalah deretan simbol sandi kotak.


Contoh 2.3


Plaintext : TUNGGU AKU


Menggunakan persamaan (2.3) setiap *Plaintext* dikonversi menjadi simbol sandi kotak yang sesuai pada Gambar 2.3.


T \Rightarrow 


U \Rightarrow 


N \Rightarrow 


G \Rightarrow 

G \Rightarrow 

U \Rightarrow 

A \Rightarrow 

K \Rightarrow 

U \Rightarrow 

Jadi, *ciphertext* untuk pesan tersebut adalah



2. Proses Dekripsi

Algoritma 2.4

Input: *Ciphertext* dalam bentuk simbol sandi kotak

Output: *Plaintext* berupa huruf alfabet A–Z

Langkah-langkah:

1. Membuat tabel kotak berbentuk grid-grid yang berisi huruf dari A- Z sesuai dengan Gambar 2.3.
2. Setiap simbol atau karakter pada *ciphertext* dikonversi menjadi *plaintext* yang sesuai huruf pada tabel sandi kotak.
3. Simbol yang berupa garis kotak tanpa titik merepresentasikan huruf pertama dari suatu posisi grid. Simbol yang sama dengan tambahan titik merepresentasikan huruf kedua dari posisi tersebut.
4. Rumus Dekripsi:

$$P = D(C) = S^{-1}(C) \quad (2.4)$$

5. Di sini, P adalah karakter *plaintext*, $D(C)$ adalah proses Dekripsi *ciphertext*, $S^{-1}(C)$ adalah fungsi invers dari substitusi *ciphertext* berdasarkan tabel sandi kotak.
6. Gabungkan semua huruf hasil konversi untuk membentuk kembali *plaintext*.

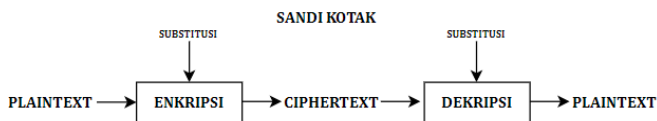
Contoh 2.4

Misalnya, *ciphertext* yang akan didekripsi yaitu hasil enkripsi dari contoh sebelumnya menggunakan jumlah kunci yang serupa.

Menggunakan persamaan (2.4) setiap simbol Sandi Kotak (*ciphertext*) akan dikonversi kembali menjadi (*plaintext*).

> ⇒ T
 V ⇒ U
 7 ⇒ N
 3 ⇒ G
 3 ⇒ G
 V ⇒ U
 7 ⇒ A
 3 ⇒ K
 V ⇒ U

Jadi, *Plaintext* yang ingin disampaikan yaitu TUNGGU AKU



Gambar 2.6. Proses Enkripsi dan Dekripsi pada Sandi Kotak

Meskipun pesan yang disandikan menggunakan metode ini sangat sulit dikenali apa yang tertulis dalam pesan, terutama bagi pihak luar yang tidak memiliki pemahaman terhadap simbol-simbolnya, sandi ini tetap dapat dipecahkan apabila pihak tersebut memiliki pengetahuan tentang bahasa atau pola yang digunakan dalam sandi kotak (Martolia dkk., 2018).

D. Hasil Penelitian yang Relevan

1. Penelitian yang dilakukan oleh Abdelmgeid, A. A., Bahgat, A. A., Seddik, A. H., & Mohamed, M. (2017) dalam *International Journal of Computer Applications* Volume 174 – No.9 yang berjudul "*Enhancing Image Steganography Methods by using New Secret Message Encoding Technique based on Pigpen Cipher (Pigpen Encoding)*". Penelitian ini mengusulkan metode pengamanan pesan dengan menyandikan teks menjadi simbol sandi kotak sebelum disisipkan ke dalam citra digital (*image steganography*). Teknik ini meningkatkan keamanan dengan memanfaatkan bentuk visual yang sulit dikenali langsung oleh pembaca biasa. Penelitian kali ini relevan karena sama-sama menggunakan sandi kotak sebagai bentuk penyamaran pesan, namun berbeda dalam media dan penerapannya. Penelitian Abdelmgeid fokus pada penyisipan ke dalam gambar, sedangkan penelitian ini mengimplementasikan sandi kotak dengan *caesar cipher* dalam bentuk teks untuk proses enkripsi dan dekripsinya.
2. Penelitian yang dilakukan oleh Darmayanti, I., Astrida, D. N., & Ariyus, D. (2018) dalam *Jurnal Ilmiah IT CIDA* Vol. 4 No. 1 yang berjudul "Penerapan Keamanan Pesan Teks Menggunakan Modifikasi Algoritma *Caisar Chiper* Ke dalam Bentuk Sandi *Morse*", mengusulkan modifikasi terhadap algoritma *caesar cipher* dengan mengkonversi hasil *ciphertext* menjadi bentuk kode *morse*. Penelitian ini memodifikasi algoritma *caesar cipher* menjadi bentuk hexadecimal yang kemudian hasil dari *ciphertext* adalah berupa sandi morse. Sedangkan penelitian kali ini sandi

morse digantikan oleh sandi kotak untuk menghasilkan simbol visual yang lebih kompleks dan menarik secara grafis. *Caesar cipher* akan dimodifikasi dengan cara, pada tahap enkripsi *ciphertext* akan dikonversi ke dalam simbol sandi kotak, lalu pada tahap dekripsi simbol sandi kotak akan dikonversi kembali ke dalam bentuk huruf.

3. Penelitian oleh Pratiwi, R., Utami, L. C., & Sakti, R. B. (2022) dalam *Bulletin of Information Technology (BIT)* Vol. 3 No. 4 yang berjudul “Perancangan Keamanan Data Pesan Dengan Menggunakan Metode Kriptografi *Caesar Cipher*”, membahas perancangan aplikasi untuk keamanan pesan menggunakan metode *caesar cipher*. Penelitian ini menggunakan perangkat lunak Visual Basic 2010 untuk merancang aplikasi keamanan pesan yang menggunakan proses enkripsi dan dekripsi serta pergeseran kunci. Sedangkan penelitian kali ini akan digunakan kombinasi dua metode kriptografi yaitu *caesar cipher* sebagai tahap penyandian dengan metode berbasis pergeseran huruf, sedangkan sandi kotak menyamarkan hurufnya menjadi simbol-simbol yang tidak langsung dikenali sebagai alfabet dan pada penelitian ini juga menggunakan GUI MATLAB sebagai alat bantu komputasinya.

