PENGKODEAN TEKS MENGGUNAKAN *HILL CIPHER* DENGAN KUNCI BERANTAI
(Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala)

TESIS

Oleh

Muhamat Abdul Rohim

NIM 201820101004

PROGRAM STUDI MAGISTER MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS JEMBER

2023

PENGKODEAN TEKS MENGGUNAKAN *HILL CIPHER* DENGAN KUNCI BERANTAI (Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala)

TESIS

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Magister (S2) Program Studi Magister Matematika Universitas Jember dan mencapai gelar Magister Sains

Oleh

Muhamat Abdul Rohim

NIM 201820101004

PROGRAM STUDI MAGISTER MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS JEMBER

2023

PERSEMBAHAN

Dengan menyebut nama Allah yang Maha Pengasih dan Maha Penyayang serta sholawat dan salam selalu tercurah kepada Nabi Muhammad SAW, penulis senantiasa persembahkan Tesis ini sebagai ungkapan kebahagiaan dan rasa terima kasih kepada:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam pengerjaan tesis;
2. Istri tercinta Nur Afifah yang selalu mendukung dalam segala hal selama pengerjaan tesis;
3. Anak tercinta, Syifa Zahira Humairoh yang selalu memberikan semangat dengan senyuman polosnya;
4. Ayahanda Muki, ibunda Rohmani, dan nenek saya Alm. Muntini yang secara tidak langsung terus memberikan dukungan dan doanya kepada penulis;
5. Difari Afreyna Fauziah dan Ferry Wiranto yang telah berkontribusi dalam akomodasi kertas *Copy Paper* 70 Gsm pada saat sidang.
6. Rekan-rekan kerja di UPT-TI Institut Teknologi dan Sains Mandala yang selalu memberikan dukungan dengan caranya sendiri;
7. Guru - guruku sejak sekolah dasar sampai dengan perguruan tinggi yang telah memberikan ilmunya kepada penulis;
8. Almamater Program Studi Magister Matematika Universitas Jember dan Fakultas Matematika dan Ilmu Pengetahuan Alam;

MOTTO

"Belajar hidup untuk sebuah kehidupan"

"Walikulli Ummatin Ajalun Fa-Idzaa Jaa-A Ajaluhum Laa Yasta'khiruuna Saa'atan Walaa Yastaqdimuun"

~ QS. Al-A'raaf:34 ~

PERYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhamat Abdul Rohim

NIM : 201820101004

menyatakan dengan sesungguhnya bahwa Tesis yang berjudul "Pengkodean Teks menggunakan *Hill Cipher* dengan Kunci Berantai (Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala" adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak mana saja serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 01 November 2022

Yang menyatakan,

Muhamat Abdul Rohim

NIM 201820101004

TESIS

PENGKODEAN TEKS MENGGUNAKAN *HILL CIPHER* DENGAN KUNCI BERANTAI (Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala)

Oleh

Muhamat Abdul Rohim

NIM 201820101004

Pembimbing:

Dosen Pembimbing Utama : Dr. Kiswara Agung Santoso S,Si., M.Kom

Dosen Pembimbing Anggota : Dr. Alfian Futuhul Hadi, S.Si., M.Si

PENGESAHAN

Tesis berjudul "Pengkodean Teks menggunakan *Hill Cipher* dengan Kunci Berantai (Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala)" karya Muhamat Abdul Rohim telah diuji dan disahkan pada:

hari, tanggal : Jumat, 27 Januari 2023

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Tim Penguji

Ketua,	Anggota I,
Dr. Kiswara Agung Santoso S,Si., M.Kom	Dr. Alfian Futuhul Hadi, S.Si., M.Si
NIP. 197209071998031003	NIP. 197407192000121001
Anggota II	Anggota III
Drs. Moh. Hasan, M.Sc., Ph.D	Prof. Drs. I Made Tirta, M.Sc., Ph.D.
NIP. 196404041988021001	NIP. 195912201985031002

Mengesahkan
Dekan,

Drs. Achmad Sjaifullah, M.sc., Ph.D
NIP. 195910091986021001

RINGKASAN

Pengkodean Teks menggunakan *Hill Cipher* dengan Kunci Berantai (Studi Kasus *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala); Muhamat Abdul Rohim, 201820101004; 2023, 70 Halaman; Program Studi Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Asosiasi Penyelenggara Jasa Internet Indonesi (APJII) merilis hasil survei tentang profil internet di Indonesia. Survei tersebut menunjukkan bahwa tingkat penetrasi pengguna internet di Indonesia selama tahun 2021 sampai dengan 2022 sebesar 77,02%. Tingginya tingkat penetrasi ini mengakibatkan tingkat pencurian data di internet juga tinggi, pada tahun 2019 terdapat 106 pencurian data pada sektor perbankan, 96 pencurian data pada sektor pinjaman *online*, disusul belanja *online* dengan kasus pencurian data sebanyak 34.

Institut Teknologi dan Sains Mandala (ITSM) sebagai salah satu perguruan tinggi swasta yang ada di kabupaten Jember memiliki layanan pendaftaran mahasiswa baru melalui internet atau berbasis daring. Pada proses pendaftarannya, calon mahasiswa akan mendaftarkan nama dan alamat email aktif melalui situs Penerimaan Mahasiswa Baru ITSM. Situs tersebut akan secara otomatis membuat *registered_id* berupa *Universally Unique Identifier* (UUID) yang unik untuk masing-masing pendaftar, kemudian menyimpannya ke dalam *database* dan mengirimkan tautan yang berisi *registered_id* tersebut ke email yang didaftarkan. *Registered_id* yang dikirim terkait langsung dengan identitas pendaftar (calon mahasiswa baru).

Identitas pendaftar menjadi sangat penting untuk dirahasiakan pada saat proses pendaftaran mahasiswa baru berbasis daring, sehingga diperlukan proses pengkodean (enkripsi) identitas dalam sistem yang berjalan. *Hill Cipher* merupakan salah satu algoritma kriptografi yang memanfaatkan operasi perkalian dan *invers* matriks didalamnya, pada *Hill Cipher* biasa, satu kunci akan digunakan untuk melakukan enkripsi keseluruhan *plaintext*. Hal ini mengakibatkan jika satu kunci itu berhasil ditemukan maka keseluruhan *plaintext* akan dapat didekripsi dengan mudah.

Penelitian ini memodifikasi *Hill Cipher* menjadi *Hill Cipher* berantai. Proses *Hill Cipher* berantai akan dilakukan pada proses enkripsi dan dekripsi *primary key* dengan cara menjadikan hasil enkripsi blok karakter sebelumnya menjadi kunci untuk enkripsi blok karakter selanjutnya, dengan kunci awal didasarkan pada kolom nama pendaftar dari *primary key* yang akan dienkripsi. Hasil implementasi *Hill Cipher* berantai ini membuat tingkat kesulitan melakukan dekripsi lebih sulit dibandingkan dengan *Hill Cipher* biasa yang hanya memakai satu kunci. Hal ini ditunjukkan dengan hasil percobaan kriptanalisis menggunakan *known-plaintext* yang dilakukan penulis terhadap kedua metode tersebut, dengan proses dan tahapan yang sama, matriks kunci K pada algoritma *Hill Cipher* biasa dapat ditemukan dan berhasil dipakai untuk mendekripsi keseluruhan *ciphertext* menjadi *plaintext* yang benar, sedangkan Algoritma *Hill Cipher* berantai, kunci yang didapatkan tidak bisa digunakan untuk keseluruhan *ciphertext*, sehingga keamanan data yang diperoleh juga semakin baik.

PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tesis dengan judul "Pengkodean Teks menggunakan *Hill Cipher* Dengan kunci Berantai (Studi Kasus: *Database* Pendaftaran Mahasiswa Baru Institut Teknologi dan Sains Mandala)". Tesis ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Pascasarjana (S2) pada Program Studi Magister Matematika Jurusan Matematika Fakultas Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan tesis ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
2. Koordinator Prodi Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
3. Dr. Kiswara Agung Santoso S,Si., M.Kom selaku dosen pembimbing utama dan Dr. Alfian Futuhul Hadi, S.Si., M.Si selaku dosen pembimbing anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam membantu penulisan tesis ini;
4. Drs. Moh. Hasan, M.Sc., Ph.D selaku dosen penguji utama dan Prof. Drs. I Made Tirta, M.Sc., Ph.D. selaku dosen penguji anggota yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan tesis ini;

Semoga bantuan, bimbingan, dan dorongan yang telah diberikan dicatat sebagai amal baik oleh Allah SWT dan mendapat balasan yang sesuai dari-Nya. Penulis menyadari bahwa Tesis ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap Tesis ini dapat bermanfaat bagi semua pihak.

Jember, November 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

Proses pendaftaran mahasiswa baru di Institut Teknologi dan Sains Mandala saat ini berbasis daring, hal ini menuntut keamanan sistem informasi yang menjamin kerahasiaan data-data pribadi milik pendaftar. Pada bab ini akan dijelaskan latar belakang kenapa data-data pendaftar itu penting dan harus dirahasiakan, permasalahan apa yang akan penulis teliti, solusi apa yang ditawarkan penulis, dan apa saja manfaat yang didapatkan pada bidang akademik, penulis sebagai peneliti, maupun pengguna yang menggunakan sistem secara langsung.

1.1 Latar Belakang

APJII (2022) merilis data tentang profil internet Indonesia pada tahun 2021 sampai 2022, data tersebut menunjukkan bahwa tingkat penetrasi internet di Indonesia mencapai 77,02% meningkat dari tahun sebelumnya yang mencapai 73,70%. Ada banyak alasan pengguna melakukan penetrasi internet, alasan terbanyak adalah untuk mengakses sosial media seperti *Facebook, WhatsApp, Telegram,* dan lain sebagainya. Media sosial juga menjadi konten internet yang paling sering diakses pada tahun 2021 hingga 2022.

Tingginya tingkat penetrasi internet di Indonesia juga mengakibatkan tingkat kebocoran data melalui internet juga bertambah. Setiawan *et al*. (2022) menjelaskan bahwa berdasarkan sumber data milik Yayasan Lembaga Konsumen Indonesia (YLKI) pada tahun 2019 terdapat 106 pencurian data pada sektor perbankan, 96 pencurian data pada sektor pinjaman *online*, disusul belanja *online* dengan kasus pencurian data sebanyak 34. Banyaknya kasus pencurian data ini mengindikasikan bahwa kerahasiaan data milik pengguna internet masih sangat rentan untuk dicuri dan dimanfaatkan oleh pihak yang tidak bertanggung jawab.

Institut Teknologi dan Sains Mandala (ITSM) sebagai salah satu perguruan tinggi swasta yang ada di kabupaten Jember memiliki layanan pendaftaran mahasiswa baru secara daring. Pada proses pendaftaran mahasiswa baru secara daring, calon mahasiswa akan mendaftarkan nama dan alamat email aktif melalui

situs Penerimaan Mahasiswa Baru ITSM. Situs tersebut akan secara otomatis membuat *registered_id* berupa *Universally Unique Identifier* (UUID) yang unik untuk masing-masing pendaftar, kemudian menyimpannya ke dalam *database* dan mengirimkan tautan yang berisi *registered_id* tersebut ke email yang didaftarkan.

Registered_id tersebut sangat penting dan bersifat rahasia agar informasi terkait data-data pribadi pendaftar tidak dicuri dan dimanfaatkan pihak ketiga yang berusaha mengambil informasi tersebut. Menurut Peraturan Menteri Komunikasi dan Informatika Nomor 20 Tahun 2016, data pribadi adalah data perseorangan tertentu yang disimpan, dirawat, dan dijaga kebenaran serta dilindungi kerahasiaannya (KOMINFO, 2016).

Kerahasiaan *registered_id* yang dikirim ke email pendaftar dapat dijaga dengan menerapkan proses kriptografi pada saat pengiriman dilakukan. Menurut Pricilla (2015) bahwa kriptografi adalah suatu ilmu untuk menjaga kerahasiaan dari suatu pesan atau informasi dengan cara mengubah bentuk (mengenkripsi) informasi tersebut ke dalam bentuk yang lain, sehingga makna yang dimiliki tidak bisa dimengerti tanpa proses pemecahan kode (dekripsi) terlebih dahulu. Ada dua proses utama pada kriptografi, yaitu proses mengkodekan pesan atau enkripsi dan proses memecah pesan atau dekripsi.

Berbagai algoritma kriptografi yang berkembang saat ini, diantaranya adalah *Hill Cipher*. Menurut Yuliandaru (2015) bahwa *Hill Cipher Cryptography* adalah algoritma kriptografi yang menggunakan teknik perkalian dan *invers* matriks untuk proses enkripsi dan dekripsi. Menurut Maryanti *et al*. (2018) matriks yang digunakan pada algoritma ini adalah matriks yang berukuran $n \times n$ dan *invertible*.

Berbagai penelitian telah dilakukan terkait dengan penggunaan *Hill Cipher* pada proses kriptografi. Hasugian (2013) pada penelitiannya menerapkan *Hill Cipher* untuk membuat sebuah aplikasi dengan bahasa pemrograman *visual basic* 6.0 dengan menggunakan matriks ordo 2×2 sebagai kunci dan hanya menyandikan *plaintext* berupa huruf, sehingga digunakan modulo 26.

Penelitian sejenis juga dilakukan oleh Hidayat dan Alawiyah (2013) yang modifikasi kunci dengan memanfaatkan matriks persegi panjang sebagai kunci pada algoritma *Hill Cipher.* Kesimpulan dari penelitian tersebut menjelaskan

bahwa *pseudo-invers* dapat digunakan untuk mencari *invers* matriks kunci $m \times n$ pada *Hill Cipher* dengan syarat $m \geq n$ dan $n \geq 1$.

Penelitian terkait *Hill Cipher* tidak hanya dilakukan pada penerapannya saja. Azhar *et al*. (2017) melakukan penelitian terkait proses kriptanalisis pada algoritma *Hill Cipher*. Penelitian tersebut menerapkan metode *known-plaintext attack* yang diimplementasikan pada sistem berbasis android, sedangkan matriks kunci yang digunakan pada penelitian ini memiliki ordo 2×2. Kesimpulan dari penelitian ini menjelaskan bahwa sistem yang dibangun dapat mencari nilai *variable* matriks kunci berdasarkan determinannya.

Penelitian sejenis juga dilakukan oleh Tuasikal *et al*. (2020) yang membandingkan penggunaan metode *known-plaintext* dan *chosen-plaintext* pada proses kriptanalisis terhadap Algoritma *Hill Cipher*. Matriks kunci yang digunakan pada penelitian tersebut memiliki ordo 2×2 dan 3×3 dan menghasilkan kesimpulan bahwa *chosen-plaintext* bekerja dengan baik pada kunci dengan ordo 2×2 dan 3×3, sedangkan *known-plaintext* bekerja dengan baik pada ordo 2×2, dan kurang baik pada ordo 3×3. Penelitian tersebut memberikan saran agar mengembangkan *Hill Cipher* menjadi *chaining Hill Cipher* agar mempersulit proses kriptanalisisnya.

Penelitian yang dilakukan oleh Widyanarko (2007) melakukan modifikasi *Hill Cipher* menjadi *Chaining Hill Cipher*. Modifikasi yang dilakukan pada penelitian tersebut adalah dengan melakukan proses penjumlahan matriks hasil enkripsi blok karakter sebelumnya dengan hasil enkripsi blok karakter selanjutnya. Proses ini menghasilkan *ciphertext* yang saling terkait satu sama lain seperti rantai sehingga diberikan nama *Chaining Hill Cipher*. Penelitian tersebut menambahkan karakter spasi, titik dan koma pada tabel konversinya, sehingga modulo yang digunakan pada penelitian ini adalah modulo 29. Kesimpulan dari penelitian tersebut menyebutkan bahwa *Chaining Hill Cipher* masih dapat dibobol menggunakan metode *known-plaintext*.

Penelitian yang dilakukan oleh Widyanarko (2007) yang telah dijelaskan sebelumnya masih menggunakan satu kunci utama yang mengakibatkan tingkat kesulitan untuk melakukan dekripsi *ciphertext* relatif rendah, katena jika kunci utama tersebut ditemukan, maka keseluruhan *ciphertext* dapat didekripsi. Salah

satu faktor yang dapat meningkatkan tingkat kesulitan dekripsi adalah dengan memodifikasi proses enkripsinya agar menggunakan lebih dari satu kunci.

Berdasarkan uraian di atas, peneliti ingin melakukan modifikasi terhadap algoritma *Hill Cipher* dengan menggunakan lebih dari satu kunci pada proses enkripsi dan dekripsinya sehingga membentuk sebuah rantai kunci yang saling terkait satu sama lain. Hasil modifikasi ini akan peneliti terapkan pada proses pengiriman *registered_id* pada *database* pendaftaran mahasiswa baru Institut Teknologi dan Sains Mandala agar kerahasiaan data pendaftar lebih bisa dijaga.

1.2 Perumusan Masalah

Berdasarkan uraian dalam latar belakang maka rumusan masalah dalam penelitian ini dapat diuraikan sebagai berikut.

1. Bagaimana proses enkripsi *Hill Cipher* berdasarkan kunci berantai?
2. Bagaimana proses enkripsi *primary key* pada tabel pendaftar di *database* situs Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala menggunakan *Hill Cipher* berantai?
3. Bagaimana perbandingan hasil kriptanalisis terhadap *Hill Cipher* dan *Hill Cipher* berantai?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini sebagai berikut.

