Faktor Exacta

Vol. 17, No. 1, March 2024, pp. 9~19
eISSN: 2502-339X, pISSN: 1979-276X, DOI: 10.30998/faktorexacta.v17i1.20680

Kombinasi Algoritma Base64 Dan Caesar Cipher Pada Aplikasi

Yudo Devianto[1], Wawan Gunawan[2], Bambang Sukowo[3], Susafa'ati[4]
[1,2,3] Program Studi Teknik Informatika, Universitas Mercu Buana, Indonesia
[4]Program Studi Teknik Informatika, Universitas Nusa Mandiri, Indonesia

Article Info

Article history:

Received Oct 13, 2023
Revised Feb 12, 2024
Accepted Apr 17, 2024

Keywords:

Base64
Caesar cipher
Cryptography
Algoritma

ABSTRACT

Crimes related to information systems are increasing. When data on a website attracts the attention of many people, it becomes vulnerable to attacks. If the stolen database is encrypted, it will make it difficult for data thieves to exploit it. This research will combine BASE64 and CAESAR CIPHER algorithms in an application to secure financial data so that it cannot be viewed by unauthorized users and cannot be misused by irresponsible parties. White box testing on the data encryption algorithm with Caesar cipher resulted in a cyclomatic complexity value of 4. Thus, it can be concluded that the system is functioning properly because the testing produced the same value.

Corresponding Author:

Wawan Gunawan,
Program Studi Teknik Informatika,
Universitas Mercu Buana,
Jl. Meruya Selatan No.1, Kembangan, Jakarta Barat.
Email: wawan.gunawan@mercubuana.ac.id

1. PENDAHULUAN

Di era yang serba digital ini sistem informasi digital bukan lagi hal yang asing dalam masyarakat. Sistem Informasi Digital merupakan sebuah transformasi dari sistem manual menuju sistem yang otomatis, sehingga meningkatkan efektivitas pola pengelolaan dari yang sebelumnya rawan salah karena hanya bergantung pada kemampuan manusia menjadi lebih baik. Dalam sistem informasi digital kemanan data juga harus diperhatikan karena bersifat rahasia [1], banyak terjadi masalah pada keamanan data yang berakibat hilangannya data atau terjadi kerusakan yang disebabkan oleh pihak yang tidak bertanggung jawab [2].

Permasalahan yang sering dihadapi mengenai keuangan adalah seringnya pihak yang tidak bertanggung jawab khususnya divisi IT yang kerap kali penasaran terhadap nominal gaji yang diterima oleh orang lain, sehingga rasa penasaran tersebut terpecahkan dengan coba masuk ke dalam struktur *database* suatu aplikasi. Jika kondisi ini tetap berlanjut, maka bisa jadi akan ada perubahan nominal yang dilakukan secara langsung melalui *database* [3]. Selain itu, akan terjadi kesenjangan sosial diantara sesama karyawan [4][5].

Pada penelitian sebelumnya mengenai data pasien dirumah sakit jiwa Prof. Dr. Muhammad Ildrem dapat dilihat oleh orang lain yang tidak memiliki kepentingan terhadap data tersebut. Hal ini menyebabkan data bisa saja menjadi sasaran tindak kejahatan oleh orang-orang yang tidak bertanggung jawab dengan menggunakan data pribadi pasien untuk tindak kejahatan seperti penipuan. Berdasarkan masalah tersebut maka diperlukan adanya sebuah pengamanan data pada *database*. Pengamanan data yang akan digunakan untuk menjaga data yang ada, yaitu dengan menggunakan kombinasi Algoritma ROT47 dan Algortima Base64 untuk mengenkripsi data pasien, di mana dengan mengkombinasi kedua algoritma tersebut dapat menjadi salah satu teknik enkripsi yang cukup rumit, karena apabila ada orang yang ingin melihat teks asli harus mengetahui kunci dan jenis kombinasi algoritma yang digunakan untuk mengenkripsi data yang ada. Hasil dari penelitian tersebut adalah sebuah aplikasi data pasien yang sudah ditambakan algoritma ROT47 dan Base64 di mana data yang dienkripsi terkait dengan data alamat, telepon, penyakit, diagnosis dan data pengobatan. [6]