BAB III

HASIL PENELITIAN DAN PEMBAHASAN

A. IMPLEMENTASI SANDI KOTAK PADA KRIPTOGRAFI CAESAR CIPHER UNTUK PROSES ENKRIPSI DAN DEKRIPSI

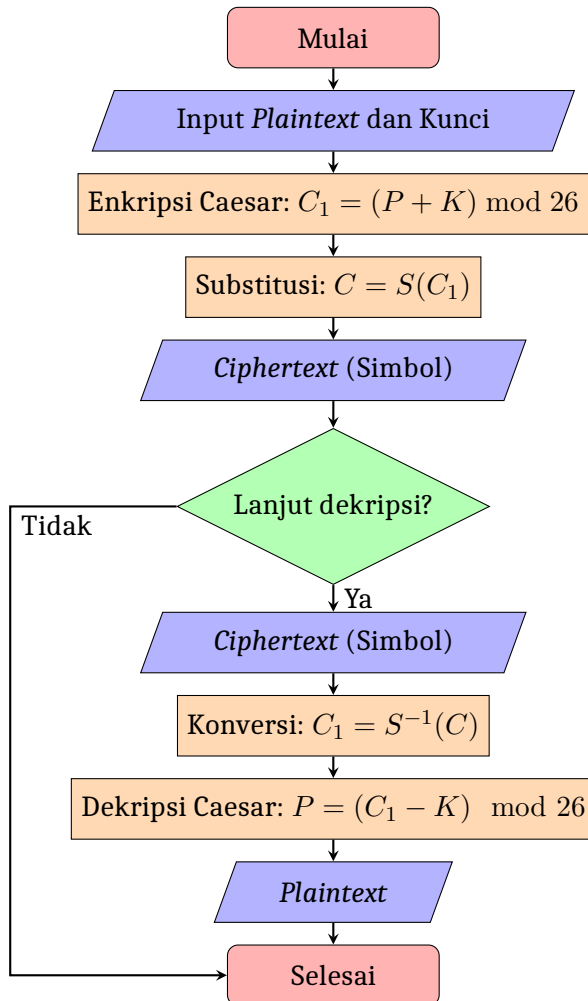
Penelitian ini menggunakan perangkat lunak MATLAB dengan antarmuka grafis (GUI) sebagai alat bantu komputasi. Tujuan dari penggunaan GUI MATLAB dalam penelitian ini adalah untuk memvisualisasikan dan menguji implementasi gabungan dua teknik kriptografi klasik, yaitu *caesar cipher* dan sandi kotak. Kombinasi kedua metode ini dimaksudkan untuk meningkatkan keamanan pesan melalui dua tahap pengkodean: pertama menggunakan metode *caesar cipher* berupa pergeseran huruf, dan yang kedua menggunakan simbol grafis (sandi kotak). Dengan menggunakan GUI MATLAB, pengguna dapat dengan mudah memahami proses pesan enkripsi dan dekripsi secara interaktif. Secara umum, proses enkripsi terdiri atas dua tahap utama:

1. Enkripsi *caesar cipher*, yaitu menggeser setiap huruf dari *plaintext* menggunakan kunci tertentu yang telah disepakati antara pengirim dan penerima pesan.
2. Konversi hasil *caesar cipher* menjadi simbol-simbol menggunakan metode sandi kotak.

Sementara itu, proses dekripsi juga dilakukan dalam dua tahap, dengan urutan yang berlawanan:

1. Simbol sandi kotak dikonversi kembali menjadi huruf sesuai tabel sandi kotak.

2. Dekripsi *caesar cipher* dilakukan terhadap hasil huruf tersebut menggunakan kunci yang serupa seperti pada proses enkripsi.



Gambar 3.1. Flowchart Sandi Kotak pada Kriptografi *Caesar Cipher*

Pada Gambar 3.1 mula-mula, pesan (*plaintext*) terlebih dahulu melalui proses enkripsi dengan memanfaatkan metode kriptografi *caesar cipher*, sehingga menghasilkan *ciphertext* 1 berupa deretan huruf yang telah digeser sesuai kunci. Selanjutnya, *ciphertext* 1 diproses kembali melalui metode sandi kotak untuk dikonversi menjadi bentuk simbol-simbol visual. Hasil akhir dari proses ini disebut sebagai *ciphertext*, yaitu pesan dalam bentuk simbol yang siap dikirimkan kepada penerima.

Untuk memperoleh kembali pesan asli, penerima harus melakukan proses dekripsi secara berurutan dan tepat, yaitu dimulai dengan mengkonversi simbol sandi kotak menjadi huruf melalui proses dekripsi sandi kotak, kemudian dilanjutkan dengan dekripsi *caesar cipher* menggunakan kunci yang serupa seperti saat enkripsi. Hasil akhir dari proses ini adalah *plaintext* yang merupakan pesan asli sebelum dienkripsi.

Proses implementasi sandi kotak pada kriptografi *caesar cipher* diklasifikasikan menjadi dua tahapan utama berdasarkan fungsinya dalam pengamanan pesan. diantaranya sebagai berikut:

1. Proses Enkripsi

Proses enkripsi pesan dengan implementasi sandi kotak pada kriptografi *caesar cipher* dinyatakan secara matematis sebagai berikut:

$$C = S(E(P, K))$$

Dengan:

- a. P adalah *plaintext*
- b. K adalah kunci untuk *caesar cipher*

- c. $E(P, K) = (P + K) \bmod 26$ adalah fungsi enkripsi *caesar cipher*.
- d. S adalah substitusi ke simbol sandi kotak.
- e. C adalah hasil akhir enkripsi dalam bentuk simbol sandi kotak.

Algoritma 3.1

Input:

1. *Plaintext* harus berupa alfabet A-Z
2. Kunci rahasia K harus berupa bilangan bulat

Output: *Ciphertext* dalam bentuk simbol sandi kotak

Langkah-langkah:

1. Konversi setiap huruf *plaintext* ke bentuk *integer*.
2. Geser setiap *integer* sebanyak kunci K ke kanan alfabet.
3. Lakukan enkripsi *caesar cipher*:

$$C_1 = (P + K) \bmod 26 \quad (3.1)$$

yaitu C_1 adalah hasil enkripsi *caesar* dalam bentuk huruf.

4. Konversi setiap huruf C_1 ke simbol menggunakan fungsi substitusi sandi kotak:

$$C = S(C_1). \quad (3.2)$$

5. Hasil akhir *ciphertext* C , berupa deretan simbol sandi kotak.

Contoh 3.1

Input *plaintext* : UIN WALISONGO

Kunci *caesar cipher* : 3

Langkah Pertama : Konversi setiap huruf pada *plaintext* menjadi bentuk *integer* sesuai Tabel 2.1.

Langkah Kedua : Lakukan proses enkripsi terhadap *plaintext* yang telah dikonversi ke dalam bentuk *integer*, dengan menggunakan persamaan (3.1).

