

PENERAPAN ALGORITMA *HILL CIPHER* MENGGUNAKAN KODE *ASCII* PADA BASIS DATA KEPEGAWAIAN BADAN PERENCANAAN PEMBANGUNAN DAERAH PROVINSI KALIMANTAN TIMUR

Muhammad Fachrian Noor¹⁾, Wahyuni²⁾ dan Hanifah Ekawati³⁾

¹⁾Teknik Informatika, STMIK Widya Cipta Dharma

¹⁾Jl. M. Yamin, Samarinda, 75123

E-mail : muhammadfachriannoor@gmail.com¹⁾, wahyuni@wicida.ac.id²⁾, hanifah@wicida.ac.id³⁾

ABSTRAK

Kriptografi merupakan salah satu teknik untuk menjaga keamanan data dalam sebuah sistem informasi. *Hill Cipher* adalah salah satu algoritma kriptografi yang menggunakan matriks sebagai kunci untuk melakukan enkripsi dan dekripsi pesan. Metode penelitian yang digunakan adalah eksperimen dengan mengambil data kepegawaian dari basis data tersebut, kemudian melakukan enkripsi menggunakan algoritma *Hill Cipher* menggunakan Kode *ASCII* dengan kunci matriks tertentu. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Hill Cipher* menggunakan Kode *ASCII* pada basis data kepegawaian Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur. Hasil dari penelitian ini menunjukkan bahwa implementasi algoritma *Hill Cipher* dengan menggunakan Kode *ASCII* pada Basis Data Kepegawaian Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur dapat meningkatkan keamanan data secara signifikan. Diharapkan hasil dari penelitian ini dapat menjadi acuan bagi institusi atau organisasi lain yang ingin meningkatkan keamanan data mereka dengan menggunakan algoritma kriptografi.

Kata Kunci : Algoritma, *Hill Cipher*, Kode *ASCII*, Kepegawaian.

1. PENDAHULUAN

Pesatnya perkembangan teknologi informasi menyebabkan data membutuhkan keamanan yang cukup baik. Sekarang, hampir semua orang dapat dengan mudah bertukar informasi dalam hal apapun termasuk diantaranya adalah berbagi pengetahuan untuk mengakses data secara ilegal. Data yang seharusnya bersifat rahasia menjadi rentan untuk dicuri ataupun diakses oleh orang yang tak bertanggung jawab. Beberapa orang tidak baik selalu mencari cara untuk mengakses data tersebut, salah satunya dengan cara mengakses secara langsung pada table database. Dengan adanya kemungkinan akan akses ilegal pada database, keamanan yang lebih baik terhadap *database* menjadi dibutuhkan.

Keamanan merupakan kebutuhan yang harusnya dipenuhi pada data digital di era sekarang, karena berubahnya bentuk dokumen yang tadi menjadi kertas kemudian menjadi berbentuk digital. Kerahasiaan data merupakan aset yang sangat berharga dan harus dijaga agar tidak diketahui pihak yang tidak memiliki kepentingan. Kerahasiaan data sudah tercetus sejak jaman dahulu tepatnya pada jaman romawi kuno dengan metode pergeseran huruf atau karakter dengan dasar nilai tertentu yang bertujuan untuk menyembunyikan pesan atau informasi agar tidak diketahui oleh pihak yang tidak berkepentingan.

Menurut UU Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi Pasal 1 Ayat 1, Data Pribadi adalah data tentang orang perseorangan yang teridentifikasi atau dapat diidentifikasi secara tersendiri atau dikombinasi dengan informasi lainnya baik secara langsung maupun tidak langsung melalui sistem elektronik atau nonelektronik. Data Kepegawaian juga merupakan bagian dari data pribadi yang harus dijaga kerahasiannya. Karena data tersebut memiliki peran penting dalam setiap kepengurusan administrasi, dan sejenisnya. Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur menyimpan data kepegawaian berbentuk digital pada komputer secara *offline* dan tidak mempunyai backup penyimpanan data yang aman. Oleh karena itu, data-data tersebut harus dijaga keamanannya agar tidak digunakan oleh pihak yang tidak bertanggungjawab.

Menjaga kerahasiaan dan keamanan sebuah data dapat menggunakan beberapa cara, seperti menggunakan sistem dengan metode Kriptografi untuk mengenkripsi atau mengubah data menjadi sebuah format yang tidak dapat dimengerti maknanya oleh orang lain. Data hanya dapat diubah kembali dengan cara didekripsi menggunakan metode yang sama agar dapat dimengerti kembali maknanya.

Menurut Magamba, dkk (2013), suatu teknik baru menggunakan Algoritma Enkripsi *Hill Cipher* diusulkan untuk menjamin keamanan bagian data pada basis data (*database*) yang diimplementasikan untuk memperkuat dan melindungi basis data tersebut. *Hill Cipher* merupakan salah satu algoritma kunci simetris yang memanfaatkan sebuah matriks K berukuran $n \times n$ sebagai kunci. Algoritma *Hill Cipher* menggunakan matriks persegi sebagai kunci untuk melakukan enkripsi dan dekripsi. Dasar teori matriks yang digunakan dalam *Hill Cipher* antara lain adalah operasi perkalian dan *invers* pada matriks. Ide awal dari *invers* matriks tergeneralisasi (*generalized inverses of matrix*) adalah menggeneralisasi pengertian *invers* matriks. Metode ini memiliki beberapa keuntungan seperti ketahanan terhadap analisis frekuensi dan *implicit* karena metode ini menggunakan perkalian matriks dan *invers* untuk enkripsi dan dekripsi.

Penelitian dilakukan pada Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur yang berada di Kota

Samarinda. Rahasia instansi adalah segala sesuatu yang berhubungan dengan keterangan mengenai data pribadi seluruh pegawai. Maka dari itu dibutuhkan sebuah sistem informasi dengan keamanan yang memadai untuk menghindari terjadinya kebocoran data pegawai atau bahkan pencurian data pegawai dikarenakan di Tahun 2005 adanya insiden pencurian data (*hacking*) di sever instansi. Berdasar latar belakang di atas maka di lakukan penelitian tentang perancangan dan penerapan aplikasi kriptografi pada basis data kepegawaian menggunakan metode *Hill Cipher* dengan mengambil studi kasus di Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur. Teknik kriptografi ini diciptakan dengan maksud untuk menciptakan *cipher* yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. Untuk semakin meningkatkan keamanan penyandian data maka penulis menggabungkan metode enkripsi algoritma *Hill Cipher* dengan menggunakan Kode *ASCII*. Kode *ASCII* (*American Standard Code for Information Interchange*) merupakan representasi numerik dari suatu karakter yang tidak tercetak.

2. RUANG LINGKUP PENELITIAN

2.1 Cakupan Permasalahan

Berdasarkan latar belakang di atas, didapatkan rumusan masalah dalam penelitian ini yaitu, “Bagaimana menerapkan algoritma *Hill Cipher* menggunakan Kode *ASCII* pada Basis Data Kepegawaian di Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur?”.

2.2 Batasan-Batasan Penelitian

- 1) Sistem ini hanya digunakan untuk pengamanan *database* pada Badan Perencanaan Pembangunan Daerah Provinsi Kalimantan Timur.
- 2) Pembuatan perangkat lunak menggunakan Bahasa pemrograman *php* berbasis *website* dengan menggunakan *Framework Laravel*.
- 3) Algoritma enkripsi yang digunakan adalah *Hill Cipher* menggunakan Kode *ASCII*.
- 4) Kunci yang digunakan dalam enkripsi dan dekripsi berbentuk matriks ordo 3x3.

