SKRIPSI

PERBANDINGAN ALGORITMA HILL CIPHER DAN AFFINE CIPHER DALAM MENJAGA KEAMANAN DATA PADA WEBSITE E-COMMERCE

Sebagai salah satu syarat untuk menyelesaikan
Program Studi Sarjana Terapan Keamanan Sistem Informasi

Oleh:

ONA MAZURA
6404221095

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BENGKALIS

TAHUN 2026

HALAMAN PENGESAHAN

SKRIPSI

PERBANDINGAN ALGORITMA HILL CIPHER DAN AFFINE CIPHER DALAM MENJAGA KEAMANAN DATA PADA WEBSITE E-COMMERCE

Oleh:

ONA MAZURA
6404221095

Telah diujikan dan dinyatakan lulus ujian skripsi pada tanggal 23 Februari 2026 oleh tim penguji Program Studi Sarjana Terapan Keamanan Sistem Informasi

Bengkalis, 23 Februari 2026

Pembimbing Utama

Nurmi Hidāyasari, ST., M.Kom
NIP. 199109012022032006

Anggota Tim Penguji

Agus Tedyyana, S.Kom., M.Kom
NIP. 198510052015041001

Mansur, S.Kom., M.Kom
NIP. 198209192021211003

Mengetahui,
Ketua Jurusan Teknik Informatika

Kasmawi, S.Kom., M.Kom
NIP. 197706072014041001

PERNYATAAN KEASLIAN SKRIPSI

Saya Menyatakan dengan sesungguhnya bahwa Skripsi ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar kesarjanaan di perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang tertulis disebutkan sumbernya dalam naskah dan dalam daftar Pustaka.

Bengkalis, 12 Januari 2026

Ona Mazura

PERBANDINGAN ALGORITMA HILL CIPHER DAN AFFINE CIPHER DALAM MENJAGA KEAMANAN DATA PADA WEBSITE E-COMMERCE

Nama Mahasiswa : Ona Mazura

NIM : 6404221095

Dosen Pembimbing : Nurmi Hidayasari, ST., M. Kom

ABSTRAK

Keamanan data pengguna merupakan aspek penting dalam sistem e-commerce karena melibatkan informasi sensitif yang rentan terhadap akses tidak sah. Salah satu metode yang dapat digunakan untuk melindungi data tersebut adalah kriptografi klasik. Penelitian ini bertujuan untuk membandingkan algoritma *Hill Cipher* dan *Affine Cipher* dalam mengamankan data pengguna pada sistem e-commerce berbasis web. Kedua algoritma diimplementasikan pada sistem e-commerce *open-source* dengan mengenkripsi data pengguna berupa username, email, dan password sebelum disimpan ke dalam database. Pengujian dilakukan dengan menganalisis performa enkripsi dan dekripsi serta keamanan ciphertext menggunakan pendekatan *Ciphertext-Only Attack* (COA) dan *Shannon Entropy*. Hasil penelitian menunjukkan bahwa *Affine Cipher* memiliki waktu proses enkripsi dan dekripsi yang lebih cepat, sedangkan *Hill Cipher* menghasilkan ciphertext dengan tingkat kompleksitas dan keacakan yang lebih tinggi. Analisis COA menunjukkan bahwa ciphertext dari kedua algoritma tidak menampilkan pola bermakna, sementara hasil pengujian *Shannon Entropy* menunjukkan bahwa algoritma *Hill Cipher* menghasilkan nilai entropy sebesar 4,55 bit yang berada pada kategori tingkat keacakan tinggi, sedangkan algoritma *Affine Cipher* menghasilkan nilai entropy sebesar 3,60 bit yang berada pada kategori tingkat keacakan sedang. Berdasarkan hasil tersebut, dapat disimpulkan bahwa *Affine Cipher* unggul dari sisi efisiensi waktu, sedangkan *Hill Cipher* lebih baik dalam menghasilkan ciphertext yang kompleks. Penelitian ini diharapkan dapat menjadi referensi dalam pemilihan algoritma kriptografi klasik untuk pengamanan data pada sistem e-commerce berbasis web.

Kata kunci: Keamanan Data, Kriptografi Klasik, *Hill Cipher*, *Affine Cipher*, *Ciphertext-Only Attack*, *Shannon Entropy*, E-Commerce.

COMPARISON OF THE HILL CIPHER AND AFFINE CIPHER ALGORITHMS IN MAINTAINING DATA SECURITY ON E-COMMERCE WEBSITES

Student Name : ONA MAZURA
Student ID : 6404221095

Supervisor : Nurmi Hidayasari, ST., M. Kom

ABSTRACT

User data security is an important aspect of e-commerce systems because it involves sensitive information that is vulnerable to unauthorized access. One method that can be used to protect such data is classical cryptography. This study aims to compare the Hill Cipher and Affine Cipher algorithms in securing user data in a web-based e-commerce system. Both algorithms are implemented in an open-source e-commerce system by encrypting user data in the form of usernames, emails, and passwords before being stored in the database. The evaluation is conducted by analyzing the performance of the encryption and decryption *processes as well as the security of the ciphertext using the Ciphertext-Only Attack (COA) approach and Shannon Entropy. The results show that the Affine Cipher has faster encryption and decryption processing times, while the Hill Cipher produces ciphertext with a higher level of complexity and randomness. COA analysis indicates that the ciphertext generated by both algorithms does not exhibit meaningful patterns. Furthermore, the Shannon Entropy test results show that the Hill Cipher produces an entropy value of 4,55 bits, which falls into the high randomness category, while the Affine Cipher produces an entropy value of 3,60 bits, which is categorized as moderate randomness. Based on these results, it can be concluded that the Affine Cipher excels in terms of time efficiency, whereas the Hill Cipher performs better in generating complex ciphertext. This study is expected to serve as a reference for selecting classical cryptographic algorithms to secure data in web-based e-commerce systems.*

Keywords: *Data Security, Classical Cryptography, Hill Cipher, Affine Cipher, Ciphertext-Only Attack, Shannon Entropy, E-Commerce.*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Perbandingan Algoritma Hill Cipher dan Affine Cipher dalam Menjaga Keamanan Data pada Website E-Commerce". Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Sarjana Terapan Keamanan Sistem Informasi, Jurusan Teknik Informatika, Politeknik Negeri Bengkalis.

Penulis menyadari bahwa dalam proses penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Johny Custer, S.T., M.T., selaku Direktur Politeknik Negeri Bengkalis.
2. Bapak Kasmawi, S.Kom., M.Kom, selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis.
3. Ibu Nurmi Hidayasari, ST., M.Kom, selaku Ketua Program Studi D-IV Keamanan Sistem Informasi sekaligus Dosen Pembimbing yang telah memberikan bimbingan, arahan, dan motivasi selama proses penyusunan skripsi ini.
4. Seluruh Bapak dan Ibu Dosen Jurusan Teknik Informatika Politeknik Negeri Bengkalis yang telah memberikan ilmu pengetahuan selama masa perkuliahan.
5. Kedua orang tua tercinta, Bapak Burhan dan Ibu Juminem beserta seluruh keluarga besar khusus kepada Kakak-kakak ku dan Adik tercinta atas doa, kasih sayang, dukungan moral, serta materi yang tiada henti diberikan kepada penulis.
6. Teman-teman seperjuangan serta semua pihak yang telah membantu, memberikan semangat, motivasi, dan dukungan hingga skripsi ini dapat diselesaikan dengan baik.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi penyempurnaan karya ini di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang keamanan sistem informasi, serta dapat dijadikan sebagai referensi untuk penelitian selanjutnya.

Bengkalis, 12 Januari 2026

Ona Mazura

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di era teknologi saat ini, keamanan data menjadi salah satu aspek terpenting dalam pengembangan sistem informasi, khususnya dalam sistem e-commerce yang menyimpan berbagai data penting seperti identitas pengguna. Tanpa adanya perlindungan yang memadai, data tersebut sangat rentan terhadap berbagai jenis serangan siber, seperti penyadapan, pencurian data, dan akses ilegal oleh pihak yang tidak berwenang [2].

Salah satu pendekatan yang umum digunakan untuk melindungi data adalah *kriptografi*, yaitu metode untuk mengubah data asli (*plaintext*) menjadi bentuk terenkripsi (*ciphertext*) yang tidak dapat dibaca tanpa kunci tertentu. Dengan cara ini, meskipun data berhasil diakses oleh pihak ketiga, isinya tetap tidak dapat dimengerti tanpa proses *dekripsi* yang tepat [3].

Secara umum, *kriptografi* terbagi menjadi dua jenis: *kriptografi* klasik dan *kriptografi* modern. Meskipun dianggap sederhana dan telah lama digunakan, *kriptografi* klasik tetap relevan dalam berbagai konteks, terutama dalam bidang pendidikan, penelitian, dan sistem berskala kecil karena kemudahannya dalam implementasi [2].

Dua algoritma *kriptografi* klasik yang umum digunakan adalah *Hill Cipher* dan *Affine Cipher*. *Hill Cipher* memanfaatkan prinsip *matriks* dan operasi matematika untuk mengkonversi blok huruf menjadi format kode. Dengan memanfaatkan blok dan *matriks* sebagai kunci, algoritma ini menciptakan *ciphertext* yang lebih rumit dan lebih sulit untuk dikenali [3]. Sementara itu, *Affine Cipher* adalah algoritma yang meggunakan rumus linier dasar dengan dua bilangan sebagai kunci, yaitu angka untuk mengalikan dan angka untuk menambah.

Beberapa penelitian sebelumnya telah menganalisis algoritma ini dari sudut pandang yang berbeda. Penelitian oleh Adnan Buyung Nasution (2020) memodifikasi *Affine Cipher* dengan menambahkan proses membalik karakter

sebelum tahap enkripsi. Hasilnya menunjukkan bahwa tingkat keacakan *ciphertext bertambah*, meskipun belum diuji untuk serangan keamanan secara langsung [5]. Nurharianna Siregar dkk (2022) menggunakan *Hill Cipher* untuk mengenkripsi file teks ASCII dengan matriks 2×2. Dengan demikian, algoritma ini mampu menyimpan pesan secara efisien dalam file teks [6]. Penelitian oleh Roman Gusmana, dkk (2023) menciptakan simulasi aplikasi enkripsi dengan *Hill Cipher* dan menemukan bahwa bentuk sandi yang dihasilkan cukup rumit dan sulit untuk ditebak [7]. Putra, dkk (2024) menganalisis perbandingan antara *Hill Cipher* dan *Affine Cipher* dalam konteks perlindungan data citra digital. Hasilnya, Hill Cipher mencatat nilai MSE sebesar 7073,40 dan PSNR sebesar 9,63 dB, yang menunjukkan bahwa *Hill Cipher* lebih efektif dalam menyembunyikan data visual [8].

Untuk menguji cara kerja kedua algoritma ini dalam melindungi keamanan data, diperlukan perbandingan secara langsung dalam sistem yang mirip dengan kondisi sebenarnya. Dalam penelitian ini, sistem e-commerce berbasis web digunakan sebagai alat simulasi untuk menerapkan algoritma *Hill Cipher* dan *Affine Cipher*. Melalui uji ini, kedua algoritma akan dianalisis dari segi proses enkripsi, dekripsi, serta ketahanan keamanan ciphertext terhadap pendekatan *Ciphertext-Only Attack* (COA) dan pengukuran tingkat keacakan menggunakan *Shannon Entropy*.

1.2 Rumusan Masalah

Dalam konteks ini, permasalahan yang dihadapi adalah keamanan data pada web e-commerce yang menyimpan informasi sensitif seperti data pengguna. Untuk mengatasi hal tersebut, algoritma kriptografi klasik seperti *Hill Cipher* dan *Affine Cipher* dapat diterapkan sebagai upaya menjaga kerahasiaan data. Namun, kedua algoritma tersebut memiliki pendekatan dan tingkat kompleksitas yang berbeda, sehingga efektivitasnya dalam menjaga keamanan belum dapat disimpulkan secara pasti. Sementara itu, masih minim penelitian yang membandingkan keduanya secara langsung dalam skenario sistem web yang nyata. Oleh karena itu, diperlukan perbandingan guna mengetahui algoritma mana yang lebih unggul serta sejauh mana masing-masing mampu melindungi data berdasarkan analisis keamanan

ciphertext melalui pendekatan *Ciphertext-Only Attack* (COA) dan pengukuran tingkat keacakan menggunakan *Shannon Entropy*.

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini lebih terarah dan tidak menyimpang dari permasalahan yang telah dirumuskan, maka ruang lingkup penelitian dibatasi sebagai berikut:

1. Penelitian ini difokuskan pada pengamanan data pengguna pada sistem e-commerce berbasis web sebagai media simulasi pengujian.
2. Algoritma kriptografi yang digunakan dalam penelitian ini dibatasi pada algoritma kriptografi klasik, yaitu *Hill Cipher* dan *Affine Cipher*.
3. Data yang diamankan dalam penelitian ini hanya berupa data teks pengguna, yang meliputi username, email, dan password.
4. Pengujian performa pada penelitian ini dibatasi pada pengukuran waktu proses enkripsi dan dekripsi.
5. Analisis keamanan difokuskan pada ciphertext yang dihasilkan oleh masing-masing algoritma, dengan pendekatan *Ciphertext-Only Attack* (COA).
6. Pengukuran tingkat keacakan ciphertext dilakukan menggunakan metode *Shannon Entropy*.

1.4 Tujuan

Tujuan penelitian ini adalah sebagai berikut:

1. Menjelaskan cara kerja algoritma *Hill Cipher* dan *Affine Cipher* sebagai bagian dari kriptografi klasik.
2. Menerapkan kedua algoritma tersebut dalam sistem web e-commerce untuk proses enkripsi dan dekripsi data.
3. Membandingkan tingkat keamanan algoritma *Hill Cipher* dan *Affine Cipher* berdasarkan analisis keamanan ciphertext menggunakan pendekatan *Ciphertext-Only Attack* (COA) dan pengukuran tingkat keacakan dengan *Shannon Entropy*.

1.5 Manfaat

Penelitian ini dapat diharapkan memberikan manfaat sebagai berikut:

1. Bagi peneliti dan pengembang, penelitian ini memberikan referensi tambahan dalam memilih algoritma kriptografi klasik yang tepat, khususnya untuk menyeimbangkan antara keamanan dan kecepatan enkripsi dekripsi pada sistem web. Dengan mengetahui performa dan tingkat keacakan masing-masing algoritma, pengembang dapat membuat keputusan yang lebih tepat sesuai kebutuhan sistem.
2. Bagi akademis, penelitian ini menambah wawasan mengenai implementasi algoritma klasik seperti Hill Cipher dan Affine Cipher, serta pengaruhnya terhadap keamanan data dan performa sistem. Temuan ini dapat dijadikan dasar untuk penelitian lanjutan atau pengembangan metode kriptografi yang lebih efisien dan aman.

BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terdahulu

Beberapa penelitian terdahulu telah membahas permasalahan terkait dengan keamanan data menggunakan algoritma kriptografi klasik, khususnya *Hill Cipher* dan *Affine Cipher*. Beberapa penelitian terdahulu yang relevan antara lain:

Berdasarkan penelitian yang dilakukan oleh Adnan Buyung Nasution (2020) dengan judul *"*Modifikasi Algoritma *Affine Cipher* untuk Mengamankan Data*"*, peneliti memodifikasi algoritma *Affine Cipher* untuk meningkatkan keamanannya. Modifikasi dilakukan dengan membagi plaintext ke dalam blok-blok, membalik urutan karakter, kemudian menerapkan rumus *affine cipher*. Hasilnya menunjukkan bahwa *ciphertext* menjadi lebih acak dan sulit ditebak. Penelitian ini menunjukkan potensi algoritma klasik jika dimodifikasi sesuai kebutuhan keamanan data saat ini [5].

Penelitian oleh Nurharianna Siregar, dkk (2022) berjudul *"*Menerapkan Algoritma *Hill Cipher* dan Matriks 2x2 dalam Mengamankan File Teks Menggunakan Kode ASCII*"*, menjelaskan penerapan algoritma *Hill Cipher* menggunakan Python untuk menyandikan file teks. Dalam penelitiannya digunakan kunci matriks 2x2 untuk proses enkripsi dan dekripsi data. Aplikasi yang dikembangkan terbukti mampu menyembunyikan isi pesan dari pihak yang tidak berkepentingan dengan baik [6].

Penelitian oleh Roman Gusmana, dkk (2023) berjudul "Implementasi Algoritma *Hill Cipher* Menggunakan Kunci Matriks 2x2 Dalam Mengamankan Data Teks" meneliti penggunaan algoritma *Hill Cipher* dalam penyandian teks dengan pendekatan kunci matriks ordo 2x2. Hasil dari penelitian menunjukkan bahwa *Hill Cipher* mampu mengenkripsi dan mendekripsi data teks secara efektif dengan karakteristik *ciphertext* yang tidak mudah dikenali, serta fleksibel dalam penggunaannya untuk berbagai bentuk teks. Selain itu, penelitian ini juga

merancang aplikasi simulasi *enkripsi-dekripsi* menggunakan Visual Studio untuk memvisualisasikan proses *algoritmik* tersebut secara praktis [3].

Selanjutnya, penelitian oleh Kamil Hidayatuloh, dkk (2021) dalam jurnal "Perbandingan Metode *Stream Cipher* dan *Hill Cipher* dalam Keamanan Data" membandingkan dua metode enkripsi *Stream Cipher* dan *Hill Cipher*. Hasil penelitian menyimpulkan bahwa *Stream Cipher* unggul dalam hal kecepatan dan efisiensi untuk data yang panjang, sedangkan *Hill Cipher* memiliki keunggulan dari segi kompleksitas dan keamanan matematis karena melibatkan operasi matriks dan *invers* modulus. Penelitian ini memberikan wawasan tentang kelebihan masing-masing algoritma berdasarkan konteks penggunaannya [10].

Penelitian oleh Ahmat Sayuti, dkk (2021) yang berjudul "Perbandingan Performa Algoritma *Hill Cipher* dengan RSA dalam Proses Enkripsi dan Dekripsi Text" melakukan evaluasi performa algoritma *Hill Cipher* dan *RSA* dari sisi waktu *enkripsi* atau dekripsi dan penggunaan memori di perangkat *Android*. Hasilnya menunjukkan bahwa *Hill Cipher* lebih unggul dari segi kecepatan proses, sementara *RSA* lebih efisien dalam penggunaan memori. Penelitian ini menegaskan bahwa pemilihan algoritma sangat bergantung pada kebutuhan sistem dan keterbatasan sumber daya perangkat [9].

Penelitian yang dilakukan oleh Roman Gusmana, dkk (2022) berjudul "Implementasi Algoritma *Affine Cipher* dan *Caesar Cipher* dalam Mengamankan Data Teks" mengusulkan kombinasi dua algoritma klasik untuk meningkatkan keamanan teks. *Affine Cipher* digunakan untuk enkripsi tahap pertama, kemudian hasilnya dienkripsi ulang dengan *Caesar Cipher*. Hasilnya menunjukkan bahwa kombinasi algoritma memberikan peningkatan dalam hal kompleksitas penyandian dan sulit untuk dipecahkan melalui *brute force attack*. Penelitian ini juga dilengkapi dengan desain aplikasi enkripsi atau dekripsi berbasis Visual Studio yang memungkinkan pengguna untuk menguji keamanan data secara praktis [7].

Selanjutnya, Penelitian yang dilakukan oleh Putra, dkk (2024) dalam penelitian berjudul *"*Perbandingan Kriptografi Klasik *Hill Cipher* dengan *Affine Cipher* dalam Pengamanan Data Citra*"* membandingkan efektivitas dua algoritma tersebut untuk data citra digital. Penelitian menggunakan lima gambar berbeda dan

mengukur kualitas hasil enkripsi menggunakan parameter MSE (*Mean Square Error)* dan PSNR *(Peak Signal to Noise Ratio*). Hasilnya menunjukkan bahwa *Hill Cipher* menghasilkan citra terenkripsi yang lebih acak dan sulit dikenali dibandingkan *Affine Cipher*, serta memperoleh nilai MSE dan PSNR yang lebih baik (misalnya MSE: 7073.40 dan PSNR: 9.63 dB). Penelitian ini membuktikan bahwa *Hill Cipher* lebih unggul dalam menyembunyikan informasi visual dibanding *Affine Cipher* [8].

Penelitian yang dilakukan oleh Adinda Fitri, dkk (2023) dari Universitas Islam Negeri Sumatera Utara, dalam jurnal *"*Algoritma *Affine Chiper* pada Enkripsi dan Deskripsi untuk Keamanan Informasi Berbasis Android*"* melakukan pengembangan aplikasi *Android* bernama Cryssage untuk mengimplementasikan algoritma *Affine Cipher* dalam menjaga kerahasiaan informasi pesan. Penelitian ini menyoroti pentingnya keamanan data dalam komunikasi mobile, mengingat kerentanan tinggi terhadap pencurian informasi melalui perangkat *smartphone*. Dengan mengaplikasikan *Affine Cipher*, aplikasi dapat mengenkripsi dan mendekripsi pesan secara efisien, serta menjaga integritas informasi selama proses pengiriman. Hasilnya menunjukkan bahwa kriptografi klasik seperti *Affine Cipher* tetap relevan jika dikemas dalam aplikasi praktis yang mudah digunakan oleh pengguna awam [4].

Penelitian yang dilakukan oleh Ilfi Nur Diana (2022) dalam jurnal *"*Algoritma *Affine Cipher* dan Modifikasi *Affine Cipher* serta Kombinasinya dengan Cipher Transposisi Grup Simetri untuk Mengamankan Pesan Teks*"* melakukan pendekatan super-enkripsi, yaitu menggabungkan algoritma substitusi *Affine Cipher* dan modifikasinya dengan transposisi (*Cipher* Transposisi Grup Simetri). Penelitian ini menunjukkan bahwa kombinasi dua teknik kriptografi klasik mampu memperkuat tingkat keamanan sistem, memperluas ruang kunci, dan menyulitkan analisis kriptografis terhadap pesan yang disandikan [11].