1. Membuat teknik pengkodean *Hill Cipher* berdasarkan kunci berantai.
2. Mengkodekan *primary key* pada tabel pendaftar di *database* situs Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala menggunakan *Hill Cipher* berantai.
3. Membandingkan hasil kriptanalisis terhadap *Hill Cipher* dan *Hill Cipher* berantai.

1.4 Manfaat Penelitian

Manfaat yang ingin didapatkan dalam penelitian ini sebagai berikut.

1. **Bidang Akademik,** penelitian yang dilakukan mampu memberikan hasil yang menjadi masukan informasi khususnya kepada jurusan Magister Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember.
2. **Bagi Penulis,** menambah pengetahuan dan melatih kemampuan untuk menerapkan ilmu yang diperoleh tentang Matriks dan Kriptografi khususnya dalam penerapan enkripsi *primary key* menggunakan metode *Hill Cipher* yang telah dimodifikasi pada proses pembentukan kuncinya sehingga menjadi *Hill Cipher* berantai.
3. **Bagi Pengguna,** menambah keamanan data pada situs Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala dengan menerapkan proses enkripsi terhadap *primary key* menggunakan *Hill Cipher* berantai.

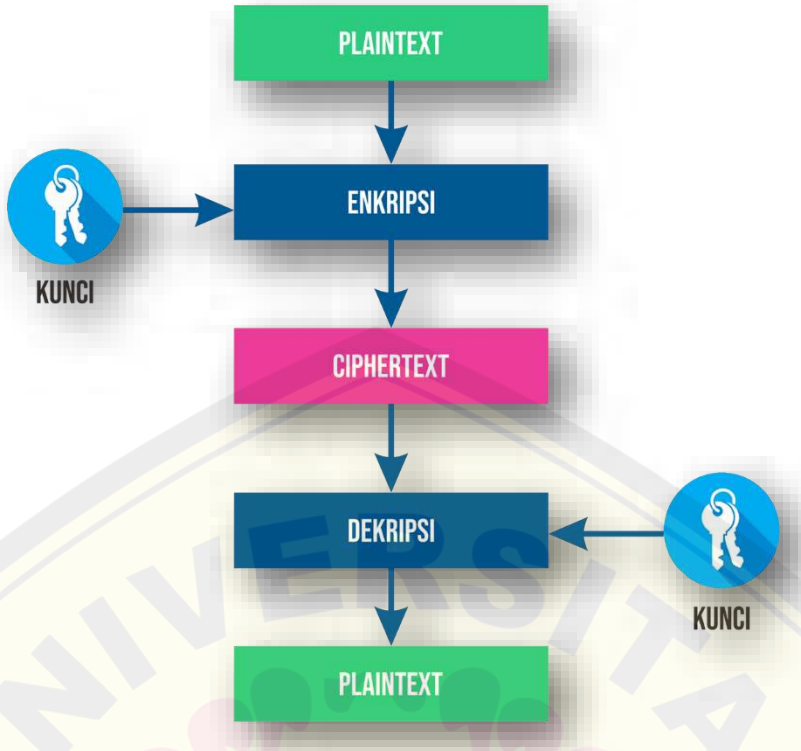
BAB 2. TINJAUAN PUSTAKA

Sistem digital yang memanfaatkan *database* pada prosesnya membutuhkan tingkat keamanan yang cukup untuk menjamin keamanan data yang disimpan di dalamnya. Keamanan tersebut dapat ditingkatkan dengan menerapkan proses kriptografi. Ada banyak teknik yang digunakan dalam kriptografi, diantaranya dengan memanfaatkan *invers* matriks. *Invers* inilah yang digunakan dalam algoritma *Hill Cipher* pada proses enkripsi dan dekripsinya. Pada bab ini akan dijelaskan tentang apa itu *database* dan istilah yang terkait di dalamnya, apa itu kriptografi, determinan, *invers* matriks, kriptanalisis dan apa serta bagaimana algoritma *Hill Cipher* bekerja.

2.1. Kriptografi

Menurut Puspita dan Bahtiar (2010), kriptografi berasal dari bahasa Yunani yaitu kripto dan graphia, kripto berarti menyembunyikan dan graphia berarti tulisan. Menurut Menezes *et al*. (1996) kriptografi artinya ilmu yang mempelajari teknik matematika untuk aspek keamanan informasi seperti kerahasiaan data, integritas data, autentikasi entitas, dan originalitas data.

Ada dua istilah yang dikenal dalam kriptografi, yaitu *plaintext* dan *ciphertext*. Menurut Pricilla (2015) *plaintext* adalah data atau informasi yang dapat dibaca dan dimengerti maknanya oleh penerima pesan, sedangkan *ciphertext* adalah suatu sandi dari pesan sehingga pesan tersebut tidak dapat dipahami maknanya. Gambaran proses kriptografi secara lebih jelas dapat dilihat pada Gambar 2.1.



Gambar 2.1 Proses Kriptografi Secara Umum

Gambar 2.1 menunjukkan bahwa proses kriptografi diawali dengan adanya *Plaintext* yang merupakan pesan rahasia yang akan disandikan. *Plaintext* akan dienkripsi menggunakan kunci tertentu sehingga menghasilkan sebuah pesan yang tidak dapat dipahami yang disebut *Ciphertext*. Proses selanjutnya adalah dekripsi *ciphertext* menggunaka kunci tertentu agar kembali menjadi *plaintext* yang dapat dipahami.

2.2. *Invers* Matriks

Menurut Pricilla (2016), jika A dan B adalah matriks persegi, dan berlaku AB = BA = I, maka dikatakan matriks A dan B saling *invers*, dimana B disebut *invers* dari A, atau ditulis A^{-1}. Menurut Anton dan Rore (2014), jika ada matriks A dengan ordo $n \times n$ sebagai berikut.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Maka, *minor* dari a_{ij} dinotasikan dengan M_{ij} merupakan determinan dari sub-matriks A setelah menghapus bari ke-i dan kolom ke-j. Jika M_{ij} adalah *minor* dari a_{ij}, maka *cofactor* dari a_{ij} dinotasikan dengan C_{ij} didefinisikan dengan persamaan (2.1).

$$C_{ij} = (-1)^{i+j} M_{ij} \tag{2.1}$$

Misal C_{ij} membentuk matriks sebagai berikut.

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

Matriks tersebut disebut dengan matriks *cofactor A*. *Transpose* dari matriks *cofactor A* disebut dengan *adjoint A* dan dinotasikan dengan $adj(A)$. Sebagai contoh misal terdapat matriks persegi A sebagai berikut:

$$A = \begin{bmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{bmatrix}$$

Misal setelah perhitungan *cofactor*, ditemukan *cofactor* sebagai berikut:

$$C_{11} = 12 \qquad C_{12} = 6 \qquad C_{13} = -16$$
$$C_{21} = 4 \qquad C_{22} = 2 \qquad C_{23} = 16$$
$$C_{31} = 12 \qquad C_{32} = -10 \qquad C_{33} = 16$$

Sehingga membentuk matriks *cofactor* sebagai berikut:

$$\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

Maka, $adj(A)$ adalah sebagai berikut:

$$adj(A) = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

Matriks A bersifat *invertible* jika dan hanya jika nilai $\det(A) \neq 0$, sebuah matriks $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $ad - bc$ inilah yang disebut determinan dari sebuah matriks, determinan ditulis dengan persamaan sebagai berikut:

$$\det(A) = ad - bc \text{ atau } \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{2.2}$$

jika matriks A *invertible* maka A^{-1} dapat dicari menggunakan persamaan (2.3) berikut:

$$A^{-1} = \frac{1}{\det(A)} adj(A) \quad (2.3)$$

Persamaan (2.3) digunakan untuk menghitung *invers* matriks secara umum, sedangkan untuk mencari *invers* matriks pada modulo *n* terlebih dahulu harus dicari *reciprocal* dan *residue* pada modulo *n*.

Jika *n* adalah bilangan bulat positif, dan *a, b* adalah sembarang bilangan bulat, maka *a* dikatakan *equivalent* dengan $b \bmod n$ jika $a - b$ adalah kelipatan dari n, dinotasikan dengan

$$a \equiv b \pmod n$$

Pada sembarang modulo *n*, setiap bilang bulat *a* akan *equivalent* dengan tepat satu bilangan bulat $0, 1, 2, \dots, n-1$ dan disebut dengan *residue* dari *a* modulo *n*. Himpunan *residue* modulo *n* dinotasikan dengan

$$Z_n = \{0,1,2, \dots, n-1\}.$$

Untuk sembarang bilangan bulat *a* dan modulo *n*, misal

$$R = remainder \text{ dari } \frac{|a|}{n}$$

maka, *residue* dari *a* modulo *n* adalah *r* dengan nilai

$$r = \begin{cases} R, \text{jika } a \geq 0 \\ (n-R), \text{jika } a < 0 \text{ dan } R \neq 0 \\ 0, \text{jika } a < 0 \text{ dan } R = 0 \end{cases}$$

Jika *a* adalah anggota dari Z_n, dan a^{-1} juga anggota dari Z_n, maka disebut dengan *reciprocal* dari *a* modulo *n* jika berlaku

$$aa^{-1} = a^{-1}a = 1 (\bmod n).$$

Reciprocal a (a^{-1}) modulo *n* dapat dicari dengan mencoba satu persatu anggota Z_n yang memenuhi syarat $aa^{-1} = a^{-1}a = 1 (\bmod n)$.

Setelah diketahui *residue* dan *reciprocal* modulo *n,* maka dapat didefinisikan bahwa matriks persegi *A* dengan elemen matriks berupa anggota Z_n akan *invertible* pada pada modulo *n* jika dan hanya jika *residue* $\det(A) \bmod n$ mempunyai *reciprocal* pada modulo *n*. *Reciprocal* $\det(A)$ atau $\det(A)^{-1}$ dapat dicari dengan bantuan *variable x* pada persamaan (2.4).

$$\det(A) \det(A)^{-1} = \det(A)\, x = 1 \pmod n \quad \dots (2.4)$$

Hitung persamaan (2.4) dengan nilai $x = \{0,1,2, \dots, n-1\}$ sehingga syarat

tersebut terpenuhi. Setelah nilai *x* diketahui, *invers* matriks dapat dihitung dengan persamaan (2.5).

$$A^{-1} = x\ adj(A)\ mod\ n \tag{2.5}$$

Menurut Marzuki (2015), selama ini yang diketahui matriks yang memiliki *invers* adalah matriks bujur sangkar dan *non-singular*, sedangkan matriks *singular* tidak memiliki *invers*. Menurut Kumar (2016) *Generalized Inverse* atau sering disebut *g-inverse*, pada dasarnya adalah perluasan untuk mencari *invers* dari matriks yang tidak bujur sangkar atau *singular*. Istilah lain yang sering digunakan untuk menyebut *g-inverse* adalah *pseudo-inverse*.

Menurut Rao dan Toutenburg (1999), jika ada matriks A dengan ordo $m \times n$, maka matriks A^- disebut *generalized invers* dari A, jika berlaku persamaan (2.6).

$$AA^-A = A \tag{2.6}$$

2.3. *Hill Cipher*

Hidayat dan Alawiyah (2013) menjelaskan bahwa *Hill Cipher* adalah salah satu algoritma kriptografi yang memanfaatkan matriks sebagai kunci pada proses enkripsi dan dekripsinya. Matriks yang digunakan dalam algoritma ini adalah matriks persegi $n \times n$. Dasar yang digunakan dalam algortima ini adalah proses perkalian matriks dengan *invers* matriksnya yang akan menghasilkan matriks identitas. Menurut Anton & Rore (2014), proses enkripsi pada algoritma *Hill Cipher* dapat dilakukan menggunakan persamaan (2.7)

$$C = KP \tag{2.7}$$

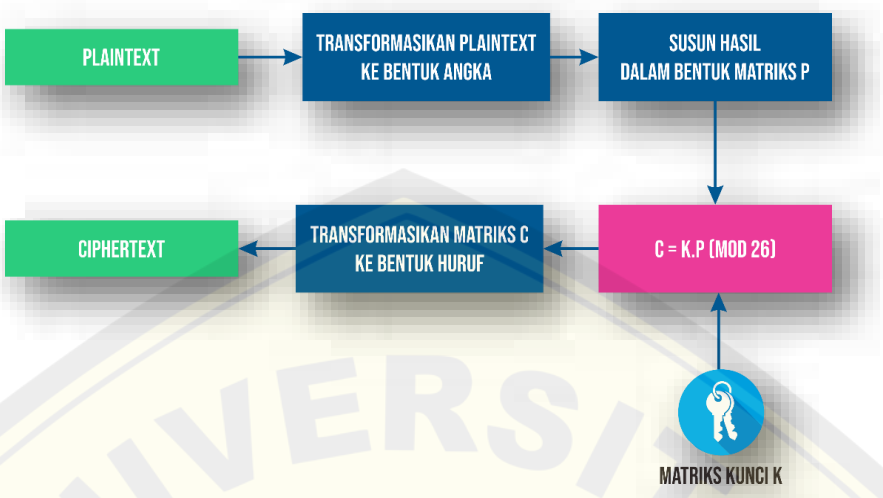
dengan

C: matriks berisi *ciphertext*

K: matriks kunci, dan

P: matriks yang berisi *plaintext*.

Proses enkripsi pada algoritma *Hill Cipher* diawali dengan *plaintext* sebagai *input* awal dan menghasilkan *ciphertext* sebagai *output* akhir. Secara umum proses enkripsi pada algoritma *Hill Cipher* dapat diilustrasikan pada Gambar 2.2.



Gambar 2. 2 Ilustrasi Enkripsi *Hill Cipher*

Berdasarkan gambar 2.2, proses enkripsi diawali dengan menentukan matriks kunci K kemudian melakukan transformasi *plaintext* menjadi bentuk nilai, kemudian disusun sehingga membentuk matriks P. Setelah matriks P ditemukan, langkah selanjutnya adalah melakukan perkalian K dan P sehingga mendapatkan matriks C. Gambar 2.2 menunjukkan bahwa matriks C ditemukan dengan melakukan operasi modulo 26 pada hasil perkalian K dan P, hal ini dilakukan jika menggunakan 26 karakter sebagai dasar transformasinya. Proses enkripsi diakhiri dengan mentransformasikan matriks C ke bentuk huruf.

Widyanarko (2007) memberikan contoh proses enkripsi pada algoritma *Hill Cipher* sebagai berikut:

Jika terdapat *plaintext* P:

$$P: BESOKMALAM$$

Maka *plaintext* tersebut ditransformasikan menjadi:

$$P: 1\ 4\ 18\ 14\ 10\ 12\ 0\ 11\ 0\ 12$$

Misal digunakan matrik $K = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$ sebagai kunci, maka *plaintext* dibagi menjadi beberapa blok yang masing-masing blok memiliki dua karakter sehingga jika dibentuk matriks, akan menjadi matriks kolom. Blok pertama dari P adalah

$$P_{1,2} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

Blok *plaintext* yang terbentuk dienkripsi menggunakan persamaan (2.7) sebagai berikut:

$$C = KP$$

$$C_{1,2} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \end{bmatrix} mod\ 26 = \begin{bmatrix} 9 \\ 7 \end{bmatrix}$$

9 dan 7 kemudian ditransformasikan ke dalam bentuk huruf, menjadi J dan H, maka *ciphertext* untuk *plaintext* BE adalah JH. Setelah melakukan enkripsi terhadap semua blok yang terbentuk, didapatkan *ciphertext* berupa JHUQIQWLYM.

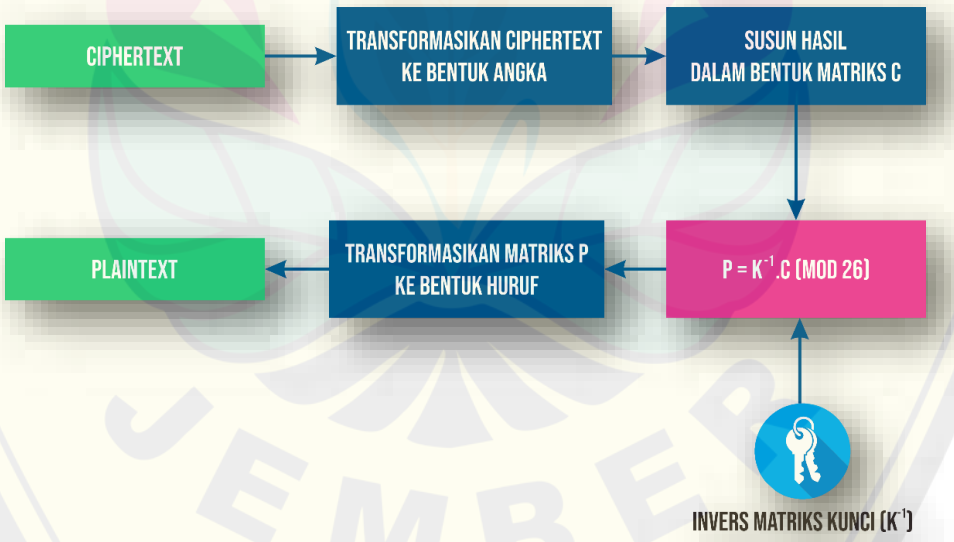
Setelah proses enkripsi, proses selanjutnya adalah proses dekripsi. Hasugian (2013) menjelaskan bahwa proses dekripsi dapat dilakukan menggunakan persamaan (2.8) berikut:

$$P = K^{-1}C \tag{2.8}$$

Dengan

K^{-1}: *invers* dari matriks kunci K

Proses dekripsi secara keseluruhan seperti pada Gambar 2.3 berikut.