3. BAHAN DAN METODE

3.1 Enkripsi dan Dekripsi

Menurut Hidayatullah dan Insanudin (2016), Enkripsi adalah sebuah proses menjadikan pesan yang dapat dibaca (*plaintext*) menjadi pesan acak yang tidak dapat dibaca (*ciphertext*). Sedangkan dekripsi merupakan proses kebalikan dari enkripsi dimana proses ini akan mengubah *ciphertext* menjadi *plaintext* dengan menggunakan algoritma ‘pembalikan’ dan key yang sama

Jadi enkripsi dan dekripsi merupakan bagian proses dari kriptografi yang masing masing berfungsi sebagai ‘pengunci’ dan ‘pembuka’ dari suatu *ciphertext*, pada enkripsi dilakukan proses pengacakan teks dari *plaintext* menjadi *ciphertext* menggunakan kunci atau ‘key’, kemudian proses dekripsi mengubah *ciphertext* menjadi *plaintext* dengan membalik proses enkripsi berdasarkan key yang digunakan untuk melakukan enkripsi.

3.2 Algoritma

Algoritma secara umum merupakan metode atau langkah yang direncanakan secara tersusun dan berurutan untuk menyelesaikan atau memecahkan permasalahan dengan sebuah intruksi atau kegiatan. Terdapat beberapa definisi lain dari algoritma yang prinsipnya selaras dengan definisi yang diungkapkan sebelumnya, antara lain:

1. Algoritma menurut Kani (2020) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu.
2. Algoritma menurut Munir dan Lidya (2016) adalah urutan langkah-langkah untuk menyelesaikan suatu persoalan.

3.3 Algoritma *Hill Cipher*

Menurut Hasugian (2013), berdasarkan jenis kunci yang dipakai, kriptografi *Hill Cipher* termasuk ke dalam Algoritma Simetrik (*Symmetric Algorithms*), karena algoritma ini menggunakan suatu kunci yang sama untuk proses enkripsi dan dekripsi pesan, Dalam melakukan proses enkripsi dan dekripsi, algoritma ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan dan menerapkan aritmatika modulo.

Dari pengertian tersebut dapat disimpulkan bahwa *Hill Cipher* adalah termasuk algoritma simetrik, dengan menggunakan suatu kunci yang sama untuk proses enkripsi dan dekripsi. Algoritma *Hill Cipher* memanfaatkan sebuah matriks K berukuran $n \times n$ sebagai kunci. Dasar teori matriks yang digunakan dalam *Hill Cipher* antara lain adalah operasi perkalian dan *invers* pada matriks.

Menurut Hidayat & Alawiyah (2013), proses enkripsi pada *Hill Cipher* dilakukan per blok *plaintext*. Ukuran blok tersebut sama dengan ukuran matriks kunci. Sebelum membagi teks menjadi deretan blok-blok, *plaintext* terlebih dahulu dikonversi sesuai tabel *ASCII*. Secara matematis, proses enkripsi pada *Hill Cipher* adalah: $C = K \cdot P$

C = *Ciphertext*

K = Kunci

P = *Plaintext*

Menurut Hidayat & Alawiyah (2013), proses dekripsi pada *Hill Cipher* pada dasarnya sama dengan proses enkripsinya. Namun matriks kunci harus dibalik (*invers*) terlebih dahulu. Secara matematis, proses dekripsi pada *Hill Cipher* dapat

diturunkan dari persamaan. Persamaan proses deskripsi yaitu: $P = K^{-1} \cdot C$

P = Plaintext

K^{-1} = Invers matriks kunci

C = Ciphertext

3.4 Kode ASCII

Menurut Injosoft (2020), *ASCII (American Standard Code for Information Interchange)* merupakan kode huruf dan simbol yang berjumlah 255 kode. Kode *ASCII* dengan nilai *ANSI (American National Standards Institute) ASCII 0-127* adalah kode untuk memanipulasi teks dan nilai *ANSI-ASCII 128-255* adalah kode untuk memanipulasi gambar/grafik.

3.5 Database

Menurut Lubis (2016), Basis Data (*Database*) adalah suatu sistem penyusunan dan pengelolaan record-record dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap pada sebuah organisasi atau perusahaan, sehingga mampu menyediakan informasi secara optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan.

3.6 Model Prototype

Menurut Rosa dan Shalahuddin (2015), Model *Prototype* digunakan untuk merancang sistem informasi. Model *Prototype* memberikan kesempatan untuk pengembang program dan objek penelitian untuk saling berinteraksi selama proses perancangan sistem.

3.7 Entity Relationship Diagram (ERD)

Menurut Yanto (2016), *ERD* adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. *ERD* juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas.

3.8 Flowchart

Menurut Mardi (2014), bagan alir (*flowchart*) merupakan kumpulan dari notasi diagram simbolik yang menunjukkan aliran data dan urutan operasi dalam sistem. Bagan alir (*flowchart*) merupakan metode teknik analisis yang dipergunakan untuk mendeskripsikan sejumlah aspek dari sistem informasi secara jelas, ringkas, dan logis.

3.9 Black Box Testing

Menurut Ardiansah dan Ahmad (2021), *Black Box Testing* adalah pengujian yang dilakukan spesifikasi fungsional yang terdapat pada aplikasi, dimana setiap *user interface* akan dilakukan pengecekan agar setiap fungsi yang ada sesuai dengan kebutuhan.

3.10 White Box Testing

Menurut Susilo dan Handy (2014), *White Box Testing* adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya white box ini menguji dengan cara melihat *Pure Code* dari suatu aplikasi atau *software* yang diuji tanpa memperdulikan Tampilan atau *UI* dari aplikasi tersebut.

4. RANCANGAN SISTEM

4.1 Tahapan Pengembangan Sistem

Pengembangan sistem enkripsi *Hill Cipher* menggunakan Kode *ASCII* berbasis *website* ini menggunakan metode *prototype*. Dalam metode *prototype* pengembangan sistem terdapat tiga tahapan berdasarkan teori yaitu: Mendengarkan Pelanggan, Membangun atau Memperbaiki *Mock-Up* dan Pelanggan Melihat dan Menguji *Mock-Up*.

Dari tiga tahapan tersebut maka dibuat Analisis Tahapan Penelitian agar lebih memudahkan dan menjadi lebih detail untuk menggunakan metode *prototype*. Dalam Analisis Tahapan Penelitian terdapat empat tahapan untuk membangun Sistem *Prototype* yaitu: Analisis Kebutuhan Sistem, Membuat *Prototype*, Membuat Sistem dan Pengujian Sistem, berikut adalah penjabaran mengenai tahapan – tahapan dalam membangun sistem enkripsi *Hill Cipher* menggunakan Kode *ASCII* berbasis *website*.

4.1.1 Analisis Kebutuhan Sistem

Pada implementasi sistem enkripsi *Hill Cipher* menggunakan Kode *ASCII* untuk mengamankan basis data kepegawaian Bappeda Provinsi Kalimantan Timur, dilakukan analisis kebutuhan sistem untuk memastikan bahwa sistem memenuhi persyaratan yang diperlukan. Berikut adalah analisis kebutuhan yang diidentifikasi:

1. Kebutuhan Keamanan Data

- 1) Sistem harus mampu mengamankan data kepegawaian agar tidak dapat diakses atau dimanipulasi oleh pihak yang tidak berwenang.