Hasil *caesar cipher* :

<i>P</i>	<i>Integer</i>	Enkripsi	Hasil	<i>C</i>
U	20	$(20 + 3) = 23 \mod 26 = 23$	23	X
I	8	$(8 + 3) = 11 \mod 26 = 11$	11	L
N	13	$(13 + 3) = 16 \mod 26 = 16$	16	Q
W	22	$(22 + 3) = 25 \mod 26 = 25$	25	Z
A	0	$(0 + 3) = 3 \mod 26 = 3$	3	D
L	11	$(11 + 3) = 14 \mod 26 = 14$	14	O
I	8	$(8 + 3) = 11 \mod 26 = 11$	11	L
S	18	$(18 + 3) = 21 \mod 26 = 21$	21	V
O	14	$(14 + 3) = 17 \mod 26 = 17$	17	R
N	13	$(13 + 3) = 16 \mod 26 = 16$	16	Q
G	6	$(6 + 3) = 9 \mod 26 = 9$	9	J
O	14	$(14 + 3) = 17 \mod 26 = 17$	17	R

Jadi, *ciphertext* C_1 dari kata UIN WALISONGO adalah XLQ ZDOLVRQJR.

Langkah Ketiga : Setiap huruf *ciphertext* C_1 dikonversi menjadi simbol sandi kotak, sesuai dengan Persamaan (3.2).

Hasil akhir (sandi kotak):

X \Rightarrow 

L \Rightarrow 

Q \Rightarrow 

Z \Rightarrow 

D \Rightarrow 

O \Rightarrow 

L \Rightarrow 

V \Rightarrow 

R \Rightarrow 

Q \Rightarrow 

J \Rightarrow 

R \Rightarrow 

Jadi, *ciphertext* akhir untuk pesan tersebut adalah



2. Proses Dekripsi

Proses dekripsi dituliskan secara matematis sebagai:

$$P = D(S^{-1}(C), K)$$

Dengan:

- C adalah *ciphertext* berupa simbol sandi kotak.
- $S^{-1}(C)$ adalah fungsi invers dari substitusi sandi kotak.

- c. $D(C, K) = (C - K) \bmod 26$ adalah fungsi dekripsi *caesar cipher*.
- d. P adalah *plaintext* hasil dekripsi.

Algoritma 3.2

Input:

1. *Ciphertext* dalam bentuk simbol sandi kotak
2. Kunci K yang sama seperti pada proses enkripsi

Output: *Plaintext* berupa huruf alfabet A-Z

Langkah-langkah:

1. Konversi setiap simbol sandi kotak ke huruf menggunakan fungsi invers substitusi sandi kotak:

$$C_1 = S^{-1}(C) \quad (3.3)$$

2. Konversi setiap huruf pada *ciphertext* 1 ke bentuk *integer*.
3. Geser setiap *integer* sebanyak kunci K ke kiri dalam alfabet.
4. Lakukan dekripsi *caesar cipher* terhadap setiap huruf C_1 menggunakan kunci yang sama seperti saat enkripsi:

$$P = (C_1 - K) \bmod 26 \quad (3.4)$$

5. Gabungkan semua huruf untuk mendapatkan *plaintext*.

Contoh 3.2

Misalnya, *ciphertext* yang didekripsi yaitu hasil enkripsi dari contoh sebelumnya menggunakan jumlah kunci yang serupa.

Input Simbol Sandi Kotak :

◁◻ ▮ ▴◻ ▮◻ ▽◻ ▮◻ ▮◻ ▮◻

Kunci *caesar cipher* : 3

Langkah Pertama : Setiap simbol (*ciphertext*) akan disubstitusikan kembali menjadi huruf alfabet, yang masih berupa pesan acak (C_1), sesuai dengan Persamaan (3.3).

Hasil Caesar Cipher :

◁◻ ⇒ X

◻◻ ⇒ L

▮ ⇒ Q

▴◻ ⇒ Z

◻◻ ⇒ D

◻ ⇒ O

◻◻ ⇒ L

▽◻ ⇒ V

▮◻ ⇒ R

▮ ⇒ Q

◻◻ ⇒ J

▮◻ ⇒ R

Jadi, hasil substitusi setiap simbol sandi kotak menghasilkan pesan XLQ ZDOLVRQJR.

Langkah Kedua : Konversi setiap huruf pada *ciphertext* 1 menjadi bentuk *integer* sesuai Tabel 2.1.

Langkah Ketiga : Lakukan proses Dekripsi terhadap *ciphertext* 1 yang telah dikonversi ke dalam bentuk *integer*, dengan menggunakan persamaan (3.4).

Hasil Akhir (Pesan Asli) :

<i>C</i>	<i>Integer</i>	Dekripsi	Hasil	<i>P</i>
X	23	$(23 - 3) = 20 \mod 26 = 20$	20	U
L	11	$(11 - 3) = 8 \mod 26 = 8$	8	I
Q	16	$(16 - 3) = 13 \mod 26 = 13$	13	N
Z	25	$(25 - 3) = 22 \mod 26 = 22$	22	W
D	3	$(3 - 3) = 0 \mod 26 = 0$	0	A
O	14	$(14 - 3) = 11 \mod 26 = 11$	11	L
L	11	$(11 - 3) = 8 \mod 26 = 8$	8	I
V	21	$(21 - 3) = 18 \mod 26 = 18$	18	S
R	17	$(17 - 3) = 14 \mod 26 = 14$	14	O
Q	16	$(16 - 3) = 13 \mod 26 = 13$	13	N
J	9	$(9 - 3) = 6 \mod 26 = 6$	6	G
R	17	$(17 - 3) = 14 \mod 26 = 14$	14	O

Jadi, *plaintext* dari kata XLQ ZDOLVRQJR adalah UIN WALISONGO

B. SIMULASI APLIKASI

1. Desain Aplikasi

Implementasi algoritma sandi kotak pada kriptografi *caesar cipher* menggunakan MATLAB dengan GUI (*Graphical User Interface*). GUI dirancang menggunakan fitur GUIDE (*Graphical User Interface Development Environment*) yang tersedia dalam MATLAB, sehingga pengguna tidak perlu mengetikkan perintah secara manual untuk menjalankan proses kriptografi.