Gambar 2. 3 Gambaran Proses Dekripsi

Berdasarkan gambar 2.3, proses dekripsi diawali dengan menghitung K^{-1} dari matriks kunci yang digunakan. Setekah K^{-1} ditemukan transformasikan *ciphertext* ke dalam bentuk nilai dan disusun sehingga menjadi matriks C. Matriks

P dicari menggunakan persamaan (2.8) dengan melakukan operasi modulo 26 sama seperti proses enkripsinya. Setelah matriks P didapatkan, langkah terakhir adalah dengan mentransformasikan matriks P kedalam bentuk huruf.

Widyanarko (2007) memberikan contoh proses dekrpisi dari hasil enkripsi yang telah dilakukan sebelumnya. Diketahui *ciphertext* dari proses enkripsi sebelumnya adalah JHUQIQWLYM, dan kunci yang digunakan adalah $K = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$, langkah awal adalah dengan menemukan *invers* dari matriks kunci. K^{-1} dicari menggunakan persamaan (2.4) dan (2.5) sebagai berikut:

$$K^{-1} = x\ adj(K)\ mod\ 26$$

Nilai $x = Z_{26}$ dicari hingga memenuhi persamaan berikut:

$$\det(A)\,.\,x = 1\ (mod\ 26)$$

misal setelah dicari, ditemukan nilai $x = 5$, maka

$$K^{-1} = 5\begin{bmatrix} 1 & -2 \\ -3 & 1 \end{bmatrix} mod\ 26$$

$$K^{-1} = \begin{bmatrix} 5 & -10 \\ -15 & 5 \end{bmatrix} mod\ 26$$

$$K^{-1} = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix}.$$

Langkah selanjutnya setelah K^{-1} ditemukan adalah melakukan transformasi *ciphertext* ke dalam bentuk angka sebagai berikut:

$$C = \text{JHUQIQWLYM}$$

Maka *plaintext* tersebut ditransformasikan menjadi:

$$P = 9\ 7\ 20\ 16\ 8\ 16\ 22\ 11\ 24\ 12$$

Proses dekripsi blok pertama adalah sebagai berikut:

$$P = K^{-1}C$$

$$P_{1,2} = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix}\begin{bmatrix} 9 \\ 7 \end{bmatrix} = \begin{bmatrix} 157 \\ 134 \end{bmatrix} mod\ 26 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

Angka 1 dan 4 jika dikonversikan ke dalam bentuk huruf menjadi B dan E, sehingga untuk *ciphertext* JH ditemukan *plaintext* BE. Setelah dilakukan dekripsi terhadap keseluruhan blok, didapatkan *plaintext* berupa BESOKMALAM.

2.4. Kriptanalisis

Azhar *et al*. (2017) menjelaskan bahwa kriptanalisis adalah ilmu untuk mengamankan, menganalisis dan membobol keamanan kriptografi. Ada beberapa metode dasar dalam kriptanalisis, menurut Keliher dan Delaney (2013), metode dasar kriptanalisis diantaranya adalah *ciphertext-only,* dan *known-plaintext*. *Ciphertext-only* dapat dilakukan jika seorang penyerang hanya mempunyai *ciphertext* saja, sedangkan *known-plaintext* dapat dilakukan jika seorang penyerang mengetahui potongan *plaintext* dan *ciphertext* yang salih berhubungan.

Algoritma *Hill Cipher* akan relatif mudah diserang menggunakan *known-plaintext,* terutama jika ordo matriks kunci yang digunakan juga diketahui oleh penyerang. *Known-plaintext* pada algoritma *Hill Cipher* dilakukan dengan mengubah/mengonversi *plaintext* dan *ciphertext* yang ada menjadi bentuk matriks, setelah itu melakukan pencarian nilai *variable* kunci yang digunakan. Widyanarko (2007) menjelaskan untuk menemukan matriks kunci, digunakan persamaan (2.9) berikut.

$$C.P^{-1} = K \tag{2.9}$$

Misal didefinisikan *plaintext*

$$P: BESOKMALAM$$

Didefinisikan *ciphertext*

$$C: JHUQIQWLYM$$

Asumsikan bahwa kriptanalis mengetahui potongan *ciphertext* dan *plaintext* yang saling terkait serta modulo yang digunakan adalah 26, berikut *ciphertext* dan *plaintext* yang diketahui oleh kriptanalis adalah $BESO = JHUQ$. Langkah selanjutnya dalah mengubah *ciphertext* dan *plaintext* menjadi bentuk matriks.

$$C = \begin{bmatrix} J & U \\ H & Q \end{bmatrix} = \begin{bmatrix} 9 & 20 \\ 7 & 16 \end{bmatrix}$$

$$P = \begin{bmatrix} B & S \\ E & O \end{bmatrix} = \begin{bmatrix} 1 & 18 \\ 4 & 14 \end{bmatrix}$$

Langkah awal, jika P *invertible*, misal setelah dilakukan perhitungan, ditemukan $P^{-1} = \begin{bmatrix} 15 & 16 \\ 5 & 15 \end{bmatrix}$, maka

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

$$K = \begin{bmatrix} 9 & 20 \\ 7 & 16 \end{bmatrix}\begin{bmatrix} 15 & 16 \\ 5 & 15 \end{bmatrix} = \begin{bmatrix} 235 & 444 \\ 185 & 352 \end{bmatrix} mod\ 26 = \begin{bmatrix} 1 & 2 \\ 3 & 14 \end{bmatrix}$$

Setelah ditemukan matriks K, dilakukan perhitungan matriks K^{-1}, misal ditemukan $K^{-1} = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix}$, maka dapat dilakukan dekripsi keseluruhan *ciphertext* sebagai berikut.

Blok pertama *ciphertext*:

$P_1 = K^{-1}C_1$

$P_1 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} J & U \\ H & Q \end{bmatrix}$

$P_1 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} 9 & 20 \\ 7 & 16 \end{bmatrix} = \begin{bmatrix} 157 & 356 \\ 134 & 300 \end{bmatrix} mod\ 26 = \begin{bmatrix} 1 & 18 \\ 4 & 14 \end{bmatrix} = \begin{bmatrix} B & S \\ E & O \end{bmatrix}$

Blok kedua *ciphertext*:

$P_2 = K^{-1}C_2$

$P_2 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} I & W \\ Q & L \end{bmatrix}$

$P_2 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} 8 & 22 \\ 16 & 11 \end{bmatrix} = \begin{bmatrix} 296 & 286 \\ 168 & 297 \end{bmatrix} mod\ 26 = \begin{bmatrix} 10 & 0 \\ 12 & 11 \end{bmatrix} = \begin{bmatrix} K & A \\ M & L \end{bmatrix}$

Blok kedua *ciphertext*:

$P_3 = K^{-1}C_3$

$P_3 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} Y \\ M \end{bmatrix}$

$$P_2 = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \begin{bmatrix} 24 \\ 12 \end{bmatrix} = \begin{bmatrix} 312 \\ 324 \end{bmatrix} mod\ 26 = \begin{bmatrix} 0 \\ 12 \end{bmatrix} = \begin{bmatrix} A \\ M \end{bmatrix}$$

Jika digabungkan, maka *ciphertext* dapat didekripsi dengan benar, yaitu BESOKMALAM.

2.5. *Database*

Menurut Supartini dan Hindarto (2016), basis data atau *database* adalah kumpulan data yang disimpan secara sistematik dalam komputer. Penggunaan *database* tidak lepas dari istilah *Database Management System* (DBMS), menurut Setyawati (2020) DBMS adalah sistem pemrosesan *database* yang dimaksudkan untuk mengatasi kelemahan-kelemahan yang ada pada sistem pemrosesan berkas secara manual.

Data dalam DBMS akan disimpan dalam bentuk tabel-tabel yang mewakili berbagai data yang disimpan di dalamnya. Setiap tabel memiliki lebih dari satu kolom yang mewakili atribut/properti data. Ada satu kolom pada masing-masing

tabel yang dijadikan acuan pada proses pengolahan data pada tabel tersebut, kolom tersebut bersifat unik untuk setiap *record* data yang disimpan, sehingga dapat merepresentasikan data secara spesifik pada *record* tertentu, kolom tersebut disebut sebagai *Primary Key*.

2.6. *Universally Unique Identifier* (UUID)

Menurut Rizaldi *et al*. (2020), UUID adalah standar identifikasi berupa 32-digit heksadesimal yang dibagi dalam 5 grup dan dipisahkan oleh tanda penghubung. Sesuai dengan namanya, UUID bersifat unik, dan berfungsi sebagai *identifier* data tertentu. Postgresql sebagai salah satu DBMS yang banyak digunakan, menyediakan UUID sebagai salah satu tipe data yang bisa digunakan sebagai *primary key* pada tabel. UUID yang ada pada Postgresql memiliki total 36 karakter, yaitu 32 karakter utama UUID, dan 4 karakter penghubung berupa tanda "-". Contoh susunan karakter UUID adalah "d57ef4b8-069c-4d4b-9393-7e3576dc2b75".

BAB 3. METODE PENELITIAN

Penelitian ini berfokus pada menggali lebih jauh bagaimana proses enkripsi, dekripsi, dan proses pembentukan kunci pada algoritma *Hill Cipher* sehingga dapat dimodifikasi menjadi *Hill Cipher* berantai serta dapat dimanfaatkan untuk meningkatkan keamanan data pada situs Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala. Bab ini akan menjelaskan tentang data apa yang digunakan dalam penelitian, kapan penelitian dilakukan, dan bagaimana tahapan penelitian akan dilakukan

3.1. Data Penelitian

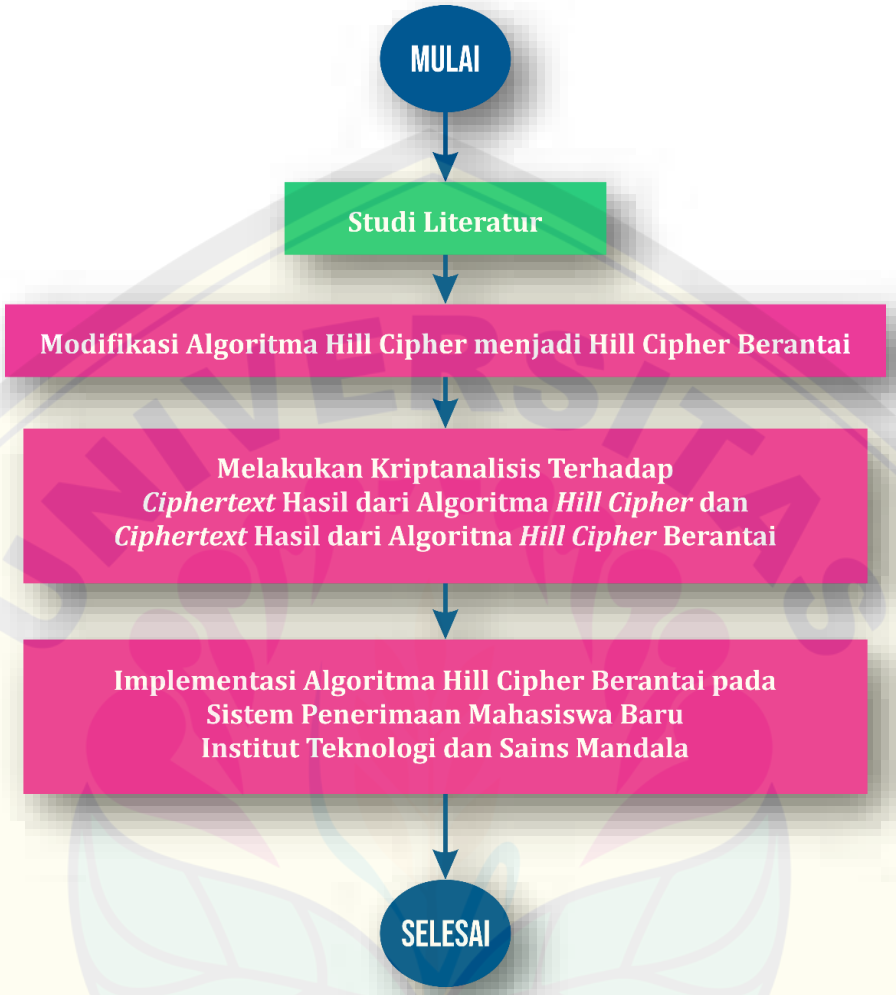
Penelitian ini mengambil studi kasus situs Penerimaan Mahasiswa Baru (PMB) Institut Teknologi dan Sains Mandala. Data yang digunakan sebagai bahan analisis pembentukan kunci berantai akan berfokus pada tabel pendaftar yang ada dalam *database* situs PMB Institut Teknologi dan Sains Mandala pada kolom yang menjadi *primary key* yang memiliki tipe data UUID. Detail karakter penyusun UUID dan konversinya dapat dilihat pada Tabel 3.1. Tabel ini akan penulis jadikan dasar dalam konversi matriks yang dilakukan selama proses penelitian.

Tabel 3. 1 Aturan Konversi

Karakter	Konversi	Karakter	Konversi	Karakter	Konversi
0	0	e	14	s	28
1	1	f	15	t	29
2	2	g	16	u	30
3	3	h	17	v	31
4	4	i	18	w	32
5	5	j	19	x	33
6	6	k	20	y	34
7	7	l	21	z	35
8	8	m	22	-	36
9	9	n	23		
a	10	o	24		
b	11	p	25		
c	12	q	26		
d	13	r	27		

3.2. Tahapan Penelitian

Penelitian ini dilakukan dengan empat tahapan, detail tahapan dalam penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Tahapan penelitian

Berdasarkan Gambar 3.1, tahapan awal penelitian adalah studi literatur. Tahapan studi literatur penulis lakukan dengan membaca dan mempelajari jurnal-jurnal terbaru, buku-buku, maupun sumber informasi lain yang terkait dengan kriptografi dan algoritma *Hill Cipher*. Setelah penulis memahami sepenuhnya algoritma *Hill Cipher*, tahap selanjutnya adalah memodifikasi algoritma *Hill Cipher* menjadi *Hill Cipher* berantai. Modifikasi ini penulis lakukan pada penggunaan kuncinya, jika *Hill Cipher* hanya menggunakan satu kunci untuk keseluruhan *plaintext*, penulis memodifikasinya dengan menggunakan hasil

enkripsi blok *plaintext* pertama sebagai kunci untuk enkripsi blok *plaintext* selanjutnya, sehingga membentuk rantai kunci yang terkait satu sama lain.

Tahap selanjutnya adalah penulis mencoba melakukan kriptanalisis terhadap *ciphertext* hasil dari algoritma *Hill Cipher,* dan *ciphertext* dari algoritma *Hill Cipher* berantai, kemudian membandingkan hasil kriptanalisis terhadap kedua algoritma tersebut. Tahap akhir penelitian ini, peneliti melakukan implementasi algoritma *Hill Cipher* berantai pada sistem penerimaan mahasiswa baru Institut Teknologi dan Sains Mandala. Pada tahap ini penulis mengimplementasikan *Hill Cipher* berantai kedalam bahasa pemrograman PHP dengan *framework* yang digunakan adalah *Laravel Framework*.

BAB 4. HASIL DAN PEMBAHASAN

Bab ini akan dijelaskan bagaimana struktur *database* dan alur kerja pendaftaran mahasiswa baru ITSM yang menjadi studi kasus, proses modifikasi *Hill Cipher* menjadi *Hill Cipher* berantai, dan bagaimana proses kriptanalisis terhadap kedua algoritma serta hasil perbandingannya.

4.1. Struktur *Database* dan Cara Kerja Sistem PMB ITS Mandala

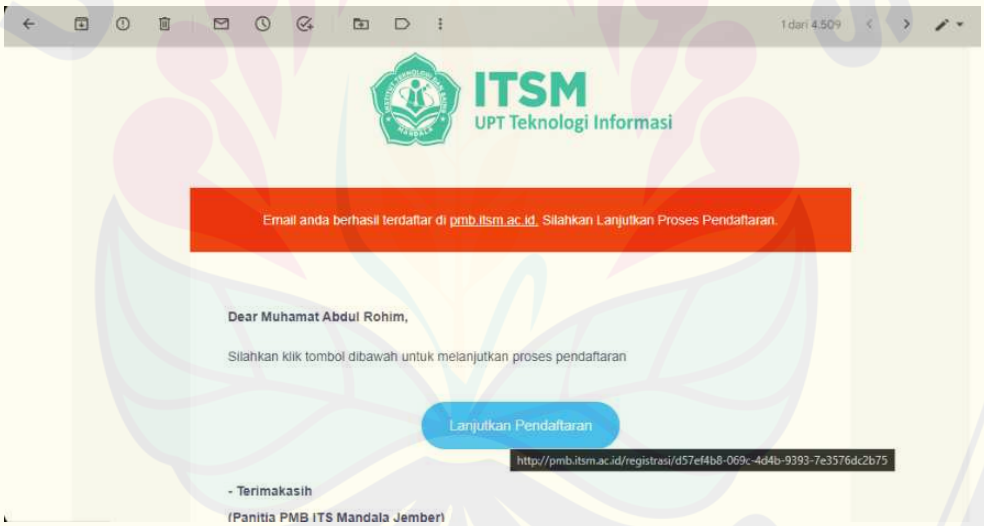
Penelitian diawali dengan melakukan analisis terhadap struktur *database* dari Sistem Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala sebagai objek studi kasus. Hasil dari proses tersebut, peneliti menemukan bahwa cara kerja sistem Penerimaan Mahasiswa Baru Institut Teknologi dan Sains Mandala diawali dengan pengisian formula pendaftaran berupa nama lengkap, email dan nomor *handphone* oleh pendaftar. Formulir pendaftaran dapat dilihat pada Gambar 4.1

Gambar 4. 1 Formulir Pendaftaran Mahasiswa Baru

Identitas pendaftar yang berupa nama lengkap, email dan nomor *handphone* tersebut kemudian disimpan pada tabel registrasi_email. Tabel registrasi_email memiliki kolom *registered_id* sebagai *primary key*, kolom ini memiliki tipe data UUID. Struktur tabel registrasi_email dapat dilihat pada Gambar 4.2.