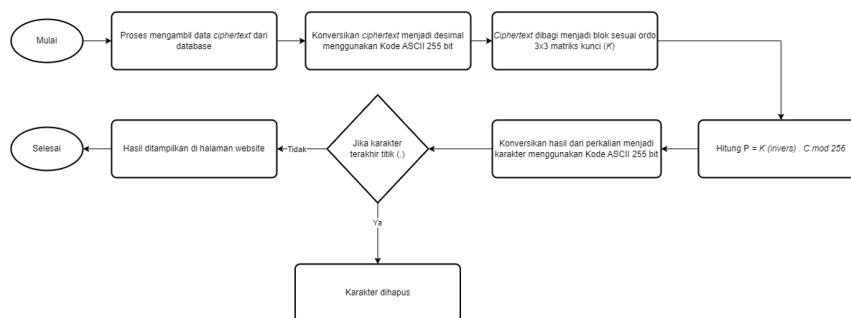
- 2) Enkripsi data kepegawaian dengan menggunakan algoritma *Hill Cipher* dengan Kode *ASCII* akan meningkatkan tingkat keamanan dan kerahasiaan informasi
2. Antarmuka Pengguna yang *User-Friendly*
 - 1) Sistem harus memiliki antarmuka pengguna yang sederhana dan mudah digunakan.
 - 2) Pengguna harus dapat dengan mudah memasukkan data kepegawaian untuk dienkripsi.
3. Pengujian Keamanan
 - 1) Sistem harus melalui pengujian keamanan untuk memastikan kualitas sistem.
 - 2) Pengujian dilakukan dengan menggunakan metode *black box* dan *white box*.
4. Pengembangan dan Perbaikan Sistem
 - 1) Setelah implementasi awal, sistem perlu dikembangkan dan diperbaiki berdasarkan hasil evaluasi dan umpan balik pengguna.
 - 2) Perbaikan dan pengembangan sistem harus dilakukan untuk meningkatkan keamanan dan fungsionalitas sistem enkripsi *Hill Cipher* menggunakan Kode *ASCII*.

4.1.2 Membuat *Prototype*

Setelah menganalisis kebutuhan sistem selanjutnya membuat *Prototype* yang dimana *Prototype* yang dibuat akan digunakan untuk perancangan sistem sesuai dengan hasil dari analisis kebutuhan sistem dan juga kebutuhan pengguna/*client*. Berikut adalah *Prototype* yang akan digunakan untuk perancangan sistem.

4.1.2.1 *Flowchart* Proses Enkripsi

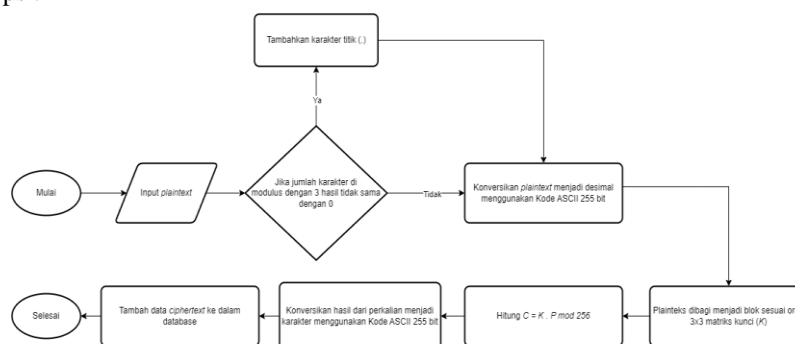
Flowchart yang dibuat ini menjabarkan Langkah-langkah proses enkripsi pada data yang ingin diamankan. Proses dimulai dengan pengguna menginput sebuah *plaintext*, kemudian *plaintext* tersebut dikonversikan menjadi bilangan desimal menggunakan Kode *ASCII* 255 bit, kemudian *plaintext* dibagi menjadi blok yang masing-masing bloknya berukuran 3 karakter karena matriks kunci berukuran 3x3, kemudian proses perhitungan sesuai dengan rumus enkripsi algoritma *Hill Cipher* yaitu $C = K \cdot P$, setelah hasil dari perkalian kunci matriks dengan *plaintext* menjadi *ciphertext* lalu ditambahkan ke dalam database. Berikut adalah gambar 4.1 dari *Flowchart* Proses Enkripsi.



Gambar 4.1 *Flowchart* Proses Enkripsi

4.1.2.2 *Flowchart* Proses Dekripsi

Flowchart yang dibuat ini menjabarkan langkah-langkah proses dekripsi pada data telah terenkripsi. Proses dimulai dengan mengambil data *ciphertext* dari *database*, kemudian *ciphertext* tersebut dikonversikan menjadi bilangan desimal menggunakan Kode *ASCII* 255 bit, kemudian *ciphertext* dibagi menjadi blok yang masing-masing bloknya berukuran 3 karakter karena matriks kunci berukuran 3x3, kemudian proses perhitungan sesuai dengan rumus dekripsi algoritma *Hill Cipher* yaitu $P = K^{-1} \cdot C$ yang dimana kunci matriks *invers* dikali dengan *ciphertext*, setelah hasil dari perkalian kunci matriks *invers* dengan *ciphertext* menjadi *plaintext* maka hasil akan ditampilkan di halaman *website*. Berikut adalah gambar 4.2 dari *Flowchart* Proses Dekripsi.



Gambar 4.2 *Flowchart* Proses Dekripsi

4.1.2.3 Perhitungan Enkripsi

Proses perhitungan manual ini akan menggunakan *plaintext* berdasarkan inputan data dari pengguna dengan nama “**Fatmawati**”. Berikut Proses perhitungan Enkripsi:

Rumus Enkripsi $C = K.P$

Plaintext : **Fatmawati**

Kunci : $\begin{bmatrix} 2 & 2 & 1 \\ 5 & 3 & 3 \\ 3 & 2 & 1 \end{bmatrix}$

Ciphertext : ???

Konversi *plaintext* menjadi bilangan desimal menggunakan Kode *ASCII*.

Tabel 4.1 Konversi *Plaintext* Bilangan Desimal

<i>Plaintext</i>	F	a	t	m	a	w	a	t	i
Hasil Konversi	70	97	116	109	97	119	97	116	105

Karena matriks kunci berukuran 3 x 3, maka *plaintext* dibagi menjadi blok yang masing-masing bloknya berukuran 3 karakter.

$$P_1 = \begin{bmatrix} 70 \\ 97 \\ 116 \end{bmatrix} P_2 = \begin{bmatrix} 109 \\ 97 \\ 119 \end{bmatrix} P_3 = \begin{bmatrix} 97 \\ 116 \\ 105 \end{bmatrix}$$

Berikut perkalian matriks kunci dengan blok berukuran 3 karakter.

$$C_1 = K.P_1 = \begin{bmatrix} 2 & 2 & 1 \\ 5 & 3 & 3 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 70 \\ 97 \\ 116 \end{bmatrix} \text{ mod } 256$$

$$= \begin{bmatrix} 194 \\ 221 \\ 8 \end{bmatrix} \text{ konversi ke ASCII } \begin{bmatrix} \hat{A} \\ \acute{Y} \\ BS \end{bmatrix}$$

Hasil enkripsi dari 3 karakter awal **Fat** adalah **ÂÝBS**.