Terakhir, penelitian yang dilakukan Ema Satriana Br Karo (2020) dalam jurnalnya yang berjudul *"*Penerapan Algoritma *Affine Cipher* dan *Electronic Code Book* (ECB) dalam Pengamanan Pesan Teks*"*, menjelaskan tentang penggunaan dua algoritma yaitu *Affine Cipher* sebagai algoritma klasik dan *Electronic Code*

Book (ECB) sebagai metode kriptografi modern untuk mengenkripsi pesan teks. Implementasi dilakukan menggunakan Visual Basic 2008, dan hasil penelitian menunjukkan bahwa proses enkripsi ganda dengan metode tersebut mampu menyamarkan pesan dengan lebih baik dan sulit untuk ditebak. Penelitian ini menekankan pentingnya penggabungan algoritma klasik dengan algoritma modern untuk meningkatkan keamanan data, terutama dalam sistem berbasis komputer yang rentan terhadap penyadapan data [12].

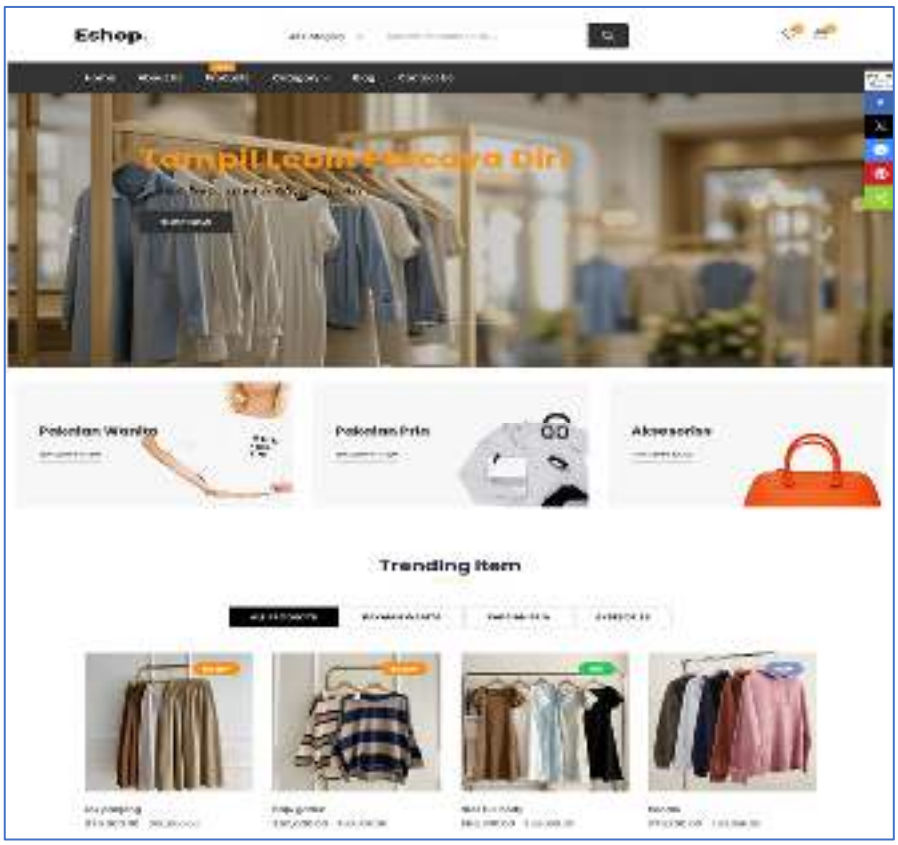
Tabel 2.1 Review Berdasarkan Penelitian Terkait

No	Judul	Penulis	Perbandingan	
			Teknologi	Hasil
1.	Modifikasi Algoritma Affine Cipher untuk Mengamankan Data	Adnan Buyung Nasution (2020)	Modifikasi dan implementasi algoritma secara manual pada desktop	Ciphertext menjadi lebih acak dan sulit ditebak setelah blok data dibalik dan disandikan ulang menggunakan Affine Cipher.
2.	Menerapkan Algoritma Hill Cipher dan Matriks 2x2 dalam Mengamankan File Teks Menggunakan Kode ASCII	Nurharianna Siregar dkk (2022)	Implementasi algoritma menggunakan Python pada platform desktop	Aplikasi dapat menyembunyikan isi pesan dengan aman dari pihak tidak berkepentingan menggunakan Hill Cipher dan kode ASCII.
3.	Implementasi Algoritma Hill Cipher Menggunakan Kunci Matriks 2x2 Dalam Mengamankan Data Teks	Roman Gusmana dkk (2023)	Implementasi dan simulasi menggunakan Visual Studio	Hill Cipher mampu mengenkripsi dan mendekripsi teks secara efektif dengan hasil ciphertext yang kompleks dan sulit dikenali.
4.	Perbandingan Metode Stream Cipher dan Hill Cipher dalam Keamanan Data	Kamil Hidayatuloh dkk (2021)	Studi literatur dan simulasi pada sistem desktop	Stream Cipher unggul dalam kecepatan untuk data panjang, Hill Cipher unggul dari sisi kompleksitas matematis dan keamanan.
5.	Perbandingan Performa Algoritma Hill Cipher dengan RSA dalam Proses Enkripsi dan Dekripsi Text	Ahmat Sayuti dkk (2021)	Implementasi pada perangkat Android	Hill Cipher lebih cepat dalam proses, RSA lebih efisien dalam penggunaan memori.
6.	Implementasi Algoritma Affine Cipher dan Caesar Cipher dalam Mengamankan Data Teks	Roman Gusmana dkk (2022)	Implementasi pada aplikasi desktop menggunakan Visual Studio	Kombinasi Affine dan Caesar Cipher memberikan kompleksitas penyandian yang tinggi dan tahan terhadap brute force attack.
7.	Perbandingan Kriptografi Klasik Hill Cipher dengan Affine Cipher dalam Pengamanan Data Citra	Putra & Astawa (2024)	Eksperimen menggunakan metode MSE & PSNR pada desktop	Hill Cipher menghasilkan citra lebih acak dengan nilai MSE dan PSNR lebih baik dibandingkan Affine Cipher.

No	Judul	Penulis	Perbandingan	
			Teknologi	Hasil
8.	Algoritma Affine Chiper pada Enkripsi dan Deskripsi untuk Keamanan Informasi Berbasis Android	Adinda Fitri dkk (2023)	Pembuatan aplikasi Android berbasis keamanan informasi	Aplikasi Android "Cryssage" menggunakan Affine Cipher berhasil menjaga kerahasiaan pesan dalam komunikasi mobile
9.	Algoritma Affine Cipher dan Modifikasi Affine Cipher serta Kombinasinya dengan Cipher Transposisi Grup Simetri untuk Mengamankan Pesan Teks	Ilfi Nur Diana (2022)	Kombinasi Affine Cipher dan Cipher Transposisi secara manual pada sistem desktop	Kombinasi dua teknik kriptografi klasik memperkuat keamanan sistem dan memperluas ruang kunci agar sulit dipecahkan.
10.	Penerapan Algoritma Affine Cipher dan Electronic Code Book (ECB) dalam Pengamanan Pesan Teks	Ema Satriana Br Karo (2020)	Implementasi algoritma Affine Cipher dan ECB menggunakan Visual Basic pada desktop	Enkripsi ganda Affine + ECB menyamarkan pesan dengan lebih baik dan sulit ditebak, cocok untuk sistem berbasis komputer yang rentan penyadapan.
11.	Perbandingan Algoritma Hill Cipher dan Affine Cipher dalam Keamanan Data pada Website E-Commerce	Ona Mazura (2025)	Implementasi Hill Cipher & Affine Cipher pada web e-commerce open-source, uji brute force dengan Burp Suite.	Membandingkan kecepatan enkripsi–dekripsi dan ketahanan algoritma terhadap serangan brute force.

2.2 Landasan Teori

2.2.1 E-Commerce

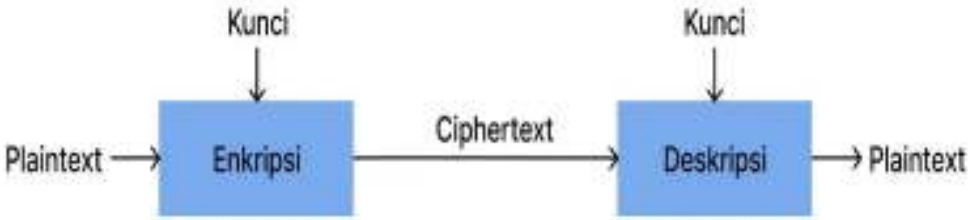


Gambar 2.1 Web Ecommerce

E-commerce merupakan bentuk kegiatan jual beli atau transaksi yang dilakukan melalui media elektronik, terutama internet, yang menjadi penghubung antara penjual dan pembeli untuk melakukan transaksi baik secara tunai maupun non-tunai [14].

Dalam perkembangannya, e-commerce tidak hanya mencakup aktivitas jual beli semata, tetapi juga mencakup seluruh proses bisnis seperti produksi, distribusi, pemasaran, hingga pengiriman barang atau jasa. Dengan dukungan teknologi digital, e-commerce mampu mempertemukan pelaku usaha dan konsumen dari berbagai belahan dunia hanya melalui perangkat elektronik, sehingga memperluas jangkauan pasar secara signifikan [15]. Contoh platform e-commerce terkenal di Indonesia meliputi Tokopedia, Shopee, Bukalapak, dan Lazada.

2.2.2 Kriptografi



Gambar 2.2 Kriptografi [14]

Kriptografi berasal dari bahasa Yunani: "*cryptós*" yang berarti "rahasia" dan "*graphein*" yang berarti "menulis". *Enkripsi* diartikan sebagai "tulisan tersembunyi". *Enkripsi* merupakan ilmu penulisan pesan tersembunyi yang bertujuan untuk menyamarkan arti pesan itu. *Enkripsi* telah berevolusi sebagai tanggapan terhadap tantangan yang dihadapi, sehingga menciptakan beberapa istilah yang digunakan untuk mendeskripsikan aktivitas pengiriman pesan rahasia. Proses mengkodekan pesan dikenal sebagai *enkripsi*, dan ketika pesan sudah dikodekan disebut dekripsi. Pesan yang belum di *enkripsi* atau dibersihkan dikenal sebagai *plaintext,* sementara pesan yang sudah terenkripsi disebut *ciphertext*.

Kriprografi merupakan ilmu yang mempelajari cara menjaga agar data atau pesan aman saat dikirim dari pengirim ke penerima tanpa adanya campur tangan pihak ketiga. Kriptografi ilmu yang mempelajari pengetahuan dan teknologi untuk melindungi keamanan data dan informasi yang dikirim, serta tentang metode untuk mendekripsi pesan yang telah di *enkripsi* (disamarkan). Berbagai istilah dan terminologi teknis sering dijumpai dalam kriptografi. Secara matematis, proses *enkripsi* dilakukan dengan menerapkan fungsi E (*enkripsi*) pada P (*plaintext*) menggunakan e (kunci *enkripsi*) untuk menghasilkan C (*ciphertext*). Proses *Enkripsi* adalah proses mengubah data menjadi bentuk yang aman, sedangkan dekripsi adalah proses untuk membaca data dalam bentuk yang semula [13].

2.2.3 Algoritma

Dunia komputasi, istilah algoritma menjadi fondasi pemikiran suatu Formulasi. Algoritma dapat diartikan sebagai metode penyusunan langkah-langkah

untuk menyelesaikan masalah yang disajikan dalam kalimat dengan jumlah kata terbatas, namun tersusun secara logis dan sistematis.

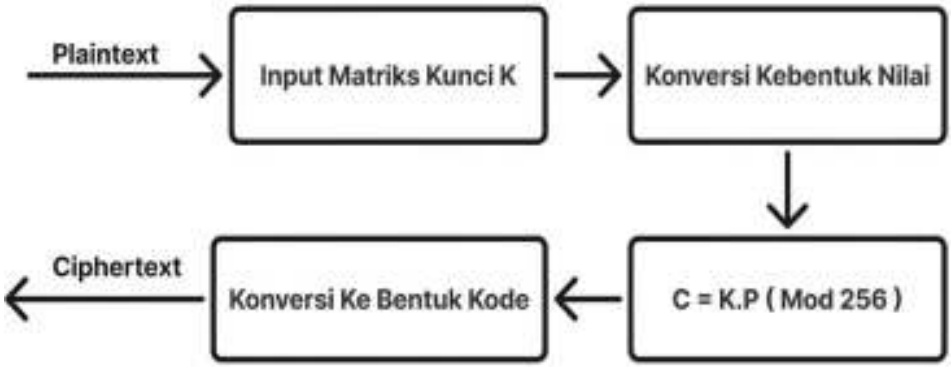
Pernyataan-pernyataan ini bisa diterjemahkan langkah demi langkah dari awal sampai selesai. Algoritma harus berhenti setelah menyelesaikan serangkaian tugas atau langkah yang terbatas, dan setiap langkah harus didefinisikan dengan jelas agar tidak memiliki makna ganda [17].

2.2.4 *Hill Cipher*

Algoritma *Hill Cipher* adalah metode kriptografi klasik dengan kunci simetris yang diterapkan untuk mengenkripsi dan mendekripsi pesan dengan memanfaatkan operasi matriks dan modulo. Metode ini diperkenalkan oleh Lester S. Hill pada tahun 1929, algoritma ini dirancang sebagai pilihan alternatif dari metode kriptografi klasik lainnya seperti *Caesar Cipher, Vigenere Cipher,* dan *Substitusi Cipher*. *Hill Cipher* diajukan untuk memastikan keamanan segmen data dalam basis data yang diterapkan, dengan tujuan memperkuat dan melindungi database. *Hill Cipher* adalah algoritma kunci simetris yang memanfaatkan matriks n x n sebagai kunci. Algoritma *Hill Cipher* memanfaatkan matriks persegi sebagai kunci untuk melaksanakan proses *enkripsi* dan dekripsi. Dasar dari teori matriks yang dipakai dalam *Hill Cipher* adalah operasi matriks perkalian dan invers [17].

2.2.4.1 Teknik Enkripsi dan Dekripsi Hill Cipher

Proses *enkripsi* dalam *Hill Cipher* dilakukan per blok Teks *Plaintext*. Ukuran blok itu setara dengan ukuran matriks kunci. Sebelum membagi teks menjadi serangkaian blok-blok, *Plaintext* terlebih dahulu dikonversi berdasarkan tabel ASCII. secara matematis, proses *enkripsi* dalam Hill Cipher adalah: C = K. P. Proses *enkripsi* secara umum adalah:



Gambar 2.3 Ilustrasi Proses *Enkripsi Hill Cipher* [17]

C = *Ciphertext*

K = Kunci

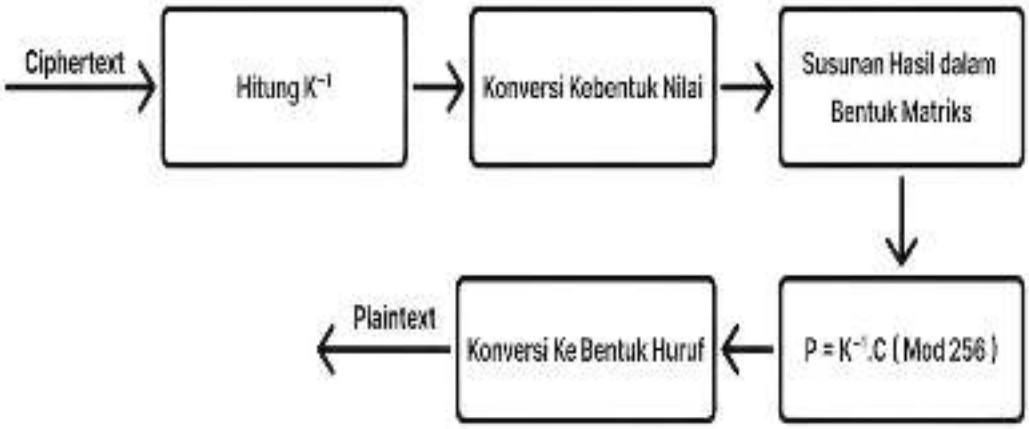
P = *Plaintext*

Langkah-langkah *enkripsi Hill Cipher* sebagai berikut:

1. Tentukan *Plaintext* (teks asli) yang akan disandikan.
2. Menentukan matriks kunci dan merupakan matriks yang *invertible* yaitu memiliki *mulltipliative inverse* K^{-1} sehingga$K.K^{-1} = I$
3. Mengubah *Plaintext* menjadi bentuk numerik sesuai dengan konversi yang telah ditetapkan.
4. Menghitung banyaknya karakter pada *Plainteks.* Selanjutnya banyaknya karakter dari *plainteks* dibagi dengan ordo dari matriks kunci yang telah ditentukan.
5. Susun *plainteks* yang berupa numerik menjadi matriks (2 x 1 jika ordo kunci 2 x 2).
6. Lakukan proses *enkripsi* dengan rumus C = K.P, sehingga diperoleh matriks baru dari hasil perkalian tersebut. Selanjutnya, modulokan hasil perkalian kedua matriks dengan 256.
7. Konversikan menjadi huruf/teks sesuai tabel konversi. Sehingga, diperoleh *ciphertext* atau karakter sandi yang merupakan hasil dari proses *enkripsi*.

Proses dekripsi pada *Hill Cipher* pada dasarnya mirip dengan proses enkripsinya. Namun, matriks kunci perlu di inversi terlebih dahulu. Secara matematis, dekripsi

dalam *Hill Cipher* bisa diperoleh dari persamaan. Persamaan untuk proses dekripsi adalah: $P = K^{-1}.C$. secara umum proses dekripsi yaitu:



Gambar 2.4 Ilustrasi Proses *Dekripsi Hill Cipher* [17]

P = K^{-1} . C

P = *Plaintext*

K^{-1} = *invers* matriks kunci

C = *ciphertext*

Adapun langkah-langkah dekripsi *Hill Cipher* sebagai berikut:

1. Mengubah *Ciphertext* menjadi bentuk numerik sesuai dengan konversi yang telah ditetapkan.
2. Menentukan *invers* matriks kunci dari matriks kunci yang digunakan pada proses *enkripsi*. Dan tentukan nilai determinan matriks kunci $K = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ maka det K = $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$
3. Tinjau lah matriks 2 × 2, K = $\left|\begin{pmatrix} a & b \\ c & d \end{pmatrix}\right.$ Jika ad - bc ≠ 0, maka: $K^{-1} = \frac{1}{ad-bc}\begin{pmatrix} a & -b \\ c & d \end{pmatrix}$ Jika determinan selain 1 atau –1 maka menentukan nilai *invers modulo*.
4. Tentukan nilai *invers modulo* $a^{-1} = 1 \bmod n$

Keterangan:

a^{-1} : nilai determinan matriks kunci (K)

n : bilangan bulat positif atau negatif

mod : modulus atau modulo → sisa bagi

5. Tentukan hasil dekripsi *Hill Cipher* (nilai *invers* matriks kunci × *invers* matriks kunci).
6. Rumus dekripsi *Hill Cipher*: $P = K^{-1} \times C$
7. Menyusun *Ciphertext* yang berupa numerik menjadi suatu matriks. Kalikan *invers* matriks dengan matriks *Ciphertext* dalam modulo 256.
8. Selanjutnya hasil numerik hasil dekripsi, diubah kembali ke teks asli kembali (Plaintext).
9. Pesan dapat dibaca kembali [17].

2.2.5 *Affine Cipher*

Affine cipher merupakan algoritma kriptografi yang termasuk dalam kategori *substitution cipher* yaitu metode kriptografi yang mengganti setiap karakter pada *plaintext* dengan karakter lain yang sesuai berdasarkan rumus matematika. Tidak seperti sandi *Caesar*, yang menggeser karakter teks biasa dengan jumlah tetap, sandi *affine* menggunakan transformasi matematika yang lebih kompleks. *Affine Cipher* memproses ekspresi *enkripsi* dan dekripsi berdasarkan operasi matematika modular. Dalam metode ini, setiap karakter teks biasa diubah menjadi angka, dan perhitungan dilakukan berdasarkan konstanta a dan b, yang akhirnya menghasilkan teks tersandi. Salah satu kelebihan *affine cipher* adalah memungkinkan penggunaan dua parameter kunci (a dan b). Oleh karena itu, sandi ini sedikit lebih kompleks dibandingkan dengan sandi *Caesar* [13].

Rumus enkripsi untuk *affine cipher*

$$E(x) = (a \cdot x + b) \; mod \; 26$$

Di mana:

a) $E(x)$ adalah hasil *enkripsi* (*ciphertext*).
b) x adalah posisi numerik dari karakter dalam alfabet (dengan $A = 0$, $B = 1$, $C = 2$, … , $Z = 25$).
c) a adalah konstanta pengali (harus coprime dengan m).
d) b adalah konstanta penambah.

e) m adalah modulus, yang biasanya 26 untuk alfabet Inggris (untuk huruf besar dan kecil).

Pada tahap dekripsi, rumus yang digunakan

$$D(y) = a - 1 \cdot (y - b) mod\ m$$

Di mana:

a) D(y) adalah hasil dekripsi (*plaintext*).
b) y adalah posisi numerik dari karakter dalam *ciphertext*.
c) $a - 1$ adalah *invers modular* dari a (*invers modular* memenuhi kondisi $a \cdot a - 1 \equiv 1(mod 26)$ [13].

2.2.6 *Ciphertext-Only Attack* (COA)

Ciphertext-Only Attack (COA) merupakan model serangan dalam kriptografi di mana penyerang hanya memiliki akses terhadap ciphertext tanpa mengetahui plaintext maupun kunci enkripsi yang digunakan. Pada pendekatan ini, penyerang berusaha menganalisis pola dan karakteristik statistik ciphertext untuk memperoleh informasi mengenai plaintext atau kelemahan algoritma enkripsi.

Dalam beberapa penelitian, COA digunakan sebagai metode evaluasi keamanan sistem enkripsi dengan mengamati sejauh mana ciphertext mampu menyamarkan informasi asli. Pengujian dilakukan dengan menganalisis kemungkinan keteraturan atau pola tertentu pada ciphertext, sehingga dapat dinilai tingkat ketahanan algoritma terhadap analisis berbasis ciphertext [19].

2.2.7 *Shannon Entropy*

Shannon Entropy merupakan metode pengukuran tingkat keacakan suatu data berdasarkan distribusi probabilitas simbol yang muncul di dalamnya. Dalam bidang kriptografi, *Shannon Entropy* digunakan untuk mengevaluasi kualitas ciphertext hasil enkripsi. Nilai entropy yang tinggi menunjukkan bahwa ciphertext memiliki tingkat keacakan yang baik dan lebih sulit dianalisis secara statistik, sehingga meningkatkan keamanan data terenkripsi [18].