Tampilan menu yang akan dirancang pada aplikasi ini meliputi :

- a. Teks input; menu ini berisi pesan yang ingin dienkripsi atau didekripsikan.
- b. Kunci; menu ini berisikan kunci untuk *caesar cipher* dan kunci harus berupa bilangan bulat.
- c. Hasil caesar; menu ini akan menampilkan hasil dari enkripsi atau dekripsi *caesar cipher*.
- d. Hasil sandi kotak; menu ini akan mengkonversi hasil *caesar cipher* menjadi bentuk simbol sandi kotak.
- e. Enkripsi; *button* ini berfungsi untuk menjalankan proses enkripsi pesan pada program.
- f. Dekripsi; *button* ini berfungsi untuk menjalankan proses dekripsi pesan pada program.
- g. Refresh; *button* ini berfungsi untuk menghapus data input beserta hasil yang telah ditampilkan oleh aplikasi.
- h. Tutup; *button* ini berfungsi untuk menutup atau keluar dari aplikasi.

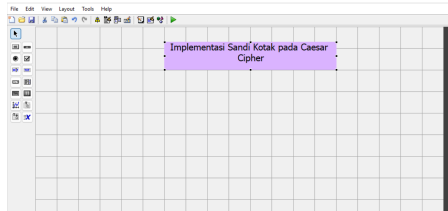
Desain antarmuka aplikasi ini ditampilkan sebagai berikut:

Gambar 3.2. Desain Antarmuka *Interface* Aplikasi

Tabel 3.3. Komponen & Fungsi GUI MATLAB

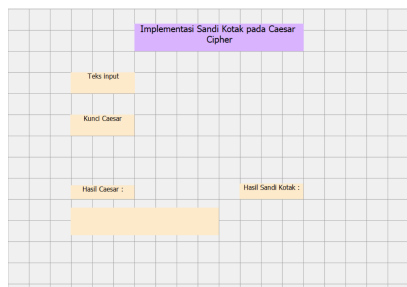
Komponen	Fungsi
<i>Static Text</i>	Label judul
<i>Static Text</i>	Label nama Teks input
<i>Static Text</i>	Label nama Kunci <i>Caesar</i>
<i>Static Text</i>	Label nama Hasil <i>Caesar</i>
<i>Static Text</i>	Label nama Hasil Sandi Kotak
<i>Static Text</i>	Menampilkan hasil <i>Caesar Cipher</i>
<i>Edit Text</i>	Memasukkan teks input
<i>Edit Text</i>	Memasukkan angka kunci <i>Caesar</i>
<i>Pushbutton</i>	Tombol enkripsi
<i>Pushbutton</i>	Tombol dekripsi
<i>Pushbutton</i>	Refresh
<i>Pushbutton</i>	Tutup
Axes	Menampilkan simbol Sandi Kotak untuk hasil enkripsi
Axes	Menampilkan simbol Sandi Kotak untuk didekripsikan

- 3) Pilih opsi *Background Colour* dan ubah warnanya menjadi ungu. Kemudian, pada bagian *String*, ketikkan teks 'Implementasi Sandi Kotak pada Caesar Cipher' dengan pengaturan *Fontname*: Tahoma dan *Fontsize*: 16.



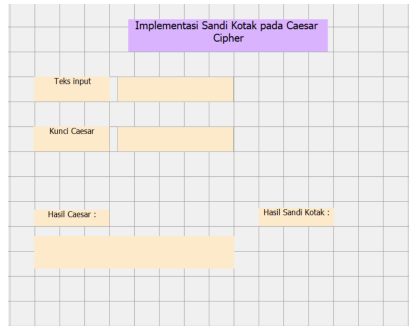
Gambar 3.5. Tampilan judul yang ditampilkan pada GUI MATLAB

- 4) Buat 5 *static text* lagi, dengan 4 *static text* tulis masing-masing string nya 'Teks input', 'Kunci Caesar', 'Hasil Caesar', 'Hasil Sandi Kotak' dengan *Background Colour* krem, *Fontname*: Tahoma, *Fontsize*: 12. Untuk *static text* satu lagi ubah *background* dengan warna cream dengan tulisan di *string* kosongkan sejajarkan di bawah Hasil Caesar.



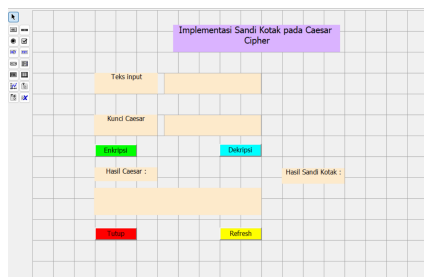
Gambar 3.6. Tampilan *Static text* GUI MATLAB

- 5) Buat 2 *edit text* lalu kosongkan *stringnya*, sejajarkan dengan Teks input dan Kunci Caesar. Berikan *Background Colour* krem, *Fontname*: Tahoma, *Fontsize*: 10.



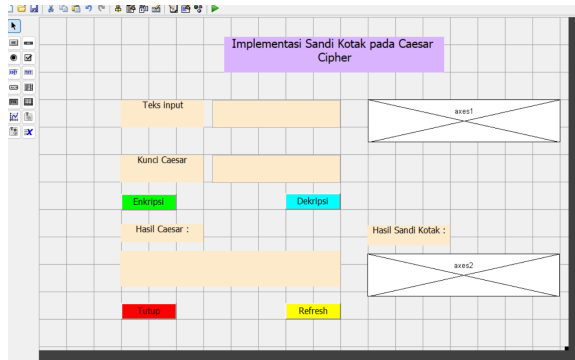
Gambar 3.7. Tampilan *Edit text* GUI MATLAB

- 6) Buat 4 *Button Pushbutton*, tulis *string*'Enkripsi' dengan *Background Colour* hijau, tulis *string*'Dekripsi' dengan *Background Colour* biru, tulis *string*'Refresh' dengan *Background Colour* kuning, tulis *string*'Tutup' dengan *Background Colour* merah. Semua menggunakan *Fontname*: Tahoma, *Fontsize*: 12.



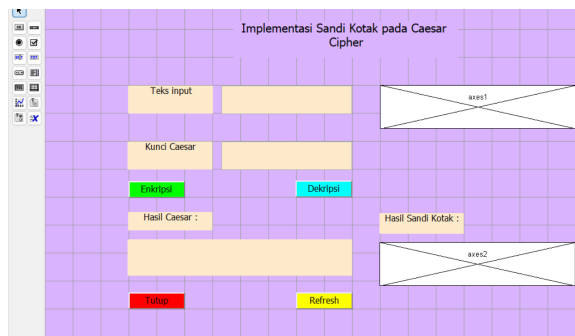
Gambar 3.8. Tampilan *Pushbutton* GUI MATLAB

- 7) Buat 2 Axes yang pertama sejajarkan di bawah Hasil Sandi kotak, yang kedua sejajarkan dengan Teks input.



Gambar 3.9. Tampilan Axes GUI MATLAB

- 8) Terakhir klik *view* pada bidang kotak-kotak dan pilih *property inspector*, klik *Background Colour* dengan warna ungu.