$$C_2 = K.P_2 = \begin{bmatrix} 2 & 2 & 1 \\ 5 & 3 & 3 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 109 \\ 97 \\ 119 \end{bmatrix} \text{ mod } 256$$

$$= \begin{bmatrix} 19 \\ 169 \\ 128 \end{bmatrix} \text{ konversi ke ASCII } \begin{bmatrix} DC3 \\ © \\ € \end{bmatrix}$$

Hasil enkripsi dari 3 karakter **maw** adalah **DC3©€**.

$$C_3 = K.P_3 = \begin{bmatrix} 2 & 2 & 1 \\ 5 & 3 & 3 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 97 \\ 116 \\ 105 \end{bmatrix} \text{ mod } 256$$

$$= \begin{bmatrix} 19 \\ 124 \\ 116 \end{bmatrix} \text{ konversi ke ASCII } \begin{bmatrix} DC3 \\ | \\ t \end{bmatrix}$$

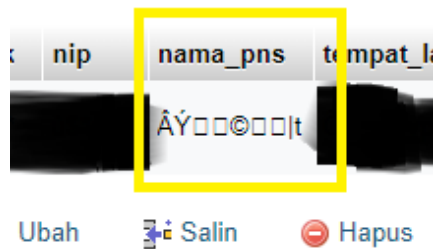
Hasil enkripsi dari 3 karakter akhir **maw** adalah **DC3|t**.

Maka hasil enkripsi *plaintext* **Fatmawati** adalah **ÂÝBSDC3©€DC3|t**.

Tabel 4.2 Hasil Enkripsi

<i>Plaintext</i>	F	a	t	m	a	w	a	t	i
Hasil Enkripsi	Â	Ý	BS	DC3	©	€	DC3	 	t

Berikut hasil enkripsi yang telah di tambahkan ke dalam *database* pada gambar 4.3 Hasil Enkripsi. Karena karakter Kode *ASCII* menjadi perintah komputer jadi beberapa tidak bisa terbaca sebagai karakter.



Gambar 4.3 Hasil Enkripsi

4.1.2.4 Perhitungan Dekripsi

Proses perhitungan manual ini akan menggunakan *ciphertext* berdasarkan data dari *database* dengan data “ÂÝBSDC3©€DC3|t”. Berikut Proses perhitungan Dekripsi:

Rumus Enkripsi $P = K^{-1} \cdot C$

Ciphertext : ÂÝBSDC3©€DC3|t

Kunci Invers : $\begin{bmatrix} 225 & 0 & 1 \\ 172 & 85 & 85 \\ 171 & 86 & 84 \end{bmatrix}$

Plaintext : ???

Konversi *ciphertext* menjadi bilangan desimal menggunakan Kode ASCII.

Tabel 4.3 Konversi *Ciphertext* Bilangan Desimal

<i>Ciphertext</i>	Â	Ý	BS	DC3	©	€	DC3		t
Hasil Konversi	194	221	8	19	169	128	19	124	116

Karena matriks kunci berukuran 3 x 3, maka *ciphertext* dibagi menjadi blok yang masing-masing bloknya berukuran 3 karakter.

$$C_1 = \begin{bmatrix} 194 \\ 221 \\ 8 \end{bmatrix} \quad C_2 = \begin{bmatrix} 19 \\ 169 \\ 128 \end{bmatrix} \quad C_3 = \begin{bmatrix} 19 \\ 124 \\ 116 \end{bmatrix}$$

Berikut perkalian *invers* matriks kunci dengan blok berukuran 3 karakter.

$$P_1 = K^{-1} \cdot C_1 = \begin{bmatrix} 255 & 0 & 1 \\ 172 & 85 & 85 \\ 171 & 86 & 84 \end{bmatrix} \begin{bmatrix} 194 \\ 221 \\ 8 \end{bmatrix} \text{ mod } 256$$

$$= \begin{bmatrix} 70 \\ 97 \\ 116 \end{bmatrix} \text{ konversi ke ASCII } \begin{bmatrix} F \\ a \\ t \end{bmatrix}$$

Hasil dekripsi dari 3 karakter awal ÂÝBS adalah Fat.

$$P_2 = K^{-1} \cdot C_2 = \begin{bmatrix} 255 & 0 & 1 \\ 172 & 85 & 85 \\ 171 & 86 & 84 \end{bmatrix} \begin{bmatrix} 19 \\ 169 \\ 128 \end{bmatrix} \text{ mod } 256$$

$$= \begin{bmatrix} 109 \\ 97 \\ 119 \end{bmatrix} \text{ konversi ke ASCII } \begin{bmatrix} m \\ a \\ w \end{bmatrix}$$

Hasil dekripsi dari 3 karakter DC3©€ adalah maw.

$$P_3 = K^{-1} \cdot C_3 = \begin{bmatrix} 255 & 0 & 1 \\ 172 & 85 & 85 \\ 171 & 86 & 84 \end{bmatrix} \begin{bmatrix} 19 \\ 124 \\ 116 \end{bmatrix} \text{ mod } 256$$

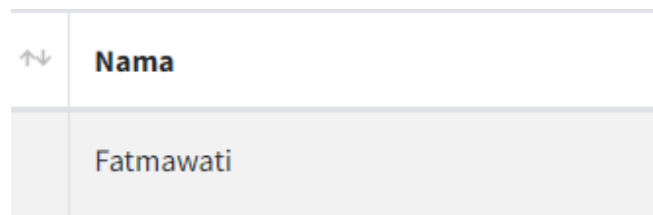
$$= \begin{bmatrix} 97 \\ 116 \\ 105 \end{bmatrix} \text{konversi ke ASCII} \begin{bmatrix} a \\ t \\ i \end{bmatrix}$$

Hasil dekripsi dari 3 karakter akhir **DC3|t** adalah **ati**.
Maka hasil enkripsi *ciphertext* **ÂÝBSDC3©€DC3|t** adalah **Fatmawati**.

Tabel 4.4 Hasil Enkripsi

<i>Ciphertext</i>	Â	Ý	BS	DC3	©	€	DC3		t
Hasil Dekripsi	F	a	t	m	a	w	a	t	i

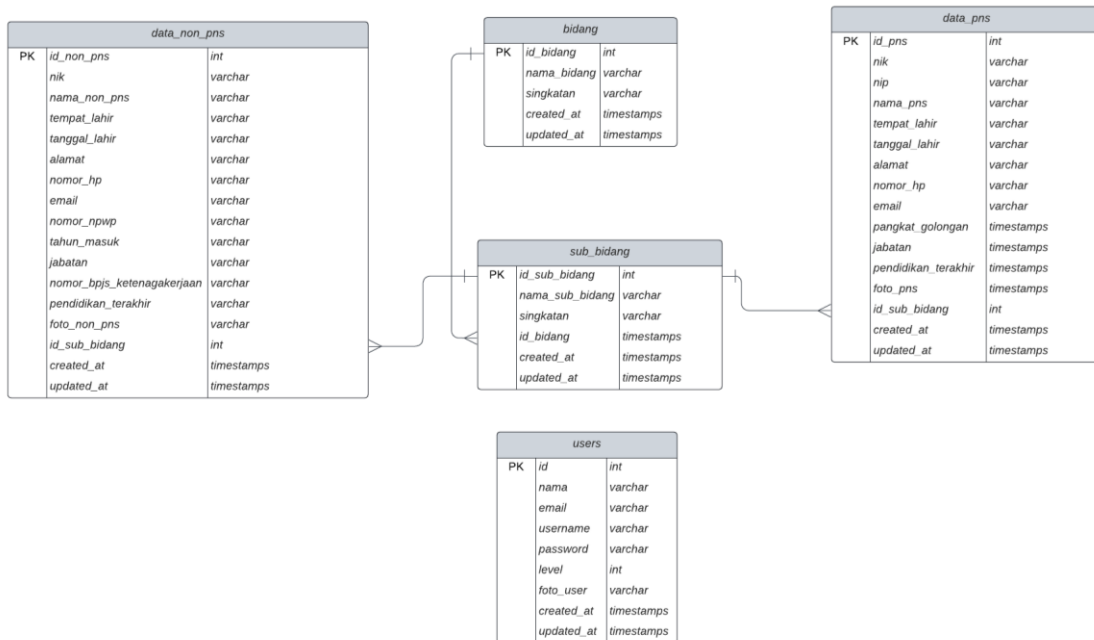
Berikut hasil Dekripsi yang telah di tampilkan di halaman *website* pada gambar 4.4 Hasil Dekripsi.