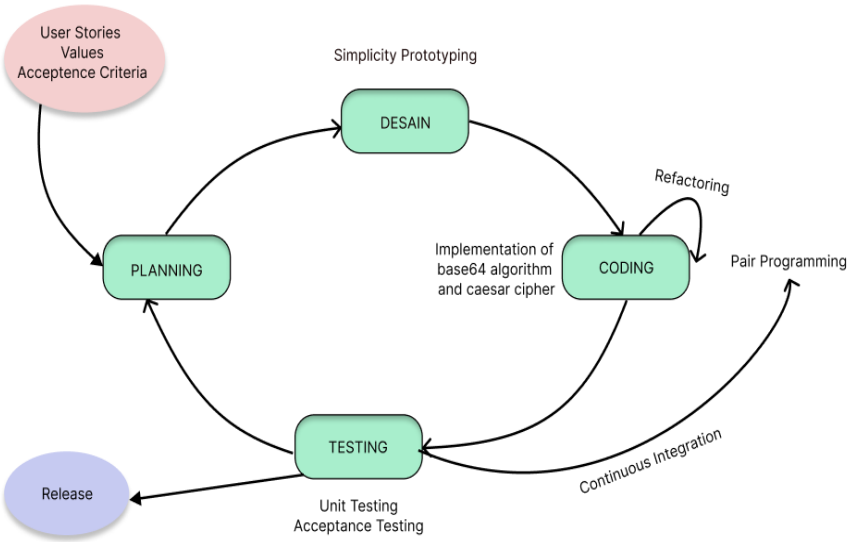
Menurut penelitian sebelumnya jumlah kejahatan yang berhubungan dengan sistem informasi terus meningkat. Ketika data yang ada disuatu website menarik perhatian banyak orang maka akan rentan menjadi target serangan, jika *database* yang dicuri telah terenkripsi maka akan mempersulit pencuri data untuk memanfaatkannya, tetapi jika tidak terenkripsi maka sangat mudah data tersebut dimanfaatkan sehingga membuat kerugian yang sangat besar terutama kepercayaan pengguna dari website tersebut. Oleh karena itu, diperlukan peningkatan keamanan pada sebuah database dengan menggunakan kombinasi algoritma. Pada penelitian tersebut menggunakan algoritma *Cest Cryptography* dan algoritma Base64 pada aplikasi jasa pengangkutan sampah, kombinasi algoritma diterapkan pada *field-field* yang dianggap penting dan sangat rentan jika diketahui oleh orang banyak seperti pada *username* dan *password*. Berdasarkan penelitian tersebut *database* sudah terlindungi karena beberapa data yang penting sudah dienkripsi dengan 2 kunci *private* dan 2 kunci *public* agar tidak mudah dibaca oleh orang yang tidak berhak, sehingga data menjadi lebih aman. Penerapan kombinasi algoritma pada sebuah sistem membutuhkan waktu yang relatif lebih lama daripada yang tidak menggunakan enkripsi [7].

Menurut penelitian lain mengenai penyebaran informasi yang sangat pesat, masalah keamanan komputer menjadi suatu bagian yang terpenting. Berbagai metode keamanan ditingkatkan seperti mengamankan pesan menjadi sesuatu yang tidak dapat dimengerti oleh orang lain, teknik tersebut dinamakan dengan kriptografi [8]. Pada penelitian tersebut peneliti memadukan kriptografi algoritma Base64 dengan metode steganografi *DCT* dengan bantuan *LSB*, proses yang dilakukan dengan menyembunyikan pesan yang telah terenkripsi menggunakan algoritma Base64 ke dalam *file* image melalui teknik steganografi *DCT* dan LSB. Dari hasil penelitian terhadap aplikasi yang sudah dibuat didapatkan bahwa dengan perpaduan enkripsi algoritma Base64 dengan metode steganografi *Distrete Cosine Transporm* (*DCT*) dalam menyisipkan pesan ke dalam gambar yang berformat png dalam 10 kali pengujian didapatkan perubahan yang terletak pada berkurangnya size sebelum dan sesudah dilakukan *encode* dengan rata-rata penurunan size yaitu 9.71 KB. Selain itu, terjadi juga perubahan resolusi ketika sebelum dan sesudah *encode*. [9]

Pada penelitian ini akan melakukan kombinasi algoritma *BASE64* dan *CAESAR CIPHER* pada aplikasi untuk menjaga keamanan data keuangan agar tidak dapat terlihat oleh pengguna yang tidak memiliki akses terhadap aplikasi dan juga tidak dapat disalahgunakan oleh pihak yang tidak bertanggung jawab.