Gambar 4. 2 Struktur Tabel registrasi_email

Jika proses pendaftaran berhasil, maka tautan yang berisi *registered_id* akan dikirimkan ke email pendaftar untuk melanjutkan proses pendaftaran. Contoh tautan pendaftaran yang dikirimkan ke email dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Email Informasi Tautan Melanjutkan Pendaftaran

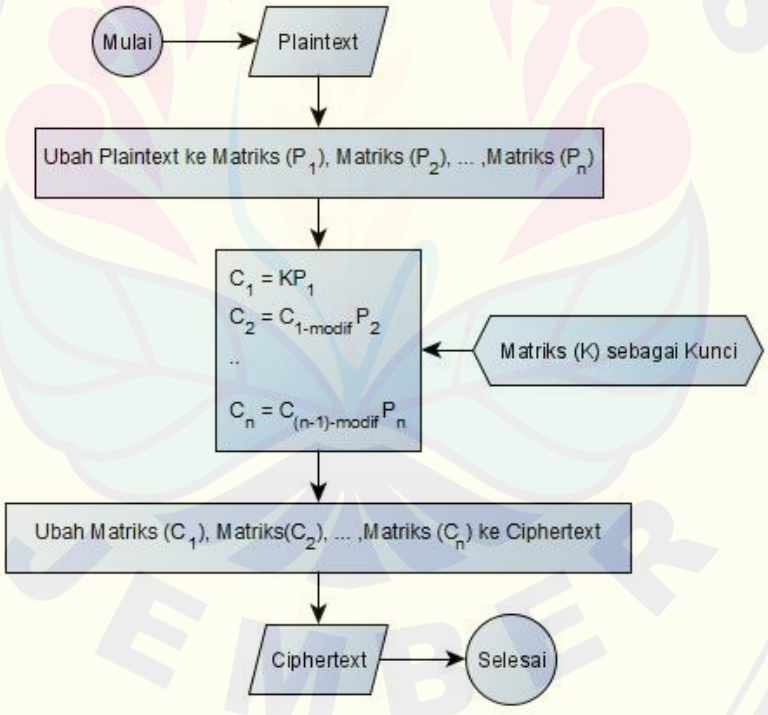
Gambar 4.3 menunjukkan bahwa *registered_id* akan digabungkan pada akhir tautan pendaftaran setelah '/registrasi/', pada contoh tersebut, tautan pendaftaran berupa http://pmb.itsm.ac.id/registrasi/d57ef4b8-069c-4d4b-9393-7e3576dc2b75 yang berarti *registered_id* yang dimiliki pendaftar tersebut adalah "d57ef4b8-069c-4d4b-9393-7e3576dc2b75". *Registered_id* yang ada pada akhir tautan ini yang akan peneliti enkripsi menggunakan algoritma *Hill Cipher* yang sudah peneliti modifikasi menjadi *Hill Cipher* berantai.

4.2. Modifikasi Algoritma *Hill Cipher* menjadi *Hill Cipher* Berantai

Modifikasi Algoritma *Hill Cipher* yang peneliti lakukan adalah pada penggunaan kunci pada proses enkripsi dan dekripsi. Proses modifikasi penulis bagi menjadi dua tahap utama, yaitu modifikasi proses enkripsi dan yang kedua adalah modifikasi proses dekripsi.

4.2.1. Modifikasi proses enkripsi

Proses enkripsi pada algoritma *Hill Cipher* menggunakan satu kunci untuk keseluruhan *plaintext*. Penulis melakukan modifikasi pada kunci yang digunakan, jika pada algoritma *Hill Cipher* hanya digunakan satu kunci, maka penulis memodifikasinya menjadi beberapa kunci yang saling terkait hingga membentuk sebuah rantai kunci yang jumlahnya akan disesuaikan dengan *plaintext* yang akan dienkripsi. Sesuai batasan masalah pada penelitian ini, maka proses modifikasi penulis lakukan berdasarkan kunci dengan ordo 2×2. Hasil modifikasi pada proses enkripsi dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Hasil Modifikasi Proses Enkripsi

Gambar 4.6 menjelaskan bahwa modifikasi yang dilakukan terletak pada proses perkalian antaran matriks P dan matriks K sebagai kunci. Jika sebelumnya matriks C_2 dihasilkan dari perkalian matriks P_2 dan K dan tidak terkait dengan

matriks C_1 maka pada hasil modifikasi ini, matriks C_2 dihasilkan dari perkalian antara matriks $C_{1-modif}$ dan matriks P_2, matriks C_3 dari perkalian matriks $C_{2-modif}$ dan matriks P_3, dan seterusnya hingga membentuk rantai kunci sampai ke matriks C_n dari perkalian matriks $C_{(n-1)-modif}$ dan matriks P_n dengan n adalah panjang blok karakter dari *plaintext*.

Matriks $C_{(n-1)-modif}$ adalah matriks dengan ordo 2×2 yang *invertible* hasil modifikasi dari matriks $C_{(n-1)}$ setelah menghitung determinannya. Berikut proses perhitungan $C_{(n-1)-modif}$ secara umum.

Misal $C_{(n-1)} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Hitung determinan $C_{(n-1)}$:

$$\det(C_{(n-1)}) = ad - bc$$

Jika $\det(C_{(n-1)}) = 0$ atau $C_{(n-1)}$ tidak *invertible*, maka

$$C_{(n-1)-modif} = \begin{bmatrix} a+1 & b \\ c & d \end{bmatrix}$$

Jika $\det(C_{(n-1)}) \neq 0$ atau $C_{(n-1)}$ *invertible*, maka

$$C_{(n-1)-modif} = C_{(n-1)} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Modifikasi $C_{(n-1)}$ perlu dilakukan karena matriks $C_{(n-1)}$ adalah matriks persegi yang belum tentu *invertible*, sedangkan syarat matriks kunci adalah bersifat *invertible*. Sebagai contoh digunakan *plaintext* yang sama dengan yang digunakan pada proses analisis algoritma sebelumnya, yaitu "d57ef4f4-069c-4d4b-9393-7e3576dc2b75", sama seperti algoritma *Hill Cipher,* proses awal adalah dengan memecah *plaintext* dan melakukan konversi seperti pada Tabel 4.1

Tabel 4. 1 Hasil Konversi *Plaintext*

$P_1 = \begin{bmatrix} d & 7 \\ 5 & e \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 7 \\ 5 & 14 \end{bmatrix}$	$P_2 = \begin{bmatrix} f & f \\ 4 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 15 \\ 4 & 4 \end{bmatrix}$
$P_3 = \begin{bmatrix} - & 6 \\ 0 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 36 & 6 \\ 0 & 9 \end{bmatrix}$	$P_4 = \begin{bmatrix} c & 4 \\ - & d \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 4 \\ 36 & 13 \end{bmatrix}$
$P_5 = \begin{bmatrix} 4 & - \\ b & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 36 \\ 11 & 9 \end{bmatrix}$	$P_6 = \begin{bmatrix} 3 & 3 \\ 9 & - \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 3 \\ 9 & 36 \end{bmatrix}$
$P_7 = \begin{bmatrix} 7 & 3 \\ e & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 3 \\ 14 & 5 \end{bmatrix}$	$P_8 = \begin{bmatrix} 7 & d \\ 6 & c \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 13 \\ 6 & 12 \end{bmatrix}$
$P_2 = \begin{bmatrix} 2 & 7 \\ b & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 7 \\ 11 & 5 \end{bmatrix}$	

Sama dengan algoritma *Hill Cipher*, langkah selanjutnya setelah proses konversi adalah mengalikan matriks P dengan matriks K, misal matriks K yang digunakan adalah $\begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}$, maka matriks C_1 adalah sebagai berikut:

$C_1 = KP_1$

$C_1 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}13 & 7\\5 & 14\end{bmatrix} = \begin{bmatrix}371 & 392\\440 & 350\end{bmatrix} mod\ 37 = \begin{bmatrix}1 & 22\\33 & 17\end{bmatrix} = \begin{bmatrix}1 & m\\x & h\end{bmatrix}$

Berdasarkan hasil modifikasi, perhitungan C_2 tidak lagi menggunakan matriks K, namun menggunakan $C_{1-modif}$, berikut perhitungan $C_{1-modif}$:

$C_1 = \begin{bmatrix}1 & 22\\33 & 17\end{bmatrix}$

$\det(C_1) = (1 \times 17) - (22 \times 33) = 17 - 726 = -709$

$\det(C_1) \neq 0$, maka C_1 *invertible*, sehingga

$$C_{1-modif} = C_1 = \begin{bmatrix}1 & 22\\33 & 17\end{bmatrix}$$

Setelah $C_{1-modif}$ ditemukan, matriks C_2 sebagai berikut:

$C_2 = C_{1-modif}P_2$

$C_2 = \begin{bmatrix}1 & 22\\33 & 17\end{bmatrix}\begin{bmatrix}15 & 15\\4 & 4\end{bmatrix} = \begin{bmatrix}103 & 103\\563 & 563\end{bmatrix} mod\ 37 = \begin{bmatrix}29 & 29\\8 & 8\end{bmatrix} = \begin{bmatrix}t & t\\8 & 8\end{bmatrix}$

Sama dengan proses pencarian C_2 yang menggunakan $C_{(n-1)-modif}$ sebagai kunci, C_3 menggunakan $C_{2-modif}$, perhitungan $C_{2-modif}$ sebagai berikut.

$C_2 = \begin{bmatrix}29 & 29\\8 & 8\end{bmatrix}$

$\det(C_2) = (29 \times 8) - (29 \times 8) = 232 - 232 = 0$

$\det(C_2) = 0$, maka C_2 tidak *invertible*, sehingga

$$C_{2-modif} = \begin{bmatrix}a+1 & b\\c & d\end{bmatrix} = \begin{bmatrix}29+1 & 29\\8 & 8\end{bmatrix} = \begin{bmatrix}30 & 29\\8 & 8\end{bmatrix}$$

Setelah $C_{2-modif}$ ditemukan, C_3 sebagai berikut:

$C_3 = C_{2-modif}P_3$

$C_3 = \begin{bmatrix}30 & 29\\8 & 8\end{bmatrix}\begin{bmatrix}36 & 6\\0 & 9\end{bmatrix} = \begin{bmatrix}1080 & 441\\288 & 120\end{bmatrix} mod\ 37 = \begin{bmatrix}7 & 34\\29 & 9\end{bmatrix} = \begin{bmatrix}7 & y\\t & 9\end{bmatrix}$

Perhitungan C_4 menggunakan $C_{3-modif}$, perhitungan $C_{3-modif}$ sebagai berikut.

$C_3 = \begin{bmatrix}7 & 34\\29 & 9\end{bmatrix}$

$\det(C_3) = (7 \times 29) - (34 \times 29) = 203 - 986 = -783$

det $(C_3) \neq 0$, maka C_3 *invertible*, sehingga

$$C_{3-modif} = C_3 = \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix}$$

Setelah $C_{3-modif}$ ditemukan, C_4 sebagai berikut:

$C_4 = C_{3-modif} P_4$

$$C_4 = \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix} \begin{bmatrix} 12 & 4 \\ 36 & 13 \end{bmatrix} = \begin{bmatrix} 1308 & 470 \\ 672 & 233 \end{bmatrix} mod\ 37 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix} = \begin{bmatrix} d & q \\ 6 & b \end{bmatrix}$$

C_5 juga menggunakan $C_{(n-1)-modif}$ yaitu $C_{4-modif}$, perhitungan $C_{4-modif}$ sebagai berikut.

$$C_4 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$$

$\det(C_4) = (13 \times 11) - (26 \times 6) = 143 - 156 = -13$

$\det(C_4) \neq 0$, maka C_4 *invertible*, sehingga

$$C_{4-modif} = C_4 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$$

Setelah $C_{4-modif}$ ditemukan, C_5 sebagai berikut:

$C_5 = C_{4-modif} P_5$

$$C_5 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix} \begin{bmatrix} 4 & 36 \\ 11 & 9 \end{bmatrix} = \begin{bmatrix} 338 & 702 \\ 145 & 315 \end{bmatrix} mod\ 37 = \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix} = \begin{bmatrix} 5 & - \\ y & j \end{bmatrix}$$

Sama dengan proses pencarian C_5 yang menggunakan $C_{(n-1)-modif}$ sebagai kunci, C_6 menggunakan $C_{5-modif}$, perhitungan $C_{5-modif}$ sebagai berikut.

$$C_5 = \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix}$$

$\det(C_5) = (5 \times 19) - (36 \times 34) = 95 - 1224 = -1129$

$\det(C_5) \neq 0$, maka C_5 *invertible*, sehingga

$$C_{5-modif} = C_5 = \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix}$$

Setelah $C_{5-modif}$ ditemukan, C_6 sebagai berikut:

$C_6 = C_{5-modif} P_6$

$$C_6 = \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 9 & 36 \end{bmatrix} = \begin{bmatrix} 339 & 1311 \\ 273 & 786 \end{bmatrix} mod\ 37 = \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix} = \begin{bmatrix} 6 & g \\ e & 9 \end{bmatrix}$$

Sama dengan blok karakter sebelumnya, C_7 juga menggunakan $C_{(n-1)-modif}$ yaitu

$C_{6-modif}$, perhitungan $C_{6-modif}$ sebagai berikut.

$C_6 = \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix}$

$\det(C_6) = (6 \times 9) - (16 \times 14) = 54 - 224 = -170$

$\det(C_6) \neq 0$, maka C_6 *invertible*, sehingga

$$C_{6-modif} = C_6 = \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix}$$

Setelah $C_{6-modif}$ ditemukan, C_7 sebagai berikut:

$C_7 = C_{6-modif} P_7$

$$C_7 = \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix} \begin{bmatrix} 7 & 3 \\ 14 & 5 \end{bmatrix} = \begin{bmatrix} 266 & 98 \\ 224 & 87 \end{bmatrix} mod\ 37 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix} = \begin{bmatrix} 7 & o \\ 2 & d \end{bmatrix}$$

Perhitungan C_8 menggunakan $C_{7-modif}$, perhitungan $C_{7-modif}$ sebagai berikut.

$C_7 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$

$\det(C_7) = (7 \times 13) - (24 \times 2) = 91 - 48 = 43$

$\det(C_7) \neq 0$, maka C_7 *invertible*, sehingga

$$C_{7-modif} = C_7 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$$

Setelah $C_{7-modif}$ ditemukan, C_8 sebagai berikut:

$C_8 = C_{7-modif} P_8$

$$C_8 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix} \begin{bmatrix} 7 & 13 \\ 6 & 12 \end{bmatrix} = \begin{bmatrix} 193 & 379 \\ 92 & 182 \end{bmatrix} mod\ 37 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix} = \begin{bmatrix} 8 & 9 \\ i & y \end{bmatrix}$$

Sama dengan proses pencarian C_8 yang menggunakan $C_{(n-1)-modif}$ sebagai kunci, C_9 menggunakan $C_{8-modif}$, perhitungan $C_{8-modif}$ sebagai berikut.

$C_8 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$

$\det(C_8) = (8 \times 34) - (9 \times 18) = 272 - 162 = 110$

$\det(C_8) \neq 0$, maka C_8 *invertible*, sehingga

$$C_{8-mod} = C_8 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$$

Setelah $C_{8-modif}$ ditemukan, C_9 sebagai berikut:

$C_9 = C_{8-modif} P_9$

$$C_9 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}\begin{bmatrix} 2 & 7 \\ 11 & 5 \end{bmatrix} = \begin{bmatrix} 115 & 101 \\ 410 & 296 \end{bmatrix} mod\ 37 = \begin{bmatrix} 4 & 27 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 4 & r \\ 3 & 0 \end{bmatrix}$$

Berdasarkan perhitungan C_1 sampai dengan C_9, maka didapatkan *ciphertext* "1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0", agar bisa di dekripsi, perlu ditambahkan hasil konversi dari matriks K pada *ciphertext*. Pada penelitian ini, penulis membagi matriks K menjadi dua bagian, yaitu kolom 1 dan kolom 2, kolom 1 penulis letakkan di depan *ciphertext* dan kolom dua penulis letakkan di akhir *ciphertext*, sehingga bentuk *ciphertext* secara umum sebagai berikut:

Jika $K\begin{bmatrix} a & c \\ b & d \end{bmatrix}$, dan *cipher* merupakan hasil perhitungan C_1 sampai dengan C_9 maka:

$$Ciphertext = ab + cipher + cd$$

Dari contoh sebelumnya didapatkan,

$K\begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$ jika dikonversi ke karakter menjadi $\begin{bmatrix} m & h \\ u & a \end{bmatrix}$,

cipher adalah "1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0", maka

$Ciphertext = ab + cipher + cd$

$Ciphertext = "mu" + "1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0" + "ha"$

$Ciphertext = "mu1xmht8t87ty9d6qb5y-\ j6eg972od8i9y43r0ha"$

Rantai kunci yang didapatkan dapat dilihat pada Tabel 4.2.