Gambar 4.4 Hasil Dekripsi

4.1.2.5 Desain Database

Dalam membentuk suatu struktur *database*, dibutuhkannya sebuah *Entity Relationship Diagram (ERD)* untuk dapat memahami struktur *database* beserta relasi antar tabel berdasarkan kebutuhan sistem. Berikut ini adalah hasil perancangan *ERD*:



Gambar 4.5 Rancangan ERD

4.1.3 Pengujian Blackbox

Dalam pengujian *blackbox*, ada beberapa pengujian yang diperlukan agar menghindari *bug* atau *error* yang terjadi ketika finalisasi aplikasi diantaranya:

4.1.3.1 Pengujian Login

Tabel 4.5 Pengujian Login

<i>Requirement</i>	Skenario uji	Hasil yang diharapkan	Hasil pengujian
<i>Login</i>	<i>Input login (jika benar)</i>	<i>Redirect</i> ke halaman dashboard	Sesuai
	<i>Input login (jika salah)</i>	Tampil notifikasi gagal login	Sesuai

4.1.3.2 Pengujian Halaman Manajemen Data PNS

Tabel 4.6 Pengujian Halaman Manajemen Data PNS

<i>Requirement</i>	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Tambah Data	<i>Input data dalam form tambah dan submit (jika lengkap semua)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data PNS	Sesuai
	<i>Input data dalam form tambah dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> , dan data tidak tersimpan	Sesuai
Edit Data	<i>Input data dalam form edit dan submit (jika lengkap)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data PNS	Sesuai
	<i>Input data dalam form modal edit dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> dan data tidak tersimpan	Sesuai
Hapus Data	Tombol Hapus	Tampil notifikasi berhasil ketika klik tombol hapus, data terhapus dan <i>refresh</i> halaman list data PNS	Sesuai
		Tampil notifikasi gagal ketika klik tombol hapus, data tidak terhapus dan <i>refresh</i> halaman list data PNS	Sesuai
Lihat Data	Tombol Lihat	Tampil Data sesuai yang dipilih dengan data yang sudah didekripsi dan <i>redirect</i> ke halaman lihat Data PNS	Sesuai

4.1.3.3 Pengujian Halaman Manajemen Data Non PNS

Tabel 4.7 Pengujian Halaman Manajemen Data Non PNS

<i>Requirement</i>	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Tambah Data	<i>Input data dalam form tambah dan submit (jika lengkap semua)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data PNS	Sesuai
	<i>Input data dalam form tambah dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> , dan data tidak tersimpan	Sesuai
Edit Data	<i>Input data dalam form edit dan submit (jika lengkap)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data PNS	Sesuai

Tabel 4.7 Pengujian Halaman Manajemen Data Non PNS (Lanjutan)

Requirement	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Edit Data	<i>Input</i> data dalam <i>form</i> modal edit dan <i>submit</i> (jika tidak lengkap)	Tampil notifikasi gagal ketika <i>submit</i> dan data tidak tersimpan	Sesuai
Hapus Data	Tombol Hapus	Tampil notifikasi berhasil ketika klik tombol hapus, data terhapus dan <i>refresh</i> halaman list data Non PNS	Sesuai
		Tampil notifikasi gagal ketika klik tombol hapus, data tidak terhapus dan <i>refresh</i> halaman list data Non PNS	Sesuai
Lihat Data	Tombol Lihat	Tampil Data sesuai yang dipilih dengan data yang sudah didekripsi dan <i>redirect</i> ke halaman lihat Data Non PNS	Sesuai

4.1.3.4 Pengujian Halaman Manajemen Data Bidang

Tabel 4.8 Pengujian Halaman Manajemen Data Bidang

Requirement	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Tambah Data	<i>Input</i> data dalam <i>form</i> tambah dan <i>submit</i> (jika lengkap semua)	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dan <i>redirect</i> ke halaman list data Bidang	Sesuai
	<i>Input</i> data dalam <i>form</i> tambah dan <i>submit</i> (jika tidak lengkap)	Tampil notifikasi gagal ketika <i>submit</i> , dan data tidak tersimpan	Sesuai
Edit Data	<i>Input</i> data dalam <i>form</i> edit dan <i>submit</i> (jika lengkap)	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data Non PNS	Sesuai
	<i>Input</i> data dalam <i>form</i> modal edit dan <i>submit</i> (jika tidak lengkap)	Tampil notifikasi gagal ketika <i>submit</i> dan data tidak tersimpan	Sesuai
Hapus Data	Tombol Hapus	Tampil notifikasi berhasil ketika klik tombol hapus, data terhapus dan <i>refresh</i> halaman list data Bidang	Sesuai
		Tampil notifikasi gagal ketika klik tombol hapus, data tidak terhapus dan <i>refresh</i> halaman list data Bidang	Sesuai

4.1.3.5 Pengujian Halaman Manajemen Data Sub Bidang

Tabel 4.9 Pengujian Halaman Manajemen Data Sub Bidang

Requirement	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Tambah Data	<i>Input</i> data dalam <i>form</i> tambah dan <i>submit</i> (jika lengkap semua)	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dan <i>redirect</i> ke halaman list data Sub Bidang	Sesuai
	<i>Input</i> data dalam <i>form</i> tambah dan <i>submit</i> (jika tidak lengkap)	Tampil notifikasi gagal ketika <i>submit</i> , dan data tidak tersimpan	Sesuai

Tabel. 4.9 Pengujian Halaman Manajemen Data Sub Bidang (Lanjutan)

<i>Requirement</i>	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Edit Data	<i>Input data dalam form edit dan submit (jika lengkap)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan redirect ke halaman list data Sub Bidang	Sesuai
	<i>Input data dalam form modal edit dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> dan data tidak tersimpan	Sesuai
Hapus Data	Tombol Hapus	Tampil notifikasi berhasil ketika klik tombol hapus, data terhapus dan <i>refresh</i> halaman list data Sub Bidang	Sesuai
		Tampil notifikasi gagal ketika klik tombol hapus, data tidak terhapus dan <i>refresh</i> halaman list data Sub Bidang	Sesuai