Secara konseptual, *Shannon Entropy* mengukur tingkat ketidakpastian (*uncertainty*) atau penyebaran informasi dalam suatu data. Semakin merata distribusi kemunculan simbol pada ciphertext, maka semakin tinggi nilai *entropy*

yang dihasilkan. Ciphertext dengan distribusi simbol yang mendekati *uniform* menunjukkan minimnya pola statistik, sehingga lebih sulit diprediksi dan dianalisis oleh pihak yang tidak berwenang.

Nilai maksimum *Shannon Entropy* bergantung pada jumlah simbol unik yang digunakan dalam ciphertext dan dirumuskan sebagai

$$\boldsymbol{H = -\sum_{i=1}^{n} P_i \ log_2(P_i)}$$

Keterangan:

H = nilai Shannon Entropy

pi = probabilitas kemunculan karakter ke-i

n = jumlah karakter unik dalam ciphertext

Pada ciphertext yang menggunakan karakter berbasis ASCII atau Base64 dengan sekitar 64 simbol, nilai *entropy* maksimum secara teoritis adalah sekitar 6 bit. Oleh karena itu, nilai *entropy* yang mendekati nilai maksimum tersebut dapat dikategorikan sebagai tingkat keacakan yang tinggi dan mencerminkan kualitas enkripsi yang baik [20].

BAB III
PERANCANGAN

3.1 Data dan Alat Penelitian

Data yang digunakan dalam penelitian ini berupa data pengguna yang umum terdapat pada sistem web e-commerce dan bersifat sensitif, seperti nama, alamat email, dan password. Pemilihan data ini didasarkan pada pentingnya menjaga kerahasiaan informasi pengguna dari pihak yang tidak berwenang.

Data tersebut akan diproses melalui enkripsi dan dekripsi menggunakan algoritma *Hill Cipher* dan *Affine Cipher*. Selanjutnya, ciphertext yang dihasilkan dianalisis untuk menguji efektivitas masing-masing algoritma dalam menjaga keamanan data melalui pendekatan *Ciphertext-Only Attack* (COA) serta pengukuran tingkat keacakan menggunakan *Shannon Entropy*.

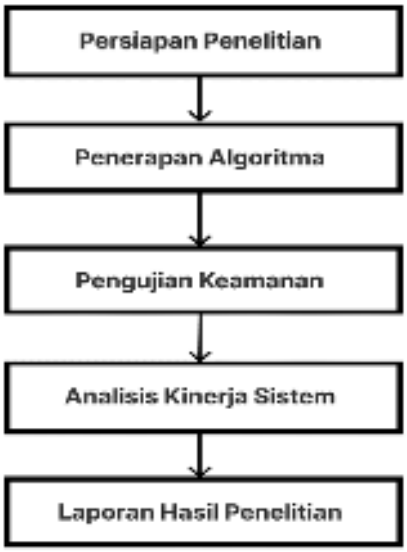
Untuk mendukung proses pengolahan data dan analisis tersebut, penelitian ini menggunakan beberapa alat penelitian yang terdiri dari perangkat keras dan perangkat lunak. Perancangan alat penelitian ini bertujuan untuk memastikan bahwa proses implementasi, pengujian, dan analisis algoritma kriptografi dapat berjalan secara optimal. Adapun kebutuhan alat penelitian yang digunakan dalam penelitian ini dibagi menjadi dua bagian, yaitu sebagai berikut.

1. Kebutuhan *Hardware*
   a) Processor intel(R) Core(TM) i3-8130U CPU @ 2.20GHz 2.21 GHz
   b) Random Acess Memory (RAM) 8 GB DDR3 1600MHz
2. Kebutuhan *Software*
   a) Sistem operasi : Windows 10 Pro
   b) Bahasa pemrograman : PHP, Laravel
   c) Text Editor : Microsoft Word 2021, Visual Studio Code
   d) Database : MySQL

3.2 Prosedur Penelitian

3.2.1 Tahapan Penelitian

Gambaran alur penelitian dalam penelitian ini disusun untuk menerapkan tahapan yang telah dirumuskan sebelumnya dalam rangka membandingkan algoritma *Hill Cipher* dan *Affine Cipher* pada pengamanan data situs web e-commerce. Sistem uji coba berbasis web yang digunakan merupakan website *open-source* yang telah dimodifikasi agar mendukung proses enkripsi dan dekripsi sesuai dengan kebutuhan penelitian. Berikut merupakan gambaran sistem perbandingan algoritma *hill cipher* dan *affine cipher* dalam menjaga keamanan data pada website e-commerce.



Gambar 3.1 Desain Sistem Tahapan Penelitian

Adapun langkah-langkah yang dilakukan dalam penelitian ini terdiri dari beberapa tahap penting yang disusun secara sistematis sebagai berikut:
a. Persiapan Penelitian

Tahap ini merupakan fondasi awal dari keseluruhan proses penelitian. Peneliti melakukan berbagai kegiatan penting untuk mempersiapkan proses implementasi dan pengujian algoritma kriptografi. Persiapan yang dilakukan meliputi beberapa aspek berikut

1. Identifikasi Masalah

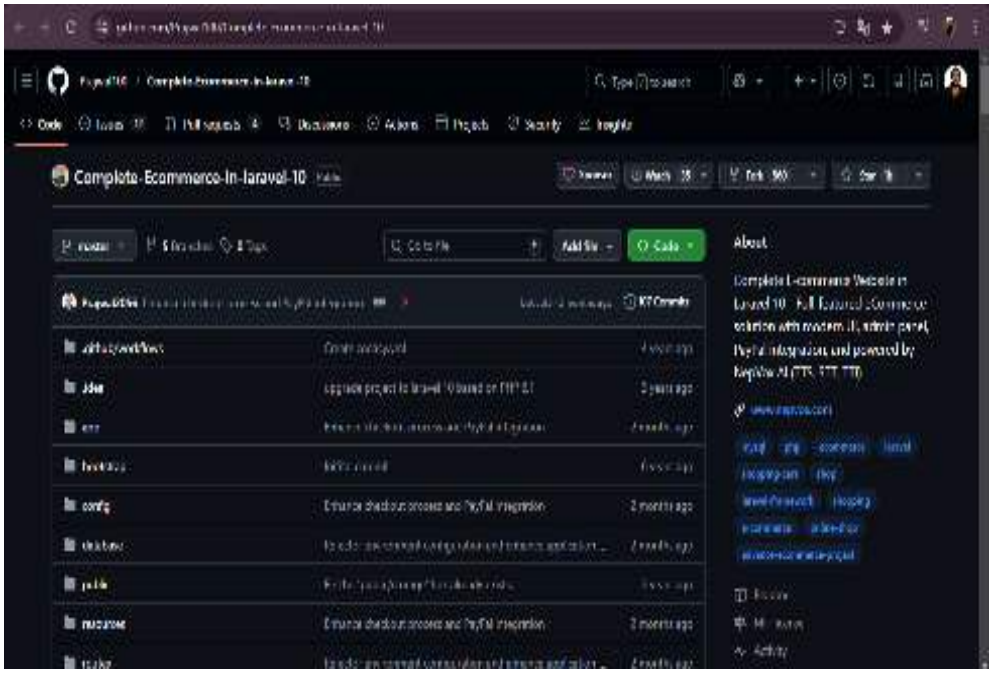
   Tahap ini merupakan langkah awal dari penelitian yang dilakukan, peneliti membuat rumusan masalah yang ditemukan dikarenakan belum diketahui algoritma kriptografi klasik mana yang lebih efektif dalam menjaga keamanan data pada platform e-commerce.

2. Kajian Literatur

   Pada tahap kajian literatur, peneliti melakukan pencarian dan pengumpulan berbagai referensi yang berkaitan dengan kriptografi, terutama algoritma *Hill Cipher, Affine Cipher*, serta *Shannon Entropy* dan *Ciphertext-Only Attack*. Tujuan dari pengumpulan literatur ini adalah untuk memperoleh pemahaman teoritis yang mendalam mengenai cara kerja masing-masing algoritma, karakteristik keamanannya, serta pendekatan yang digunakan untuk mengukur tingkat keacakan ciphertext dan ketahanan algoritma terhadap analisis berbasis ciphertext.

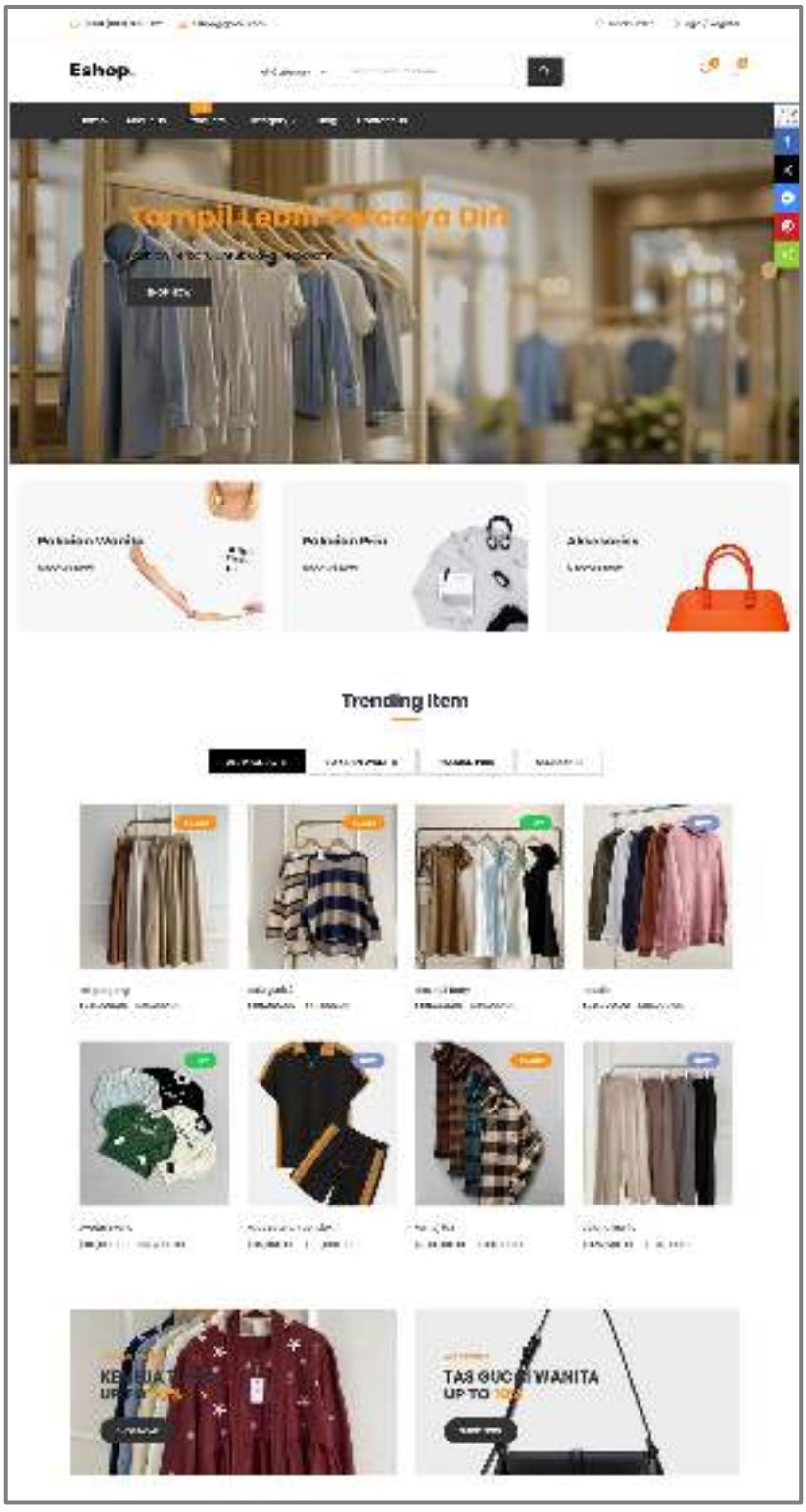
3. Pemilihan Objek Penelitian

   Pada tahap ini, peneliti melakukan pencarian dan pemilihan terhadap beberapa website e-commerce *open-source* dari GitHub untuk digunakan sebagai media simulasi penerapan algoritma *Hill Cipher* dan *Affine Cipher*. Website yang dipilih harus memenuhi beberapa kriteria, seperti bersifat open-source, memiliki struktur kode yang jelas dan mudah dimodifikasi, berbasis PHP atau Laravel, serta mendukung input dan penyimpanan data pengguna. Setelah menjalani evaluasi dengan kriteria tersebut, dipilih satu website e-commerce yang dianggap paling sesuai dan dikonfigurasi ke dalam dua proyek terpisah: satu untuk menerapkan algoritma *Hill Cipher* dan satu lagi untuk *Affine Cipher*. Pemisahan ini dilakukan agar proses pengujian tetap fokus dan hasil perbandingan antar algoritma dapat dianalisis secara objektif.

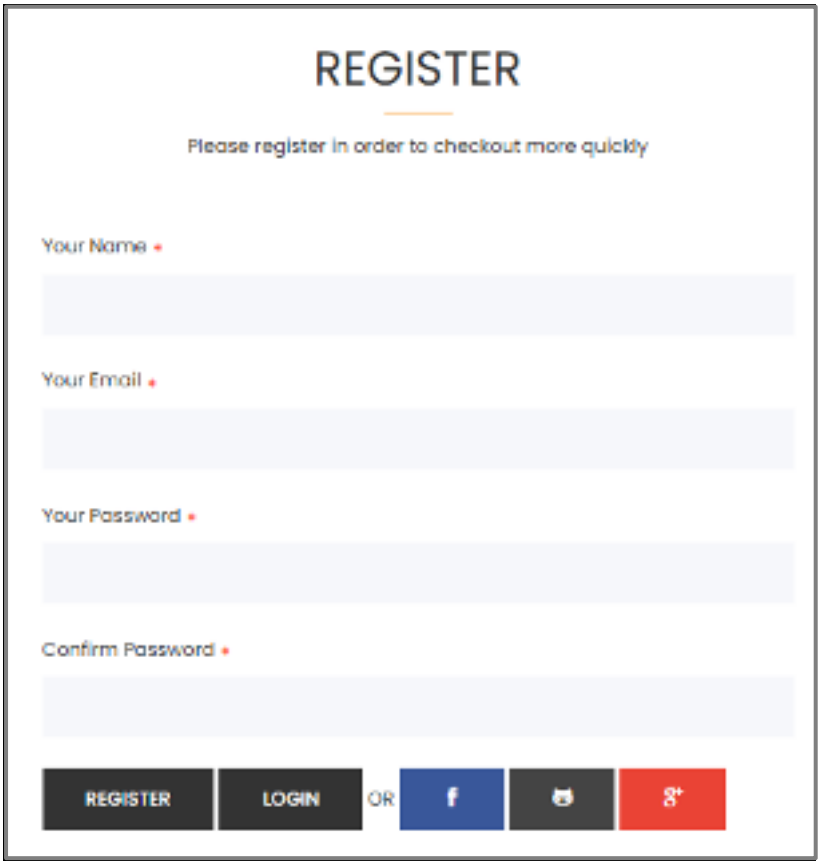


Gambar 3.2 Repository *Complete E-Commerce in Laravel 10* pada GitHub.

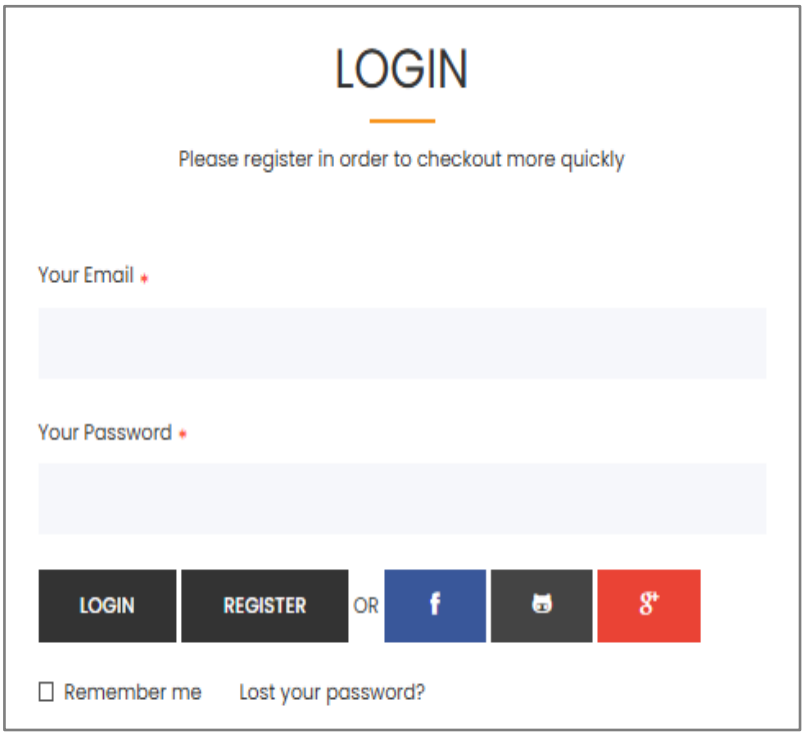
a. Tampilan Website



Gambar 3.3 Halaman Home

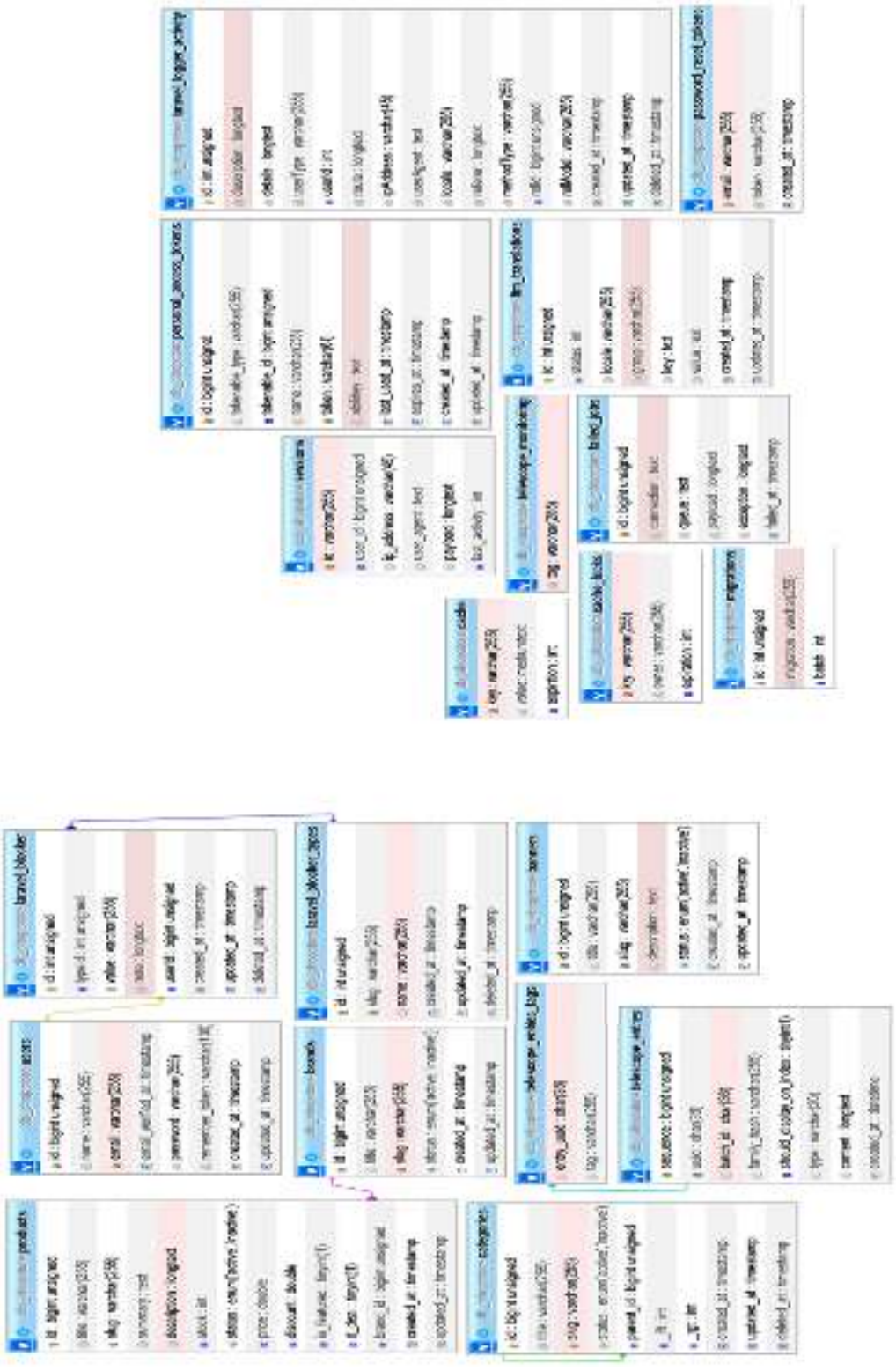


Gambar 3.4 Halaman Register



Gambar 3.5 Halaman Login User

b. Database Website



Gambar 3.6 Struktur Database

1. Tabel users

Tabel ini menyimpan data utama pengguna sistem, baik admin maupun customer. Informasi yang disimpan mencakup identitas pengguna seperti *name*, *email*, dan *password*. Selain itu, terdapat kolom *email_verified_at* untuk mencatat status verifikasi email, serta *remember_token* untuk mendukung autentikasi berkelanjutan.

2. Tabel products

Tabel ini digunakan untuk menyimpan data produk yang tersedia di toko online. Informasi yang dicatat mencakup *title* produk, *slug* untuk URL unik, *description*, *price*, *discount*, *stock*, serta status produk (*active* atau *inactive*). Produk juga terhubung dengan kategori dan brand tertentu melalui relasi ke tabel *categories* dan *brands*.

3. Tabel categories

Berfungsi untuk menyimpan data kategori produk. Tabel ini mendukung struktur *parent_id* sehingga memungkinkan pembentukan kategori dan sub-kategori secara hierarkis. Informasi tambahan meliputi *slug*, *status*, serta *created_at* dan *updated_at* untuk pencatatan waktu.