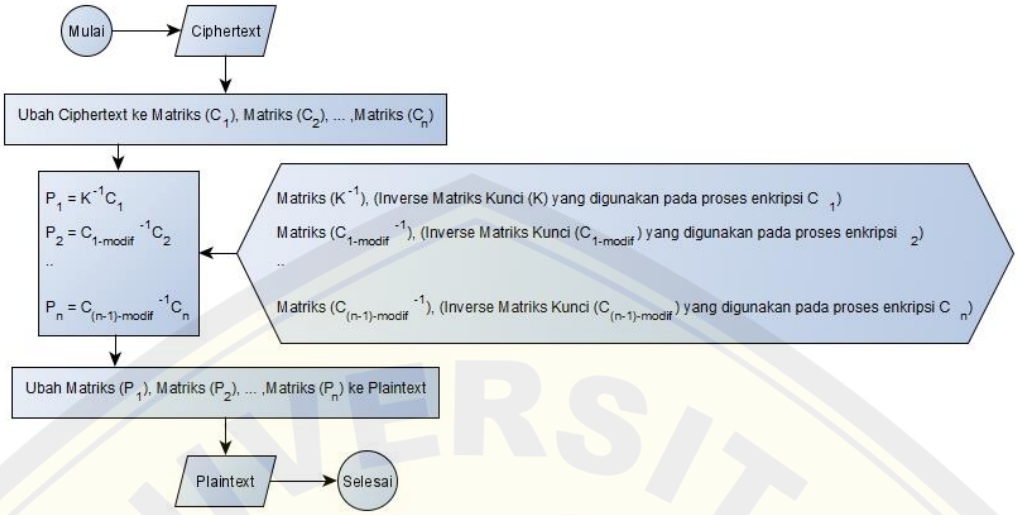
Tabel 4. 2 Hasil Rantai Kunci Enkripsi

$K = \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$	$C_{1-modif} = \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix}$	$C_{2-modif} = \begin{bmatrix} 30 & 29 \\ 8 & 8 \end{bmatrix}$
$C_{3-modif} = \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix}$	$C_{4-modif} = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$	$C_{5-modif} = \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix}$
$C_{6-modif} = \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix}$	$C_{7-modif} = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$	$C_{8-modif} = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$

4.2.2. Modifikasi proses dekripsi

Proses dekripsi pada algoritma *Hill Cipher* menggunakan satu *invers* kunci untuk keseluruhan *plaintext*. Proses enkripsi telah penulis modifikasi pada pembentukan kunci hingga membentuk kunci berantai, hal ini berakibat *invers*

matriks kunci yang digunakan untuk melakukan dekripsi juga berubah sesuai dengan jumlah kunci yang dibentuk pada proses enkripsi. Hasil modifikasi pada proses dekripsi dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Hasil Modifikasi Proses Dekripsi

Pada proses enkripsi, ditambahkan dua karakter di depan dan dua karakter di belakang *ciphertext* asli sebagai kunci utama, maka langkah awal dari proses dekripsi adalah mendapatkan kunci tersebut. Pada proses 4.2.1 didapatkan *ciphertext* "mu1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0ha", maka kunci utama pada *ciphertext* tersebut adalah $\begin{bmatrix} m & h \\ u & a \end{bmatrix}$ jika dikonversi ke angka menjadi $\begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$. Setelah dikurangi kunci utama, maka *ciphertext* menjadi "1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0", *ciphertext* ini yang dipecah dan didekripsi. Hasil pemecahan *ciphertext* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Hasil Konversi *Ciphertext*

$C_1 = \begin{bmatrix} 1 & m \\ x & h \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix}$	$C_2 = \begin{bmatrix} t & t \\ 8 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 29 & 29 \\ 8 & 8 \end{bmatrix}$
$C_3 = \begin{bmatrix} 7 & y \\ t & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix}$	$C_4 = \begin{bmatrix} d & q \\ 6 & b \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$
$C_5 = \begin{bmatrix} 5 & - \\ y & j \end{bmatrix} \rightarrow \begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix}$	$C_6 = \begin{bmatrix} 6 & g \\ e & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix}$
$C_7 = \begin{bmatrix} 7 & o \\ 2 & d \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$	$C_8 = \begin{bmatrix} 8 & 9 \\ i & y \end{bmatrix} \rightarrow \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$
$C_9 = \begin{bmatrix} 4 & r \\ 3 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 27 \\ 3 & 0 \end{bmatrix}$	

Langkah selanjutnya setelah matriks C ditemukan adalah dengan mengalikan matriks C dengan *invers* matriks yang digunakan sebagai kunci pada proses enkripsi. Kunci yang digunakan pada proses enkripsi adalah kunci berantai, sehingga proses dekripsi dilakukan secara terbalik mulai dari P_n sampai dengan P_1, atau pada penelitian ini mulai dari P_9 sampai dengan P_1.

Berdasarkan Gambar 4.5, untuk mencari P_n digunakan *invers* dari kunci $C_{(n-1)-modif}$, yaitu matriks $C^{-1}_{(n-1)-modif}$. Secara umum, $C_{(n-1)-modif}$ didapatkan dengan perhitungan sebagai berikut:

Jika $C_{(n-1)} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$,

hitung determinan $C_{(n-1)}$:

$$\det(C_{(n-1)}) = ad - bc$$

jika $\det(C_{(n-1)}) = 0$ atau $C_{(n-1)}$ tidak *invertible*, maka

$$C_{(n-1)-modif} = \begin{bmatrix} a+1 & b \\ c & d \end{bmatrix}$$

jika $\det(C_{(n-1)}) \neq 0$ atau $C_{(n-1)}$ *invertible*, maka

$$C_{(n-1)-mod} = C_{(n-1)} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Berdasarkan uraian sebelumnya, pada penelitian ini dekripsi dimulai dengan mencari nilai P_9 dengan menggunakan $C^{-1}_{8-modif}$, $C_{8-modif}$ sebagai berikut:

$C_8 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$

$\det(C_8) = (8 \times 34) - (9 \times 18) = 272 - 162 = 110$

$\det(C_8) \neq 0$, maka C_8 *invertible*, sehingga

$$C_{8-modif} = C_8 = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}.$$

Setelah $C_{8-modif}$ ditemukan, untuk menghitung $C^{-1}_{8-modif}$ modulo n dapat dihitung menggunakan persamaan (2.4), sedangkan nilai x diambil dari anggota Z_{37} nilai x memenuhi persamaan berikut:

$$\det(C_{8-modif}) . x = 1 \pmod{37}$$

misal nilai $x = 36$, maka.

$$C_{8-modif}^{-1} = x\ adj(C_{8-mod})\ mod\ n$$

$$C_{8-modif}^{-1} = 36\begin{bmatrix} 34 & -9 \\ -18 & 8 \end{bmatrix}\ mod\ 37$$

$$C_{8-modif}^{-1} = \begin{bmatrix} 1224 & -324 \\ -648 & 288 \end{bmatrix}\ mod\ 37$$

$$C_{8-modif}^{-1} = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}$$

Maka nilai P_9 adalah.

$$P_9 = C_{8-modif}^{-1} C_9$$

$$P_9 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 4 & 27 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 39 & 81 \\ 159 & 486 \end{bmatrix} mod\ 37 = \begin{bmatrix} 2 & 7 \\ 11 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 7 \\ b & 5 \end{bmatrix}$$

Sama dengan perhitungan nilai P_9, P_8 dicari dengan menggunakan $C_{7-modif}^{-1}$, $C_{7-modif}$ sebagai berikut.

$$C_7 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$$

$$\det(C_7) = (7 \times 13) - (24 \times 2) = 91 - 48 = 43$$

$\det(C_7) \neq 0$, maka C_7 *invertible*, sehingga

$$C_{7-modif} = C_7 = \begin{bmatrix} 7 & 24 \\ 2 & 13 \end{bmatrix}$$

Setelah $C_{7-modif}$ ditemukan, sebelum menghitung $C_{7-modif}^{-1}$, terlebih dahulu dihitung nilai $x \in Z_{37}$ hingga memenuhi persamaan.

$$\det(C_{7-modif}).x = 1\ (\text{mod } 37)$$

Jika ditemukan nilai $x = 31$, maka:

$$C_{7-modif}^{-1} = x\ adj(C_{7-modif})\ mod\ n$$

$$C_{7-modif}^{-1} = 31\begin{bmatrix} 13 & -24 \\ -2 & 7 \end{bmatrix}\ mod\ 37$$

$$C_{7-modif}^{-1} = \begin{bmatrix} 403 & -744 \\ -62 & 217 \end{bmatrix}\ mod\ 37$$

$$C_{7-modif}^{-1} = \begin{bmatrix} 33 & 33 \\ 12 & 32 \end{bmatrix}$$

Maka nilai P_8 adalah:

$$P_8 = C_{7-modif}^{-1} C_8$$

$$P_8 = \begin{bmatrix}33 & 33\\12 & 32\end{bmatrix}\begin{bmatrix}8 & 9\\18 & 34\end{bmatrix} = \begin{bmatrix}858 & 1419\\672 & 1196\end{bmatrix} mod\ 37 = \begin{bmatrix}7 & 13\\6 & 12\end{bmatrix} = \begin{bmatrix}7 & d\\6 & c\end{bmatrix}$$

P_7 dicari dengan menggunakan $C_{6-modif}^{-1}$, $C_{6-modif}$ sebagai berikut:

$C_6 = \begin{bmatrix}6 & 16\\14 & 9\end{bmatrix}$

$\det(C_6) = (6 \times 9) - (16 \times 14) = 54 - 224 = -170$

$\det(C_6) \neq 0$, maka C_6 *invertible*, sehingga

$$C_{6-modif} = C_6 = \begin{bmatrix}6 & 16\\14 & 9\end{bmatrix}$$

Setelah $C_{6-modif}$ ditemukan, $C_{6-modif}^{-1}$ modulo n dihitung menggunakan *variable* bantuan $x \in Z_{37}$ hingga x memenuhi persamaan berikut.

$$\det(C_{6-modif}).x = 1 \pmod{37}$$

misal nilai $x = 5$, maka.

$$C_{6-modif}^{-1} = x\ adj(C_{6-modif})\ mod\ n$$

$$C_{6-modif}^{-1} = 5\begin{bmatrix}9 & -16\\-14 & 6\end{bmatrix} mod\ 37$$

$$C_{6-modif}^{-1} = \begin{bmatrix}45 & -80\\-70 & 30\end{bmatrix} mod\ 37$$

$$C_{6-modif}^{-1} = \begin{bmatrix}8 & 31\\4 & 30\end{bmatrix}$$

Maka nilai P_7 adalah:

$P_7 = C_{6-mod}^{-1} C_7$

$$P_7 = \begin{bmatrix}8 & 31\\4 & 30\end{bmatrix}\begin{bmatrix}7 & 24\\2 & 13\end{bmatrix} = \begin{bmatrix}118 & 595\\88 & 486\end{bmatrix} mod\ 37 = \begin{bmatrix}7 & 3\\14 & 5\end{bmatrix} = \begin{bmatrix}7 & 3\\e & 5\end{bmatrix}$$

Pencarian P_6 menggunakan $C_{5-modif}^{-1}$, $C_{5-modif}$ sebagai berikut.

$C_5 = \begin{bmatrix}5 & 36\\34 & 19\end{bmatrix}$

$\det(C_5) = (5 \times 19) - (36 \times 34) = 95 - 1224 = -1129$

$\det(C_5) \neq 0$, maka C_5 *invertible*, sehingga

$$C_{5-mod} = C_5 = \begin{bmatrix}5 & 36\\34 & 19\end{bmatrix}$$

Setelah $C_{5-modif}$ ditemukan, untuk menghitung $C_{5-modif}^{-1}$ modulo n digunakan $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut.

$$\det(C_{5-modif}) . x = 1 \pmod{37}$$

Jika nilai $x = 35$, maka:

$$C_{5-modif}^{-1} = x\ adj(C_{5-modif})\ mod\ n$$

$$C_{5-modif}^{-1} = 35 \begin{bmatrix} 19 & -36 \\ -34 & 5 \end{bmatrix} mod\ 37$$

$$C_{5-modif}^{-1} = \begin{bmatrix} 665 & -1260 \\ -1190 & 175 \end{bmatrix} mod\ 37$$

$$C_{5-modif}^{-1} = \begin{bmatrix} 36 & 35 \\ 31 & 27 \end{bmatrix}$$

Maka nilai P_6 adalah:

$$P_6 = C_{5-modif}^{-1} C_6$$

$$P_6 = \begin{bmatrix} 36 & 35 \\ 31 & 27 \end{bmatrix} \begin{bmatrix} 6 & 16 \\ 14 & 9 \end{bmatrix} = \begin{bmatrix} 706 & 891 \\ 564 & 739 \end{bmatrix} mod\ 37 = \begin{bmatrix} 3 & 3 \\ 9 & 36 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 9 & - \end{bmatrix}$$

P_5 dicari dengan menggunakan $C_{4-modif}^{-1}$, $C_{4-modif}$ sebagai berikut:

$$C_4 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$$

$\det(C_4) = (13 \times 11) - (26 \times 6) = 143 - 156 = -13$

$\det(C_4) \neq 0$, maka C_4 *invertible*, sehingga

$$C_{4-modif} = C_4 = \begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix}$$

Setelah $C_{4-modif}$ ditemukan, $C_{4-modif}^{-1}$ modulo n dihitung dengan $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut.

$$\det(C_{4-modif}) . x = 1 \pmod{37}$$

Jika nilai $x = 17$, maka:

$$C_{4-modif}^{-1} = x\ adj(C_{4-modif})\ mod\ n$$

$$C_{4-modif}^{-1} = 17 \begin{bmatrix} 11 & -26 \\ -6 & 13 \end{bmatrix} mod\ 37$$

$$C_{4-modif}^{-1} = \begin{bmatrix} 187 & -442 \\ -102 & 221 \end{bmatrix} mod\ 37$$

$$C_{4-modif}^{-1} = \begin{bmatrix} 2 & 2 \\ 9 & 36 \end{bmatrix}$$

Maka nilai P_5 adalah:

$$P_5 = C_{4-modif}^{-1} C_5$$

$$P_5 = \begin{bmatrix} 2 & 2 \\ 9 & 36 \end{bmatrix}\begin{bmatrix} 5 & 36 \\ 34 & 19 \end{bmatrix} = \begin{bmatrix} 78 & 110 \\ 1269 & 1008 \end{bmatrix} mod\ 37 = \begin{bmatrix} 4 & 36 \\ 11 & 9 \end{bmatrix} = \begin{bmatrix} 4 & - \\ b & 9 \end{bmatrix}$$

Perhitungan P_4 menggunakan $C_{3-modif}^{-1}$, $C_{3-modif}$ sebagai berikut:

$C_3 = \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix}$

$\det(C_3) = (7 \times 9) - (34 \times 29) = 63 - 986 = -923$

$\det(C_3) \neq 0$, maka C_3 *invertible*, sehingga

$$C_{3-modif} = C_3 = \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix}$$

Setelah $C_{3-modif}$ ditemukan, $C_{3-modif}^{-1}$ modulo n dihitung menggunakan $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut:

$$\det(C_{3-modif}).x = 1 \pmod{37}$$

misal ditemukan nilai $x = 19$, maka:

$$C_{3-modif}^{-1} = x\ adj(C_{3-modif})\ mod\ n$$

$$C_{3-modif}^{-1} = 19\begin{bmatrix} 9 & -34 \\ -29 & 7 \end{bmatrix} mod\ 37$$

$$C_{3-modif}^{-1} = \begin{bmatrix} 171 & -646 \\ -551 & 133 \end{bmatrix} mod\ 37$$

$$C_{3-modif}^{-1} = \begin{bmatrix} 23 & 20 \\ 4 & 22 \end{bmatrix}$$

Maka nilai P_4 adalah:

$P_4 = C_{3-modif}^{-1} C_4$

$$P_4 = \begin{bmatrix} 23 & 20 \\ 4 & 22 \end{bmatrix}\begin{bmatrix} 13 & 26 \\ 6 & 11 \end{bmatrix} = \begin{bmatrix} 419 & 818 \\ 184 & 346 \end{bmatrix} mod\ 37 = \begin{bmatrix} 12 & 4 \\ 36 & 13 \end{bmatrix} = \begin{bmatrix} c & 4 \\ - & d \end{bmatrix}$$

Sama dengan perhitungan nilai P_4, P_3 dicari dengan menggunakan $C_{2-modif}^{-1}$, $C_{2-modif}$ sebagai berikut:

$C_2 = \begin{bmatrix} 29 & 29 \\ 8 & 8 \end{bmatrix}$

$\det(C_2) = (29 \times 8) - (29 \times 8) = 232 - 232 = 0$

$\det(C_2) = 0$, maka C_2 *invertible*, sehingga

$$C_{2-modif} = \begin{bmatrix} a+1 & b \\ c & d \end{bmatrix} = \begin{bmatrix} 29+1 & 29 \\ 8 & 8 \end{bmatrix} = \begin{bmatrix} 30 & 29 \\ 8 & 8 \end{bmatrix}$$

Setelah $C_{2-modif}$ ditemukan, untuk menghitung $C_{2-modif}^{-1}$ modulo n dapat

dihitung menggunakan *variable* $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut.