4.1.3.6 Pengujian Halaman Manajemen Data Akun

Tabel 4.10 Pengujian Halaman Manajemen Data Akun

<i>Requirement</i>	Skenario uji	Hasil yang diharapkan	Hasil pengujian
Tambah Data	<i>Input data dalam form tambah dan submit (jika lengkap semua)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan redirect ke halaman list data Akun	Sesuai
	<i>Input data dalam form tambah dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> , dan data tidak tersimpan	Sesuai
Edit Data	<i>Input data dalam form edit dan submit (jika lengkap)</i>	Tampil notifikasi berhasil ketika <i>submit</i> , data tersimpan dengan dienkripsi dan <i>redirect</i> ke halaman list data Non PNS	Sesuai
	<i>Input data dalam form modal edit dan submit (jika tidak lengkap)</i>	Tampil notifikasi gagal ketika <i>submit</i> dan data tidak tersimpan	Sesuai
Hapus Data	Tombol Hapus	Tampil notifikasi berhasil ketika klik tombol hapus, data terhapus dan <i>refresh</i> halaman list data Akun	Sesuai
		Tampil notifikasi gagal ketika klik tombol hapus, data tidak terhapus dan <i>refresh</i> halaman list data Akun	Sesuai

4.1.4 Pengujian Whitebox

Untuk menguji *white box* di aplikasi ini, maka dibutuhkannya *flowgraph*. *White box* menguji tiap fungsi pernyataan yang ada di aplikasi ini seperti diantaranya adalah *basis path* dan *loop testing*. Pengujian ini diuji dengan menggunakan perhitungan *Cyclomatic Complexity* yang memiliki rumus:

$$V(G) = E - N + 2P$$

Keterangan:

E = Jumlah *edges* (anak panah pada *flowgraph*)

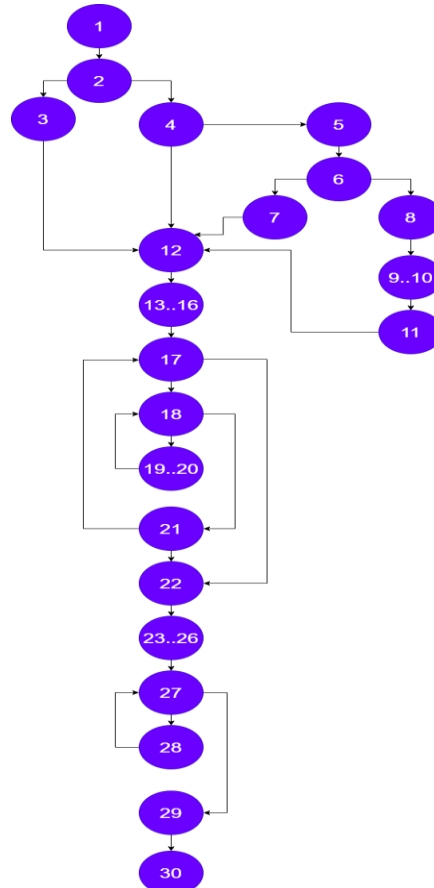
N = Jumlah *nodes* (lingkaran pada *flowgraph*)

P = Jumlah *predicate node* (koneksi antar *nodes*)

4.1.4.1 Proses Enkripsi

```
1 ~ @php$check_pt = strlen($plaintext); //menghitung panjang sebuah karakter
2 ~ if ($check_pt % 3 == 0) { //proses modulus atau sisa hasil bagi
3     $str_split = str_split($plaintext); //mengubah string menjadi array (per karakter)
4 ~ } else { //jika tidak
5     $plaintext .= '.'; //menambahkan karakter titik
6 ~ if ($check_pt % 3 == 0) { //lalu di cek lagi dengan modulus atau sisa hasil bagi
7     $str_split = str_split($plaintext); //mengubah string menjadi array (per karakter)
8 ~ } else {
9     $plaintext .= '.'; //menambahkan karakter titik
10    $str_split = str_split($plaintext); //mengubah string menjadi array (per karakter)
11    }
12    }
13    $array[][] = ''; //mendeklarasikan array
14    $bagikarakter = count($str_split) / 3; //membagi karakter 3
15    $divide = round($bagikarakter);
16    $index_arr = 0; //variabel index array str_split
17    for ($i = 0; $i < $divide; $i++) { //looping matriks
18    for ($j = 0; $j < 3; $j++) {
19        $array[$i][$j] = ord($str_split[$index_arr]);
20        $index_arr++;
21    }
22    }
23    $dividekurang = $divide - 1;
24    $hasil_kali = hitungKali(kuncimatriks(), $array, $dividekurang); //proses perkalian matriks kunci dengan array blok yang diatas
25    $hasil_enkrip = Desimaltoascii($hasil_kali); //proses mengubah desimal menjadi karakter ascii
26    $toString = '';
27    for ($i = 0; $i < count($hasil_enkrip); $i++) { //looping array matriks menjadi string
28        $toString .= $hasil_enkrip[$i][0] . '' . $hasil_enkrip[$i][1] . '' . $hasil_enkrip[$i][2];
29    }
30    return $toString;
31 ~ @endphp
```

Gambar 4.6 Pseudocode dan line number Proses Enkripsi



Gambar 4.7 Flowgraph Proses Enkripsi

Berdasarkan *Flowgraph* diatas dapat dihitung *Cyclomatic Complexity* (CC) berikut:

$$V(G) = E - N + 2P$$

$$V(G) = 26 - 22 + 2$$

$$V(G) = 6$$

Maka, jalur independent yang didapat:

Path 1 : 1 – 2 – 3 – 12 – 13..16 – 17 – 18 – 19..20 – 18 – 21 – 17 – 22 – 23..26 – 27 – 28 – 27 – 29 – 30

Path 2 : 1 – 2 – 4 – 12 – 13..16 – 17 – 18 – 19..20 – 18 – 21 – 17 – 22 – 23..26 – 27 – 28 – 27 – 29 – 30

Path 3 : 1 – 2 – 4 – 5 – 6 – 7 – 12 – 13..16 – 17 – 18 – 19..20 – 18 – 21 – 17 – 22 – 23..26 – 27 – 28 – 27 – 29 – 30

Path 4 : 1 – 2 – 4 – 5 – 6 – 8 – 9..10 – 11 – 12 – 13..16 – 17 – 18 – 19..20 – 18 – 21 – 17 – 22 – 23..26 – 27 – 28 – 27 – 29 – 30

Path 5 : 1 – 2 – 3 – 12 – 13..16 – 17 – 18 – 21 – 17 – 22 – 23..26 – 27 – 29 – 30

Path 6 : 1 – 2 – 3 – 12 – 13..16 – 17 – 21 – 17 – 22 – 23..26 – 27 – 29 – 30

Setelah menghitung CC maka jumlah tes yang dilakukan adalah sebanyak 6 kali dan berdasarkan jumlah CC yang dihitung bahwa *pseudocode* yang ada di proses enkripsi masih dalam kategori modul sederhana dengan resiko kecil sehingga *pseudocode* dapat dikatakan aman.