2. METODE

Pengembangan sistem pada penelitian ini terlihat seperti pada Gambar 1 yang menggunakan metode Extreme Programming (XP), yang merupakan satu dari sekian banyak metodologi yang dapat digunakan untuk menerapkan prinsip pengembangan perangkat lunak berbasis agile. Agile sendiri adalah prinsip pengembangan perangkat lunak yang mengutamakan adaptasi terhadap perubahan, mementingkan fungsional aplikasi daripada dokumentasi, dan prinsip-prinsip agile lainnya.

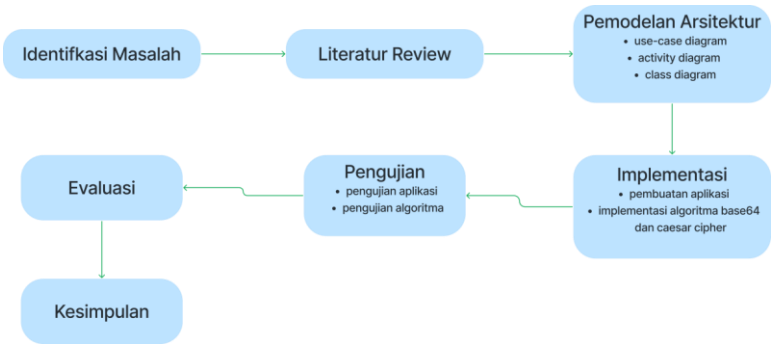


Gambar 1. Pengembangan sistem *extreme programming*

Diagram pada Gambar 1 merupakan representasi dari siklus pengembangan perangkat lunak yang mengikuti metodologi *Agile*, dengan tahapan sebagai berikut:

1. Perencanaan (*Planning*): Tahap ini melibatkan penentuan cerita pengguna (*User Stories*) dan kriteria penerimaan (*Acceptance Criteria*). Fase di mana kebutuhan dan fitur-fitur yang harus dikembangkan diidentifikasi dan direncanakan.
2. Desain (*Design*): Setelah perencanaan, proses desain dimulai dengan *prototyping* yang sederhana untuk merancang solusi secara visual dan fungsional sebelum memulai proses *coding*.
3. Pengkodean (*Coding*): Fase implementasi, di mana algoritma berbasis kriteria telah ditentukan dan fitur lainnya dikembangkan.
4. Pengujian (*Testing*): Setelah pengkodean, produk akan melalui serangkaian pengujian termasuk unit *testing* dan *acceptance* testing untuk memastikan bahwa perangkat lunak berfungsi seperti yang diharapkan dan memenuhi kriteria yang telah ditetapkan.
5. Rilis *(Release)*: Ini adalah tahap akhir di mana perangkat lunak yang sudah diuji dan siap digunakan dirilis ke pasar atau dikirimkan kepada pengguna.

Diagram tersebut juga menunjukkan bahwa pengembangan perangkat lunak adalah proses iteratif dan inkremental dengan umpan balik yang kontinu dan integrasi berkelanjutan (*Continuous Integration*), yang memungkinkan perubahan pada setiap tahap dari siklus hidup pengembangan perangkat lunak. Sedangkan dalam penelitian ini tahapan yang dilakukan dimulai dari identifikasi masalah, *literature review*, pemodelan arsitektur, implementasi, evaluasi dan kesimpulan seperti tampak pada gambar 2.



Gambar 2. Tahapan penelitian

Diagram alur yang ditampilkan pada gambar 2 terkait proses penelitian atau pengembangan sistem yang sistematis, berikut tahapan dalam diagram tersebut:

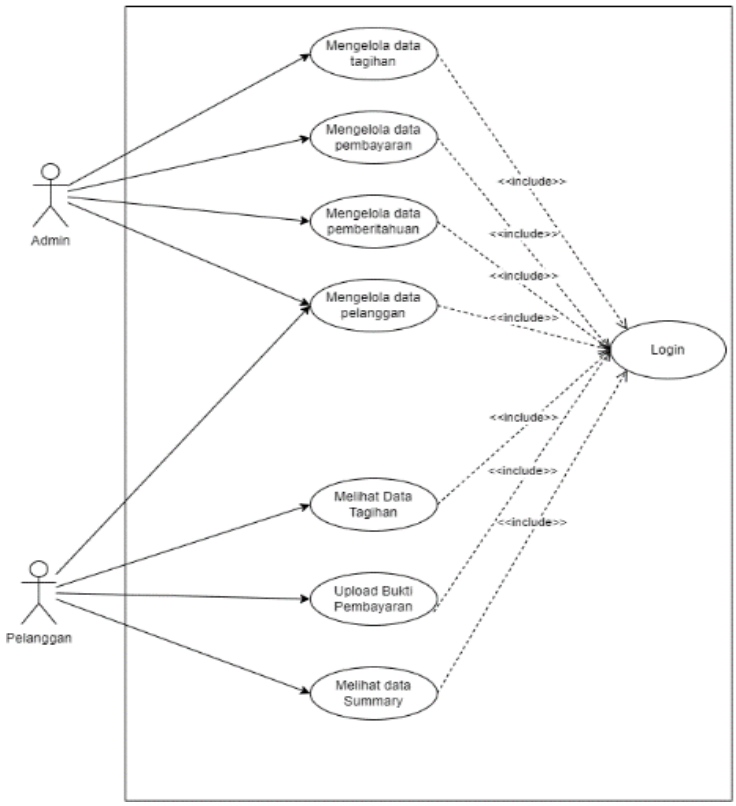
1. Identifikasi Masalah: Proses dimulai dengan mengidentifikasi masalah yang akan ditangani dan merupakan tahap penting di mana peneliti atau pengembang menentukan fokus dan tujuan dari penelitian yang akan dilakukan.
2. *Literatur Review*: Setelah masalah teridentifikasi, dilakukan penelaahan literatur yang meliputi pengumpulan dan analisis publikasi yang relevan untuk mendapatkan pemahaman yang lebih mendalam tentang topik dan untuk menemukan celah penelitian yang ada.
3. Pemodelan Arsitektur: Tahap ini melibatkan perancangan struktur sistem dengan menggunakan diagram-diagram seperti *use-case diagram* dan *class diagram* untuk membantu dalam visualisasi dan strukturisasi komponen sistem serta interaksi antara mereka.
4. Implementasi: Berdasarkan model arsitektur yang telah dirancang, tahap implementasi melibatkan pembuatan aplikasi dan penerapan algoritma yang spesifik, seperti algoritma *base64* dan *caesar cipher*.
5. Pengujian: Setelah implementasi, sistem atau aplikasi yang telah dibangun akan diuji untuk memastikan bahwa semua fungsi bekerja sebagaimana mestinya.
6. Evaluasi: Evaluasi adalah tahap di mana hasil pengujian dianalisis untuk menilai apakah yang dikembangkan berhasil memenuhi tujuan yang ditetapkan.
7. Kesimpulan: Tahap terakhir menarik kesimpulan dari seluruh proses penelitian atau pengembangan, kesimpulan mencakup rangkuman dari temuan, pembahasan mengenai implikasi dari hasil pengembangan atau penelitian, serta rekomendasi untuk penelitian atau pengembangan lebih lanjut.

 DOI: 10.30998/faktorexacta.v17i1.20680

3. HASIL DAN PEMBAHASAN

3.1. *Use Case Diagram*

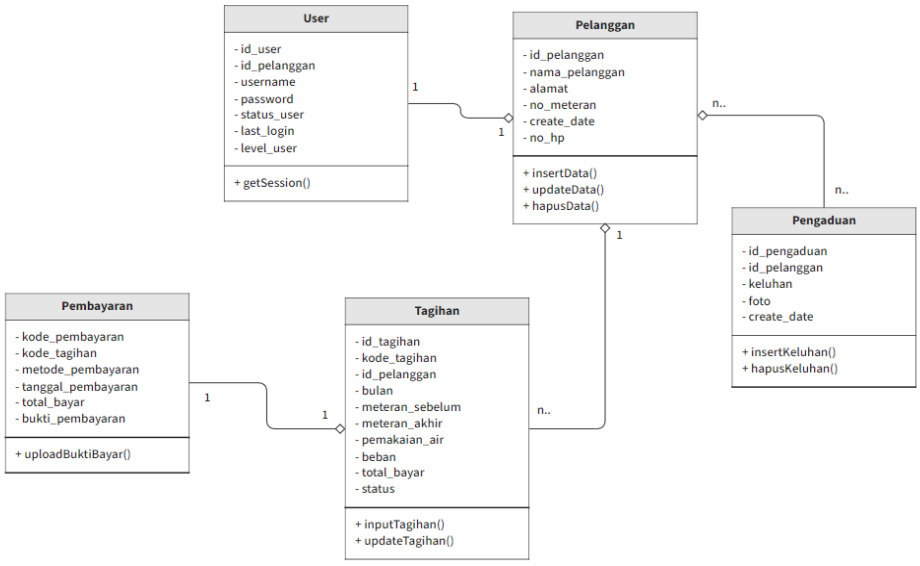
Gambar 3 merupakan *Use Case Diagram* yang menggambarkan aktivitas pengguna terhadap sistem yang dikembangkan. Pada aplikasi yang dibuat terdapat 2 aktor yang akan menggunakan sistem tersebut yaitu admin dan pelanggan.