$$\det(C_{2-modif}) . x = 1 \pmod{37}$$

Jika nilai $x = 14$, maka:

$$C_{2-modif}^{-1} = x\ adj(C_{2-modif})\ mod\ n$$

$$C_{2-modif}^{-1} = 14 \begin{bmatrix} 8 & -29 \\ -8 & 30 \end{bmatrix} mod\ 37$$

$$C_{2-modif}^{-1} = \begin{bmatrix} 112 & -406 \\ -112 & 420 \end{bmatrix} mod\ 37$$

$$C_{2-modif}^{-1} = \begin{bmatrix} 1 & 1 \\ 36 & 13 \end{bmatrix}$$

Maka nilai P_3 adalah:

$$P_3 = C_{2-modif}^{-1} C_3$$

$$P_3 = \begin{bmatrix} 1 & 1 \\ 36 & 13 \end{bmatrix} \begin{bmatrix} 7 & 34 \\ 29 & 9 \end{bmatrix} = \begin{bmatrix} 36 & 43 \\ 629 & 1341 \end{bmatrix} mod\ 37 = \begin{bmatrix} 36 & 6 \\ 0 & 9 \end{bmatrix} = \begin{bmatrix} - & 6 \\ 0 & 9 \end{bmatrix}$$

P_2 dicari dengan menggunakan $C_{1-modif}^{-1}$, $C_{1-modif}$ sebagai berikut:

$$C_1 = \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix}$$

$$\det(C_1) = (1 \times 17) - (22 \times 33) = 17 - 726 = -709$$

$\det(C_1) \neq 0$, maka C_1 *invertible*, sehingga

$$C_{1-modif} = C_1 = \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix}$$

Setelah $C_{1-modif}$ ditemukan, $C_{1-modif}^{-1}$ modulo n dihitung menggunakan $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut:

$$\det(C_{1-modif}) . x = 1 \pmod{37}$$

Jika nilai $x = 6$, maka:

$$C_{1-modif}^{-1} = x\ adj(C_{1-modif})\ mod\ n$$

$$C_{1-modif}^{-1} = 6 \begin{bmatrix} 17 & -22 \\ -33 & 1 \end{bmatrix} mod\ 37$$

$$C_{1-modif}^{-1} = \begin{bmatrix} 102 & -132 \\ -198 & 6 \end{bmatrix} mod\ 37$$

$$C_{1-modif}^{-1} = \begin{bmatrix} 28 & 16 \\ 24 & 6 \end{bmatrix}$$

Maka nilai P_2 adalah:

$$P_2 = C_{1-modif}^{-1} C_2$$

$$P_2 = \begin{bmatrix} 28 & 16 \\ 24 & 6 \end{bmatrix} \begin{bmatrix} 29 & 29 \\ 8 & 8 \end{bmatrix} = \begin{bmatrix} 940 & 940 \\ 744 & 744 \end{bmatrix} mod\ 37 = \begin{bmatrix} 15 & 15 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} f & f \\ 4 & 4 \end{bmatrix}$$

Berbeda dengan perhitungan nilai P_2 sampai dengan P_9, P_1 dicari dengan menggunakan K^{-1}, Berdasarkan pemecahan kunci di awal dekripsi, didapatkan K sebagai berikut:

$$K = \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$$

Setelah K ditemukan, untuk menghitung K^{-1} modulo n digunakan $x \in Z_{37}$ hingga x memenuhi persamaan berikut:

$$\det(K) \,.\, x = 1 \pmod{37}$$

Jika nilai $x = 31$, maka:

$$K^{-1} = x\ adj(K)\ mod\ n$$

$$K^{-1} = 31 \begin{bmatrix} 10 & -17 \\ -30 & 22 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 310 & -527 \\ -930 & 682 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix}$$

Maka nilai P_1 adalah:

$$P_1 = K^{-1} C_1$$

$$P_1 = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix} \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix} = \begin{bmatrix} 938 & 784 \\ 560 & 976 \end{bmatrix} mod\ 37 = \begin{bmatrix} 13 & 7 \\ 5 & 14 \end{bmatrix} = \begin{bmatrix} d & 7 \\ 5 & e \end{bmatrix}$$

Berdasarkan perhitungan P_1 sampai dengan P_9, maka didapatkan *plaintext* "d57ef4f4-069c-4d4b-9393-7e3576dc2b75". Kumpulan *invers* yang digunakan untuk melakukan dekripsi dapat dilihat pada Tabel 4.4.

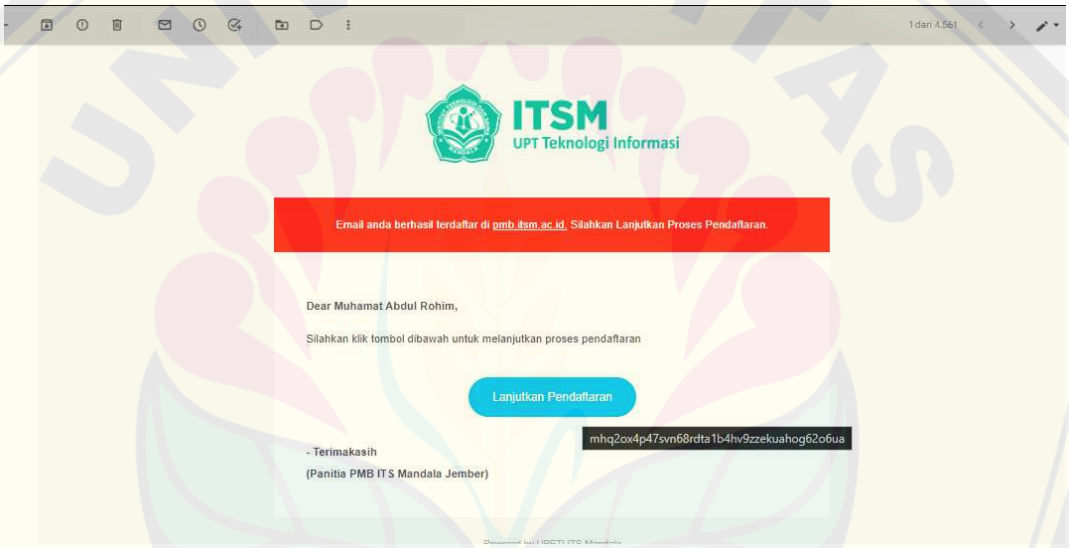
Tabel 4. 4 Kumpulan *Invers* Matriks Kunci

$K^{-1} = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix}$	$C_{1-modif}^{-1} = \begin{bmatrix} 28 & 16 \\ 24 & 6 \end{bmatrix}$	$C_{2-modif}^{-1} = \begin{bmatrix} 1 & 1 \\ 36 & 13 \end{bmatrix}$
$C_{3-modif}^{-1} = \begin{bmatrix} 23 & 20 \\ 4 & 22 \end{bmatrix}$	$C_{4-modif}^{-1} = \begin{bmatrix} 2 & 2 \\ 9 & 36 \end{bmatrix}$	$C_{5-modif}^{-1} = \begin{bmatrix} 36 & 35 \\ 31 & 27 \end{bmatrix}$
$C_{6-modif}^{-1} = \begin{bmatrix} 8 & 31 \\ 4 & 30 \end{bmatrix}$	$C_{7-modif}^{-1} = \begin{bmatrix} 33 & 33 \\ 12 & 32 \end{bmatrix}$	$C_{8-modif}^{-1} = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}$

4.3. Implementasi *Hill Cipher* Berantai pada situs PMB ITS Mandala

Implementasi yang dilakukan penulis adalah dengan menuliskan algoritma *Hill Cipher* berantai ke dalam bahasa pemrograman PHP. Bahasa pemrograman PHP penulis pilih untuk mempermudah pengaplikasiannya pada situs penerimaan mahasiswa bari di Institut Teknologi dan Sains Mandala yang juga menggunakan bahasa pemrograman PHP dengan *Laravel Framework*.

Proses enkripsi penulis implementasikan pada fungsi pengiriman email. Sebelum implementasi, fungsi ini akan mengirim langsung *registered_id* ke email pendaftar tanpa ada proses enkripsi. Setelah implementasi, ditambahkan proses enkripsi di dalamnya, sehingga yang dikirimkan ke email pendaftar adalah *ciphertext*. Tampilan email yang diterima pendaftar setelah proses implementasi dilakukan dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Tampilan Email Setelah Implementasi *Hill Cipher* Berantai

Berdasarkan pembahasan 4.2.1, dalam proses enkripsi dibutuhkan kunci awal yang sudah ditentukan terlebih dahulu. Pada implementasi ini, penulis menggunakan empat karakter pertama pada nama pendaftar sebagai kunci awal proses enkripsi. Sebagai contoh, misal nama pendaftar “muhamat abdul rohim”, maka kunci awal yang terbentuk sebagai berikut:

$$K = \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} m & h \\ u & a \end{bmatrix} = \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$$

Jika nama pendaftar hanya mengandung dua huruf, maka matriks kunci dibentuk dengan cara sebagai berikut:

$$K = \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} a & a+1 \\ b & b+1 \end{bmatrix}$$

Sebagai contoh, misalkan nama pendaftar hanya "mu", maka kunci yang terbentuk adalah:

$$K = \begin{bmatrix} m & \_ \\ u & \_ \end{bmatrix} = \begin{bmatrix} 22 & 22+1 \\ 30 & 30+1 \end{bmatrix} = \begin{bmatrix} 22 & 23 \\ 30 & 31 \end{bmatrix}$$

4.4. Perbandingan hasil Kriptanalisis *Ciphertext*

Pada tahap ini penulis terlebih dahulu melakukan enkripsi *plaintext* yang sama dengan menggunakan algoritma *Hill Cipher* dan *Hill Cipher* berantai. Langkah selanjutnya adalah penulis melakukan kriptanalisis menggunakan metode *known plaintext* terhadap *ciphertext* yang dihasilkan dari kedua algoritma tersebut kemdian membandingkan hasil keduanya.

4.4.1. Enkripsi Algoritma *Hill Cipher*

Misal *plaintext* yang akan di enkripsi sama dengan yang dienkripsi pada pembahasan 4.2.1, yaitu "d57ef4f4-069c-4d4b-9393-7e3576dc2b75", dengan matriks kunci $K \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$. Langkah pertama adalah memecah *plaintext* menjadi blok karakter kemudian konversi ke dalam bentuk matriks dengan aturan pada Tabel 3.1. Blok karakter dan konversi dari *plaintext* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Hasil Konversi *Plaintext* ke Matriks

$P_1 = \begin{bmatrix} d & 7 \\ 5 & e \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 7 \\ 5 & 14 \end{bmatrix}$	$P_2 = \begin{bmatrix} f & f \\ 4 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 15 \\ 4 & 4 \end{bmatrix}$
$P_3 = \begin{bmatrix} - & 6 \\ 0 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 36 & 6 \\ 0 & 9 \end{bmatrix}$	$P_4 = \begin{bmatrix} c & 4 \\ - & d \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 4 \\ 36 & 13 \end{bmatrix}$
$P_5 = \begin{bmatrix} 4 & - \\ b & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 36 \\ 11 & 9 \end{bmatrix}$	$P_6 = \begin{bmatrix} 3 & 3 \\ 9 & - \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 3 \\ 9 & 36 \end{bmatrix}$
$P_7 = \begin{bmatrix} 7 & 3 \\ e & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 3 \\ 14 & 5 \end{bmatrix}$	$P_8 = \begin{bmatrix} 7 & d \\ 6 & c \end{bmatrix} \rightarrow \begin{bmatrix} 7 & 13 \\ 6 & 12 \end{bmatrix}$
$P_2 = \begin{bmatrix} 2 & 7 \\ b & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 7 \\ 11 & 5 \end{bmatrix}$	

Setelah matriks P ditemukan, matriks C sebagai berikut.

Blok pertama

$C_1 = KP_1$

$$C_1 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}13 & 7\\5 & 14\end{bmatrix} = \begin{bmatrix}371 & 392\\440 & 350\end{bmatrix} mod\ 37 = \begin{bmatrix}1 & 22\\33 & 17\end{bmatrix} \rightarrow \begin{bmatrix}1 & m\\x & h\end{bmatrix}$$

Blok kedua

$C_2 = KP_2$

$$C_2 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}15 & 15\\4 & 4\end{bmatrix} = \begin{bmatrix}398 & 398\\490 & 490\end{bmatrix} mod\ 37 = \begin{bmatrix}28 & 28\\9 & 9\end{bmatrix} \rightarrow \begin{bmatrix}s & s\\9 & 9\end{bmatrix}$$

Blok ketiga

$C_3 = KP_3$

$$C_3 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}36 & 6\\0 & 9\end{bmatrix} = \begin{bmatrix}792 & 285\\1080 & 270\end{bmatrix} mod\ 37 = \begin{bmatrix}15 & 26\\7 & 11\end{bmatrix} \rightarrow \begin{bmatrix}f & q\\7 & b\end{bmatrix}$$

Blok keempat

$C_4 = KP_4$

$$C_4 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}12 & 4\\36 & 13\end{bmatrix} = \begin{bmatrix}876 & 309\\720 & 250\end{bmatrix} mod\ 37 = \begin{bmatrix}25 & 13\\17 & 28\end{bmatrix} \rightarrow \begin{bmatrix}p & d\\h & s\end{bmatrix}$$

Blok kelima

$C_5 = KP_5$

$$C_5 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}4 & 36\\11 & 9\end{bmatrix} = \begin{bmatrix}275 & 945\\230 & 1170\end{bmatrix} mod\ 37 = \begin{bmatrix}16 & 20\\8 & 23\end{bmatrix} \rightarrow \begin{bmatrix}g & k\\8 & n\end{bmatrix}$$

Blok keenam

$C_6 = KP_6$

$$C_6 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}3 & 3\\9 & 36\end{bmatrix} = \begin{bmatrix}219 & 678\\180 & 450\end{bmatrix} mod\ 37 = \begin{bmatrix}34 & 12\\32 & 6\end{bmatrix} \rightarrow \begin{bmatrix}y & c\\w & 6\end{bmatrix}$$

Blok ketujuh

$C_7 = KP_7$

$$C_7 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}7 & 3\\4 & 5\end{bmatrix} = \begin{bmatrix}392 & 151\\350 & 140\end{bmatrix} mod\ 37 = \begin{bmatrix}22 & 3\\17 & 29\end{bmatrix} \rightarrow \begin{bmatrix}m & 3\\h & t\end{bmatrix}$$

Blok kedelapan

$C_8 = KP_8$

$$C_8 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}7 & 13\\6 & 12\end{bmatrix} = \begin{bmatrix}256 & 490\\270 & 510\end{bmatrix} mod\ 37 = \begin{bmatrix}34 & 9\\11 & 29\end{bmatrix} \rightarrow \begin{bmatrix}y & 9\\b & t\end{bmatrix}$$

Blok kesembilan

$C_9 = KP_9$

$$C_9 = \begin{bmatrix}22 & 17\\30 & 10\end{bmatrix}\begin{bmatrix}2 & 7\\11 & 5\end{bmatrix} = \begin{bmatrix}231 & 239\\170 & 260\end{bmatrix} mod\ 37 = \begin{bmatrix}9 & 17\\22 & 1\end{bmatrix} \rightarrow \begin{bmatrix}9 & h\\m & 1\end{bmatrix}$$

Berdasarkan perhitungan C_1 sampai C_9, *ciphertext* yang dihasilkan adalah sebagai berikut.

"1xmhs9s9f7qbphdsg8knywc6mh3tyb9t9mh1"

4.4.2. Kriptanalisis *Ciphertext Hill Cipher*

Sebelum menggunakan metode *known plaintext*, seorang kriptanalis harus mengetahui panjang kunci, potongan *plaintext* yang benar dari *ciphertext,* serta modulo yang digunakan. Berdasarkan pembahasan 4.4.1, diasumsikan kriptanalis mengetahui panjang kunci yang digunakan adalah 2×2, modulo yang digunakan adalah modulo 37, dan potongan *plaintext* dan *ciphertext* yang saling terkait sebagai berikut.

Ciphertext : "9mh1"

Plaintext : "2b75"

Langkah selanjutnya adalah merubah *plaintext* dan *ciphertext* menjadi bentuk matriks.

$$P = \begin{bmatrix} 2 & 7 \\ b & 5 \end{bmatrix} = \begin{bmatrix} 2 & 7 \\ 11 & 5 \end{bmatrix}$$

$$C = \begin{bmatrix} 9 & h \\ m & 1 \end{bmatrix} = \begin{bmatrix} 9 & 17 \\ 22 & 1 \end{bmatrix}$$

Berdasarkan persamaan (2.9), matriks kunci K sebagai berikut:

$$K = CP^{-1}$$

P^{-1} modulo n dapat dihitung menggunakan $x \in Z_{37}$ hingga x memenuhi persamaan berikut:

$$\det(P) \,.\, x = 1 \pmod{37}$$

Misal ditemukan nilai $x = 16$, maka:

$$P^{-1} = x\ adj(P)\ mod\ n$$

$$P^{-1} = 16 \begin{bmatrix} 5 & -7 \\ -11 & 2 \end{bmatrix} mod\ 37$$

$$P^{-1} = \begin{bmatrix} 80 & -112 \\ -176 & 32 \end{bmatrix} mod\ 37$$

$$P^{-1} = \begin{bmatrix} 6 & 36 \\ 9 & 32 \end{bmatrix}$$

Maka nilai K adalah

$$K = CP^{-1}$$

$$K = \begin{bmatrix} 9 & 17 \\ 22 & 1 \end{bmatrix}\begin{bmatrix} 6 & 36 \\ 9 & 32 \end{bmatrix} = \begin{bmatrix} 207 & 868 \\ 141 & 824 \end{bmatrix} mod\ 37 = \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$$

Ditemukan matriks kunci:

$$K = \begin{bmatrix} 22 & 17 \\ 30 & 10 \end{bmatrix}$$

Mencari K^{-1} dengan $x \in Z_{37}$ hingga x memenuhi persamaan berikut:

$$\det(K) \,.\, x = 1 \pmod{37}$$

Jika ditemukan nilai $x = 31$, maka:

$$K^{-1} = x\ adj(K)\ mod\ n$$

$$K^{-1} = 31\begin{bmatrix} 10 & -17 \\ -30 & 22 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 310 & -527 \\ -930 & 682 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix}$$

Setelah ditemukan K^{-1}, langkah selanjutnya adalah mendekripsi keseluruhan *ciphertext* menggunakan K^{-1}. Berikutt proses dekripsi keseluruhan *ciphertext*.