4. Tabel brands

Menyimpan data merek produk (brand) yang tersedia dalam sistem. Setiap produk dapat dihubungkan dengan satu brand tertentu melalui relasi *brand_id*.

5. Tabel banners

Digunakan untuk menyimpan data banner promosi yang akan ditampilkan pada halaman utama website. Informasi mencakup *title*, *slug*, *description*, *status*, serta waktu pembuatan dan pembaruan data.

6. Tabel sessions

Mencatat informasi sesi pengguna yang sedang aktif di sistem. Data yang disimpan mencakup *user_id*, alamat IP (*ip_address*), serta payload aktivitas untuk mendukung monitoring keamanan dan akses.

7. Tabel personal_access_tokens

Tabel ini menyimpan token autentikasi berbasis API untuk pengguna, mendukung integrasi sistem dengan layanan eksternal atau mobile apps.

8. Tabel password_reset_tokens

   Berfungsi menyimpan token reset password ketika pengguna melakukan permintaan untuk mengganti kata sandi.

9. Tabel migrations

   Menyimpan catatan seluruh migrasi database yang telah dijalankan oleh sistem Laravel. Hal ini penting untuk manajemen versi database.

10. Tabel failed_jobs

    Berisi log kegagalan eksekusi *queue jobs* pada sistem, digunakan untuk memantau error dalam pemrosesan background task.

11. Tabel cache dan cache_locks

    Digunakan untuk menyimpan data sementara (cache) yang mempercepat proses akses data dalam sistem. *cache_locks* digunakan untuk mengelola kunci akses pada data cache agar tidak terjadi konflik.

12. Tabel telescope_entries dan telescope_monitoring

    Menyimpan data monitoring dan debugging dari Laravel Telescope. Informasi ini berguna bagi developer untuk memantau aktivitas sistem, query, dan log error.

13. Tabel laravel_logger_activity

    Mencatat seluruh aktivitas pengguna dalam sistem, termasuk *user_id*, *route*, *ipAddress*, *userAgent*, serta waktu aktivitas. Hal ini digunakan untuk audit trail dan keamanan sistem.

1. Penerapan Algoritma

Setelah sistem simulasi web e-commerce berhasil disiapkan, langkah selanjutnya adalah menerapkan algoritma kriptografi klasik pada masing-masing sistem. Penelitian ini menerapkan dua algoritma kriptografi yaitu *Hill Cipher* dan *Affine Cipher*, dengan pendekatan penerapan yang berbeda meskipun keduanya diterapkan dalam satu website yang sama. Pemisahan ini dilakukan untuk memastikan setiap algoritma diuji secara terfokus tanpa saling memengaruhi hasil.

1. Perhitungan Hill Cipher

Algoritma *Hill Cipher* diterapkan pada sistem e-commerce berbasis web untuk menjaga kerahasiaan data pengguna yang tersimpan dalam database. Penerapan algoritma *Hill Cipher* difokuskan pada bagaimana data sensitif seperti nama, email, dan *password* dapat dienkripsi sebelum disimpan, sehingga isi data tidak dapat dipahami langsung oleh pihak yang tidak berwenang. Proses algoritma *Hill Cipher* dilakukan dengan cara mengubah setiap karakter *plainteks* menjadi bilangan ASCII, kemudian mengalikan setiap blok plainteks dengan matriks kunci, dan hasilnya dimodulo 256 untuk memperoleh *ciphertext*. Proses dekripsi dilakukan dengan mengalikan invers matriks kunci dengan *ciphertext* sehingga *plainteks* dapat dipulihkan kembali.

Pada penelitian ini, matriks kunci yang digunakan adalah matriks yang memiliki determinan 1 atau –1 agar dapat dilakukan proses invers. Sistem secara otomatis menyimpan hasil *enkripsi* ke dalam *database*, sehingga pada saat dilakukan proses login atau autentikasi, sistem akan memanggil *ciphertext* tersebut dan melakukan dekripsi sesuai kunci yang tersedia. Dengan cara ini, proses *enkripsi* dan *dekripsi* dapat berjalan secara otomatis dalam aplikasi. Contoh dibuat yaitu berupa data *record* dalam *database* yang terdiri dari beberapa tabel dan beberapa field. Jika sebuah *database* dengan nama db_shopstore mempunyai tabel users yang terdiri dari atribut dan field sebagai berikut:

Tabel 3. 1 Data Penguna

Username	Password	Email
ONA MAZURA	Onaaa123#	Onaaja@gmail.com

Pada tabel 3.1 ini berupa record dari database db_shopstore pada tabel users, kemudian akan dikonversikan terlebih dahulu menjadi bilangan decimal berdasarkan kode ASCII 256.

Tabel 3.2 Kode ASCII Data Penguna

Username	Password	Email
79, 78, 65, 77, 65, 90, 85, 82, 65.	79, 110, 97, 97, 97, 49, 50, 51, 35.	79, 110, 97, 97, 106, 97, 64, 103, 109, 97, 105, 108, 46, 99, 111, 109.

Plainteks yang telah dikonversikan ke dalam angka desimal pada tabel 3.3 akan dienkripsi dengan menggunakan teknik *Hill Cipher* dengan kunci K yang merupakan matriks 2x2. Setiap *plainteks* yang telah dikonversikan akan dibagi per blok, dan setelah itu tiap blok akan dienkripsi dengan kunci K.

a. Perhitungan Enkripsi Algoritma Hill Cipher

1. Enkripsi Data Penguna

Field pertama Username dengan *Plainteks* ONA MAZURA maka *Plainteks* akan dibagi perblok yang masing-masing bloknya berukuran 2 karakter sehingga perhitungan *enkripsi* adalah sebagai berikut:

Tabel 3.3 Enkripsi *Hill Cipher* Data Pengguna Field 1

1	2	3	4	5
79, 78	65, 77	65, 90	85, 82	65, 88

Blok 1

$$C^1 = \boldsymbol{K}.\boldsymbol{P^1}$$

$$= \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix} \begin{pmatrix} 79 \\ 78 \end{pmatrix} Mod\ 256$$

$$= \begin{pmatrix} 313 \\ 1.643 \end{pmatrix} Mod\ 256$$

Setelah itu hasil di mod 256 seperti berikut:

= 313 mod 256 = 57

= 1.643 mod 256 = 107

Karakter yang terkorespondensi dengan 57 dan 107 adalah 9 dan k

Blok 2

$C^2 = \boldsymbol{K}.\boldsymbol{P^2}$

$= \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix} \begin{pmatrix} 65 \\ 77 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 296 \\ 1.557 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 296 mod 256 = 40

= 1.557 mod 256 = 21

Karakter yang terkorespondensi dengan 40 dan 21 adalah (dan NAK

Blok 3

$C^3 = \boldsymbol{K}.\boldsymbol{P^3}$

$= \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix} \begin{pmatrix} 65 \\ 90 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 335 \\ 1.765 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 335 mod 256 = 79

= 1.765 mod 256 = 229

Karakter yang terkorespondensi dengan 79 dan 229 adalah O dan ….(kosong)

Blok 4

$C^4 = \boldsymbol{K}.\boldsymbol{P^4}$

$= \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix} \begin{pmatrix} 85 \\ 82 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 331 \\ 1.737 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 331 mod 256 = 75

= 1.737 mod 256 = 201

Karakter yang terkorespondensi dengan 75 dan 201 adalah K dan É

Blok 5

$C^5 = \boldsymbol{K}.\boldsymbol{P^5}$

$= \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix} \begin{pmatrix} 65 \\ 88 \end{pmatrix}$

$= \begin{pmatrix} 329 \\ 1.733 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 329 mod 256 = 73

= 1.733 mod 256 = 197

Karakter yang terkorespondensi dengan 73 dan 197 adalah I dan Å. *Enkripsi* yang didapatkan dari *Username* ONA MAZURA adalah 9k(NAKOKÉIÅ

Field kedua *Password* dengan *Plainteks* Onaaa123# maka *Plainteks* akan dibagi perblok yang masing-masing bloknya berukuran 2 karakter sehingga perhitungan *enkripsi* adalah sebagai berikut:

Tabel 3.4 Enkripsi *Hill Cipher* Data Pengguna Field 2

1	2	3	4	5
79, 110	97, 97	97, 49	50, 51	35, 88

Blok 1

$C^1 = \boldsymbol{K}.\boldsymbol{P^1}$

$= \begin{pmatrix} 3 & 5 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 79 \\ 110 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 787 \\ 928 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 787 mod 256 = 19

= 928 mod 256 = 160

Karakter yang terkorespondensi dengan 19 dan 160 adalah DC3 dan

Blok 2

$C^2 = \boldsymbol{K}.\boldsymbol{P^2}$

$= \begin{pmatrix} 3 & 5 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 97 \\ 97 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 776 \\ 873 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 776 mod 256 = 8

= 873 mod 256 = 105

Karakter yang terkorespondensi dengan 8 dan 105 adalah BS dan i

Blok 3

$C^3 = \boldsymbol{K}.\boldsymbol{P^3}$

$= \begin{pmatrix} 3 & 5 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 97 \\ 49 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 536 \\ 537 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 536 mod 256 = 24

= 537 mod 256 = 25

Karakter yang terkorespondensi dengan 24 dan 25 adalah CAN dan EM

Blok 4

$C^4 = \boldsymbol{K}.\boldsymbol{P^4}$

$= \begin{pmatrix} 3 & 5 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 50 \\ 51 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 405 \\ 457 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 405 mod 256 = 149

= 457 mod 256 = 201

Karakter yang terkorespondensi dengan 149 dan 201 adalah • dan É

Blok 5

$C^5 = \boldsymbol{K}.\boldsymbol{P^5}$

$= \begin{pmatrix} 3 & 5 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 35 \\ 88 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 545 \\ 686 \end{pmatrix} Mod\ 2\ 467$

Setelah itu hasil di mod 256 seperti berikut:

= 545 mod 256 = 33

= 686 mod 256 =174

Karakter yang terkorespondensi dengan 73 dan adalah ! dan ®. Enkripsi yang didapatkan dari Password Onaaa123# adalah DC3BSiCANEM•É!®

Field ketiga Email dengan Plainteks Onaaja@gmail.com maka Plainteks akan dibagi perblok yang masing-masing bloknya berukuran 2 karakter sehingga perhitungan enkripsi adalah sebagai berikut:

Tabel 3.5 Enkripsi *Hill Cipher* Data Pengguna Field 3

1	2	3	4	5	6	7	8
79, 110	97, 97	106, 97	64, 103	109, 97	105, 108	46, 99	111, 109

Blok 1

$C^1 = \boldsymbol{K}.\boldsymbol{P^1}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 79 \\ 110 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 1.151 \\ 1.103 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.151 mod 256 = 127

= 1.103 mod 256 = 79

Karakter yang terkorespondensi dengan 127 dan 79 adalah DEL dan O

Blok 2

$C^2 = \boldsymbol{K}.\boldsymbol{P^2}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 97 \\ 97 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 1.261 \\ 1.164 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.261 mod 256 = 237

= 1.164 mod 256 = 140

Karakter yang terkorespondensi dengan 237 dan 140 adalah í dan CE

Blok 3

$C^3 = \boldsymbol{K}.\boldsymbol{P^3}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 106 \\ 97 \end{pmatrix} Mod\ 256$

$= \begin{pmatrix} 1.342 \\ 1.227 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.342 mod 256 = 62

= 1.227 mod 256 = 203

Karakter yang terkorespondensi dengan 62 dan 203 adalah > dan Ë

Blok 4

$C^4 = \boldsymbol{K}.\boldsymbol{P^4}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 64 \\ 103 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 988 \\ 963 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 988 mod 256 = 220

= 963 mod 256 = 195

Karakter yang terkorespondensi dengan 220 dan 195 adalah Ü dan Ã

Blok 5

$C^5 = \boldsymbol{K}.\boldsymbol{P^5}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 109 \\ 97 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 1.369 \\ 1.248 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.369 mod 256 = 89

= 1.248 mod 256 =22

Karakter yang terkorespondensi dengan 89 dan 224 adalah Y dan à.

Blok 6

$C^6 = \boldsymbol{K}.\boldsymbol{P^6}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 105 \\ 108 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 1.377 \\ 1.275 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.377 mod 256 = 97

= 1.275 mod 256 =251

Karakter yang terkorespondensi dengan 97 dan 251 adalah a dan û

Blok 7

$C^7 = \boldsymbol{K}.\boldsymbol{P^7}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 46 \\ 99 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 810 \\ 817 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 810 mod 256 = 42

= 817 mod 256 = 49

Karakter yang terkorespondensi dengan 42 dan 49 adalah * dan 1.

Blok 8

$C^8 = \boldsymbol{K}.\boldsymbol{P^8}$

$= \begin{pmatrix} 9 & 4 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 111 \\ 109 \end{pmatrix} \ Mod\ 256$

$= \begin{pmatrix} 1.435 \\ 1.322 \end{pmatrix} Mod\ 256$

Setelah itu hasil di mod 256 seperti berikut:

= 1.435 mod 256 = 155

= 1.322 mod 256 =42

Karakter yang terkorespondensi dengan 155 dan 42 adalah › dan *.

Enkripsi yang didapatkan dari Email Onaaja@gmail.com adalah DELOíCE>ËÜÃYàaû*1›*.

2. Dekripsi Data Penguna

Dari *ciphertext* yang dihasilkan terlihat bahwa algoritma *Hill Cipher* menghasilkan *ciphertext* yang tidak memiliki pola yang mirip dengan Plaintextnya. Proses dekripsi pada *Hill Cipher* hampir sama dengan proses *enkripsi*, tetapi matriks kunci harus dibalik (*invers*) terlebih dahulu, secara sistematis, proses dekripsi pada algoritma *Hill Cipher* sebagai berikut:

$$C = \boldsymbol{K}.\boldsymbol{P}$$

$$K^{-1}.C = \boldsymbol{K^{-1}}\ .\boldsymbol{K}.\boldsymbol{P}$$

$$K^{-1}.C = \boldsymbol{I}.\boldsymbol{P}$$

$$P = \boldsymbol{K^{-1}}\ .\boldsymbol{C}$$

Menjadi persamaan deskripsi:

$$\mathrm{P} = \boldsymbol{K^{-1}}\ .\boldsymbol{C}$$

Dengan menggunakan kunci *matriks*

$$K = \begin{pmatrix} 1 & 3 \\ 5 & 16 \end{pmatrix}$$

Maka:

Menghitung nilai determinan kunci (K) dan menghitung invers *matriks* kunci K^{-1}

$$\mathrm{Det(K)} = (1.16)-(3.5) = 16 - 15 = 1$$

$$\boldsymbol{K^{-1}} = \frac{1}{det(K)}\begin{pmatrix} 16 & -3 \\ -5 & 1 \end{pmatrix} = \begin{pmatrix} 16 & -3 \\ -5 & 1 \end{pmatrix} = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix} (Mod\ 256)$$

Maka

$$K^{-1} = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix}$$

Field pertama yang dimulai untuk melakukan dekripsi yaitu *ciphertext* dari Username 9k(NAKOKÉIÅ maka akan didekripsikan sebagai berikut:

Tabel 3.6 Dekripsi *Hill Cipher* Data Pengguna Field 1

1	2	3	4	5
9 k	(NAK	O	K É	I Å

Setiap huruf pada ciphetext yang berupa karakter kode ASCII dikonversikan ke dalam bentuk desimal.

$$C_1 = \begin{pmatrix} 9 \\ k \end{pmatrix} = \begin{pmatrix} 57 \\ 107 \end{pmatrix} \qquad C_2 = \begin{pmatrix} (\\ NAK \end{pmatrix} = \begin{pmatrix} 40 \\ 21 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} O \end{pmatrix} = \begin{pmatrix} 79 \\ 229 \end{pmatrix} \qquad C_4 = \begin{pmatrix} K \\ \text{É} \end{pmatrix} = \begin{pmatrix} 75 \\ 201 \end{pmatrix}$$

$$C_5 = \begin{pmatrix} \text{I} \\ \text{Å} \end{pmatrix} = \begin{pmatrix} 73 \\ 197 \end{pmatrix}$$

Selanjutnya ciphertext didekripsi dengan menggunakan kunci $K^{-1} = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix}$. Proses dekripsi dilakukan blok perblok seperti proses enkripsi.

Blok 1

$$C_1 = \begin{pmatrix} 9 \\ k \end{pmatrix} = \begin{pmatrix} 57 \\ 107 \end{pmatrix}$$

Dekripsi:

$$P_1 = K^{-1}.C$$

$$P_1 = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix}\begin{pmatrix} 57 \\ 107 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 27.983 \\ 14.414 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 27.983 mod 256 = 79

= 14.414 mod 256 = 78

Karakter yang terkorespondensi dengan 79 dan 78 adalah O dan N

Blok 2

$$C_2 = \begin{pmatrix} (\\ NAK \end{pmatrix} = \begin{pmatrix} 40 \\ 21 \end{pmatrix}$$

Dekripsi:

$$P_2 = K^{-2}.C$$

$$P_2 = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix} \begin{pmatrix} 40 \\ 21 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 5.953 \\ 10.061 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 5.953 mod 256 = 65

= 10.061 mod 256 = 77

Karakter yang terkorespondensi dengan 65 dan 77 adalah A dan M

Blok 3

$$C_3 = \begin{pmatrix} 0 \end{pmatrix} = \begin{pmatrix} 79 \\ 229 \end{pmatrix}$$

Dekripsi:

$$P_3 = K^{-3}.C$$

$$P_3 = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix} \begin{pmatrix} 79 \\ 229 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 59.201 \\ 20.058 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 59.201 mod 256 = 65

= 20.058 mod 256 = 90

Karakter yang terkorespondensi dengan 65 dan 90 adalah A dan Z

Blok 4

$$C_4 = \begin{pmatrix} K \\ \text{É} \end{pmatrix} = \begin{pmatrix} 75 \\ 201 \end{pmatrix}$$

Dekripsi:

$$P_4 = K^{-4}.C$$

$$P_4 = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix} \begin{pmatrix} 75 \\ 201 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 52.053 \\ 19.026 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 52.053 mod 256 = 85

= 19.026 mod 256 = 82

Karakter yang terkorespondensi dengan 85 dan 82 adalah U dan R

Blok 5

$$C_5 = \begin{pmatrix} I \\ \text{Å} \end{pmatrix} = \begin{pmatrix} 73 \\ 197 \end{pmatrix}$$

Dekripsi:

$$P_5 = K^{-5}.C$$

$P_5 = \begin{pmatrix} 16 & 253 \\ 251 & 1 \end{pmatrix} \begin{pmatrix} 73 \\ 197 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 51.009 \\ 18.520 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 51.009 mod 256 = 65

= 18.520 mod 256 = 88

Karakter yang terkorespondensi dengan 65 dan 88 adalah A dan X . Dekripsi yang didapatkan dari Username 9k(NAKOKÉIÅ adalah ONA MAZURAX.

Field kedua yang dimulai untuk melakukan dekripsi yaitu ciphertext dari Password DC3BSiCANEM•É!® maka akan didekripsikan sebagai berikut:

Tabel 3.7 Dekripsi *Hill Cipher* Data Pengguna Field 2

1	2	3	4	5
DC3	BS i	CAN EM	• É	! ®

Setiap huruf pada ciphetext yang berupa karakter kode ASCII dikonversikan ke dalam bentuk desimal.

$$C_1 = \begin{pmatrix} DC3 \end{pmatrix} = \begin{pmatrix} 19 \\ 160 \end{pmatrix} \qquad C_2 = \begin{pmatrix} BS \\ i \end{pmatrix} = \begin{pmatrix} 8 \\ 105 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} CAN \\ EM \end{pmatrix} = \begin{pmatrix} 24 \\ 25 \end{pmatrix} \qquad C_4 = \begin{pmatrix} \bullet \\ \text{É} \end{pmatrix} = \begin{pmatrix} 149 \\ 201 \end{pmatrix}$$

$$C_5 = \begin{pmatrix} ! \\ \text{Å} \end{pmatrix} = \begin{pmatrix} 33 \\ 174 \end{pmatrix}$$

Selanjutnya *ciphertext* didekripsi dengan menggunakan kunci $K^{-1} = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix}$. Proses dekripsi dilakukan blok perblok seperti proses enkripsi.

Blok 1

$$C_1 = \begin{pmatrix} DC3 \end{pmatrix} = \begin{pmatrix} 19 \\ 160 \end{pmatrix}$$

Dekripsi:

$$P_1 = K^{-1}.C$$

$$P_1 = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix} \begin{pmatrix} 19 \\ 160 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 35.663 \\ 40.814 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 35.663 mod 256 = 79

= 40.814 mod 256 = 110

Karakter yang terkorespondensi dengan 79 dan 110 adalah O dan n

Blok 2

$$C_2 = \begin{pmatrix} BS \\ i \end{pmatrix} = \begin{pmatrix} 8 \\ 105 \end{pmatrix}$$

Dekripsi:

$$P_2 = K^{-2}.C$$

$$P_2 = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix} \begin{pmatrix} 8 \\ 105 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 22.881 \\ 25.953 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 22.881 mod 256 = 97

= 25.953 mod 256 = 97

Karakter yang terkorespondensi dengan 97 dan 97 adalah a dan a

Blok 3

$$C_3 = \begin{pmatrix} CAN \\ EM \end{pmatrix} = \begin{pmatrix} 24 \\ 25 \end{pmatrix}$$

Dekripsi:

$$P_3 = K^{-3}.C$$

$$P_3 = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix} \begin{pmatrix} 24 \\ 25 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 8.033 \\ 10.289 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 8.033 mod 256 = 97

= 10.289 mod 256 = 49

Karakter yang terkorespondensi dengan 97 dan 49 adalah a dan 1

Blok 4

$$C_4 = \begin{pmatrix} \dot{\text{E}} \end{pmatrix} = \begin{pmatrix} 149 \\ 201 \end{pmatrix}$$

Dekripsi:

$$P_4 = K^{-4}.C$$

$P_4 = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix} \begin{pmatrix} 149 \\ 201 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 59.442 \\ 74.547 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 59.442 mod 256 = 50

= 74.547 mod 256 = 51

Karakter yang terkorespondensi dengan 50 dan 51 adalah 2 dan 3

Blok 5

$C_5 = \begin{pmatrix} ! \\ ® \end{pmatrix} = \begin{pmatrix} 33 \\ 174 \end{pmatrix}$

Dekripsi:

$P_5 = K^{-5}.C$

$P_5 = \begin{pmatrix} 117 & 209 \\ 186 & 233 \end{pmatrix} \begin{pmatrix} 33 \\ 174 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 46.680 \\ 40.389 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 46.680 mod 256 = 35

= 40.389 mod 256 = 88

Karakter yang terkorespondensi dengan 35 dan 88 adalah # dan X . Dekripsi yang didapatkan dari Password DC3BSiCANEM•É!® adalah Onaaa123#X.