Gambar 3.10. Tampilan akhir GUI MATLAB sebelum dijalankan

2. Langkah Memasukkan *Script* Aplikasi

1) Memulai persiapan MATLAB

- (a) Buka MATLAB.
- (b) Atur direktori kerja ke folder tempat menyimpan file `.fig`, `.m`, dan folder `pigpen_symbols`.
- (c) Buat folder `pigpen_symbols` untuk menyimpan gambar simbol sandi kotak (A.png hingga Z.png).

2) Membuat file GUI menggunakan GUIDE

- (a) Ketik `guide` di *Command Window* MATLAB, lalu pilih *Blank GUI (Default)*.
- (b) Tambahkan komponen utama GUI: dua *Edit Text* (`edit_input`, `edit_key`), satu *Static Text* (`text_result`), dua *Axes* (`axes_input`, `axes_result`), dan empat *Push Button* (`push_encrypt`, `push_decrypt`, `push_clear`, `push_close`).
- (c) Simpan file sebagai `pigpen_gui.fig`, yang akan menghasilkan `pigpen_gui.m`.

3) Memasukkan *script* inisialisasi GUI

- (a) Buka `pigpen_gui.m`.
- (b) Salin kode inisialisasi dari Lampiran 1 ke fungsi `pigpen_gui_OpeningFcn` dan `pigpen_gui_OutputFcn`. Kode ini mengatur properti awal GUI dan memeriksa folder `pigpen_symbols`.
- (c) Pastikan folder `pigpen_symbols` ada untuk menghindari peringatan.

- 4) Memasukkan *script callback* input dan kunci
 - (a) Di editor GUIDE, tambahkan fungsi *callback* untuk `edit_input` dan `edit_key` melalui *View Callback*.
 - (b) Salin kode dari Lampiran 2 untuk mengatur input *plaintext* dan validasi kunci (hanya menerima bilangan bulat).
- 5) Memasukkan *script* tombol Enkripsi dan Dekripsi
 - (a) Tambahkan fungsi *callback* untuk tombol `push_encrypt` dan `push_decrypt` di editor GUIDE.
 - (b) Salin kode dari Lampiran 3 untuk memanggil fungsi *caesar cipher* dan display sandi kotak, serta menampilkan hasil di `text_result` dan `axes_result`.
- 5) Memasukkan *script* fungsi *Caesar Cipher*
 - (a) Tambahkan fungsi *callback* untuk *caesar cipher*
 - (b) Salin kode dari Lampiran 4 untuk menangani enkripsi dan dekripsi.
- 6) Memasukkan *script* fungsi Display Sandi Kotak
 - (a) Tambahkan fungsi *callback* untuk fungsi display sandi kotak
 - (b) Salin kode dari Lampiran 5 untuk menampilkan simbol Sandi Kotak di `axesHandle` berdasarkan teks input dan folder `pigpen_symbols`.
 - (c) Pastikan gambar A.png hingga Z.png ada di folder `pigpen_symbols`.

7) Memasukkan *script* tombol Refresh dan Tutup

- (a) Tambahkan fungsi *callback* untuk tombol `push_clear` dan `push_close` di editor GUIDE.
- (b) Salin kode dari Lampiran 6 untuk mengosongkan input dan menutup aplikasi dengan konfirmasi.

8) Menguji dan menjalankan aplikasi

- (a) Simpan semua file (`pigpen_gui.fig`, `pigpen_gui.m`, `pigpen_symbols`) di direktori kerja.
- (b) Jalankan `pigpen_gui` di *Command Window*.
- (c) Uji fungsi dengan memasukkan *plaintext* (misalnya, "UIN WALISONGO") dan kunci (misalnya, 3), hasil sesuai Lampiran 7.

3. Pengujian Fungsional Aplikasi Implementasi Sandi Kotak pada *Caesar Cipher*

Berikut ini hasil Pengujian aplikasi menggunakan metode *black-box testing*:

1. Pengujian Fungsional Pada Aplikasi

a. Pengujian kolom input pada Teks Input

Langkah pengoperasian pada kolom input dilakukan dengan cara memasukkan pesan:

- 1) Klik Pada Teks Input
- 2) Masukkan hanya huruf A–Z. Karakter non-alfabet akan diabaikan.
- 3) Tampilan seperti yang terlihat pada Gambar 3.11.

The screenshot shows a web application titled "Implementasi Sandi Kotak pada Caesar Cipher". It features a text input field labeled "Teks Input :" containing the text "UIN123". Below it is a label "Kunci Caesar (angka) :" with a value of "3". There are two buttons: a green "Enkripsi" button and a blue "Dekripsi" button. Below the buttons, the "Hasil Caesar :" field displays "XLQ". To the right, the "Hasil Sandi Kotak :" field shows a visual representation of the cipher using symbols: a less-than sign, a dot, a square with a dot, and a square with a corner. At the bottom, there are "Tutup" (red) and "Refresh" (yellow) buttons.

Gambar 3.11. Tampilan Pengujian pada Teks Input

b. Pengujian kolom input pada Kunci *Caesar Cipher*

Langkah pengoperasian kolom input dilakukan dengan cara memasukkan kunci :

- 1) Klik Pada Kunci *Caesar*

- 2) Masukkan kunci harus berupa bilangan bulat. Jika tidak, program akan menampilkan peringatan, seperti terlihat pada Gambar 3.12.



Gambar 3.12. Tampilan Pengujian pada *Kunci Caesar*

c. Pengujian *Button* 'Enkripsi'

Langkah proses Pengujian fungsi *Button* 'Enkripsi':

- 1) Klik *Pushbutton* 'Enkripsi'
- 2) Tampilan hasil Enkripsi akan selalu huruf kapital, seperti yang terlihat pada Gambar 3.13.



Gambar 3.13. Tampilan Pengujian *Button* 'Enkripsi'

d. Pengujian *Button* 'Dekripsi'

Langkah proses Pengujian fungsi *Button* 'Dekripsi':

- 1) Klik *Pushbutton* 'Dekripsi'
- 2) Tampilan hasil Dekripsi akan selalu huruf kapital, seperti yang terlihat pada Gambar 3.14.

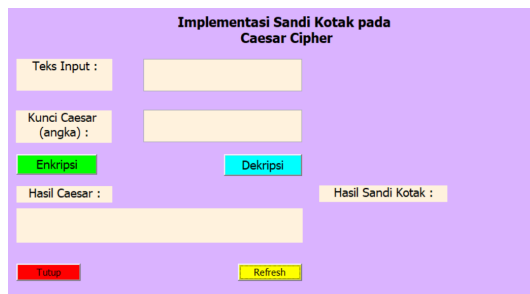


Gambar 3.14. Tampilan Pengujian *Button* 'Dekripsi'

e. Pengujian *Button* 'Refresh'

Langkah proses Pengujian fungsi *Button* 'Refresh':

- 1) Klik *Pushbutton* 'Refresh'
- 2) Menghapus semua input dan hasil aplikasi, seperti yang terlihat pada Gambar 3.15.