Blok pertama:

$$C_1 = \begin{bmatrix} 1 & m \\ x & h \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix}$$

$$P_1 = K^{-1}C_1$$

$$P_1 = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix}\begin{bmatrix} 1 & 22 \\ 33 & 17 \end{bmatrix} = \begin{bmatrix} 938 & 784 \\ 560 & 976 \end{bmatrix} mod\ 37 = \begin{bmatrix} 13 & 7 \\ 5 & 14 \end{bmatrix} = \begin{bmatrix} d & 7 \\ 5 & e \end{bmatrix}$$

Blok kedua:

$$C_2 = \begin{bmatrix} s & s \\ 9 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 28 & 28 \\ 9 & 9 \end{bmatrix}$$

$$P_2 = K^{-1}C_2$$

$$P_2 = \begin{bmatrix} 14 & 28 \\ 32 & 16 \end{bmatrix}\begin{bmatrix} 28 & 28 \\ 9 & 9 \end{bmatrix} = \begin{bmatrix} 644 & 644 \\ 1040 & 1040 \end{bmatrix} mod\ 37 = \begin{bmatrix} 15 & 15 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} f & f \\ 4 & 4 \end{bmatrix}$$

Blok ketiga:

$$C_3 = \begin{bmatrix} f & q \\ 7 & b \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 26 \\ 7 & 11 \end{bmatrix}$$

$P_3 = K^{-1}C_3$

$$P_3 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}15 & 26\\7 & 11\end{bmatrix} = \begin{bmatrix}406 & 672\\592 & 1008\end{bmatrix} mod\ 37 = \begin{bmatrix}36 & 6\\0 & 9\end{bmatrix} = \begin{bmatrix}- & 6\\0 & 9\end{bmatrix}$$

Blok keempat:

$$C_4 = \begin{bmatrix}p & d\\h & s\end{bmatrix} \rightarrow \begin{bmatrix}25 & 13\\17 & 28\end{bmatrix}$$

$P_4 = K^{-1}C_4$

$$P_4 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}25 & 13\\17 & 28\end{bmatrix} = \begin{bmatrix}826 & 966\\1072 & 864\end{bmatrix} mod\ 37 = \begin{bmatrix}12 & 4\\36 & 13\end{bmatrix} = \begin{bmatrix}c & 4\\- & d\end{bmatrix}$$

Blok kelima:

$$C_5 = \begin{bmatrix}g & k\\8 & n\end{bmatrix} \rightarrow \begin{bmatrix}16 & 20\\8 & 23\end{bmatrix}$$

$P_5 = K^{-1}C_5$

$$P_5 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}16 & 20\\8 & 23\end{bmatrix} = \begin{bmatrix}448 & 924\\640 & 1008\end{bmatrix} mod\ 37 = \begin{bmatrix}4 & 36\\11 & 9\end{bmatrix} = \begin{bmatrix}4 & -\\b & 9\end{bmatrix}$$

Blok keenam:

$$C_6 = \begin{bmatrix}y & c\\w & 6\end{bmatrix} \rightarrow \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}$$

$P_6 = K^{-1}C_6$

$$P_6 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}14 & 28\\32 & 16\end{bmatrix} = \begin{bmatrix}1372 & 336\\1600 & 480\end{bmatrix} mod\ 37 = \begin{bmatrix}3 & 3\\9 & 36\end{bmatrix} = \begin{bmatrix}3 & 3\\9 & -\end{bmatrix}$$

Blok ketujuh:

$$C_7 = \begin{bmatrix}m & 3\\h & t\end{bmatrix} \rightarrow \begin{bmatrix}22 & 3\\17 & 29\end{bmatrix}$$

$P_7 = K^{-1}C_7$

$$P_7 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}22 & 3\\17 & 29\end{bmatrix} = \begin{bmatrix}784 & 854\\976 & 560\end{bmatrix} mod\ 37 = \begin{bmatrix}7 & 3\\14 & 5\end{bmatrix} = \begin{bmatrix}7 & 3\\e & 5\end{bmatrix}$$

Blok kedelapan:

$$C_8 = \begin{bmatrix}y & 9\\b & t\end{bmatrix} \rightarrow \begin{bmatrix}34 & 9\\11 & 29\end{bmatrix}$$

$P_8 = K^{-1}C_8$

$$P_8 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}34 & 9\\11 & 29\end{bmatrix} = \begin{bmatrix}784 & 938\\1264 & 752\end{bmatrix} mod\ 37 = \begin{bmatrix}7 & 13\\6 & 12\end{bmatrix} = \begin{bmatrix}7 & d\\6 & c\end{bmatrix}$$

Blok kesembilan:

$$C_9 = \begin{bmatrix}9 & h\\m & 1\end{bmatrix} \rightarrow \begin{bmatrix}9 & 17\\22 & 1\end{bmatrix}$$

$P_9 = K^{-1}C_9$

$$P_9 = \begin{bmatrix}14 & 28\\32 & 16\end{bmatrix}\begin{bmatrix}9 & 17\\22 & 1\end{bmatrix} = \begin{bmatrix}742 & 266\\640 & 560\end{bmatrix} mod\ 37 = \begin{bmatrix}2 & 7\\11 & 5\end{bmatrix} = \begin{bmatrix}2 & 7\\b & 5\end{bmatrix}$$

Setelah ditemukan P_1 sampai dengan P_3, ditemukan *plaintext* yang benar yaitu "d57ef4f4-069c-4d4b-9393-7e3576dc2b75".

4.4.3. Kriptanalisis *Ciphertext Hill Cipher* berantai

Sama dengan pada pembahasan 4.4.2, diasumsikan bahwa kriptanalis mengetahui panjang kunci 2×2, modulo 37 dan potongan *plaintext* yang saling terkait dengan *ciphertext*. *Ciphertext* yang digunakan pada proses kriptanalisis ini adalah *ciphertext* hasil enkripsi pada pembahasan 4.2.1, yaitu "mu1xmht8t87ty9d6qb5y-j6eg972od8i9y43r0ha". Misal *plaintext* dan *ciphertext* yang diketahui kritanalis adalah sebagai berikut.

Ciphertext : "43r0"

Plaintext : "2b75"

Langkah selanjutnya adalah mengubah *plaintext* dan *ciphertext* menjadi bentuk matriks.

$$P = \begin{bmatrix}2 & 7\\b & 5\end{bmatrix} = \begin{bmatrix}2 & 7\\11 & 5\end{bmatrix}$$

$$C = \begin{bmatrix}4 & r\\3 & 0\end{bmatrix} = \begin{bmatrix}4 & 27\\3 & 0\end{bmatrix}$$

Berdasarkan persamaan (2.9), matriks kunci K sebagai berikut:

$K = CP^{-1}$

Misal ditemukan $P^{-1} = \begin{bmatrix}6 & 36\\9 & 32\end{bmatrix}$, maka nilai K sebagai berikut.

$K = CP^{-1}$

$$K = \begin{bmatrix}4 & 27\\3 & 0\end{bmatrix}\begin{bmatrix}6 & 36\\9 & 32\end{bmatrix} = \begin{bmatrix}267 & 1008\\18 & 108\end{bmatrix} mod\ 37 = \begin{bmatrix}8 & 9\\18 & 34\end{bmatrix}$$

Ditemukan matriks kunci:

$$K = \begin{bmatrix} 8 & 9 \\ 18 & 34 \end{bmatrix}$$

Mencari K^{-1} dengan $x \in Z_{37}$ hingga nilai x memenuhi persamaan berikut.

$$\det(K) \,.\, x = 1 \pmod{37}$$

Misal ditemukan nilai $x = 36$, maka:

$$K^{-1} = x\, adj(K)\, mod\, n$$

$$K^{-1} = 36 \begin{bmatrix} 34 & -9 \\ -18 & 8 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 1224 & -324 \\ -648 & 288 \end{bmatrix} mod\ 37$$

$$K^{-1} = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}$$

Setelah K^{-1}, langkah selanjutnya adalah melakukan dekripsi keseluruhan *ciphertext*. Berikut proses dekripsinya.

Blok pertama:

$$C_1 = \begin{bmatrix} m & 1 \\ u & x \end{bmatrix} \rightarrow \begin{bmatrix} 22 & 1 \\ 30 & 33 \end{bmatrix}$$

$$P_1 = K^{-1} C_1$$

$$P_1 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix} \begin{bmatrix} 22 & 1 \\ 30 & 33 \end{bmatrix} = \begin{bmatrix} 336 & 300 \\ 1266 & 975 \end{bmatrix} mod\ 37 = \begin{bmatrix} 3 & 4 \\ 8 & 15 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 8 & d \end{bmatrix}$$

Blok kedua:

$$C_2 = \begin{bmatrix} m & t \\ h & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 22 & 29 \\ 17 & 8 \end{bmatrix}$$

$$P_2 = K^{-1} C_2$$

$$P_2 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix} \begin{bmatrix} 22 & 29 \\ 17 & 8 \end{bmatrix} = \begin{bmatrix} 219 & 159 \\ 889 & 754 \end{bmatrix} mod\ 37 = \begin{bmatrix} 34 & 11 \\ 1 & 14 \end{bmatrix} = \begin{bmatrix} y & b \\ 1 & e \end{bmatrix}$$

Blok ketiga:

$$C_3 = \begin{bmatrix} t & 7 \\ 8 & t \end{bmatrix} \rightarrow \begin{bmatrix} 29 & 7 \\ 8 & 29 \end{bmatrix}$$

$$P_3 = K^{-1} C_3$$

$$P_3 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix} \begin{bmatrix} 29 & 7 \\ 8 & 29 \end{bmatrix} = \begin{bmatrix} 159 & 282 \\ 754 & 967 \end{bmatrix} mod\ 37 = \begin{bmatrix} 11 & 23 \\ 14 & 5 \end{bmatrix} = \begin{bmatrix} b & n \\ e & 5 \end{bmatrix}$$

Blok keempat:

$$C_4 = \begin{bmatrix} y & d \\ 9 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 34 & 13 \\ 9 & 6 \end{bmatrix}$$

$P_4 = K^{-1}C_4$

$$P_4 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 34 & 13 \\ 9 & 6 \end{bmatrix} = \begin{bmatrix} 183 & 93 \\ 873 & 408 \end{bmatrix} mod\ 37 = \begin{bmatrix} 35 & 19 \\ 22 & 1 \end{bmatrix} = \begin{bmatrix} z & j \\ m & 1 \end{bmatrix}$$

Blok kelima:

$$C_5 = \begin{bmatrix} q & 5 \\ b & y \end{bmatrix} \rightarrow \begin{bmatrix} 26 & 5 \\ 11 & 34 \end{bmatrix}$$

$P_5 = K^{-1}C_5$

$$P_5 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 26 & 5 \\ 11 & 34 \end{bmatrix} = \begin{bmatrix} 177 & 321 \\ 787 & 1076 \end{bmatrix} mod\ 37 = \begin{bmatrix} 29 & 25 \\ 10 & 3 \end{bmatrix} = \begin{bmatrix} t & p \\ a & 3 \end{bmatrix}$$

Blok keenam:

$$C_6 = \begin{bmatrix} - & 6 \\ j & e \end{bmatrix} \rightarrow \begin{bmatrix} 36 & 6 \\ 19 & 14 \end{bmatrix}$$

$P_6 = K^{-1}C_6$

$$P_6 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 36 & 6 \\ 19 & 14 \end{bmatrix} = \begin{bmatrix} 279 & 144 \\ 1199 & 514 \end{bmatrix} mod\ 37 = \begin{bmatrix} 20 & 33 \\ 15 & 33 \end{bmatrix} = \begin{bmatrix} k & x \\ f & x \end{bmatrix}$$

Blok ketujuh:

$$C_7 = \begin{bmatrix} g & 7 \\ 9 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 16 & 7 \\ 9 & 2 \end{bmatrix}$$

$P_7 = K^{-1}C_7$

$$P_7 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 16 & 7 \\ 9 & 2 \end{bmatrix} = \begin{bmatrix} 129 & 39 \\ 549 & 184 \end{bmatrix} mod\ 37 = \begin{bmatrix} 18 & 2 \\ 31 & 36 \end{bmatrix} = \begin{bmatrix} i & 2 \\ v & - \end{bmatrix}$$

Blok kedelapan:

$$C_8 = \begin{bmatrix} o & 8 \\ d & i \end{bmatrix} \rightarrow \begin{bmatrix} 24 & 8 \\ 13 & 18 \end{bmatrix}$$

$P_8 = K^{-1}C_8$

$$P_8 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 24 & 8 \\ 13 & 18 \end{bmatrix} = \begin{bmatrix} 189 & 186 \\ 809 & 666 \end{bmatrix} mod\ 37 = \begin{bmatrix} 4 & 1 \\ 32 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ w & 0 \end{bmatrix}$$

Blok kesembilan:

$$C_9 = \begin{bmatrix} 9 & 4 \\ y & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 9 & 4 \\ 34 & 3 \end{bmatrix}$$

$P_9 = K^{-1}C_9$

$$P_9 = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 9 & 4 \\ 34 & 3 \end{bmatrix} = \begin{bmatrix} 333 & 39 \\ 1148 & 159 \end{bmatrix} mod\ 37 = \begin{bmatrix} 0 & 2 \\ 1 & 11 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 1 & b \end{bmatrix}$$

Blok kesepuluh:

$$C_{10} = \begin{bmatrix} r & h \\ 0 & a \end{bmatrix} \rightarrow \begin{bmatrix} 27 & 17 \\ 0 & 10 \end{bmatrix}$$

$P_{10} = K^{-1}C_{10}$

$$P_{10} = \begin{bmatrix} 3 & 9 \\ 18 & 29 \end{bmatrix}\begin{bmatrix} 27 & 17 \\ 0 & 10 \end{bmatrix} = \begin{bmatrix} 81 & 141 \\ 486 & 596 \end{bmatrix} mod\ 37 = \begin{bmatrix} 7 & 30 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} 7 & u \\ 5 & 4 \end{bmatrix}$$

Setelah P_1 sampai dengan P_4 ditemukan, didapatkan *plaintext* "384dy1beben5zmj1tap3kfxxiv2-4w10012b75u4", sedangkan *plaintext* yang asli sesuai pembahasan 4.2.1 adalah "d57ef4f4-069c-4d4b-9393-7e3576dc2b75".

4.4.4. Hasil Perbandingan

Kriptanalisis yang dilakukan memberikan hasil yang berbeda pada masing-masing algoritma. Pada algoritma *Hill Cipher*, matriks kunci K berhasil ditemukan, sehingga K^{-1} juga ditemukan dan dapat digunakan untuk mendekripsi semua *ciphertext* hingga mendapatkan *plaintext* yang benar. Pada Algoritma *Hill Cipher* berantai, setelah didapatkan matriks kunci K, K^{-1} dari kunci tersebut tidak dapat digunakan untuk melakukan dekripsi yang benar pada semua blok karakter *ciphertext* yang ada, sehingga hasil dekripsi tidak mencerminkan *plaintext* sama sekali. K^{-1} tidak bisa digunakan karena setiap blok karakter pada algoritma *Hill Cipher* berantai di enkripsi dengan kunci yang berbeda. Berdasarkan kedua hasil tersebut, dapat disimpulkan bahwa *Hill Cipher* berantai lebih sulit dibongkar dibandingkan dengan *Hill Cipher* biasa.

BAB 5. KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran yang diberikan dapat digunakan sebagai acuan dalam penelitian selanjutnya.

5.1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, dapat penulis ambil kesimpulan sebagai berikut.

1. Penggunaan hasil enkripsi blok karakter pertama sebagai kunci enkripsi blok karakter selanjutnya, tidak bisa diimplementasikan secara langsung karena hasil enkripsi blok sebelumnya belum tentu *invertible*. Jika hasil enkripsi blok sebelumnya tidak *invertible* maka perlu dilakukan modifikasi dengan cara menambahkan elemen pertama pada hasil enkripsi dengan angka 1, sehingga matriksnya akan bersifat *invertible*.
2. Proses enkripsi *primary key* pada tabel pendaftar di situs penerimaan mahasiswa baru Institut Teknologi dan Sains Mandala menggunakan modulo 37, hal ini dilakukan karena *primary key* yang dienkripsi menggunakan *type data* UUID yang merupakan kombinasi huruf dan angka ditambah karakter "-". Proses enkripsi dilakukan dengan memecah *primary key* menjadi kelompok karakter dengan masing-masing empat karakter, sehingga terbentuk 9 kelompok karakter dengan 9 kunci yang digunakan. Pada akhir enkripsi, kunci awal ditambahkan ke *ciphertext* dengan aturan kolom pertama kunci awal diletakkan di awal, dan kolom kedua diletakkan di akhir *ciphertext*
3. Proses kriptanalisis pada dua metode tersebut menunjukkan bahwa tingkat keamanan *Hill Cipher* berantai lebih baik dibandingkan dengan *Hill Cipher* biasa. Hal ini ditunjukkan dengan *ciphertext Hill Cipher* biasa dapat di dekripsi secara keseluruhan, sedangkan *Hill Cipher* berantai *plaintext* yang didapatkan tidak sesuai dengan aslinya.