Field ketiga yang dimulai untuk melakukan dekripsi yaitu ciphertext dari Email DELOíCE>ËÜÃYàaû*1›* maka akan didekripsikan sebagai berikut:

Tabel 3.8 Dekripsi *Hill Cipher* Data Pengguna Field 3

1	2	3	4	5	6	7	8
DEL O	Í CE	> Ë	Ü Ã	Y à	a û	* 1	› *

Setiap huruf pada *ciphetext* yang berupa karakter kode ASCII dikonversikan ke dalam bentuk desimal.

$$C_1 = \begin{pmatrix} \text{DEL} \\ O \end{pmatrix} = \begin{pmatrix} 127 \\ 79 \end{pmatrix} \qquad C_2 = \begin{pmatrix} \text{Í} \\ \text{CE} \end{pmatrix} = \begin{pmatrix} 237 \\ 140 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} > \\ \text{Ë} \end{pmatrix} = \begin{pmatrix} 62 \\ 203 \end{pmatrix} \qquad C_4 = \begin{pmatrix} \text{Ü} \\ \text{Ã} \end{pmatrix} = \begin{pmatrix} 220 \\ 195 \end{pmatrix}$$

$$C_5 = \begin{pmatrix} \text{Y} \\ \text{à} \end{pmatrix} = \begin{pmatrix} 89 \\ 224 \end{pmatrix} \qquad C_6 = \begin{pmatrix} \text{a} \\ \text{û} \end{pmatrix} = \begin{pmatrix} 97 \\ 251 \end{pmatrix}$$

$$C_7 = \begin{pmatrix} * \\ 1 \end{pmatrix} = \begin{pmatrix} 42 \\ 49 \end{pmatrix} \qquad C_8 = \begin{pmatrix} › \\ * \end{pmatrix} = \begin{pmatrix} 155 \\ 42 \end{pmatrix}$$

Selanjutnya *ciphertext* didekripsi dengan menggunakan kunci $K^{-1} = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix}$. Proses dekripsi dilakukan blok perblok seperti proses enkripsi.

Blok 1

$$C_1 = \begin{pmatrix} DEL \\ O \end{pmatrix} = \begin{pmatrix} 127 \\ 79 \end{pmatrix}$$

Dekripsi:

$$P_1 = K^{-1}.C$$

$$P_1 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 127 \\ 79 \end{pmatrix} \text{ Mod } 256$$

$= \begin{pmatrix} 27.727 \\ 22.894 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 27.727 mod 256 = 79

= 22.894 mod 256 = 110

Karakter yang terkorespondensi dengan 79 dan 110 adalah O dan n

Blok 2

$C_2 = \begin{pmatrix} \acute{I} \\ CE \end{pmatrix} = \begin{pmatrix} 237 \\ 140 \end{pmatrix}$

Dekripsi:

$P_2 = K^{-2}.C$

$P_2 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 237 \\ 140 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 51.297 \\ 41.825 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 51.297 mod 256 = 97

= 41.825 mod 256 = 97

Karakter yang terkorespondensi dengan 97 dan 97 adalah a dan a

Blok 3

$C_3 = \begin{pmatrix} > \\ \ddot{E} \end{pmatrix} \begin{pmatrix} 62 \\ 203 \end{pmatrix}$

Dekripsi:

$P_3 = K^{-3}.C$

$P_3 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 62 \\ 203 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 23.402 \\ 31.073 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 23.402 mod 256 = 106

= 31.073 mod 256 = 97

Karakter yang terkorespondensi dengan 106 dan 97 adalah j dan a

Blok 4

$C_4 = \begin{pmatrix} \text{Ü} \\ \text{Ã} \end{pmatrix} = \begin{pmatrix} 220 \\ 195 \end{pmatrix}$

Dekripsi:

$P_4 = K^{-4}.C$

$P_4 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 220 \\ 195 \end{pmatrix}$ Mod 256

$= \begin{pmatrix} 51.520 \\ 46.695 \end{pmatrix}$

Setelah itu hasil di mod 256 seperti berikut:

= 51.520 mod 256 = 69

= 46.695 mod 256 = 103

Karakter yang terkorespondensi dengan 69 dan 103 adalah @ dan g

Blok 5

$$C_5 = \begin{pmatrix} \text{Y} \\ \text{à} \end{pmatrix} = \begin{pmatrix} 89 \\ 224 \end{pmatrix}$$

Dekripsi:

$$P_5 = K^{-5}.C$$

$$P_5 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 89 \\ 224 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 29.549 \\ 36.449 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 29.549 mod 256 = 109

= 36.449 mod 256 = 97

Karakter yang terkorespondensi dengan 109 dan 97 adalah m dan a.

Blok 6

$$C_5 = \begin{pmatrix} \text{a} \\ \text{û} \end{pmatrix} = \begin{pmatrix} 97 \\ 251 \end{pmatrix}$$

Dekripsi:

$$P_6 = K^{-6}.C$$

$$P_6 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 97 \\ 251 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 32.617 \\ 40.556 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 32.617 mod 256 = 105

= 40.556 mod 256 = 108

Karakter yang terkorespondensi dengan 105 dan 108 adalah i dan l.

Blok 7

$$C_7 = \begin{pmatrix} * \\ 1 \end{pmatrix} = \begin{pmatrix} 42 \\ 49 \end{pmatrix}$$

Dekripsi:

$$P_7 = K^{-7}.C$$

$$P_5 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 42 \\ 49 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 10.542 \\ 10.339 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 10.542 mod 256 = 46

= 10.339 mod 256 = 99

Karakter yang terkorespondensi dengan 46 dan 99 adalah . dan c.

Blok 8

$$C_8 = \begin{pmatrix} \rangle \\ * \end{pmatrix} = \begin{pmatrix} 155 \\ 42 \end{pmatrix}$$

Dekripsi:

$$P_8 = K^{-8}.C$$

$$P_8 = \begin{pmatrix} 181 & 60 \\ 105 & 121 \end{pmatrix} \begin{pmatrix} 155 \\ 42 \end{pmatrix} \text{ Mod } 256$$

$$= \begin{pmatrix} 30.575 \\ 21.357 \end{pmatrix}$$

Setelah itu hasil di mod 256 seperti berikut:

= 30.575 mod 256 = 111

= 21.357 mod 256 = 109

Karakter yang terkorespondensi dengan 111 dan 109 adalah o dan m. Dekripsi yang didapatkan dari Email DELOíCE>ËÜÃYàaû*1›* adalah Onaaja@gmail.com

2. Perhitungan Affine Cipher

Algoritma Affine Cipher diterapkan dengan cara mengubah setiap karakter plainteks ke dalam bilangan ASCII, kemudian diolah menggunakan pasangan kunci tertentu sehingga menghasilkan ciphertext. Proses dekripsi dilakukan dengan memanfaatkan invers dari kunci tersebut agar plainteks dapat dipulihkan kembali. Sistem secara otomatis menyimpan ciphertext ke dalam database, dan ketika pengguna melakukan login, data yang terenkripsi akan didekripsi kembali sesuai kunci yang digunakan.

Contoh perhitungan dilakukan pada field data pengguna dalam database db_shopstore pada tabel users. Data berupa *username*, *password*, dan *email* dikonversi menjadi kode ASCII, kemudian dienkripsi menggunakan kunci Affine Cipher. Hasil enkripsi ditampilkan dalam bentuk ciphertext, sedangkan proses dekripsi menunjukkan bahwa data asli dapat dipulihkan kembali.

Tabel 3.9 Plaintext *Affine Cipher*

Username	Password	Email
ONA MAZURA	Onaaa123#	Onaaja@gmail.com

Tabel 3.10 Angka ASCII dari Data Pengguna

Username	Password	Email
79, 78, 65, 77, 65, 90, 85, 82, 65.	79, 110, 97, 97, 97, 49, 50, 51, 35.	79, 110, 97, 97, 106, 97, 64, 103, 109, 97, 105, 108, 46, 99, 111, 109.

a. Perhitungan Enkripsi Algoritma *Affine Cipher*

1. Enkripsi Data Penguna

Tabel 3.11 Enkripsi *Affine Cipher* Data Pengguna Username

O	N	A	M	A	Z	U	R	A
79	78	64	77	65	90	85	82	65

E(O) = (5 . 79 + 8) Mod 256

= 403 Mod 256 = 147

Karakter yang terkorespondensi dengan 147 adalah “

E(N) = (5 . 78 + 8) Mod 256

= 398 Mod 256 = 142

Karakter yang terkorespondensi dengan 142 adalah Ž

E(A) = (5 . 54 + 8) Mod 256

= 333 Mod 256 = 77

Karakter yang terkorespondensi dengan 77 adalah M

E(M) = (5 . 77 + 8) Mod 256

= 393 Mod 256 = 137

Karakter yang terkorespondensi dengan 137 adalah ‰

E(A) = (5 . 065+ 8) Mod 256

= 333 Mod 256 = 77

Karakter yang terkorespondensi dengan 77 adalah M

E(Z) = (5 . 90 + 8) Mod 256

= 458 Mod 256 = 202

Karakter yang terkorespondensi dengan 202 adalah Ê

E(U) = (5 . 85 + 8) Mod 256

= 433 Mod 256 = 177

Karakter yang terkorespondensi dengan 177 adalah ±

E(R) = (5 . 82 + 8) Mod 256

= 418 Mod 256 = 162

Karakter yang terkorespondensi dengan 162 adalah ¢

E(A) = (5 . 65 + 8) Mod 256

= 333 Mod 256 = 77

Karakter yang terkorespondensi dengan 77 adalah M

Enkripsi yang didapatkan dari Username ONA MAZURA adalah “ŽM‰MÊ±¢M

Tabel 3.12 *Affine Cipher* dari Data Pengguna bagian Password

O	n	a	a	a	1	2	3	#
79	110	97	97	97	49	50	51	35

E(O) = (5 . 79 + 8) Mod 256

= 403 Mod 256 = 147

Karakter yang terkorespondensi dengan 147 adalah “

E(n) = (5 . 110 + 8) Mod 256

= 558 Mod 256 = 46

Karakter yang terkorespondensi dengan 46 adalah .

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(1) = (5 . 49 + 8) Mod 256

= 253 Mod 256 = 253

Karakter yang terkorespondensi dengan 253 adalah ý

E(2) = (5 . 50 + 8) Mod 256

= 258 Mod 256 = 2

Karakter yang terkorespondensi dengan 2 adalah STX

E(3) = (5 . 51 + 8) Mod 256

= 263 Mod 256 = 7

Karakter yang terkorespondensi dengan 7 adalah BEL

E(#) = (5 . 35 + 8) Mod 256

= 183 Mod 256 = 183

Karakter yang terkorespondensi dengan 183 adalah .

Enkripsi yang didapatkan dari Password Onaaa123# adalah ".íííýSTXBEL·

Tabel 3.13 *Affine Cipher* Data Pengguna Email

O	n	a	a	j	a	@	g	m	a	i
79	110	97	97	106	97	64	103	109	97	105

l	.	c	o	m
108	46	99	111	109

E(O) = (5 . 79 + 8) Mod 256

= 403 Mod 256 = 147

Karakter yang terkorespondensi dengan 147 adalah “

E(n) = (5 . 110 + 8) Mod 256

= 558 Mod 256 = 46

Karakter yang terkorespondensi dengan 46 adalah .

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(j) = (5 . 106 + 8) Mod 256

= 538 Mod 256 = 26

Karakter yang terkorespondensi dengan 26 adalah SUB

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(@) = (5 . 64 + 8) Mod 256

= 328 Mod 256 = 72

Karakter yang terkorespondensi dengan 72 adalah H

E(g) = (5 . 103 + 8) Mod 256

= 523 Mod 256 = 11

Karakter yang terkorespondensi dengan 11 adalah VT

E(m) = (5 . 109 + 8) Mod 256

= 553 Mod 256 = 41

Karakter yang terkorespondensi dengan 41 adalah)

E(a) = (5 . 97 + 8) Mod 256

= 493 Mod 256 = 237

Karakter yang terkorespondensi dengan 237 adalah í

E(i) = (5 . 105 + 8) Mod 256

= 533 Mod 256 = 21

Karakter yang terkorespondensi dengan 21 adalah NAK

E(l) = (5 . 108 + 8) Mod 256

= 548 Mod 256 = 36

Karakter yang terkorespondensi dengan 36 adalah $

E(.) = (5 . 46 + 8) Mod 256

= 238 Mod 256 = 238

Karakter yang terkorespondensi dengan 238 adalah í

E(c) = (5 . 99 + 8) Mod 256

= 503 Mod 256 = 247

Karakter yang terkorespondensi dengan 247 adalah ÷

E(o) = (5 . 111 + 8) Mod 256

= 563 Mod 256 = 51

Karakter yang terkorespondensi dengan 51 adalah 3

E(m) = (5 . 109 + 8) Mod 256

= 553 Mod 256 = 41

Karakter yang terkorespondensi dengan 41 adalah) , Enkripsi yang didapatkan dari Email Onaaja@gmail.com adalah “.ííSUBíHVT) íNAK$í÷3)

Tabel 3.14 Dekripsi *Affine Cipher* Data Pengguna Username

“	Ž	M	‰	M	Ê	±	¢	M
147	142	77	137	77	202	177	162	77

Deskripsi D(y) = a – 1 . (y – b) Mod 256

D(“) = 205 . (147 - 8) Mod 256

= 205 . (139) Mod 256

= 28.495 Mod 256 = 79

Karakter yang terkorespondensi dengan 79 adalah O

D(Ž) = 205 . (142 - 8) Mod 256

= 205 . (134) Mod 256

= 27.470 Mod 256 = 78

Karakter yang terkorespondensi dengan 78 adalah N

D(M) = 205 . (77 - 8) Mod 256

= 205 . (69) Mod 256

= 14.145 Mod 256 = 65

Karakter yang terkorespondensi dengan 65 adalah A

D(‰) = 205 . (137 - 8) Mod 256

= 205 . (129) Mod 256

= 26.445 Mod 256 = 77

Karakter yang terkorespondensi dengan 77 adalah M

D(M) = 205 . (77 - 8) Mod 256

= 205 . (69) Mod 256

= 14.145 Mod 256 = 65

Karakter yang terkorespondensi dengan 65 adalah A

D(Ê) = 205 . (202 - 8) Mod 256

= 205 . (194) Mod 256

= 39.770 Mod 256 = 90

Karakter yang terkorespondensi dengan 90 adalah Z

D(±) = 205 . (177 - 8) Mod 256

= 205 . (169) Mod 256

= 34.645 Mod 256 = 85

Karakter yang terkorespondensi dengan 85 adalah U

D(¢) = 205 . (162 - 8) Mod 256

= 205 . (154) Mod 256

= 31.570 Mod 256 = 82

Karakter yang terkorespondensi dengan 82 adalah R

D(M) = 205 . (77 - 8) Mod 256

= 205 . (69) Mod 256

= 14.145 Mod 256 = 65

Karakter yang terkorespondensi dengan 65 adalah A

Dekripsi yang didapatkan dari Username “ŽM‰MÊ±¢M adalah ONA MAZURA.

Tabel 3.15 Dekripsi Affine Cipher Data Pengguna Password

“	.	í	í	í	ý	STX	BEL	·
147	46	237	237	237	253	2	7	183

Deskripsi D(y) = a – 1 . (y – b) Mod 256

D(“) = 205 . (147 - 8) Mod 256

= 205 . (139) Mod 256

= 28.495 Mod 256 = 79

Karakter yang terkorespondensi dengan 79 adalah O

D(.) = 205 . (46 - 8) Mod 256

= 205 . (38) Mod 256

= 7.790 Mod 256 = 110

Karakter yang terkorespondensi dengan 110 adalah n

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 65

Karakter yang terkorespondensi dengan 97 adalah a

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 65

Karakter yang terkorespondensi dengan 65 adalah a

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 65

Karakter yang terkorespondensi dengan 65 adalah a

D(ý) = 205 . (253 - 8) Mod 256

= 205 . (245) Mod 256

= 50.225 Mod 256 = 49

Karakter yang terkorespondensi dengan 49 adalah 1

D(STX) = 205 . (2 - 8) Mod 256

= 205 . (-6) Mod 256

= -1.230 Mod 256 = 50

Karakter yang terkorespondensi dengan 50 adalah 2

D(BEL) = 205 . (7 - 8) Mod 256

= 205 . (-1) Mod 256

= -205 Mod 256 = 51

Karakter yang terkorespondensi dengan 51 adalah 3

D(.) = 205 . (183 - 8) Mod 256

= 205 . (175) Mod 256

= 35.875 Mod 256 = 35

Karakter yang terkorespondensi dengan 35 adalah #

Dekripsi yang didapatkan dari Password “.ÍíÍýSTXBEL· adalah Onaaa123#.

Tabel 3.16 Dekripsi Affine Cipher Data Pengguna Email

“	.	í	í	SUB	í	H	VT	)	í	$	î
147	46	237	237	26	237	72	11	41	237	36	238

÷	3	)
247	51	41

Deskripsi D(y) = a – 1 . (y – b) Mod 256

D(“) = 205 . (147 - 8) Mod 256

= 205 . (139) Mod 256

= 28.495 Mod 256 = 79

Karakter yang terkorespondensi dengan 79 adalah O

D(.) = 205 . (46 - 8) Mod 256

= 205 . (38) Mod 256

= 7.790 Mod 256 = 110

Karakter yang terkorespondensi dengan 110 adalah n

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 97

Karakter yang terkorespondensi dengan 97 adalah a

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 97

Karakter yang terkorespondensi dengan 97 adalah a

D(SUB) = 205 . (26 - 8) Mod 256

= 205 . (18) Mod 256

= 3.690 Mod 256 = 106

Karakter yang terkorespondensi dengan 106 adalah j

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 97

Karakter yang terkorespondensi dengan 97 adalah a

D(H) = 205 . (72 - 8) Mod 256

= 205 . (64) Mod 256

= 13.120 Mod 256 = 64

Karakter yang terkorespondensi dengan 64 adalah @

D(VT) = 205 . (11 - 8) Mod 256

= 205 . (3) Mod 256

= 615 Mod 256 = 103

Karakter yang terkorespondensi dengan 103 adalah g

D( )) = 205 . (41 - 8) Mod 256

= 205 . (33) Mod 256

= 6.765 Mod 256 = 103

Karakter yang terkorespondensi dengan 103 adalah m

D(í) = 205 . (237 - 8) Mod 256

= 205 . (229) Mod 256

= 46.945 Mod 256 = 97

Karakter yang terkorespondensi dengan 97 adalah a

D(NAK) = 205 . (21 - 8) Mod 256

= 205 . (13) Mod 256

= 2.665 Mod 256 = 105

Karakter yang terkorespondensi dengan 105 adalah i

D($) = 205 . (36 - 8) Mod 256

= 205 . (2) Mod 256

= 5.740 Mod 256 = 108

Karakter yang terkorespondensi dengan 108 adalah l

D(î) = 205 . (238 - 8) Mod 256

= 205 . (230) Mod 256

= 47.150 Mod 256 = 46

Karakter yang terkorespondensi dengan 46 adalah .

D(÷) = 205 . (247 - 8) Mod 256

= 205 . (239) Mod 256

= 48.995 Mod 256 = 99

Karakter yang terkorespondensi dengan 99 adalah c

D(3) = 205 . (51 - 8) Mod 256

= 205 . (43) Mod 256

= 8.815 Mod 256 = 111

Karakter yang terkorespondensi dengan 111 adalah o

D( )) = 205 . (41 - 8) Mod 256

= 205 . (33) Mod 256

= 6.765 Mod 256 = 109

Karakter yang terkorespondensi dengan 109 adalah m

Dekripsi yang didapatkan dari Email ".ííSUBíHVT)íNAK$î÷3) adalah Onaaja@gmail.com

b. Pengujian Keamanan

Tahap pengujian dilakukan untuk mengevaluasi ketahanan masing-masing algoritma kriptografi *Hill Cipher* dan *Affine Cipher* dalam mengamankan data pengguna pada sistem simulasi berbasis web e-commerce. Proses pengujian mencakup dua jenis uji utama, yaitu uji performa enkripsi dan dekripsi serta uji keamanan ciphertext. Uji performa dilakukan untuk mengukur waktu proses enkripsi dan dekripsi pada masing-masing algoritma, sehingga dapat diketahui efisiensi algoritma dalam pengolahan data.

Selanjutnya, uji keamanan dilakukan menggunakan pendekatan *Ciphertext-Only Attack* (COA), di mana analisis difokuskan pada ciphertext yang dihasilkan tanpa mengetahui plaintext maupun kunci enkripsi. Serta pengujian juga dilengkapi dengan analisis *Shannon Entropy* untuk mengukur tingkat keacakan ciphertext.

1. Performa Enkripsi dan Dekripsi

Pada tahap ini, dilakukan pengukuran kecepatan proses enkripsi serta dekripsi data dengan menggunakan algoritma *Hill Cipher* dan *Affine Cipher*. Informasi pengguna seperti nama, alamat *email*, dan password akan diproses dengan kedua algoritma, kemudian waktu yang diperlukan untuk setiap proses dicatat. Hasil ini akan memberikan ilustrasi tentang efisiensi algoritma dari segi waktu, yang merupakan indikator krusial dalam sistem berbasis web yang memerlukan kecepatan dan responsivitas.