Gambar 3. *Use Case Diagram* pengembangan sistem

3.2. *Class Diagram*

Class diagram memperlihatkan hubungan antar kelas dan menggambarkan keadaan (atribut/properti) dari suatu sistem yang terdapat pada Gambar 4.



Gambar 4. *Class Diagram* pengembangan sistem

3.3. Implementasi Algoritma

Penelitian ini menggabungkan algoritma *Base64* dengan algoritma *Caesar Cipher*. Untuk prosesnya sendiri yang pertama dilakukan adalah enkripsi menggunakan *Base64* terlebih dahulu selanjutnya dibungkus lagi dengan enkripsi *Caesar chipper*. Adapun langkah-langkah yang dilakukan untuk membentuk chiperteks dengan *Caesar Cipher* adalah:

a. Menentukan besarnya pergeseran karakter yang digunakan dalam membentuk *cipherteks* ke *plainteks*
b. Menukarkan karakter pada plainteks menjadi *cipherteks* dengan berdasarkan pada pergeseran yang telah ditentukan sebelumnya.

Algoritma dari *Caesar Cipher* adalah C = E (P) = (P + K) mod 26 untuk fungsi enkripsi. Sedangkan untuk fungsi dekripsi adalah P = D (C) = (C - K) mod 26 dapat dilihat pada gambar 5.

```php
function geser_teks($string, $key) {
    return implode('', array_map(function ($char) use ($key) {
        return $this->geser_karakter($char, $key);
    }, str_split($string)));
}

function geser_karakter($char, $shift) {
    $shift = $shift % 25;
    $ascii = ord($char);
    $shifted = $ascii + $shift;

    if ($ascii >= 65 && $ascii <= 90) {
        return chr($this->geser_huruf_besar($shifted));
    }elseif ($ascii >= 97 && $ascii <= 122) {
        return chr($this->geser_huruf_kecil($shifted));
    }elseif ($ascii >= 33 && $ascii <= 58) {
        return chr($this->geser_angka($shifted));
    }

    return chr($ascii);
}

function geser_angka($ascii) {
    if ($ascii < 33) {
        $ascii = 59 - (33 - $ascii);
    }elseif ($ascii > 58) {
        $ascii = ($ascii - 58) + 32;
    }
    return $ascii;
}

function geser_huruf_besar($ascii) {
    if ($ascii < 65) {
        $ascii = 91 - (65 - $ascii);
    }elseif ($ascii > 90) {
        $ascii = ($ascii - 90) + 64;
    }

    return $ascii;
}

function geser_huruf_kecil($ascii) {
    if ($ascii < 97) {
        $ascii = 123 - (97 - $ascii);
    }elseif ($ascii > 122) {
        $ascii = ($ascii - 122) + 96;
    }

    return $ascii;
}
```

```php
function enkripsi_caesar($plaintext, $key = 12) {
    return $this->geser_teks($plaintext, $key);
}

function dekripsi_caesar($ciphertext, $key = -12) {
    return $this->geser_teks($ciphertext, -$key);
}
```

Gambar 5. *Function enkripsi caesar*

Gambar 5 merupakan *function* yang digunakan untuk proses enkripsi dari *Caesar Cipher*. Untuk proses enkripsi *Base64* peneliti menggunakan *library* bawaan dari *library codeigniter*, algoritma tersebut diterapkan pada saat menyimpan data pembayaran ke *database*. Sebelum data di simpan ke *database*, data terlebih dahulu dienkripsi sehingga data masuk ke *database* berupa kata acak yang tidak bisa dibaca seperti ditampilkan pada gambar 6.

```php
if($this->upload->do_upload('file')){
    $data_pembayaran = array(
        'kode_pembayaran' => $this->enkripsi_caesar(base64_encode($newKodeBayar),9),
        'id_tagihan' => $this->encrypt->decode($this->input->post('id_tagihan_add')),
        'no_rekening' => $this->enkripsi_caesar(base64_encode($this->input->post('noreq')),9),
        'metode_pembayaran' =>$this->enkripsi_caesar(base64_encode($metodeBayar),9),
        'tanggal_pembayaran' => date('Y-m-d h:i:s'),
        'total_bayar' =>$this->enkripsi_caesar(base64_encode($this->input->post('total_bayar')),9),
        'status' => "Menunggu",
        'bukti_bayar' => $this->enkripsi_caesar(base64_encode($surat),9),
        'id_users_create' => $this->session->userdata('id_user'),
        'create_date' => date('Y-m-d h:i:s'),
    );
```