5.2. Saran

Adapun saran yang penulis berikan untuk dijadikan masukan pada penelitian selanjutnya adalah sebagai berikut.

1. Pada penelitian ini digunakan matriks kunci dengan ordo 2×2, penelitian selanjutnya bisa menguji apakah *Hill Cipher* berantai dapat di aplikasikan menggunakan kunci yang miliki ordo lebih tinggi.
2. Perbandingan antara *Hill Cipher* dan *Hill Cipher* berantai pada penelitian ini menggunakan satu metode kriptanalisis, yaitu *known plaintext*, pada penelitian selanjutnya bisa di uji kembali menggunakan metode yang lain.

DAFTAR PUSTAKA

Anton, H., dan Rore, C. 2014. *Elementary Linear Algebra Application 1a*.

APJII. 2022. Profil Internet Indonesi, *Survei Profil Internet Indonesia 2022*.

Azhar, W. Y., Supriyadi, dan Yanitasari, Y. 2017. Kriptanalisis Hill Cipher Terhadap Known Plaintext Attack Menggunakan Metode Determinan Matriks Berbasis Android. *Jurnal SIMETRIS*, *8*, 579–592.

Hasugian, A. H. 2013. Implementasi Algoritma Hill Cipher dalam Penyandian Data. *Pelita Informatika Budi Darma*, *IV*(2), 115–122.

Hidayat, A., dan Alawiyah, T. 2013. Enkripsi dan Dekripsi Teks menggunakan Algoritma Hill Cipher dengan Kunci Matriks Persegi Panjang. *Jurnal Matematika Integratif*, *9*(1), 39–51.

Keliher, L., dan Delaney, A. Z. 2013. Cryptanalysis of the toorani-falahati hill ciphers. *Proceedings - IEEE Symposium on Computers and Communications*, 436–440. https://doi.org/10.1109/ISCC.2013.6754985

KOMINFO. 2016. *Peraturan Menteri Komunikasi dan Informatika Nomor 20 Tahun 2016*. Kementrian Komunikasi Dan Infornatika.

Maryanti, S., Rakhman, A., dan Suroso. 2018. Perancangan Aplikasi Kerahasiaan Pesan dengan Algoritma Hill Cipher. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, 70–74.

Menezes, A., Oorschot, P. van, dan Vanstone, S. 1996. *Handbook of Applied Cryptography*.

Pricilla, C. 2015. Aplikasi Perkalian dan Invers Matriks dalam Kriptografi Hill Cipher. *Aljabar Geometri*, 1–5.

Puspita, N. P., dan Bahtiar, N. 2010. Kriptografi Hill Cipher dengan Menggunakan Operasi Matriks. *Jurnal Matematika*, 2–6.

Rao, C. R., dan Toutenburg, H. 1999. *Linear Models: Least Squares and Alternatives, Second Edition*.

Rizaldi, B., Pambudi, D. S., dan Bariyah, T. 2020. Implementasi Teknologi Bluetooth Low Energy dan Metode Triliterasi untuk Pencarian Indoor. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *18*(2), 57. https://doi.org/10.12962/j24068535.v18i2.a897

Setiawan, H. B., Fatma, dan Najicha, U. 2022. Perlindungan Data Pribadi Warga Negara Indonesia Terkait dengan Kebocoran Data. *Jurnal Kewarganegaraan*, *6*(1), 976–982.

Supartini, W., dan Hindarto. 2016. Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosis Dini Penyakit Tuberkulosis di JawaTimur. *KINETIK*, *1*(3), 147–154.

Tuasikal, A. R., Indra, D., dan Fattah, F. 2020. Analisis Perbandingan Known Plaintext dan Chosen Plaintext pada Metode Hill Chiper. *Buletin Sistem Informasi Dan Teknologi Islam*, *1*(1), 1–5.

Widyanarko, A. 2007. Studi dan Analisis mengenai Hill Cipher, Teknik Kriptanalisis dan Upaya Penanggulangannya. *Jurnal Program Studi Teknik Informatika*, 1–6.

Yuliandaru, A. R. 2015. Teknik Kriptografi Hill Cipher Menggunakan Matriks. *Aljabar Geometri*, 1–6.

LAMPIRAN

Lampiran 1. Baris *Code* Implementasi Hill Cipher Berantai

```php
<?php

namespace App\Models\Metode;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class HillChiperChain extends Model
{
    use HasFactory;

    private function split_text($text)
    {
        return str_split($text, 4);
    }

    private function convert_to_matrix($karakter)
    {
        $karakter = strtolower($karakter);
        if ($karakter === 'a') $karakter = '10';
        elseif ($karakter === 'b') $karakter = '11';
        elseif ($karakter === 'c') $karakter = '12';
        elseif ($karakter === 'd') $karakter = '13';
        elseif ($karakter === 'e') $karakter = '14';
        elseif ($karakter === 'f') $karakter = '15';
        elseif ($karakter === 'g') $karakter = '16';
        elseif ($karakter === 'h') $karakter = '17';
        elseif ($karakter === 'i') $karakter = '18';
        elseif ($karakter === 'j') $karakter = '19';
        elseif ($karakter === 'k') $karakter = '20';
        elseif ($karakter === 'l') $karakter = '21';
        elseif ($karakter === 'm') $karakter = '22';
        elseif ($karakter === 'n') $karakter = '23';
        elseif ($karakter === 'o') $karakter = '24';
        elseif ($karakter === 'p') $karakter = '25';
        elseif ($karakter === 'q') $karakter = '26';
        elseif ($karakter === 'r') $karakter = '27';
        elseif ($karakter === 's') $karakter = '28';
        elseif ($karakter === 't') $karakter = '29';
        elseif ($karakter === 'u') $karakter = '30';
        elseif ($karakter === 'v') $karakter = '31';
        elseif ($karakter === 'w') $karakter = '32';
        elseif ($karakter === 'x') $karakter = '33';
        elseif ($karakter === 'y') $karakter = '34';
        elseif ($karakter === 'z') $karakter = '35';
        elseif ($karakter === '-') $karakter = '36';
        return $karakter;
    }

    private function convert_to_character($matrix_element)
    {
        $matrix_element = $matrix_element % 37;
```

```
        if ($matrix_element < 0) {
            $matrix_element += abs(37);
        }
        if ($matrix_element === 10) $matrix_element = 'a';
        elseif ($matrix_element === 11) $matrix_element = 'b';
        elseif ($matrix_element === 12) $matrix_element = 'c';
        elseif ($matrix_element === 13) $matrix_element = 'd';
        elseif ($matrix_element === 14) $matrix_element = 'e';
        elseif ($matrix_element === 15) $matrix_element = 'f';
        elseif ($matrix_element === 16) $matrix_element = 'g';
        elseif ($matrix_element === 17) $matrix_element = 'h';
        elseif ($matrix_element === 18) $matrix_element = 'i';
        elseif ($matrix_element === 19) $matrix_element = 'j';
        elseif ($matrix_element === 20) $matrix_element = 'k';
        elseif ($matrix_element === 21) $matrix_element = 'l';
        elseif ($matrix_element === 22) $matrix_element = 'm';
        elseif ($matrix_element === 23) $matrix_element = 'n';
        elseif ($matrix_element === 24) $matrix_element = 'o';
        elseif ($matrix_element === 25) $matrix_element = 'p';
        elseif ($matrix_element === 26) $matrix_element = 'q';
        elseif ($matrix_element === 27) $matrix_element = 'r';
        elseif ($matrix_element === 28) $matrix_element = 's';
        elseif ($matrix_element === 29) $matrix_element = 't';
        elseif ($matrix_element === 30) $matrix_element = 'u';
        elseif ($matrix_element === 31) $matrix_element = 'v';
        elseif ($matrix_element === 32) $matrix_element = 'w';
        elseif ($matrix_element === 33) $matrix_element = 'x';
        elseif ($matrix_element === 34) $matrix_element = 'y';
        elseif ($matrix_element === 35) $matrix_element = 'z';
        elseif ($matrix_element === 36) $matrix_element = '-';
        return $matrix_element;
    }

    private function get_ciphertext($k_matrix, $P)
    {
        $ciphertext = "";
        $temp_key = $k_matrix;
        $first = $this->convert_to_character($k_matrix['a']) .
$this->convert_to_character($k_matrix['b']);
        $last = $this->convert_to_character($k_matrix['c']) .
$this->convert_to_character($k_matrix['d']);
        for ($i = 0; $i < sizeof($P); $i++) {
            $P_0 = $this->convert_to_matrix(substr($P[$i], 0, 1));
            $P_1 = $this->convert_to_matrix(substr($P[$i], 1, 1));
            $P_2 = $this->convert_to_matrix(substr($P[$i], 2, 1));
            $P_3 = $this->convert_to_matrix(substr($P[$i], 3, 1));
            $C_0 = ($temp_key['a'] * $P_0) + ($temp_key['b'] *
$P_1);
            $C_1 = ($temp_key['c'] * $P_0) + ($temp_key['d'] *
$P_1);
            $C_2 = ($temp_key['a'] * $P_2) + ($temp_key['b'] *
$P_3);
            $C_3 = ($temp_key['c'] * $P_2) + ($temp_key['d'] *
$P_3);
            $temp_key = ['a' => $C_0 % 37, 'b' => $C_2 % 37, 'c'
=> $C_1 % 37, 'd' => $C_3 % 37];
            if (!$this->is_invertibel($temp_key))
                $temp_key['a'] = $temp_key['a'] + 1;
            $ciphertext = $ciphertext . $this-
```

```
>convert_to_character((int)$C_0) . $this-
>convert_to_character((int)$C_1) . $this-
>convert_to_character((int)$C_2) . $this-
>convert_to_character((int)$C_3);
        }
        return $first . $ciphertext . $last;
    }

    public function encrypt($name, $uuid)
    {
        $name = str_split($name);
        $a = 22;
        $b = 17;
        $c = 30;
        $d = 10;
        if (isset($name[0])) $a = $this-
>convert_to_matrix($name[0]);
        if (isset($name[1])) $c = $this-
>convert_to_matrix($name[1]);
        if (isset($name[2])) $b = $this-
>convert_to_matrix($name[2]);
        if (isset($name[3])) $d = $this-
>convert_to_matrix($name[3]);
        $k_matrix = ['a' => $a, 'c' => $c, 'b' => $b, 'd' => $d];
        $P = $this->split_text($uuid);
        $ciphertext = $this->get_ciphertext($k_matrix, $P);
        return $ciphertext;
    }

    private function get_invers($matrix)
    {
        $det = ($matrix['a'] * $matrix['d']) - ($matrix['b'] *
$matrix['c']);
        $k = -1;
        do {
            $k = $k + 1;
            $x = (($k * 37) + 1) / $det;
        } while (!is_int($x));
        $inv['a'] = ($matrix['d'] * $x) % 37;
        if ($inv['a'] < 0)
            $inv['a'] += 37;
        $inv['b'] = (-$matrix['b'] * $x) % 37;
        if ($inv['b'] < 0)
            $inv['b'] += 37;
        $inv['c'] = (-$matrix['c'] * $x) % 37;
        if ($inv['c'] < 0)
            $inv['c'] += 37;
        $inv['d'] = ($matrix['a'] * $x) % 37;
        if ($inv['d'] < 0)
            $inv['d'] += 37;
        return $inv;
    }

    private function get_plaintext($C, $k_matrix)
    {
        $plaintext = "";
        for ($i = sizeof($C)-1; $i >= 0; $i--) {
            if ($i > 0) {
                $temp_key = ['a' => $this-
```

```
>convert_to_matrix(substr($C[$i - 1], 0, 1)), 'c' => $this-
>convert_to_matrix(substr($C[$i - 1], 1, 1)), 'b' => $this-
>convert_to_matrix(substr($C[$i - 1], 2, 1)), 'd' => $this-
>convert_to_matrix(substr($C[$i - 1], 3, 1))];
                if (!$this->is_invertibel($temp_key))
                    $temp_key = ['a' => $this-
>convert_to_matrix(substr($C[$i - 1], 0, 1)) + 1, 'c' => $this-
>convert_to_matrix(substr($C[$i - 1], 1, 1)), 'b' => $this-
>convert_to_matrix(substr($C[$i - 1], 2, 1)), 'd' => $this-
>convert_to_matrix(substr($C[$i - 1], 3, 1))];
            } else
                $temp_key = $k_matrix;
            $inv = $this->get_invers($temp_key, $i);
            $C_0 = $this->convert_to_matrix(substr($C[$i], 0, 1));
            $C_1 = $this->convert_to_matrix(substr($C[$i], 1, 1));
            $C_2 = $this->convert_to_matrix(substr($C[$i], 2, 1));
            $C_3 = $this->convert_to_matrix(substr($C[$i], 3, 1));
            $P_0 = ($inv['a'] * $C_0) + ($inv['b'] * $C_1);
            $P_1 = ($inv['c'] * $C_0) + ($inv['d'] * $C_1);
            $P_2 = ($inv['a'] * $C_2) + ($inv['b'] * $C_3);
            $P_3 = ($inv['c'] * $C_2) + ($inv['d'] * $C_3);
            $plaintext = $this->convert_to_character((int)$P_0) .
$this->convert_to_character((int)$P_1) . $this-
>convert_to_character((int)$P_2) . $this-
>convert_to_character((int)$P_3) . $plaintext;
        }
        return $plaintext;
    }

    public function decrypt($ciphertext)
    {
        $C = str_split($ciphertext, 2);
        $k_matrix = ['a' => $this->convert_to_matrix(substr($C[0],
0, 1)), 'b' => $this->convert_to_matrix(substr($C[0], 1, 1)), 'c'
=> $this->convert_to_matrix(substr($C[sizeof($C) - 1], 0, 1)), 'd'
=> $this->convert_to_matrix(substr($C[sizeof($C) - 1], 1, 1))];
        unset($C[sizeof($C)-1], $C[0]);
        $C = $this->split_text(join("", $C));
        $plaintext = $this->get_plaintext($C, $k_matrix);
        return $plaintext;
    }

    private function is_invertibel($matrix)
    {
        $det = ($matrix['a'] * $matrix['d']) - ($matrix['b'] *
$matrix['c']);
        if ($det != 0) return true;
        else return false;
    }
}
```

Lampiran 2. Baris *Code* Proses Enkripsi pada Fungsi Pengiriman Email**.**

```
public function store_mail(Request $request)
{
    $request->validate([
        'nama_lengkap' => 'required',
        'email_daftar' => 'required',
        'nomor_hp' => 'required'
    ]);
    try {
        $registrasi = RegistrasiEmail::add_registrasi_email(
            $request->email_daftar, $request->nama_lengkap,
$request->nomor_hp);
        if ($registrasi->is_success) {
            $HillCipher = new HillChiperChain();
            $ciphertext = $HillCipher->encrypt($registrasi-
>nama_lengkap, $registrasi->registered_id);
            Mail::to($request->email_daftar)->send(new
RegistrationMail($registrasi,
                $ciphertext));
            Session::flash('success_message',
                "Email berhasil didaftarkan, silahkan periksa
kotak masuk/spam email anda !");
        } else
            Session::flash('failed_message', $registrasi->result);
    } catch (InvalidEmail $e) {
        Session::flash('failed_message', "Email gagal dikirim,
pastikan alamat email valid");
    }
    return redirect()->back();
}
```

Lampiran 3. Baris *Code* Fungsi Pembaca Identitas Pendaftar

```
public function registrasi($registered_id)
{
    $HillCipher = new HillChiperChain();
    $registered_id = $HillCipher->decrypt($registered_id);
    $registrasi =
RegistrasiEmail::get_registrasi_email_by_id($registered_id);
    if ($registrasi->is_success) {
        $menu = "Daftar";
        $kota = Kotakabupaten::list_kotakabupaten();
        $jalur_masuk = SeleksiMaba::get_seleksi_aktif(false,
'2023');
        return view('front-page.form_pendaftaran', compact('menu',
            'kota', 'jalur_masuk', 'registrasi'));
    } else {
        Session::flash('failed_message', "Tidak Ditemukan ID");
        return redirect('/');
    }
}
```

AUTOBIOGRAFI

Muhamat Abdul Rohim

Lahir di Lumajang, 08 Februari 1995, memperoleh gelar Sarjana Komputer dari Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember tahun 2018. Sejak dinyatakan lulus setelah sidang skripsi bulan Oktober 2018 sampai dengan April 2019, mendapat kesempatan untuk bekerja di PT. Catur Mukti Pratama sebagai *Junior Programmer* untuk mengembangkan sistem pengelolaan Lampu Penerangan Jalan berbasis IoT milik perusahaan. Agustus 2019 bekerja sebagai Staff UPT-TI di Institut Teknologi dan Sains Mandala. Sejak kuliah hingga bekerja banyak menggunakan bahasa pemrograman PHP dan sering bersinggungan dengan proses keamanan sistem yang dikembangkan diantaranya terkait proses enkripsi/dekripsi data yang ada di dalamnya.