Untuk memperoleh hasil yang terukur secara sistematis, maka pengujian dilakukan melalui beberapa langkah berikut:

a) Menentukan data uji berupa username, email, dan password yang dimasukkan pada form registrasi website. Data ini dipilih karena bersifat sensitif dan sering menjadi target penyalahgunaan.
b) Melakukan proses enkripsi data dengan algoritma *Hill Cipher* dan *Affine Cipher* pada website.
c) Menyimpan hasil enkripsi berupa ciphertext ke dalam database sehingga tidak ada plaintext yang disimpan langsung.
d) Melakukan proses dekripsi terhadap ciphertext tersebut untuk memastikan bahwa algoritma mampu mengembalikan data ke plaintext semula.
e) Mencatat lama waktu eksekusi enkripsi dan dekripsi menggunakan fungsi pencatat waktu (misalnya microtime dalam PHP atau log waktu pada Laravel).
f) Melakukan percobaan beberapa kali uji coba untuk masing-masing algoritma, sehingga dapat dihitung rata-rata waktu proses.
g) Membandingkan hasil rata-rata dari kedua algoritma untuk menilai algoritma mana yang lebih efisien dalam kinerja.

2. Ciphertext-Only Attack (COA)

Pada tahap ini dilakukan pengujian keamanan menggunakan pendekatan *Ciphertext-Only Attack* (COA), yaitu skenario pengujian di mana analisis dilakukan hanya berdasarkan ciphertext tanpa mengetahui plaintext maupun kunci

enkripsi yang digunakan. Pendekatan ini bertujuan untuk menilai sejauh mana algoritma Hill Cipher dan Affine Cipher mampu menyamarkan informasi asli sehingga tidak mudah dianalisis berdasarkan pola ciphertext.

Langkah-langkah pengujian *Ciphertext-Only Attack* (COA) adalah sebagai berikut:

a) Mengambil ciphertext hasil enkripsi data pengguna yang tersimpan di dalam database sistem.

b) Melakukan analisis terhadap pola dan distribusi karakter ciphertext untuk mengidentifikasi kemungkinan keteraturan atau kemiripan pola tertentu.

c) Membandingkan karakteristik ciphertext yang dihasilkan oleh *Hill Cipher* dan *Affine Cipher* untuk menilai tingkat kesulitan analisis berbasis ciphertext.

d) Menarik kesimpulan mengenai ketahanan masing-masing algoritma terhadap analisis berbasis ciphertext.

3. Analisis Shannon Entropy

Tahap ini dilakukan analisis *Shannon Entropy* untuk mengukur tingkat keacakan ciphertext yang dihasilkan oleh algoritma *Hill Cipher* dan *Affine Cipher*. *Shannon Entropy* digunakan sebagai metode kuantitatif untuk mengevaluasi distribusi karakter dalam ciphertext.

Langkah-langkah analisis *Shannon Entropy* meliputi:

a) Menghitung nilai *Shannon Entropy* dari ciphertext hasil enkripsi menggunakan *Hill Cipher* dan *Affine Cipher*.

b) Membandingkan nilai entropy yang diperoleh dari kedua algoritma.

c) Menafsirkan hasil pengujian, di mana nilai entropy yang lebih tinggi menunjukkan tingkat keacakan ciphertext yang lebih baik serta tingkat keamanan yang lebih tinggi terhadap analisis statistik.

d. Analisis Kinerja Sistem

Analisis kinerja dalam penelitian ini direncanakan untuk dilakukan setelah tahapan implementasi dan pengujian selesai dilaksanakan. Tujuan dari analisis ini adalah untuk memperoleh pemahaman yang komprehensif mengenai efektivitas

algoritma *Hill Cipher* dan *Affine Cipher* dalam menjaga keamanan data pengguna pada sistem berbasis web e-commerce.

Parameter yang akan digunakan dalam analisis meliputi:

1. Waktu Proses Enkripsi dan Dekripsi
   a) Hasil pengukuran waktu enkripsi dan dekripsi dari tahap pengujian dicatat dalam satuan milidetik.
   b) Nilai rata-rata dari beberapa kali uji coba digunakan untuk memastikan hasil yang stabil.
   c) Indikator penilaian: semakin cepat waktu yang dibutuhkan, semakin efisien algoritma dalam mendukung performa aplikasi web.

2. Tingkat Keamanan Ciphertext
   a) Tingkat keamanan dianalisis menggunakan pendekatan *Ciphertext-Only Attack* (COA), yaitu dengan mengevaluasi karakteristik ciphertext tanpa mengetahui plaintext maupun kunci enkripsi.
   b) Selain itu, *Shannon Entropy* digunakan sebagai parameter kuantitatif untuk mengukur tingkat keacakan ciphertext yang dihasilkan oleh masing-masing algoritma. Nilai entropy dihitung berdasarkan distribusi karakter dalam ciphertext hasil enkripsi.
   c) Indikator penilaian: semakin tinggi nilai *Shannon Entropy* dan semakin sulit ciphertext dianalisis secara statistik, maka semakin baik tingkat keamanan algoritma.

Berdasarkan kedua parameter tersebut, penelitian ini tidak hanya menilai kinerja algoritma *Hill Cipher* dan *Affine Cipher* secara individual, tetapi juga melakukan perbandingan untuk memperoleh gambaran yang lebih menyeluruh mengenai efektivitas dan efisiensi masing-masing algoritma dalam melindungi data pengguna pada sistem web e-commerce.

Berikut adalah contoh gambaran hasil penelitian dari (Ahmat Sayuti, dkk 2019 dan Cahyo Prawiro, dkk 2024)

Tabel 3.17 Contoh Perbandingan Performa *Hill Cipher* dan *RSA* [9]

Algoritma	Waktu Enkripsi	Memori Enkripsi	Waktu Dekripsi	Memori Dekripsi
Hill Cipher	43.8 ms	27.9 kb	32.3 ms	23.4 kb
RSA	1829.5 ms	27 kb	2141.4 ms	27.3 kb

Keterangan: warna hijau lebih baik

Melalui analisis kinerja ini, penelitian diharapkan dapat memberikan evaluasi akademis yang terukur mengenai efektivitas *Hill Cipher* dan *Affine Cipher*. Hasil dari analisis akan digunakan sebagai dasar untuk menyimpulkan algoritma yang memiliki efisiensi waktu yang lebih baik serta tingkat keamanan ciphertext yang lebih tinggi berdasarkan pendekatan *Ciphertext-Only Attack* dan *Shannon Entropy*, sehingga dapat direkomendasikan sebagai solusi pengamanan data pada sistem web e-commerce.

e. Laporan Hasil Penelitian

Pada tahap akhir penelitian, disusun laporan hasil yang mencakup seluruh tahapan penelitian, mulai dari persiapan sistem, implementasi algoritma, pengujian performa enkripsi dan dekripsi, hingga analisis keamanan ciphertext menggunakan pendekatan *Ciphertext-Only Attack* (COA) dan *Shannon Entropy*. Laporan hasil penelitian menyajikan perbandingan antara algoritma *Hill Cipher* dan *Affine Cipher* berdasarkan parameter efisiensi proses enkripsi–dekripsi, tingkat keacakan ciphertext, serta ketahanan terhadap analisis berbasis ciphertext. Penarikan kesimpulan dilakukan secara sistematis berdasarkan data hasil pengujian yang telah diperoleh.

Diharapkan, penelitian ini dapat memberikan kontribusi dalam pengembangan teknologi keamanan data berbasis kriptografi klasik, serta menjadi referensi akademik bagi peneliti dan praktisi yang tertarik dalam bidang keamanan sistem informasi, khususnya pada penerapan algoritma kriptografi dalam lingkungan web e-commerce.

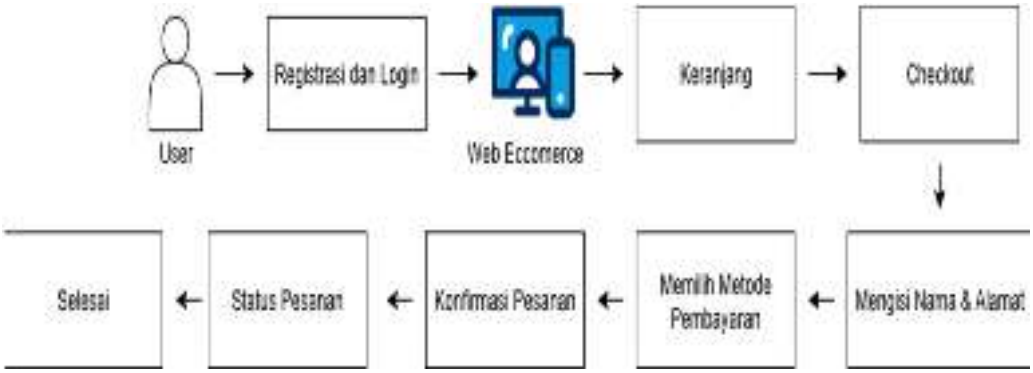
3.3 Analisis Kebutuhan

Analisis kebutuhan dalam penelitian ini dilakukan untuk menentukan kebutuhan yang diperlukan dalam proses pengujian dan perbandingan algoritma *Hill Cipher* dan *Affine Cipher* dalam menjaga keamanan data pada website e-commerce. Berdasarkan tujuan penelitian, diperlukan lingkungan simulasi berupa website e-commerce yang mampu menerima input data pengguna serta mendukung proses enkripsi dan dekripsi data sebelum disimpan ke dalam database. Sistem yang digunakan harus mampu menghasilkan ciphertext dari masing-masing algoritma sehingga dapat dilakukan analisis tingkat keacakan menggunakan metode *Shannon Entropy* serta pengujian ketahanan terhadap *Ciphertext-Only Attack*. Selain itu, diperlukan perangkat lunak dan perangkat keras yang stabil agar proses pengujian dapat berjalan secara konsisten dan menghasilkan data yang objektif untuk keperluan analisis perbandingan kedua algoritma.

3.4 Perancangan

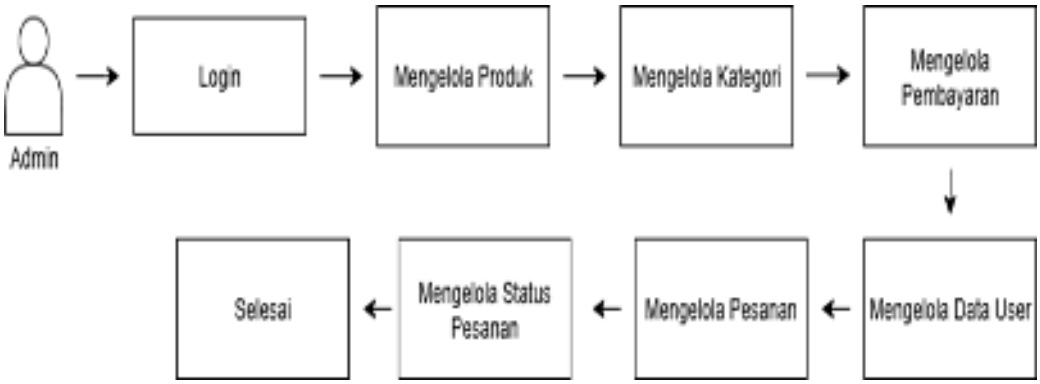
Perancangan dalam penelitian ini dilakukan dengan memanfaatkan website e-commerce sebagai media simulasi untuk menerapkan dan menguji algoritma *Hill Cipher* dan *Affine Cipher* dalam menjaga keamanan data. Sistem e-commerce yang digunakan telah memiliki alur operasional dasar seperti interaksi user dan admin, yang melibatkan proses input, pengolahan, serta penyimpanan data. Alur tersebut dimanfaatkan sebagai konteks penelitian untuk menguji bagaimana algoritma kriptografi diterapkan dalam mengamankan data sensitif sebelum disimpan ke dalam database. Dengan demikian, perancangan pada penelitian ini tidak berfokus pada pengembangan fitur bisnis e-commerce, melainkan pada alur penerapan dan pengujian keamanan data menggunakan algoritma *Hill Cipher* dan *Affine Cipher*.

a. Alur Sistem



Gambar 3.7 Alur Sistem User

Alur sistem pada gambar di atas menjelaskan proses transaksi yang dilakukan oleh user dalam website e-commerce. Pertama, user melakukan registrasi dan login untuk mendapatkan akses ke dalam sistem. Setelah berhasil masuk, user dapat menelusuri produk, menambahkannya ke dalam keranjang, lalu melanjutkan ke tahap checkout. Pada proses checkout, user diminta untuk mengisi data nama dan alamat pengiriman serta memilih metode pembayaran yang tersedia. Setelah itu, sistem akan menampilkan halaman konfirmasi pesanan yang berisi ringkasan transaksi. Selanjutnya, user dapat memantau perkembangan pembelian melalui fitur status pesanan hingga transaksi dinyatakan selesai. Dengan alur ini, sistem e-commerce memberikan pengalaman berbelanja yang terstruktur mulai dari login hingga penyelesaian pesanan.

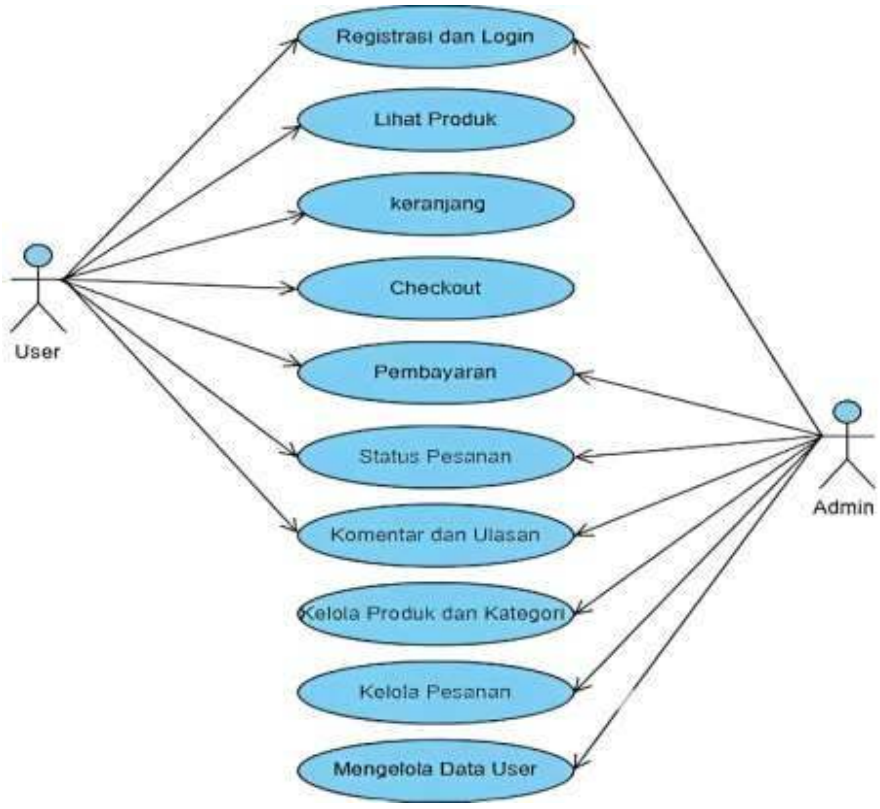


Gambar 3.8 Alur Sistem Admin

Alur sistem admin pada website e-commerce diawali dengan proses login untuk melakukan autentikasi. Setelah berhasil masuk, admin dapat menjalankan

berbagai fungsi pengelolaan, seperti mengelola produk dan kategori, serta memastikan transaksi melalui fitur pengelolaan pesanan dan status pesanan. Admin juga memiliki wewenang untuk mengelola pembayaran, baik dalam bentuk verifikasi maupun pencatatan transaksi. Selain itu, admin dapat meninjau serta mengatur komentar dan ulasan dari user untuk menjaga kualitas layanan. Yang tidak kalah penting, admin berperan dalam mengelola data user, yaitu menambahkan, memperbarui, atau menonaktifkan akun pengguna sesuai kebutuhan sistem. Dengan demikian, admin menjadi aktor utama yang memastikan seluruh data dan aktivitas dalam e-commderce tetap berjalan teratur dan terkontrol.

b. Usecase



Gambar 3.9 Usecase

Gambar 3.9 di atas menggambarkan interaksi antara dua aktor utama, yaitu User dan Admin, dalam sistem e-commerce berbasis website. Diagram ini bertujuan untuk mendeskripsikan fitur-fitur utama yang dapat diakses oleh masing-

masing pengguna sesuai dengan hak akses dan perannya.

1. User

User berperan sebagai aktor eksternal yang menggunakan sistem untuk melakukan aktivitas transaksi pembelian produk. User dapat melakukan registrasi dan login untuk memperoleh akses ke dalam sistem. Selanjutnya, user memiliki kemampuan untuk melihat produk yang tersedia, menambahkan produk ke dalam keranjang, serta melanjutkan ke proses checkout. Pada tahap berikutnya, user dapat melakukan pembayaran sesuai dengan metode yang disediakan sistem. Selain itu, user juga dapat memantau status pesanan yang sedang diproses serta memberikan komentar dan ulasan terhadap produk yang telah dibeli. Dengan demikian, user memiliki peran utama dalam memanfaatkan sistem untuk memenuhi kebutuhan belanja secara daring.

2. Admin

Admin merupakan aktor internal yang memiliki tanggung jawab dalam pengelolaan sistem secara keseluruhan. Admin berwenang untuk mengelola data produk dan kategori, termasuk melakukan penambahan, pembaruan, maupun penghapusan produk. Selain itu, admin juga bertugas dalam pengelolaan data user yang terdaftar pada sistem, serta menangani proses pengelolaan pesanan dari user, mulai dari verifikasi pembayaran hingga konfirmasi status pesanan. Admin juga dapat memantau interaksi user dengan sistem, seperti komentar dan ulasan produk, guna memastikan kualitas layanan dan kepuasan pelanggan. Dengan demikian, admin memiliki peran sentral dalam memastikan kelancaran operasional serta ketersediaan informasi yang akurat pada sistem.

Secara keseluruhan, setiap aktor dalam sistem diwajibkan untuk melakukan login terlebih dahulu sebelum dapat mengakses fitur yang disediakan sesuai perannya. User berfokus pada aktivitas transaksi dan konsumsi produk, sedangkan Admin berperan pada aspek manajerial dan pengendalian data dalam sistem.

BAB IV
HASIL DAN PENGUJIAN

4.1 Hasil

Penelitian ini menghasilkan temuan perbandingan algoritma Hill Cipher dan Affine Cipher berdasarkan performa enkripsi/dekripsi, ketahanan terhadap Ciphertext Only Attack (COA), serta tingkat keacakan menggunakan Shannon Entropy. Hasil pengujian menunjukkan bahwa kedua algoritma mampu menghasilkan ciphertext yang tidak menyerupai plaintext, sehingga cukup aman dari serangan dasar. Dari sisi performa, Affine Cipher memiliki waktu enkripsi dan dekripsi yang lebih cepat dibandingkan Hill Cipher.

Namun, dari sisi keamanan, Hill Cipher menunjukkan tingkat keacakan yang lebih tinggi dengan nilai Shannon Entropy sebesar 6,49 bit, sedangkan Affine Cipher memperoleh 4,59 bit. Hal ini menunjukkan bahwa Hill Cipher menghasilkan ciphertext yang lebih kompleks dan lebih sulit ditebak. Dengan demikian, Hill Cipher lebih unggul dari sisi keamanan, sementara Affine Cipher lebih unggul dari sisi kecepatan, sehingga pemilihannya dapat disesuaikan dengan kebutuhan sistem.

4.1.1 Hasil Implementasi Algoritma

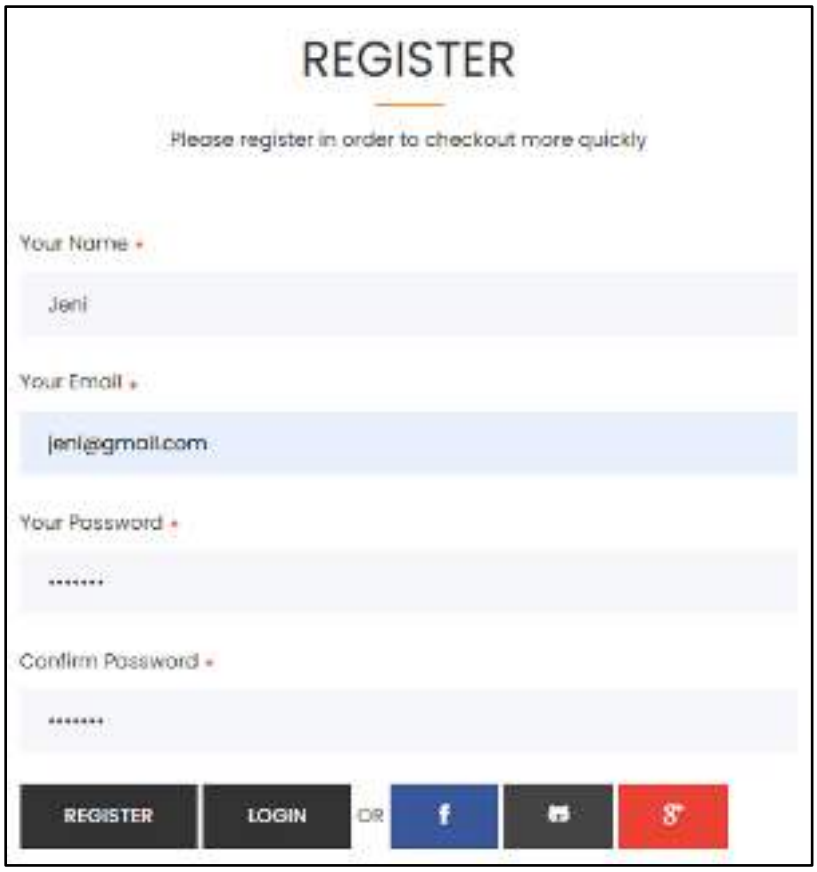
Implementasi algoritma dalam penelitian ini dilakukan dengan mengintegrasikan algoritma *Hill Cipher* dan *Affine Cipher* ke dalam sistem e-commerce sebagai mekanisme pengamanan data pengguna. Proses enkripsi dilakukan sebelum data disimpan ke dalam database, sedangkan proses dekripsi dilakukan saat sistem melakukan verifikasi data pada proses login.

Tujuan dari implementasi ini adalah untuk memastikan bahwa data pengguna tidak tersimpan dalam bentuk plaintext, sehingga meningkatkan tingkat keamanan data apabila terjadi akses tidak sah terhadap database.