Gambar 6. Penerapan Enkripsi Algoritma *Base64* dan *Caesar Cipher*

Beberapa data yang akan diinput ke database terlebih dahulu di enkripsi menggunakan *Base64* dan *Caesar Cipher*, sehingga hasilnya dapat dilihat pada gambar 7.

Data Output | Explain | Messages | Notifications

	kode_pembayaran [PK] text	metode_pembayaran text	tanggal_pembayaran date	total_bayar text	status character varying	bukti_bayar text	id_users_cr integer	create_date date	no_rekening text	id_tagihan integer
1	S0hBWVZCOTkl	Q2Fqd2JvbmE=	2022-12-15	IScIlzk5	Menunggu	Ym56ZG53bG5f...	8	2022-12-15	JyclIylhliUkJ...	8
2	S0hBWVZCU1NZ	Q2Fqd2JvbmE=	2022-12-16	OiMkJTk5	Menunggu	a2psdHBheGR3...	7	2022-12-16	JyclIylhliUkJ...	9

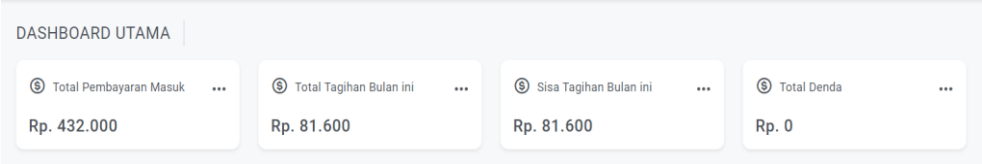
Gambar 7. Hasil Enkripsi Data

Berdasarkan gambar 7 beberapa data yang disimpan ke dalam *database* sudah dienkripsi sehingga data yang tersimpan kedatabase berupa huruf acak yang tidak dapat dibaca dan tidak ada artinya.

```
if(count($getTotalTagihan)>0){
    foreach($getTotalTagihan as $en ){
        $de64 = base64_decode($en->total_bayar);
        $caesar = $this->dekripsi_caesar($en->total_bayar,9);
        $totalBayar[] = $this->dekripsi_caesar($de64,9);
        $jml = array_sum($totalBayar);
    }
}else{
    $jml = 0;
}

if(count($getTotalDenda)>0){
    foreach($getTotalDenda as $td ){
        $de64Denda = base64_decode($td->denda);
        $totalDenda[] = $this->dekripsi_caesar($de64Denda,9);
        $jmlDenda = array_sum($totalDenda);
    }
}else{
    $jmlDenda = 0;
}
```

Gambar 8. Penerapan Dekripsi Algoritma *Base64* dan *Caesar Cipher*

Pada penggunaan dekripsi dari *plaintext* yang sudah dienkripsi yang pertama dilakukan adalah kebalikan dari proses enkripsi yaitu dimulai dari *decode Base64* lalu dekripsi *Caesar* dengan kode program dapat dilihat pada gambar 8, sedangkan hasil dari proses dekripsi terlihat pada gambar 9.



Gambar 9. Hasil Dekripsi pada total pembayaran

Pada tahap implementasi algoritma dilakukan juga uji coba untuk membandingkan proses manual dengan algoritma yang sudah di implementasikan pada sistem. Berikut adalah uji coba hasil manual dan sistem dengan menggunakan nilai 20000. Proses implementasi tersebut dapat dilihat pada Tabel 1 sampai Tabel 5.
Plaintext yang diuji : 20000

1) Langkah pertama *plaintext* diubah menjadi biner terlebih dahulu seperti pada Tabel 1

Tabel 1. Mengubah Plaintext menjadi biner

2	0	0	0	0
00110010	00110000	00110000	00110000	00110000

Pada tabel 1 dijelaskan nilai pada masing-masing nilai 20000 dengan masing-masing bilangan binernya, nilai 2 dengan biner 00110010, nilai 0 dengan biner 00110000.