Gambar 3.15. Tampilan Pengujian *Button* 'Refresh'

f. Pengujian *Button* 'Tutup'

Langkah proses Pengujian fungsi *Button* 'Tutup':

- 1) Klik *Pushbutton* 'Tutup'
- 2) Muncul menu konfirmasi dengan pilihan 'ya' atau 'tidak', jika memilih 'ya' maka aplikasi tertutup.
- 3) Tampilan seperti yang terlihat pada Gambar 3.16.



Gambar 3.16. Tampilan Pengujian *Button* 'Tutup'

2. Pengujian Kesesuaian Antara Hasil Aplikasi dan Rancangannya

Berikut merupakan proses Pengujian untuk mengevaluasi apakah hasil akhir aplikasi telah sesuai dengan rancangan yang telah dibuat, sebagai bagian dari validasi menggunakan metode *black-box testing*:

Tabel 3.4. Tabel Unit Aplikasi dan Rancangan Proses

No.	Unit Aplikasi	Rancangan Proses	Hasil yang diharapkan	Keterangan
1.	Menu pada kolom input	Klik Teks Input	Tampil pesan yang diketik berupa alfabet	Sesuai
		Klik Kunci Caesar	Tampil kunci yang diketik berupa bilangan bulat	Sesuai
2.	Menu Proses Enkripsi & Dekripsi	Klik <i>Button</i> 'Enkripsi'	Tampilan hasil Enkripsi	Sesuai
		Klik <i>Button</i> 'Dekripsi'	Tampilan hasil Dekripsi	Sesuai
3.	Menu Hapus Data	Klik <i>Button</i> 'Refresh'	Menghapus semua input-an dan hasil aplikasi	Sesuai
4.	Menu Tutup Aplikasi	Klik <i>Button</i> 'Tutup'	Tampil menu konfirmasi pilihan 'ya' atau 'tidak', jika memilih 'ya' maka aplikasi tertutup	Sesuai

4. Analisis Hasil

Berdasarkan hasil simulasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa implementasi sandi kotak dan *caesar cipher* dalam bentuk GUI MATLAB telah berhasil dijalankan dengan baik. Kombinasi dua metode kriptografi ini mampu meningkatkan kerumitan dan keamanan dalam penyandian pesan. *Caesar cipher* sebagai tahap penyandian dengan metode berbasis pergeseran huruf, sedangkan sandi kotak menyamarkan huruf menjadi simbol-simbol yang tidak langsung dikenali sebagai alfabet.

Analisis juga menunjukkan bahwa penggunaan MATLAB memiliki kemampuan pengolahan string dan visualisasi grafis yang kuat. MATLAB juga menyediakan fitur GUI yang mendukung pembuatan aplikasi dengan tampilan interaktif dan hasil yang dapat ditampilkan secara *real-time*. Berdasarkan penelitian ini, MATLAB sangat efektif untuk memvisualisasikan simbol sandi kotak, yang merupakan bagian dari sandi non-huruf dan memerlukan tampilan grafis.

Namun demikian, terdapat beberapa keterbatasan. Salah satunya adalah keterbatasan GUI untuk menerima input simbol sandi kotak secara manual jika pengguna ingin melakukan dekripsi. Selain itu, karakter khusus dan angka, belum dikodekan dalam algoritma, sehingga hanya alfabet A–Z yang dapat diproses.

Secara keseluruhan, hasil analisis menunjukkan bahwa simulasi menggunakan GUI MATLAB sangat mendukung pemahaman serta penerapan algoritma kriptografi klasik. Implementasi ini dapat digunakan sebagai media pembelajaran interaktif maupun sebagai dasar untuk pengembangan sistem keamanan informasi yang lebih kompleks.

BAB IV

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian serta simulasi yang telah dilakukan, dapat ditarik beberapa simpulan sebagai berikut:

Proses enkripsi pesan dengan implementasi sandi kotak pada kriptografi *caesar cipher* secara matematis yaitu :

$$C = S(E(P, K))$$

Enkripsi *plaintext* P menggunakan *caesar cipher* dengan kunci K , dan kemudian hasilnya diubah ke simbol-simbol sandi kotak melalui fungsi substitusi S , sehingga menghasilkan *ciphertext* C dalam bentuk simbol sandi kotak.

Proses dekripsi pesan dengan implementasi sandi kotak pada kriptografi *caesar cipher* secara matematis yaitu :

$$P = D(S^{-1}(C), K)$$

Dekripsi dengan mengubah simbol sandi kotak *ciphertext* C menjadi huruf melalui fungsi invers sandi kotak S^{-1} , kemudian hasil huruf tersebut didekripsi menggunakan algoritma *caesar cipher* dengan kunci K sehingga diperoleh kembali *plaintext* P .

B. Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini di masa mendatang adalah sebagai berikut:

1. Mengembangkan sistem menggunakan sandi kotak 2 atau sandi kotak 3, agar dapat dibandingkan efektivitas dan keamanannya dengan sandi kotak 1 yang dipakai dalam penelitian ini. Penggunaan metode kriptografi lain juga dapat dipertimbangkan untuk meningkatkan fleksibilitas dan keamanan sistem.
2. Menambahkan dukungan untuk karakter selain huruf alfabet, seperti angka dan tanda baca, agar program lebih fleksibel dalam menangani pesan yang kompleks.
3. Mengembangkan fitur agar simbol sandi kotak dapat dikenali dan diketik secara manual oleh pengguna jika ingin melakukan dekripsi pesan.
4. Mengintegrasikan algoritma ini ke dalam aplikasi berbasis web atau versi mobile agar aplikasinya dapat diakses secara lebih luas dan tidak terbatas hanya pada pengguna MATLAB.