Tabel 4.12 Skenario Jalur Independen Proses Enkripsi

<i>Path</i>	Skenario yang diharapkan	Skenario yang sebenarnya	Validasi
1	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka tidak perlu ditambah titik (.) atau karakter tambahan	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka tidak perlu ditambah titik (.) atau karakter tambahan	Sesuai
2	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 1 karakter titik (.)	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 1 karakter titik (.)	Sesuai
3	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 1 karakter titik (.), pengecekan 2 kali karena didalam else	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 1 karakter titik (.), pengecekan 2 kali karena didalam else	Sesuai
4	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 2 karakter titik (.)	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka ditambah 2 karakter titik (.)	Sesuai
5	Ketika hasil jumlah <i>plaintext</i> dibagi menjadi per 3 karakter jumlahnya lebih dari 1 maka terjadi perulangan sesuai dengan jumlah dibagi tersebut	Ketika jumlah karakter plainteks mod 3 tidak sama dengan 0 maka tidak perlu ditambah titik (.) atau karakter tambahan	Sesuai
6	Ketika hasil jumlah <i>plaintext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja	Ketika hasil jumlah <i>plaintext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja	Sesuai

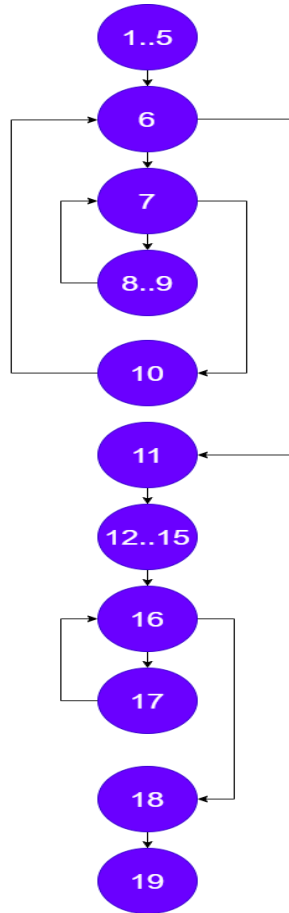
4.1.4.2 Proses Dekripsi

```

1 ~ @php$array[] = ''; //mendeklarasikan array
2     $string = str_split($ciphertext); //mengubah string menjadi array (per karakter)
3     $bagikarakter = count($string) / 3; //membagi karakter 3
4     $divide = round($bagikarakter);
5     $index_arr = 0; //variabel index array str_split
6 ~     for ($i = 0; $i < $divide; $i++) { //looping matrix
7 ~         for ($j = 0; $j < 3; $j++) {
8             $array[$i][$j] = ord($string[$index_arr]);
9             $index_arr++;
10        }
11    }
12    $dividekurang = $divide - 1;
13    $hasil_kali = hitungKali(inverskuncimatriks(), $array, $dividekurang); //proses perkalian matriks kunci dengan array blok yang diatas
14    $hasil_enkrip = DesimaltoAscii($hasil_kali); //proses mengubah desimal menjadi karakter ascii
15    $tostring = '';
16 ~     for ($i = 0; $i < count($hasil_enkrip); $i++) { //looping array matriks menjadi string
17         $tostring .= $hasil_enkrip[$i][0] . ' ' . $hasil_enkrip[$i][1] . ' ' . $hasil_enkrip[$i][2];
18     }
19    return rtrim($tostring, ' '); //menghilangkan karakter titik(.) paling kanan
20 ~ @endphp

```

Gambar 4.8 Pseudocode dan line number Proses Dekripsi



Gambar 4.9 Flowgraph Proses Dekripsi

Berdasarkan *Flowgraph* diatas dapat dihitung *Cyclomatic Complexity* (CC) berikut:

$$V(G) = E - N + 2P$$

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

Maka, jalur independent yang didapat:

Path 1 : 1..5 – 6 – 7 – 8..9 – 7 – 10 – 6 – 11 – 12..15 – 16 – 17 – 16 – 18 – 19

Path 2 : 1..5 – 6 – 7 – 10 – 6 – 11 – 12..15 – 16 – 17 – 16 – 18 – 19

Path 3 : 1..5 – 6 – 7 – 10 – 6 – 11 – 12..15 – 16 – 18 – 19

Path 4 : 1..5 – 6 – 11 – 12..15 – 16 – 18 – 19

Setelah menghitung CC maka jumlah tes yang dilakukan adalah sebanyak 4 kali dan berdasarkan jumlah CC yang dihitung bahwa *pseudocode* yang ada di proses dekripsi masih dalam kategori modul sederhana dengan resiko kecil sehingga *pseudocode* dapat dikatakan aman.

Tabel 4.13 Skenario Jalur Independen Proses Dekripsi

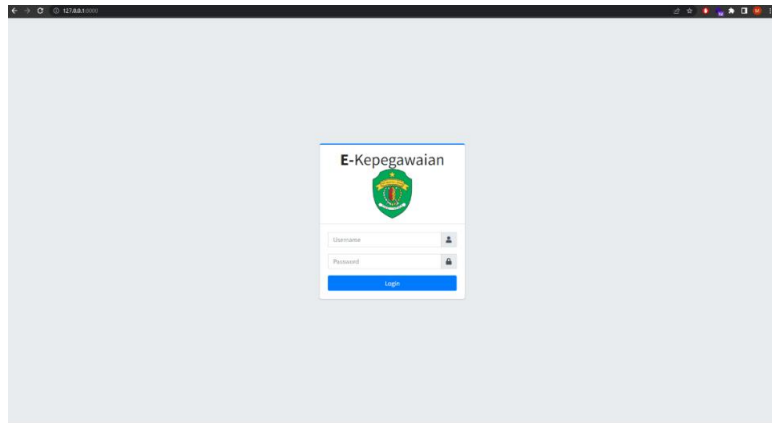
<i>Path</i>	Skenario yang diharapkan	Skenario yang sebenarnya	Validasi
1	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya lebih dari 1 maka terjadi perulangan sesuai dengan jumlah dibagi tersebut, dan juga di perulangan bercabang terjadi perulangan dua kali lipat dari jumlah dibagi.	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya lebih dari 1 maka terjadi perulangan sesuai dengan jumlah dibagi tersebut, dan juga di perulangan bercabang terjadi perulangan dua kali lipat dari jumlah dibagi.	Sesuai
2	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja dan juga di perulangan bercabang terjadi perulangan 3 kali saja karena proses memasukkan ke dalam array	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja dan juga di perulangan bercabang terjadi perulangan 3 kali saja karena proses memasukkan ke dalam array	Sesuai

Tabel 4.13 Skenario Jalur Independen Proses Dekripsi (Lanjutan)

<i>Path</i>	Skenario yang diharapkan	Skenario yang sebenarnya	Validasi
3	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja dan di perulangan akhir terjadi lebih dari 1 perulangan	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja dan di perulangan akhir terjadi lebih dari 1 perulangan	Sesuai
4	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja	Ketika hasil jumlah <i>ciphertext</i> dibagi menjadi per 3 karakter jumlahnya 1 maka terjadi perulangan 1 kali saja	Sesuai

5. IMPLEMENTASI

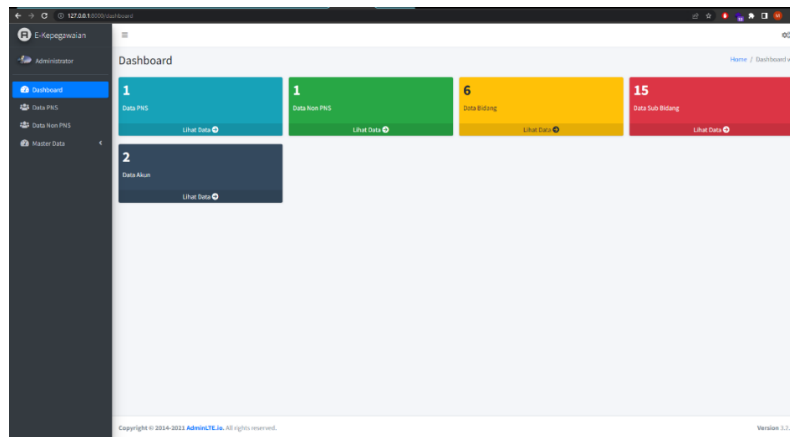
5.1 Halaman *Login*



Gambar 5.1 Halaman *Login*

Halaman *login* merupakan halaman yang akan diakses terlebih dahulu baik itu *admin* maupun *member* dikarenakan halaman *login* digunakan agar dapat menjaga *data privacy* yang digunakan oleh pihak OPD. *Input* yang diperlukan dihalaman ini adalah *username* dan *password*.