2) Binner yang sudah didapat berupa binner 8 bit, akan dilakukan perubahan ke kelompok biner 6 bit seperti yang terlihat pada Tabel 2.

Tabel 2. Mengubah 8 bit menjadi 6 bit

00110010	00110000	00110000	00110000	00110000		
001100	100011	000000	110000	001100	000011	000000

3) Convert masing-masing kelompok 6 bit ke decimal seperti pada Tabel 3

Tabel 3. Konvert 6 bit ke decimal

001100	100011	000000	110000	001100	000011	000000
12	35	0	48	12	3	0

Sebagai contoh untuk biner 001100 akan dilakukan konversi dari nilai masing-masing kanan ke kiri menjadi nilai 12 seperti cara berikut:

Bit paling kanan (posisi 0) adalah 0, jadi $0\times2^0=0$
Bit berikutnya (posisi 1) adalah 0, jadi $0\times2^1=0$
Bit berikutnya (posisi 2) adalah 1, jadi $1\times2^2=4$
Bit berikutnya (posisi 3) adalah 1, jadi $1\times2^3=8$
Bit berikutnya (posisi 4) adalah 0, jadi $0\times2^4=0$
Bit berikutnya (posisi 5) adalah 0, jadi $0\times2^5=0$

Setelah mendapatkan hasilnya, jumlahkan semua hasilnya: 0+0+4+8+0+0=12

4) Gunakan masing-masing decimal untuk mencari kode karakter pada index *Base64* seperti pada Tabel 5 dengan rujukan untuk tabel index Base64 seperti pada tabel 4

Tabel 4. index Base64

Index	Value	Index	Value	Index	Value	Index	Value	Index	Value
0	A	14	O	28	c	42	q	56	4
1	B	15	P	29	d	43	r	57	5
2	C	16	Q	30	e	44	s	58	6
3	D	17	R	31	f	45	t	59	7
4	E	18	S	32	g	46	u	60	8
5	F	19	T	33	h	47	v	61	9
6	G	20	U	34	i	48	w	62	+
7	H	21	V	35	j	49	x	63	-
8	I	22	W	36	k	50	y		
9	J	23	X	37	l	51	z		
10	K	24	Y	38	m	52	0		
11	L	25	Z	39	n	53	1		
12	M	26	a	40	o	54	2		
13	N	27	b	41	p	55	3		

Berdasarkan hasil perhitungan biner ke bilangan desimal untuk nilai 12, maka kita merujuk pada tabel 4 untuk mencari nilai terkait index ke-12, maka dihasilkan untuk index tersebut adalah nilai M.

Tabel 5. Mencari kode karakter *Base64*

12	35	0	48	12	3	0

M	j	A	w	M	D	A

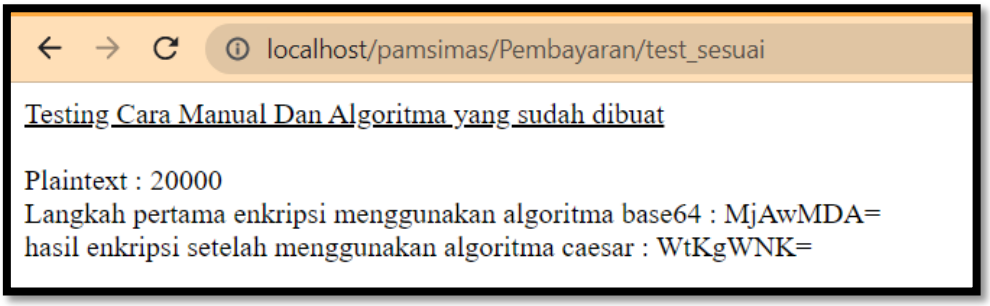
5) Hasil enkripsi dari *Base64* dari plaintext 20000 adalah : **MjAwMDA=**
6) Selanjutnya enkripsi lagi menggunakan *Caesar ciper* dengan menggunakan key 10, dengan tabel abjad seperti pada tabel 6

Tabel 6. Tabel abjad

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

7) Hasil dari enkripsi berdasarkan Caesar ciper adalah : **WtKgWNK=**
Proses yang dapat dilakukan dari hasil **MjAwMDA** untuk karakter **M** yang berada pada index ke-12, selanjutnya dilakukan pergeseran ke kanan sebanyak 10 maka akan dihasilkan pada index ke 22 berdasarkan hasil 12+10. Selanjutnya perubahan karakter-karakter tersebut disesuaikan dengan huruf awalnya yang didapatkan pada enkripsi *Base64*. Jika bentuk awal adalah huruf kapital, maka hasil caesar ciper menggunakan huruf kapital, pun demikian sebaliknya jika hasil dari *Base64* adalah huruf kecil maka pada caesar ciper akan menjadi huruf kecil.