DAFTAR PUSTAKA

- Abdelmgeid, A. A., Bahgat, A. A., Saad, A. H. S., & Gomaa, M. M. (2017). Enhancing Image Steganography Methods By Using New Secret Message Encoding Technique Based on Pigpen cipher (Pigpen Encoding). *International Journal of Computer Applications*, 975, 8887.
- Ariyus, D. (2006). *Kriptografi Keamanan Data dan Kriptografi*. Yogyakarta: Penerbit Andi Offset.
- Ariyus, D. (2008). *Pengantar Ilmu Kriptografi (Teori, Analisis, dan Implementasi)*. S. Suyantoro, Ed. CV Andi Offset.
- Azhar, A. H., Destari, R. A., Negoro, W. S., & Ardansyah, A. (2024). Pelatihan Teknik *Caesar Cipher* Terhadap Keamanan Informasi. *Jurnal Tunas*, 5(2), 88-92.
- Darmayanti, I., Astrida, D. N., & Ariyus, D. (2018). Penerapan Keamanan Pesan Teks Menggunakan Modifikasi Algoritma *Caisar Chiper* Kedalam Bentuk Sandi Morse. *Jurnal Ilmiah IT CIDA*, 4(1).
- Gurning, R. R. A. (2014). Perancangan Aplikasi Pengamanan Pesan dengan Algoritma *Caesar Chiper*. *Jurnal Pelita Informatika Budi Darma*, 6(4), 106–110.
- Martolia, M., Dimri, P., & Arora, N. (2018). PIGPEN: A Novel Approach for Securing Data. *International Journal of Research in Engineering, IT and Social Sciences*, 08(4), 43-46.
- Munir, R. 2010. *Matematika Diskrit (Revisi Keempat)*. Bandung: Informatika.

- Nasution, A. B., Efendi, S., & Suwilo, S. (2018). Image Steganography in Securing Sound File Using Arithmetic Coding Algorithm, Triple Data Encryption Standard (3DES) and Modified Least Significant Bit (MLSB). *Journal of Physics: Conference Series*, 1007(1).
- Pratiwi, R., Utami, L. C., & Sakti, R. B. (2022). Perancangan Keamanan Data Pesan Dengan Menggunakan Metode Kriptografi Caesar Cipher. *Bulletin of Information Technology (BIT)*, 3(4), 367-373.
- Prayitno, A., & Nurdin, N. (2017). Analisis Dan Implementasi Kriptografi Pada Pesan Rahasia Menggunakan Algoritma Cipher Transposition. *Jurnal elektronik sistem informasi dan komputer*, 3(1), 1-10.
- Purnomo, H. D., & Sembiring, I. (2022). Modifikasi Algoritma Caesar Cipher pada Kode ASCII dalam Meningkatkan Keamanan Pesan Teks. *Journal of Information Technology*, 2(1), 16-22.
- Seftyanto, D., Apriani, M., & Haryanto, T. (2012, November). Peran Algoritma Caesar Cipher Dalam Membangun Karakter Akan Kesadaran Keamanan Informasi. In *Seminar Nasional Matematika dan Pendidikan Matematika FMIPA UNY* (pp. 883-890).
- Siregar, T. (2025). *Matematika Sains dan Teknologi (Saintek)*. Goresan Pena.

- Wrixon, F.B (2005). *Codes, Ciphers, Secrets and Cryptic Communication: Making and Breaking Sercet Messages from Hieroglyphocs to the Internet*. Black Dog & Leventhal Pub.
- Zubaidi, U. (2018). *Taklukkan Syarat-Syarat Kecakapan Umum: Pramuka Penegak Bantara Laksana*. Zooba.ID.

Lampiran 1. **SCRIPT** GUI MATLAB Inisialisasi GUI (Opening dan Output)

```

1  function varargout = pigpen_gui(varargin)
2  gui_Singleton = 1;
3  gui_State = struct('gui_Name',       mfilename, ...
4                    'gui_Singleton',   gui_Singleton, ...
5                    'gui_OpeningFcn', @pigpen_gui_OpeningFcn, ...
6                    'gui_OutputFcn',  @pigpen_gui_OutputFcn, ...
7                    'gui_LayoutFcn',  [], ...
8                    'gui_Callback',    []);
9  if nargin && ischar(varargin{1})
10     gui_State.gui_Callback = str2func(varargin{1});
11 end
12
13 if nargin
14     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
15 else
16     gui_mainfcn(gui_State, varargin{:});
17 end
18
19 function pigpen_gui_OpeningFcn(hObject, eventdata, handles, varargin)
20 handles.output = hObject;
21
22 cla(handles.axes_input);
23 axis(handles.axes_input, 'on');
24
25 cla(handles.axes_result);
26 axis(handles.axes_result, 'on');
27
28 handles.pigpenFolder = 'pigpen_symbols';
29 if ~exist(handles.pigpenFolder, 'dir')
30     mkdir(handles.pigpenFolder);
31     warndlg('Folder_"pigpen_symbols"_dibuat._Silakan_tambahkan_gambar_A.png
32     _____sampai_Z.png', 'Peringatan');
33 end
34 guidata(hObject, handles);
35 function varargout = pigpen_gui_OutputFcn(hObject, eventdata, handles)
36 varargout{1} = handles.output;

```


Lampiran 2. *SCRIPT* GUI MATLAB Callback Input dan Kunci

```

1  function edit_input_Callback(hObject, eventdata, handles)
2  textInput = upper(get(hObject, 'String'));
3  function edit_input_CreateFcn(hObject, eventdata, handles)
4  if ispc && isequal(get(hObject,'BackgroundColor'),
5  get(0,'defaultUiControlBackgroundColor'))
6      set(hObject,'BackgroundColor','white');
7  end
8  function edit_key_Callback(hObject, eventdata, handles)
9  keyStr = get(hObject, 'String');
10 keyStr = strrep(keyStr, ',', '.');
11 if ~isempty(keyStr)
12     key = str2double(keyStr);
13     if isnan(key) || ~isreal(key) || mod(key, 1) ~= 0
14         errordlg('Kunci_harus_berupa_bilangan_bulat!', 'Error');
15     else
16         set(hObject, 'String', num2str(key));
17     end
18 end
19 function edit_key_CreateFcn(hObject, eventdata, handles)
20 if ispc && isequal(get(hObject,'BackgroundColor'),
21 get(0,'defaultUiControlBackgroundColor'))
22     set(hObject,'BackgroundColor','white');
23 end
24 function edit_key_KeyPressFcn(hObject, eventdata, handles)
25 currentText = get(hObject,'String');
26 allowedKeys = {'0','1','2','3','4','5','6','7','8','9','-','...
27     'backspace','delete','leftarrow','rightarrow'};
28 if any(eventdata.Character == ',.') && contains(currentText, ['. ', ', '])
29     set(hObject, 'UserData', 'ignore');
30     return;
31 end
32 if ~any(strcmp(eventdata.Key, allowedKeys)) && ...
33     ~isempty(eventdata.Character) && ...
34     ~any(strcmp(eventdata.Character, {',', '.', ' '}))
35     set(hObject, 'UserData', 'ignore');
36 else
37     set(hObject, 'UserData', 'allow');
38 end