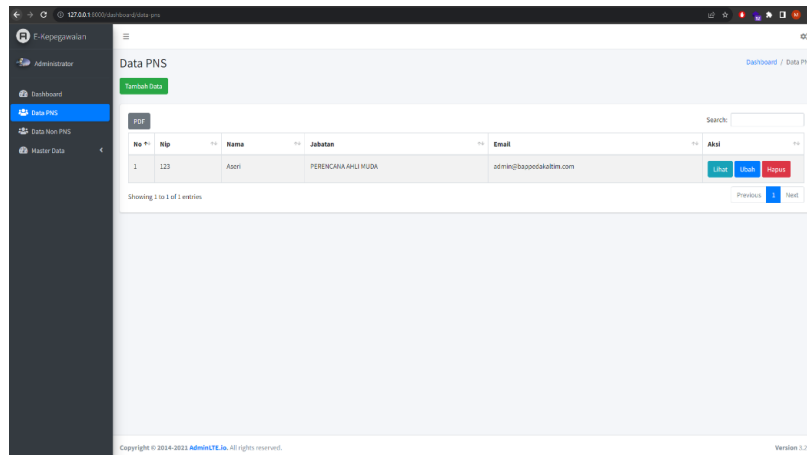
5.2 Halaman *Dashboard*



Gambar 5.2 Halaman *Dashboard*

Halaman *Dashboard* merupakan tampilan awal aplikasi setelah berhasil *login*. Halaman ini menampilkan jumlah setiap data yang ada agar dapat mudah mengetahui data yang sudah dikelola.

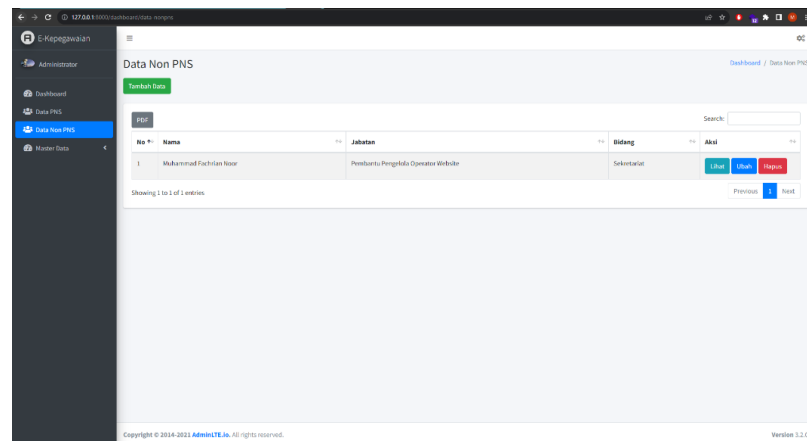
5.3 Halaman Manajemen Data PNS



Gambar 5.3 Halaman Manajemen Data PNS

Halaman Manajemen Data PNS merupakan halaman untuk mengatur data PNS yang ada di aplikasi ini. Dalam halaman ini juga dapat melakukan operasi CRUD (*Create, Read, Update and Delete*).

5.4 Halaman Manajemen Data Non PNS



Gambar 5.4 Halaman Manajemen Data Non PNS

Halaman Manajemen Data Non PNS merupakan halaman untuk mengatur data Non PNS yang ada di aplikasi ini. Dalam halaman ini juga dapat melakukan operasi CRUD (*Create, Read, Update and Delete*).

6. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka diambil kesimpulan sebagai berikut:

- 1) Hasil Enkripsi Algoritma *Hill Cipher* sulit dipecahkan/ditebak (didekripsi) karena pada Kode *ASCII* ini sebuah simbol, spasi, operator dan sebagainya dapat dikodekan menjadi suatu bilangan
- 2) Algoritma *Hill Cipher* hanya digunakan pada Data PNS dan Non PNS
- 3) Proses Enkripsi secara otomatis dilakukan didalam sistem pada saat menginput Data PNS dan Data Non PNS.
- 4) Proses Dekripsi secara otomatis juga dilakukan didalam sistem agar mudah dibaca pada saat di sistem.
- 5) Beberapa Karakter Kode *ASCII* menjadi perintah komputer, oleh sebab itu tidak bisa terbaca sebagai karakter.
- 6) Kunci matriks bersifat statis atau sudah ditetapkan untuk seluruh proses enkripsi.
- 7) Pengujian sistem menggunakan *black box* dan *white box*, tujuan dari pengujian ini yaitu untuk memastikan semua kesalahan masukan yang dilakukan oleh pengguna dapat ditangani oleh sistem.

7. SARAN

Beberapa saran untuk pengembangan aplikasi yang lebih baik dan relevan di masa mendatang adalah:

- 1) Diharapkan untuk kedepannya untuk menjadikan kunci matriks menjadi dinamis karena didalam sistem masih menggunakan kunci matriks statis.
- 2) Pada penelitian selanjutnya diharapkan dapat menampilkan proses enkripsi dan dekripsi pada sistem yang dibuat dikarenakan pada penelitian ini memiliki kendala pada *hardware* yang terbatas untuk menampilkan proses tersebut.

8. DAFTAR PUSTAKA

- Ardiansah, Irfan dan Ahmad, Faisal. 2021. *Perancangan Sistem Informasi Penjualan Kopi Berbasis Object Oriented Programming (OOP)*. Bandung: CV. Cendekia Press.
- Hasugian, Abdul Halim. 2013. *Implementasi Algoritma Hill Cipher Dalam Penyandian Data*. Jurnal Pelita Informatika Budi Darma. Vol 4. No. 2.
- Hidayat, Akik, Tuty Alawiyah. 2013. *Enkripsi dan Dekripsi Teks menggunakan Algoritma Hill Cipher dengan Kunci Matriks Persegi Panjang*. Jurnal Matematika Integratif. Vol. 9. No. 1.
- Hidayatullah, Ary dan Insanudin, Entik. 2016. *Pengenalan Kriptografi Dan Pemakaiannya Sehari-hari*. Bandung: Universitas Islam Negeri Sunan Gunung Djati.
- Injosoft. 2020. *ASCII Code*. <http://www.ascii-code.com/>. Diakses pada 10 Agustus 2023 pukul 15:12.
- Kani. 2020. *Algoritma dan Pemograman*. Tangerang: Universitas Terbuka.
- Lubis, Adyanata. 2016. *Basis Data Dasar*. Yogyakarta: Deepublish.
- Magamba, Kondwani, Solomon Kadaleka dan Ansley Kasambara. 2013. *Variable-length Hill Cipher with MDS Key Matrix.*, University of Malawi.
- Mardi. 2014. *Sistem Informasi Akuntansi*. Bogor: Ghalia.
- Munir, Rinaldi dan Leoda Lidya. 2016. *Algoritma dan Pemrograman Dalam Bahasa Pascal, C dan C++*. Bandung: Informatika.
- Rosa, A. S, dan Shalahuddin, M. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Yanto, Robi. 2016. *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: Deepublish.