2. Implementasi Algoritma *Hill Cipher*

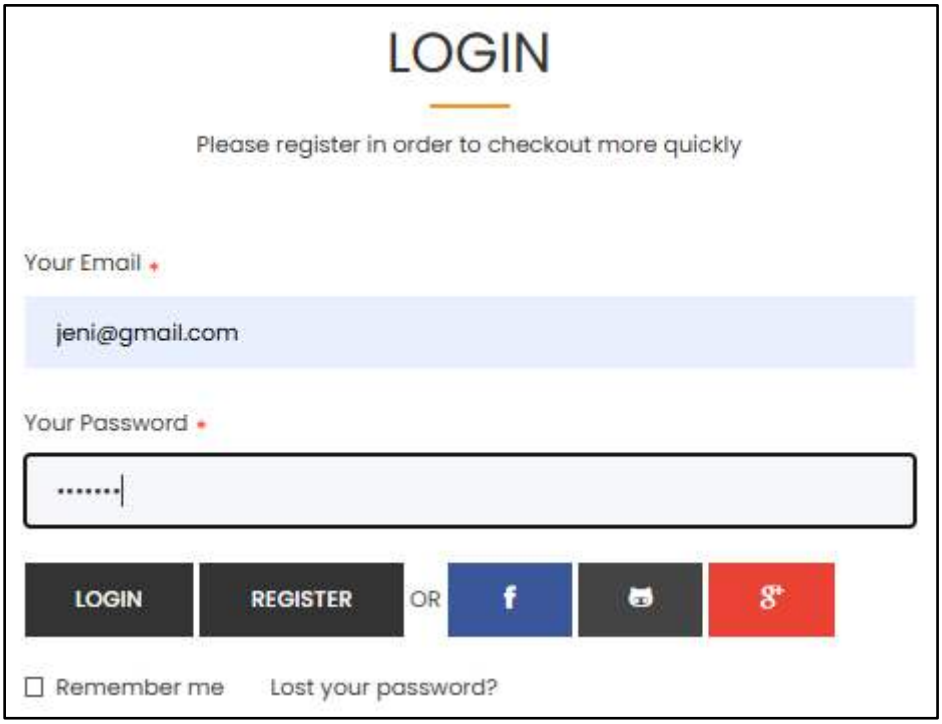
Pada penerapan algoritma *Hill Cipher*, data pengguna berupa username, email, dan password yang diinputkan pada halaman registrasi akan diproses dan dienkripsi terlebih dahulu sebelum disimpan ke dalam database.

Pada saat proses login, sistem melakukan dekripsi terhadap data terenkripsi untuk keperluan autentikasi pengguna. Proses enkripsi dan dekripsi dilakukan secara otomatis pada sisi server menggunakan bahasa pemrograman PHP, sehingga data yang tersimpan di database berbentuk ciphertext dan tidak dapat dibaca secara langsung.



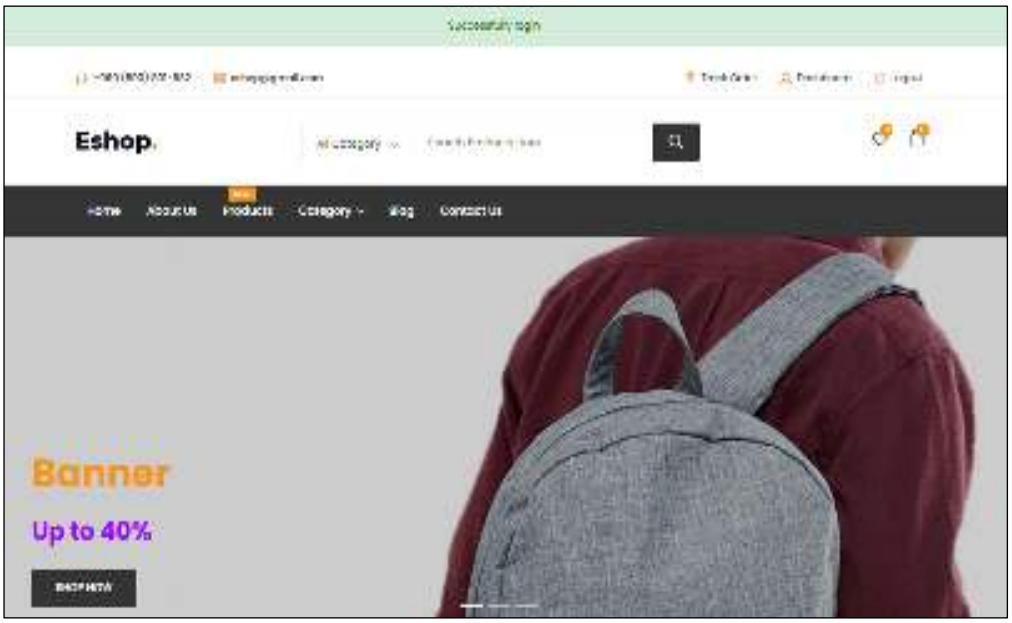
Gambar 4.1 Halaman Registrasi Data Pengguna (*Hill Cipher*)

Setelah proses registrasi berhasil, pengguna dapat melakukan login dengan memasukkan email dan password yang telah didaftarkan. Sistem akan memproses data login tersebut dan melakukan dekripsi terhadap data terenkripsi untuk proses pencocokan.



Gambar 4.2 Halaman login Pengguna (*Hill Cipher*)

Apabila data yang dimasukkan sesuai, maka pengguna berhasil masuk ke dalam sistem dan diarahkan ke halaman utama.



Gambar 4.3 Halaman Home Setelah Berhasil Login (H*ill Cipher*)

Proses enkripsi data pengguna dilakukan pada sisi server menggunakan *source code* yang menerapkan algoritma *Hill Cipher*. Setiap data yang dikirimkan dari form diproses dan dienkripsi sebelum disimpan ke dalam database.

```php
<?php

namespace App\Helpers;

class HillCipher
{
    private static $key = [
        [3, 3],
        [2, 5]
    ];

    private static $mod = 256;

    private static function textToNumbers(string $text): array
    {
        return array_map('ord', str_split($text));
    }

    private static function numbersToText(array $nums): string
    {
        return implode('', array_map('chr', $nums));
    }

    // ================= ENCRYPT =================
    public static function encrypt(string $plaintext): string
    {
        $nums = self::textToNumbers($plaintext);

        if (count($nums) % 2 !== 0) {
            $nums[] = 0;
        }

        $cipher = [];

        for ($i = 0; $i < count($nums); $i += 2) {
            $p1 = $nums[$i];
            $p2 = $nums[$i + 1];
```

```
            $c1 = (self::$key[0][0] * $p1 + self::$key[0][1] *
$p2) % self::$mod;
            $c2 = (self::$key[1][0] * $p1 + self::$key[1][1] *
$p2) % self::$mod;

            $cipher[] = $c1;
            $cipher[] = $c2;
        }

        return bin2hex(self::numbersToText($cipher));
    }

    // ================= DECRYPT =================
    public static function decrypt(string $ciphertext): string
    {
        if ($ciphertext === '') return '';

        $ciphertext = hex2bin($ciphertext);

        $nums = self::textToNumbers($ciphertext);

        $det = (self::$key[0][0] * self::$key[1][1]) -
(self::$key[0][1] * self::$key[1][0]);
        $det = ($det % self::$mod + self::$mod) % self::$mod;

        $invDet = self::modInverse($det, self::$mod);
        if ($invDet === null) {
            throw new \Exception("Determinant tidak memiliki
invers modulo.");
        }

        $inv = [
            [ (self::$key[1][1] * $invDet) % self::$mod, ((-
self::$key[0][1]) * $invDet) % self::$mod ],
            [ ((-self::$key[1][0]) * $invDet) % self::$mod,
(self::$key[0][0] * $invDet) % self::$mod ],
        ];

        foreach ($inv as $r => $row) {
            foreach ($row as $c => $val) {
                $inv[$r][$c] = ($val + self::$mod) % self::$mod;
            }
        }

        $plain = [];
```

```
        for ($i = 0; $i < count($nums); $i += 2) {
            if (!isset($nums[$i + 1])) break;

            $c1 = $nums[$i];
            $c2 = $nums[$i + 1];

            $p1 = ($inv[0][0] * $c1 + $inv[0][1] * $c2) %
self::$mod;
            $p2 = ($inv[1][0] * $c1 + $inv[1][1] * $c2) %
self::$mod;

            $plain[] = $p1;
            $plain[] = $p2;
        }

        return rtrim(self::numbersToText($plain), "\0");
    }
    private static function modInverse(int $a, int $m): ?int
    {
        $a = ($a % $m + $m) % $m;
        if ($a === 0) return null;

        $m0 = $m;
        $x0 = 0;
        $x1 = 1;

        while ($a > 1) {
            $q = intdiv($a, $m);
            $temp = $m;

            $m = $a % $m;
            $a = $temp;

            $temp = $x0;
            $x0 = $x1 - $q * $x0;
            $x1 = $temp;
        }

        if ($x1 < 0) {
            $x1 += $m0;
        }

        return $x1;
    }
}
```

Setelah muncul notifikasi bahwa data berhasil disimpan, seluruh data pengguna tersimpan di dalam database dalam bentuk ciphertext, yaitu kombinasi angka dan karakter acak hasil enkripsi menggunakan algoritma Hill Cipher.

email	password
85e9e3027cef5ecfcebe76f147da	85e9b6b72f633b71
9704e3027cef5ecfcebe76f147da	37c4b3b20517
88dfac24fb127cef5ecfcebe76f147da	289fac2476df2f69
6dcd85e9f5836abf7feeb34b94ff	0d8d85e9295c9966
94f397f5f5836abf7feeb34b94ff	e3657fe5b3b2ab72

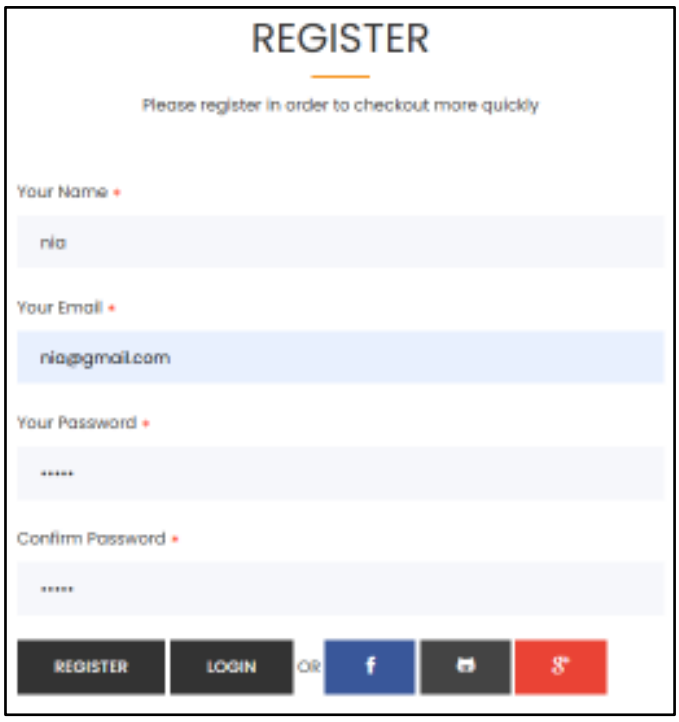
Gambar 4.4 Data *Users* Terenkripsi pada Database (*Hill Cipher*)

Terlihat pada gambar 4.4 diatas data pengguna tersebut tersimpan dalam bentuk terenkripsi. Hal ini menunjukkan bahwa algoritma Hill Cipher telah berhasil diterapkan dalam proses pengamanan data pengguna pada sistem e-commerce.

3. Implementasi Algoritma *Affine Cipher*

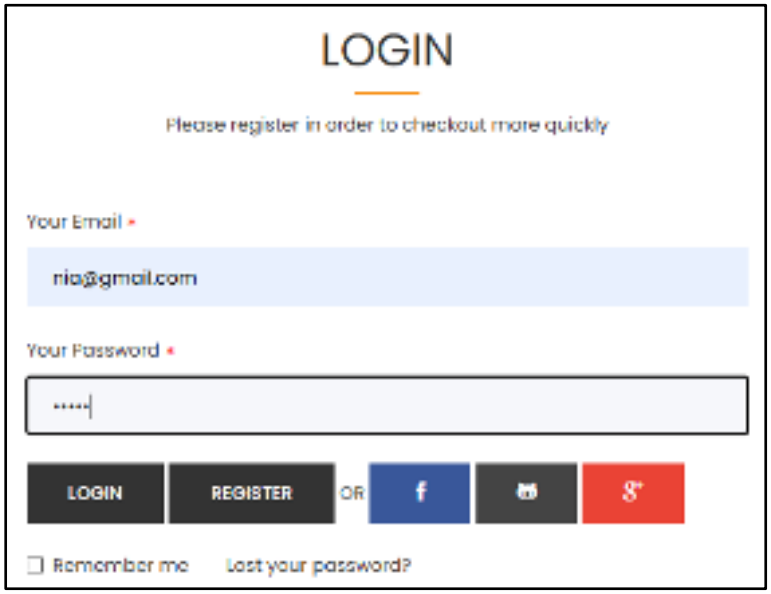
Pada Implementasi algoritma *Affine Cipher* dilakukan dengan mengenkripsi data pengguna sebelum disimpan ke dalam database. Data berupa username, email, dan password diproses menggunakan rumus enkripsi *Affine Cipher* pada sisi server.

Pada proses login, sistem melakukan dekripsi terhadap data terenkripsi untuk keperluan autentikasi pengguna. Proses enkripsi dan dekripsi berjalan secara otomatis, sehingga data yang tersimpan di dalam database berbentuk ciphertext dan tidak dapat dibaca secara langsung.



Gambar 4.5 Halaman Registrasi Input Data Pengguna (*Affine Cipher*)

Setelah data berhasil tersimpan, pengguna dapat melakukan proses login ke dalam sistem. Sistem akan memproses data login dengan melakukan dekripsi terhadap data terenkripsi menggunakan algoritma *Affine Cipher* untuk mencocokkan data yang dimasukkan.



Gambar 4.6 Halaman Login Pengguna (*Affine Cipher*)

Apabila proses login berhasil, pengguna akan diarahkan ke halaman utama sistem.

Gambar 4.7 Halaman Home Setelah Berhasil Login (*Affine Cipher*)

Selanjutnya, proses enkripsi dan dekripsi data pengguna dilakukan pada sisi server menggunakan source code yang menerapkan algoritma *Affine Cipher*. Source code tersebut digunakan oleh sistem untuk memproses data pengguna sebelum disimpan ke dalam database serta pada saat proses login.

```
<?php
namespace App\Helpers;

class HillCipher
{
    private static $key = [
        [3, 3],
        [2, 5]
    ];

    private static $mod = 256;

    private static function textToNumbers(string $text): array
    {
        return array_map('ord', str_split($text));
    }
```

```
    private static function numbersToText(array $nums): string
    {
        return implode('', array_map('chr', $nums));
    }

    // ================= ENCRYPT =================
    public static function encrypt(string $plaintext): string
    {
        $nums = self::textToNumbers($plaintext);

        if (count($nums) % 2 !== 0) {
            $nums[] = 0;
        }

        $cipher = [];

        for ($i = 0; $i < count($nums); $i += 2) {
            $p1 = $nums[$i];
            $p2 = $nums[$i + 1];

            $c1 = (self::$key[0][0] * $p1 + self::$key[0][1] *
$p2) % self::$mod;
            $c2 = (self::$key[1][0] * $p1 + self::$key[1][1] *
$p2) % self::$mod;

            $cipher[] = $c1;
            $cipher[] = $c2;
        }

        return bin2hex(self::numbersToText($cipher));
    }

    // ================= DECRYPT =================
    public static function decrypt(string $ciphertext): string
    {
        if ($ciphertext === '') return '';

        $ciphertext = hex2bin($ciphertext);

        $nums = self::textToNumbers($ciphertext);

        $det = (self::$key[0][0] * self::$key[1][1]) -
(self::$key[0][1] * self::$key[1][0]);
        $det = ($det % self::$mod + self::$mod) % self::$mod;
```

```
        $invDet = self::modInverse($det, self::$mod);
        if ($invDet === null) {
            throw new \Exception("Determinant tidak memiliki
invers modulo.");
        }

        $inv = [
            [ (self::$key[1][1] * $invDet) % self::$mod, ((-
self::$key[0][1]) * $invDet) % self::$mod ],
            [ ((-self::$key[1][0]) * $invDet) % self::$mod,
(self::$key[0][0] * $invDet) % self::$mod ],
        ];

        foreach ($inv as $r => $row) {
            foreach ($row as $c => $val) {
                $inv[$r][$c] = ($val + self::$mod) %
self::$mod;
            }
        }

        $plain = [];

        for ($i = 0; $i < count($nums); $i += 2) {
            if (!isset($nums[$i + 1])) break;

            $c1 = $nums[$i];
            $c2 = $nums[$i + 1];

            $p1 = ($inv[0][0] * $c1 + $inv[0][1] * $c2) %
self::$mod;
            $p2 = ($inv[1][0] * $c1 + $inv[1][1] * $c2) %
self::$mod;

            $plain[] = $p1;
            $plain[] = $p2;
        }

        return rtrim(self::numbersToText($plain), "\0");
    }

    private static function modInverse(int $a, int $m): ?int
    {
        $a = ($a % $m + $m) % $m;
        if ($a === 0) return null;
```

```
        $m0 = $m;
        $x0 = 0;
        $x1 = 1;

        while ($a > 1) {
            $q = intdiv($a, $m);
            $temp = $m;

            $m = $a % $m;
            $a = $temp;

            $temp = $x0;
            $x0 = $x1 - $q * $x0;
            $x1 = $temp;
        }

        if ($x1 < 0) {
            $x1 += $m0;
        }

        return $x1;
    }
}
```

Data pengguna yang telah diproses selanjutnya disimpan ke dalam database dalam bentuk ciphertext sebagai hasil dari proses enkripsi menggunakan algoritma *Affine Cipher*.

name	email	password
2e15ed	2e15ed480b29ed1524eef73329	2e15edfd02070c11
1a012e15	1a012e15480b29ed1524eef73329	1a012e15fd0207
332eed	332eed480b29ed1524eef73329	932eedf80cb7
47154c15	47154c15480b29ed1524eef73329	48ed2415edf825
2e51425124	2e51425124480b29ed1524eef73329	2e51425124f8fd

Gambar 4.8 Data *Users* Terenkripsi pada Database (*Affine Cipher*)

Tabel 4.1 Hasil Enkripsi Algoritma Hill Cipher dan Affine Cipher

Algoritma	Data	Plaintext	Ciphertext
Hill Cipher	Username	nia	85e923c2
	Password	nia@gmail.com	85e9b6b72f633b71
	Email	nia12345	85e9e3027cef5ecfcebe76f147da
Affine Cipher	Username	nia	2e15ed
	Password	nia@gmail.com	2e15edfd02070c11
	Email	nia12345	2e15ed480b29ed1524eef73329

4.2 Pengujian

Pengujian dilakukan untuk mengevaluasi kinerja serta tingkat keamanan algoritma *Hill Cipher* dan *Affine Cipher* dalam mengamankan data pengguna pada sistem e-commerce berbasis web. Pengujian difokuskan pada dua aspek utama, yaitu performa proses enkripsi dan dekripsi serta keamanan ciphertext yang dihasilkan. Data yang digunakan dalam pengujian berupa data pengguna yang meliputi username, email, dan password.

4.2.1 Deskripsi Pengujian

Pengujian dalam penelitian ini terdiri dari dua jenis pengujian utama, yaitu pengujian performa dan pengujian keamanan ciphertext. Pengujian performa bertujuan untuk mengetahui waktu yang dibutuhkan oleh algoritma *Hill Cipher* dan *Affine Cipher* dalam melakukan proses enkripsi dan dekripsi data pengguna. Pengujian keamanan bertujuan untuk menganalisis kualitas ciphertext yang dihasilkan oleh masing-masing algoritma. Pengujian keamanan dilakukan menggunakan pendekatan *Ciphertext-Only Attack* (COA) untuk mengamati kemungkinan pola pada ciphertext serta pengujian *Shannon Entropy* untuk mengukur tingkat keacakan ciphertext sebagai indikator kualitas enkripsi.

4.2.2 Prosedur Pengujian

Prosedur pengujian dilakukan dengan menerapkan algoritma *Hill Cipher* dan *Affine Cipher* secara terpisah pada sistem e-commerce berbasis web yang sama. Data pengguna berupa username, email, dan password dienkripsi terlebih dahulu

sebelum disimpan ke dalam database. Pada saat proses login, sistem melakukan dekripsi terhadap data terenkripsi untuk keperluan autentikasi pengguna.

Pengujian performa dilakukan dengan mengukur waktu yang dibutuhkan untuk proses enkripsi dan dekripsi menggunakan pencatat waktu pada sistem. Pengujian dilakukan terhadap sejumlah data uji dan diulang beberapa kali untuk memperoleh hasil yang stabil. Pengujian keamanan dilakukan dengan menganalisis ciphertext hasil enkripsi menggunakan metode *Ciphertext-Only Attack* (COA) serta menghitung nilai *Shannon Entropy* dari ciphertext yang dihasilkan oleh masing-masing algoritma.