```

Lampiran 3. *SCRIPT* GUI MATLAB Tombol Enkripsi dan Dekripsi

```

1  function push_encrypt_Callback(hObject, eventdata, handles)
2  textInput = upper(get(handles.edit_input, 'String'));
3  keyStr = get(handles.edit_key, 'String');
4  keyStr = strrep(keyStr, ',', '.');
5  key = str2double(keyStr);
6
7  if isnan(key) || ~isreal(key) || mod(key, 1) ~= 0
8      errordlg('Kunci_harus_berupa_bilangan_bulat!', 'Error');
9      return;
10 end
11
12 cipher = caesar_cipher(textInput, key, 'encrypt');
13 set(handles.text_result, 'String', cipher);
14
15 cla(handles.axes_input);
16 display_pigpen(cipher, handles.axes_result, handles.pigpenFolder);
17
18 function push_decrypt_Callback(hObject, eventdata, handles)
19 textInput = upper(get(handles.edit_input, 'String'));
20 keyStr = get(handles.edit_key, 'String');
21 keyStr = strrep(keyStr, ',', '.');
22 key = str2double(keyStr);
23
24 if isnan(key) || ~isreal(key) || mod(key, 1) ~= 0
25     errordlg('Kunci_harus_berupa_bilangan_bulat!', 'Error');
26     return;
27 end
28
29 plain = caesar_cipher(textInput, key, 'decrypt');
30 set(handles.text_result, 'String', plain);
31
32 cla(handles.axes_result);
33 display_pigpen(textInput, handles.axes_input, handles.pigpenFolder);

```

Lampiran 4. *SCRIPT GUI MATLAB Fungsi Caesar Cipher*

```

1  function out = caesar_cipher(text, key, mode)
2  out = '';
3  for i = 1:length(text)
4      ch = upper(text(i));
5      if isletter(ch)
6          base = double('A');
7          idx = double(ch) - base;
8          if strcmp(mode, 'encrypt')
9              idx = mod(idx + key, 26);
10             else
11                 idx = mod(idx - key + 26, 26);
12             end
13             out(end+1) = char(base + idx);
14         elseif ch == ' '
15             out(end+1) = ' ';
16         end
17     end

```

Lampiran 5. *SCRIPT* GUI MATLAB Fungsi Display Sandi Kotak

```

1  function display_pigpen(text, axesHandle, folder)
2  cla(axesHandle);
3  axis(axesHandle, 'off');
4
5  if isempty(text)
6      text(0.5, 0.5, 'Tidak_ada_teks_untuk_ditampilkan', ...
7          'HorizontalAlignment', 'center', 'Parent', axesHandle);
8      return;
9  end
10
11 if ~exist(folder, 'dir')
12     text(0.5, 0.5, 'Folder_simbol_tidak_ditemukan', ...
13         'HorizontalAlignment', 'center', 'Parent', axesHandle);
14     return;
15 end
16
17 pigpenImages = {};
18 missingChars = [];
19
20 for i = 1:length(text)
21     ch = upper(text(i));
22     if ch >= 'A' && ch <= 'Z'
23         imgPath = fullfile(folder, [ch '.png']);
24         if exist(imgPath, 'file')
25             pigpenImages(end+1) = imread(imgPath);
26         else
27             img = 255 * ones(100, 100, 3, 'uint8');
28             pigpenImages(end+1) = insertText(img, [30 40], ch, 'FontSize',
29                 40, 'BoxColor', 'white', 'TextColor', 'black');
30             missingChars = [missingChars ch];
31         end
32     elseif ch == ' '
33         pigpenImages(end+1) = 255 * ones(100, 30, 3, 'uint8');
34     end
35 end
36 if ~isempty(missingChars)
37     warndlg(['Simbol_untuk_huruf_berikut_tidak_ditemukan:_'
38         unique(missingChars)], 'Peringatan');
39 end
40

```

```

41 if ~isempty(pigpenImages)
42     imgSize = size(pigpenImages{1});
43     for i = 2:length(pigpenImages)
44         pigpenImages{i}= imresize(pigpenImages{i},[imgSize(1) imgSize(2)]);
45     end
46
47     result = pigpenImages{1};
48     for i = 2:length(pigpenImages)
49         result = [result pigpenImages{i}];
50     end
51
52     axes(axesHandle);
53     imshow(result);
54     axis(axesHandle, 'off');
55 else
56     text(0.5, 0.5, 'Tidak_ada_teks_untuk_ditampilkan', ...
57         'HorizontalAlignment', 'center', 'Parent', axesHandle);
58 end

```

Lampiran 6. *SCRIPT* GUI MATLAB Tombol Refresh dan Tutup

```

1  function push_clear_Callback(hObject, eventdata, handles)
2  set(handles.edit_input, 'String', '');
3  set(handles.edit_key, 'String', '');
4  set(handles.text_result, 'String', '');
5
6  cla(handles.axes_input);
7  axis(handles.axes_input, 'off');
8
9  cla(handles.axes_result);
10 axis(handles.axes_result, 'off');
11
12 function push_clear_CreateFcn(hObject, eventdata, handles)
13
14 function push_close_Callback(hObject, eventdata, handles)
15 selection = questdlg('Apakah_Anda_yakin_ingin_menutup_aplikasi?',...
16                     'Konfirmasi_Penutupan',...
17                     'Ya', 'Tidak', 'Tidak');
18
19 switch selection
20     case 'Ya'
21         delete(handles.figure1);
22     case 'Tidak'
23         return;
24 end

```

Lampiran 7. Hasil Uji Aplikasi

Tampilan Hasil Enkripsi

Implementasi Sandi Kotak pada Caesar Cipher

Teks Input : UIN WALISONGO

Kunci Caesar (angka) : 3

Enkripsi Dekripsi

Hasil Caesar : XLQ ZDOLVRQJR

Hasil Sandi Kotak :

< . [^ _] n e v r q j r

Tutup Refresh

Tampilan Hasil Dekripsi

Implementasi Sandi Kotak pada Caesar Cipher

Teks Input : XLQ ZDOLVRQJR

Kunci Caesar (angka) : 3

Enkripsi Dekripsi

Hasil Caesar : UIN WALISONGO

Hasil Sandi Kotak :

< . [^ _] n e v r q j r

Tutup Refresh

Lampiran 8. Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



A. Identitas

Nama : Fadillah Al Musyaffa
 Tempat, Tanggal Lahir : Bekasi, 19 September 2002
 Jenis Kelamin : Laki-laki
 Agama : Islam
 Alamat : Jl. Pramuka No. 43, Kec. Tambun Selatan, Bekasi
 Email : 2108046052@student.walisongo.ac.id

B. Riwayat Pendidikan

2009–2015 : SD Negeri Mangunjaya 02
 2015–2018 : SMP Negeri 7 Tambun Selatan
 2018–2021 : SMA Negeri 2 Tambun Utara
 2021–2025 : Universitas Islam Negeri Walisongo