4.2.3 Data Hasil Pengujian

4.2.3.1 Performa Enkripsi dan Dekripsi

Pengujian performa dilakukan dengan mengukur waktu yang dibutuhkan oleh masing-masing algoritma dalam proses enkripsi dan dekripsi data pengguna. Pengukuran dilakukan menggunakan fungsi pencatat waktu pada sistem dan dilakukan beberapa kali pengujian untuk memperoleh hasil yang stabil.

a. Hasil Pengujian Performa *Hill Cipher*

Tabel 4.2 Hasil Pengujian Performa *Hill Cipher*

No	Sampel Data	Rata-Rata Waktu Enkripsi (ms)	Rata-Rata Waktu Dekripsi (ms)
1.	Data 1–100	0.018 ms	0.026 ms
	Total Rata-Rata	**0.018 ms**	**0.026 ms**

Berdasarkan Tabel 4.2 algoritma *Hill Cipher* berhasil melakukan proses enkripsi dan dekripsi data pengguna dengan baik. Seluruh ciphertext pada data uji dapat dikembalikan ke bentuk plaintext secara tepat tanpa ditemukan kesalahan selama proses pengujian. Waktu proses enkripsi dan dekripsi yang dihasilkan relatif stabil pada seluruh data uji.

b. Hasil Pengujian Performa *Affine Cipher*

Tabel 4.3 Hasil Pengujian Performa *Affine Cipher*

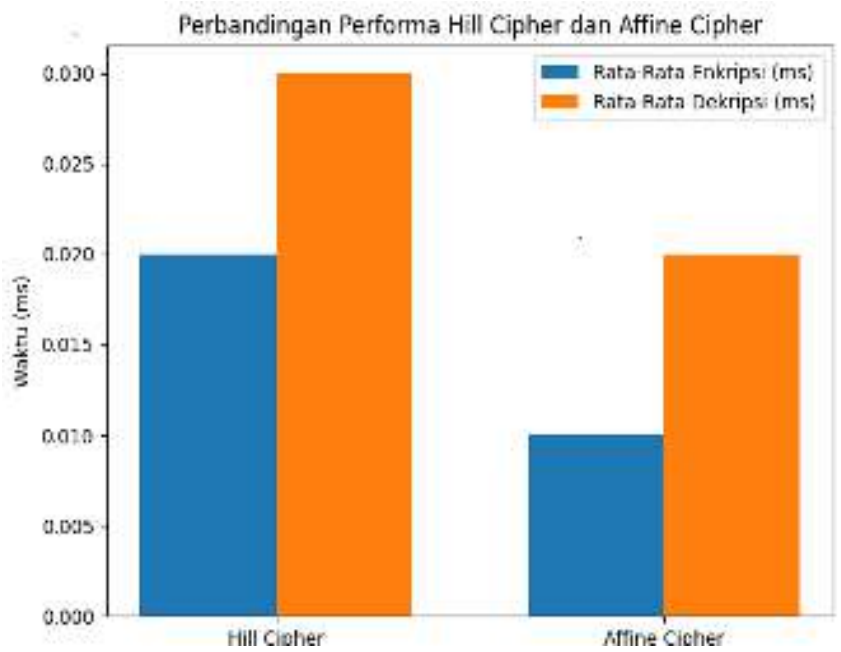
No	Sampel Data	Rata-Rata Waktu Enkripsi (ms)	Rata-Rata Waktu Dekripsi (ms)
1.	Data 1–100	0.010 ms	0.014 ms
	Total Rata-Rata	**0.010 ms**	**0.014 ms**

Hasil pengujian algoritma *Affine Cipher* menunjukkan waktu proses enkripsi dan dekripsi yang lebih cepat, serta mampu mengembalikan seluruh data uji ke bentuk plaintext dengan benar tanpa ditemukan kesalahan selama proses pengujian.

c. Hasil Perbandingan Performa *Hill Cipher* dan *Affine Cipher*

Tabel 4.4 Perbandingan Performa *Hill Cipher* dan *Affine Cipher*

Algoritma	Rata-Rata Enkripsi (ms)	Rata-Rata Dekripsi (ms)	Kecepatan Eksekusi	Kompleksitas Ciphertext
Hill Cipher	0.018 ms	0.026 ms	Lebih lambat	Lebih kompleks
Affine Cipher	0.010 ms	0.014 ms	Lebih cepat	Kurang kompleks



Gambar 4.9 Hasil Perbandingan Performa *Hill Cipher* dan *Affine Cipher*

Berdasarkan Gambar 4.9 perbandingan nilai rata-rata waktu enkripsi dan dekripsi menunjukkan bahwa algoritma Affine Cipher memiliki waktu proses yang lebih cepat dibandingkan algoritma Hill Cipher. Hal ini ditunjukkan oleh nilai rata-rata waktu enkripsi dan dekripsi Affine Cipher yang lebih rendah.

Meskipun demikian, algoritma Hill Cipher menghasilkan ciphertext dengan tingkat kompleksitas yang lebih tinggi karena menggunakan operasi matriks pada proses enkripsi. Perbedaan ini menunjukkan bahwa Affine Cipher unggul dari sisi kecepatan eksekusi, sedangkan Hill Cipher memiliki karakteristik ciphertext yang lebih kompleks dalam skenario pengujian yang dilakukan.

4.2.3.2 Pengujian *Ciphertext Only Attack* (COA)

Tabel 4.5 Hasil Pengujian *Ciphertext Only Attack* (COA)

Algoritma	Data	Plaintext	Ciphertext	Keterangan
Hill Cipher	Username	nia	85e923c2	Ciphertext bersifat acak dan tidak menyerupai data asli
	Password	nia@gmail.com	85e9b6b72f63	Ciphertext bersifat acak dan tidak menyerupai data asli
	Email	nia123	85e9e3027cef5ecfcebe76f147da	Ciphertext bersifat acak dan tidak menyerupai data asli
Affine Cipher	Username	nia	2e15ed	Ciphertext bersifat acak dan tidak menyerupai data asli
	Password	nia@gmail.com	2e15edfd0207	Ciphertext bersifat acak dan tidak menyerupai data asli
	Email	nia123	2e15ed480b29ed1524eef73329	Ciphertext bersifat acak dan tidak menyerupai data asli

Hasil pengujian menunjukkan bahwa ciphertext yang dihasilkan oleh algoritma *Hill Cipher* maupun *Affine Cipher* tidak menampilkan pola bermakna yang dapat digunakan untuk menebak atau merekonstruksi plaintext. Ciphertext yang dihasilkan tidak menyerupai format data asli dan tidak menunjukkan hubungan linier yang jelas dengan plainteks. Namun demikian, secara visual dan struktural, ciphertext hasil *Hill Cipher* tampak memiliki tingkat kompleksitas yang sedikit

lebih tinggi dibandingkan *Affine Cipher*, yang disebabkan oleh proses transformasi matriks yang menyebarkan informasi plainteks ke dalam beberapa karakter ciphertext.

4.2.3.3 Pengujian *Shannon Entropy*

Pengujian *Shannon Entropy* dilakukan untuk mengukur tingkat keacakan ciphertext yang dihasilkan oleh algoritma *Hill Cipher* dan *Affine Cipher* sebagai pendekatan kuantitatif terhadap kualitas enkripsi. Ciphertext yang digunakan dalam pengujian ini merupakan hasil enkripsi gabungan data pengguna yang terdiri dari username, email, dan password, sebagaimana diterapkan pada sistem e-commerce berbasis web.

Sebagai contoh perhitungan manual, ciphertext hasil enkripsi menggunakan algoritma Affine Cipher adalah sebagai berikut:

2e15ed2e15edfd02072e15ed480b29ed1524eef73329

Keterangan: Ciphertext tersebut memiliki panjang 36 karakter

4. Rumus Shannon Entropy

   Perhitungan Shannon Entropy dilakukan menggunakan rumus berikut:

$$\boldsymbol{H = -\sum_{i=1}^{n} P_i \ log_2(P_i)}$$

   Keterangan:

   H = nilai Shannon Entropy

   pi = probabilitas kemunculan karakter ke-i

   n = jumlah karakter unik dalam ciphertext

5. Perhitungan Frekuensi dan Probabilitas Karakter

   Jumlah seluruh karakter pada ciphertext adalah N=36. Probabilitas kemunculan setiap karakter dihitung menggunakan persamaan:

$$\boldsymbol{P_i = \frac{f_i}{44}}$$

dengan fi merupakan frekuensi kemunculan karakter ke-i.

Berdasarkan hasil perhitungan frekuensi kemunculan karakter pada ciphertext, diperoleh 3 kelompok frekuensi Karakter dengan frekuensi 9 kali, 7 kali, 5 kali, 4 kali, 3 kali, 2 kali dan 1 kali.

3. Perhitungan Shannon Entropy
   a) Karakter dengan frekuensi 9

$$p = \frac{9}{44} = 0{,}2045$$

$$H = -(0{,}2045\ x\ log_2(0{,}2045)) = 0{,}4683$$

   b) Karakter dengan frekuensi 7

$$p = \frac{7}{44} = 0{,}1590$$

$$H = -(0{,}1590\ x\ log_2(0{,}1590)) = 0{,}4219$$

   c) Karakter dengan frekuensi 5

$$p = \frac{5}{44} = 0{,}1136$$

$$H = -(0{,}1136\ x\ log_2(0{,}1136)) = 0{,}3565$$

   d) Karakter dengan frekuensi 4

$$p = \frac{4}{44} = 0{,}0909$$

$$H = -(0{,}0909\ x\ log_2(0{,}0909)) = 0{,}3145$$

   e) Karakter dengan frekuensi 3

$$p = \frac{3}{44} = 0{,}0681$$

$$H = -(0{,}0681\ x\ log_2(0{,}0681)) = 0{,}2642$$

f) Karakter dengan frekuensi 2

$$p = \frac{2}{44} = 0{,}0454$$

$$H = -(0{,}0454\ x\ log_2(0{,}0454)) = 0{,}2027$$

g) Karakter dengan frekuensi 1

$$p = \frac{1}{44} = 0{,}0227$$

$$H = -(0{,}0277\ x\ log_2(0{,}0277)) = 0{,}1241$$

Sehingga total Shannon Entropy dihitung sebagai berikut:

$$H = (1\ x\ 0{,}4683\ ) + (1\ x\ 0{,}4219) + (1\ x\ 0{,}3565) + (2\ x\ 0{,}3145) + (1\ x\ 0{,}2642) + (6\ x\ 0{,}2027) + (2\ x\ 0{,}1241)$$

$$H = 0{,}4683 + 0{,}4219 + 0{,}3565 + 0{,}629 + 0{,}2642 + 1{,}2162 + 0{,}2482$$

$$H = 3{,}6043$$

Nilai tersebut kemudian dibulatkan menjadi 3,60 bit.

2e15ed2e15edfd02072e15ed480b29ed1524eef73329	2	2	7	0,1591	0,4219
	e	e	9	0,2045	0,4683
	1	1	4	0,0909	0,3145
	5	5	4	0,0909	0,3145
	e	d	5	0,1136	0,3565
	d	f	2	0,0455	0,2027
	2	0	3	0,0682	0,2642
	e	7	2	0,0455	0,2027
	1	4	2	0,0455	0,2027
	5	8	1	0,0227	0,1241
	e	b	1	0,0227	0,1241
	d	9	2	0,0455	0,2027
	f	3	2	0,0455	0,2027
	d	9	2	0,0455	0,2027
	0				3,6043
	2				
	0				
	7				
	2				
	e				
	1				
	5				
	e				
	d				
	4				
	8				
	0				
	b				
	2				
	9				
	e				
	d				
	1				
	5				
	2				
	4				
	e				
	e				
	f				
	7				
	3				
	3				
	2				

Gambar 4.10 Hasil Perhitungan *Shannon Entropy*

Secara teoritis, nilai maksimum Shannon Entropy ditentukan oleh jumlah simbol yang digunakan dalam representasi data. Pada data berbasis byte, nilai entropy maksimum adalah 8 bit, sedangkan nilai yang mendekati 0 menunjukkan tingkat keacakan yang rendah. Nilai entropy yang mendekati nilai maksimum menunjukkan tingkat keacakan yang tinggi dan ciphertext sulit diprediksi [21].

Dalam penelitian ini, nilai Shannon Entropy yang diperoleh berada di bawah nilai maksimum teoritis tersebut, sehingga tingkat keacakan ciphertext dikategorikan berdasarkan rentang nilai entropy yang dihasilkan, yaitu tingkat keacakan tinggi untuk Hill Cipher dan sedang untuk Affine Cipher.

Tabel 4.6 Hasil Pengujian *Shannon Entropy*

Algoritma	Ciphertext	Entropy (bit)	Tingkat Keacakan
Hill Cipher	85e923c285e9b6b72f6385e9e3027cef5ecfcebe76f147da	4,55	Tinggi
Affine Cipher	2e15ed2e15edfd02072e15ed480b29ed1524eef73329	3,60	Sedang

Hasil pengujian menunjukkan bahwa algoritma Hill Cipher menghasilkan nilai Shannon Entropy sebesar 4,55 bit, yang berada pada kategori tingkat keacakan tinggi, sedangkan algoritma Affine Cipher menghasilkan nilai Shannon Entropy sedikit lebih rendah sebesar 3,60 bit, yang berada pada kategori tingkat keacakan sedang juga.

Nilai entropy Hill Cipher menunjukkan bahwa algoritma ini memiliki distribusi karakter yang sedikit lebih merata dan tingkat penyebaran informasi yang lebih baik dibandingkan Affine Cipher. Sementara itu, nilai entropy Affine Cipher yang berada pada kategori sedang mengindikasikan bahwa ciphertext masih memiliki pola kemunculan karakter tertentu, yang merupakan karakteristik dari algoritma kriptografi klasik.

4.3 Analisis Perbandingan *Hill Cipher* dan *Affine Cipher*

Analisis perbandingan dilakukan untuk mengevaluasi perbedaan karakteristik algoritma Hill Cipher dan Affine Cipher. Berdasarkan hasil pengujian yang telah diperoleh evaluasi ini mencakup aspek performa enkripsi–dekripsi, ketahanan terhadap uji Ciphertext-Only Attack (COA), serta tingkat keacakan ciphertext yang diukur menggunakan Shannon Entropy. Ringkasan hasil pengujian tersebut disajikan pada Tabel berikut:

Tabel 4.7 Perbandingan *Hill Cipher* dan *Affine Cipher*

Algoritma	Performa Enkripsi Dekripsi	COA	Shannon Entropy	Keterangan
Hill Cipher	Enkripsi: 0,018 ms Dekripsi: 0,026 ms	Ciphertext tidak menyerupai plaintext dan bersifat acak	4,55 bit (Tingkat keacakan tinggi)	Lebih Unggul dari sisi keamanan karena distribusi karakter lebih merata dan kompleksitas ciphertext sedikit lebih tinggi
Affine Cipher	Enkripsi: 0,010 ms Dekripsi: 0,014 ms	Ciphertext tidak menyerupai plaintext dan bersifat acak	3,60 bit (Tingkat keacakan sedang)	Lebih Unggul dari sisi kecepatan eksekusi, namun tingkat keacakan lebih rendah

Berdasarkan Tabel 4.7, Affine Cipher menunjukkan waktu enkripsi dan dekripsi yang lebih cepat dibandingkan Hill Cipher karena menggunakan operasi aritmetika linier sederhana. Sebaliknya, Hill Cipher memerlukan operasi matriks sehingga waktu proses relatif lebih lama.

Dari sisi keamanan, kedua algoritma tidak menunjukkan pola bermakna pada uji COA. Namun, Hill Cipher memiliki nilai Shannon Entropy yang lebih tinggi, yang menunjukkan tingkat keacakan dan distribusi karakter ciphertext yang lebih merata. Hal ini mengindikasikan bahwa Hill Cipher memberikan tingkat penyamaran data yang lebih baik dibandingkan Affine Cipher.

Secara keseluruhan, Affine Cipher lebih unggul dalam efisiensi proses, sedangkan Hill Cipher lebih unggul dalam kualitas keamanan ciphertext. Oleh karena itu, pemilihan algoritma perlu disesuaikan dengan kebutuhan sistem, apakah lebih mengutamakan performa atau tingkat keamanan data.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Algoritma *Hill Cipher* dan *Affine Cipher* sebagai bagian dari kriptografi klasik memiliki mekanisme kerja yang berbeda, di mana *Hill Cipher* menggunakan operasi matriks dalam proses enkripsi dan dekripsi, sedangkan *Affine Cipher* menggunakan operasi aritmatika linear. Perbedaan mekanisme ini memengaruhi karakteristik proses enkripsi dan bentuk ciphertext yang dihasilkan oleh masing-masing algoritma.
2. Algoritma *Hill Cipher* dan *Affine Cipher* berhasil diterapkan pada sistem web e-commerce untuk melakukan proses enkripsi dan dekripsi data pengguna sebelum disimpan ke dalam database. Implementasi ini menunjukkan bahwa kedua algoritma dapat digunakan sebagai mekanisme pengamanan data pada sistem web sesuai dengan kebutuhan penelitian.
3. Hasil perbandingan tingkat keamanan dan performa menunjukkan bahwa kedua algoritma mampu menghasilkan ciphertext tanpa pola bermakna berdasarkan pendekatan Ciphertext-Only Attack (COA). Berdasarkan pengujian Shannon Entropy, Hill Cipher menghasilkan ciphertext dengan tingkat keacakan tinggi, sedangkan Affine Cipher berada pada kategori tingkat keacakan sedang. Dari sisi performa, Affine Cipher memiliki waktu proses enkripsi dan dekripsi yang lebih cepat, sedangkan Hill Cipher menghasilkan ciphertext dengan tingkat kompleksitas yang lebih tinggi akibat penggunaan operasi matriks.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penelitian selanjutnya disarankan untuk menggunakan algoritma kriptografi yang lebih modern atau mengombinasikan algoritma kriptografi klasik dengan metode keamanan lain guna

meningkatkan tingkat pengamanan data. Selain itu, pengujian keamanan dapat dikembangkan dengan pendekatan kriptanalisis yang lebih beragam, seperti analisis statistik lanjutan terhadap ciphertext atau pengujian *Shannon Entropy* menggunakan dataset yang lebih besar agar diperoleh hasil analisis yang lebih komprehensif. Penelitian berikutnya juga dapat mengkaji penerapan algoritma kriptografi dalam konteks sistem e-commerce dengan mempertimbangkan mekanisme keamanan tambahan pada tingkat aplikasi, seperti kontrol akses atau monitoring data terenkripsi, sehingga diperoleh gambaran perlindungan data yang lebih menyeluruh tanpa terbatas pada pengembangan sistem secara langsung.

DAFTAR PUSTAKA

[1] Dwi Haryono, et al. (2020). Keamanan Data dengan Teknik Kriptografi.Yogyakarta: Deepublish.

[2] Syarif, A. (2020). Kriptografi Klasik dan Modern. Jakarta: Media Teknika.

[3] Roman Gusmana, dkk (2023). "Implementasi Algoritma Hill Cipher Menggunakan Kunci Matriks 2x2 Dalam Mengamankan Data Teks". Jurnal Teknik Informatika, Vol. 10, No. 1.

[4] Fitri, A., et al. (2023). "Algoritma Affine Cipher pada Enkripsi dan Deskripsi untuk Keamanan Informasi Berbasis Android". Jurnal Teknologi Informasi dan Komputer, Vol. 9, No. 2.

[5] Adnan Buyung Nasution . (2020). "Modifikasi Algoritma Affine Cipher untuk Mengamankan Data". Jurnal RESTI, Vol. 4, No. 5.

[6] Nurharianna Siregar, dkk. (2022). "Menerapkan Algoritma Hill Cipher dan Matriks 2x2 dalam Mengamankan File Teks Menggunakan Kode ASCII". Jurnal Informatika, Vol. 8, No. 3.

[7] Roman Gusmana, et al. (2022). "Implementasi Algoritma Affine Cipher dan Caesar Cipher dalam Mengamankan Data Teks". Jurnal Teknologi dan Sistem Komputer, Vol. 10, No. 4.

[8] Putra, dkk (2024). "Perbandingan Kriptografi Klasik Hill Cipher dengan Affine Cipher dalam Pengamanan Data Citra". Jurnal Ilmu Komputer dan Informatika, Vol. 11, No. 1.

[9] Ahmad Sayuti, dkk. (2021). "Perbandingan Performa Algoritma Hill Cipher dengan RSA dalam Proses Enkripsi dan Dekripsi Teks". Jurnal Rekayasa Sistem Komputer, Vol. 6, No. 1.

[10] Kamil Hidayatuloh., Yustantina, & Kusmadi. (2021). "Perbandingan Metode Stream Cipher dan Hill Cipher dalam Keamanan Data". Jurnal Infotronik, Vol. 6, No. 1.

[11] Ilfi Nur Diana. (2022)."Algoritma Affine Cipher dan Modifikasi serta Kombinasinya dengan Cipher Transposisi Grup Simetri untuk Mengamankan Pesan Teks".Jurnal Informatika dan Rekayasa Perangkat Lunak,Vol.8,No. 1.

[12] Ema Satriana Br Karo. (2020). "Penerapan Algoritma Affine Cipher dan Electronic Code Book (ECB) dalam Pengamanan Pesan Teks". Jurnal Sistem Informasi, Vol. 7, No. 2.

[13] Mohammad Azmi Abdussyukur, dkk (2025). “Analisis Metode Affine Cipher dengan Keystream Acak untuk Meningkatkan Keamanan Data dalam Sistem Kriptografi”. JATI (Jurnal Mahasiswa Teknik Informatika), Vol. 9, No. 2.

[14] Feriyanto, dkk (2024). Kemudahan dan Keamanan Transaksi E-Commerce dan Inovasi Pembayaran Digital.Jurnal STAR,Universitas Teknologi Digital.

[15] Sumarni, dkk (2025). Menyesuaikan Konsep Web E-Commerce dengan Tujuan dan Model Bisnis yang di Inginkan. Jurnal Manajemen dan Bisnis Ekonomi, Vol. 3, No. 1.

[16] Aam Alfain Hidayatullah, (2024).Kombinasi Algoritma Caesar Cipher dan Algoritma Affine Cipher. Maliki Interdisciplinary Journal (MIJ),Vol.2,No. 2.

[17] Abidah Nabilah, (2023). Penerapan algoritma Hill Cipher untuk keamanan data aplikasi pembuatan surat tanah kantor desa (Skripsi, Politeknik Negeri Bengkalis, Program Studi Rekayasa Perangkat Lunak).

[18] Lestari, A., Hadiana, A. I., & Melina. “Implementasi Algoritma Rivest Shamir Adleman (RSA) dan Zero-Knowledge Proofs (ZKP) untuk Meningkatkan Keamanan Data Rekam Medis Elektronik,” Jurnal Algoritma, vol. 22, no. 2.

[19] Muhammad Naufal Zhafran Soleh, Asep Id Hadiana, dan Fatan Kasyidi (2024). Kriptografi Homomorfik dalam Anonimisasi Data untuk Pengolahan Data pada Sistem E-Voting. Jurnal Masyarakat Informatika, Vol. 15, No. 2.

[20] Behrouz Zolfaghari, Khodakhast Bibak, dan Takeshi Koshiba, “The Odyssey of Entropy: Cryptography,” Entropy, vol. 24, no. 3, 2022.

[21] Hasan Shadzily dan Bambang Sujatmiko, “Analisis Tingkat Keamanan File Dokumen Menggunakan Algoritma Advanced Encryption Standard (AES) Berdasarkan Nilai Shannon Entropy,” INOVATE: Jurnal Ilmiah Inovasi Teknologi Informasi, vol. 5, no. 1, 2025.