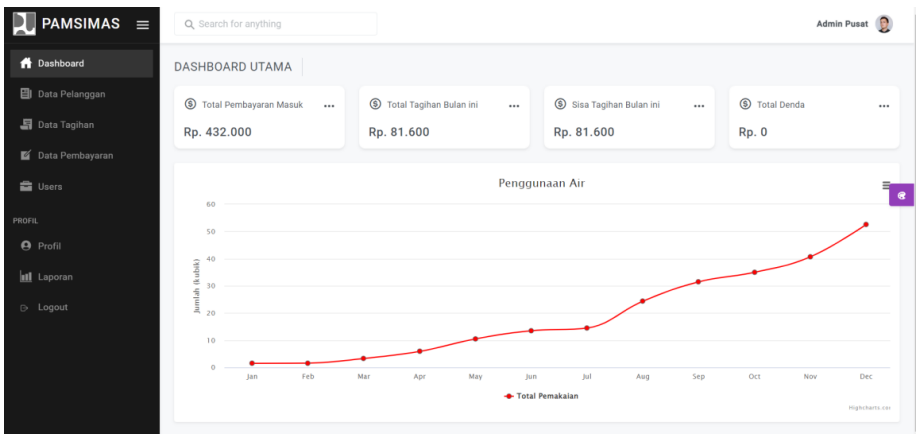
8) Hasil berdasarkan implementasi algoritma pada sistem seperti ditampilkan pada gambar 9.



Gambar 8. Penerapan Algoritma *Base64* dan *Caesar Cipher*

3.4. *User Interfase*

Penerapan algoritma *Base64* dan *Caesar Ciper* pada aplikasi dapat dilihat pada gambar 9. Gambar tersebut merupakan hasil proses dekripsi dari algoritma yang digunakan, sehingga pengguna dapat melihat nilai secara *realtime* terkait dengan biaya yang harus dibayarkan.



Gambar 9. Halaman *dashboard*

3.5. Pengujian Aplikasi

Pengujian yang dilakukan adalah menggunakan pengujian *Black box* di mana pengujian tersebut berfokus kepada melihat fungsi-fungsi yang ada pada sistem tanpa harus mengetahui bagaimana fungsi sistem tersebut dibuat. Setelah itu membandingkan hasil keluaran sistem dengan hasil yang diharapkan jika hasil pengujian sesuai dengan yang diharapkan berarti aplikasi yang sudah dibuat sudah sesaui dengan desain yang sudah dibuat sebelumnya, jika hasil belum sesuai maka perlu dilakukan pengecekan lanjut dan melakukan perbaikan pada aplikasi. Pengujian tersebut dapat dilihat pada tabel 7.

Tabel 7. Pengujian pengolahan data pembayaran

Pengujian	Kondisi Pengujian	Yang diharapkan	Hasil Pengujian
		Admin	
Validasi Pembayaran	Klik tombol validasi untuk mengecek apakah bukti yang diinput sesuai dengan jumlah tagihan	Tampil sebuah form pengecekan jika admin menyatakan bahwa bukti valid maka status pembayaran akan berganti dari "menunggu konfirmasi admin" ke "Terbayar"	**Sesuai**
		Pelanggan	
Melakukan Pembayaran	Klik tombol bayar untuk lanjut ke proses selanjutnya	Sistem akan menampilkan halaman rincian pembayaran, pemilihan metode bayar dan upload bukti pembayaran	**Sesuai**
Informasi Tagihan	Klik menu pembayaran	Jika terdapat tagihan maka sistem akan menampilkan total tagihan yang harus dibayar oleh pelanggan, jika tidak ada tagihan maka pesan tidak ada tagihan akan muncul	**Sesuai**
Riwayat Pembayaran	Klik menu pembayaran	Menampilkan list data riwayat pembayaran yang sudah dilakukan	**Sesuai**
		Hasil uji dengan data yang salah	
Upload bukti bayar	Tidak menambahkan file bukti dan langsung klik bayar	Sistem akan menampilkan *alert eror* karena *field* yang diminta belum di*upload*	**Sesuai**

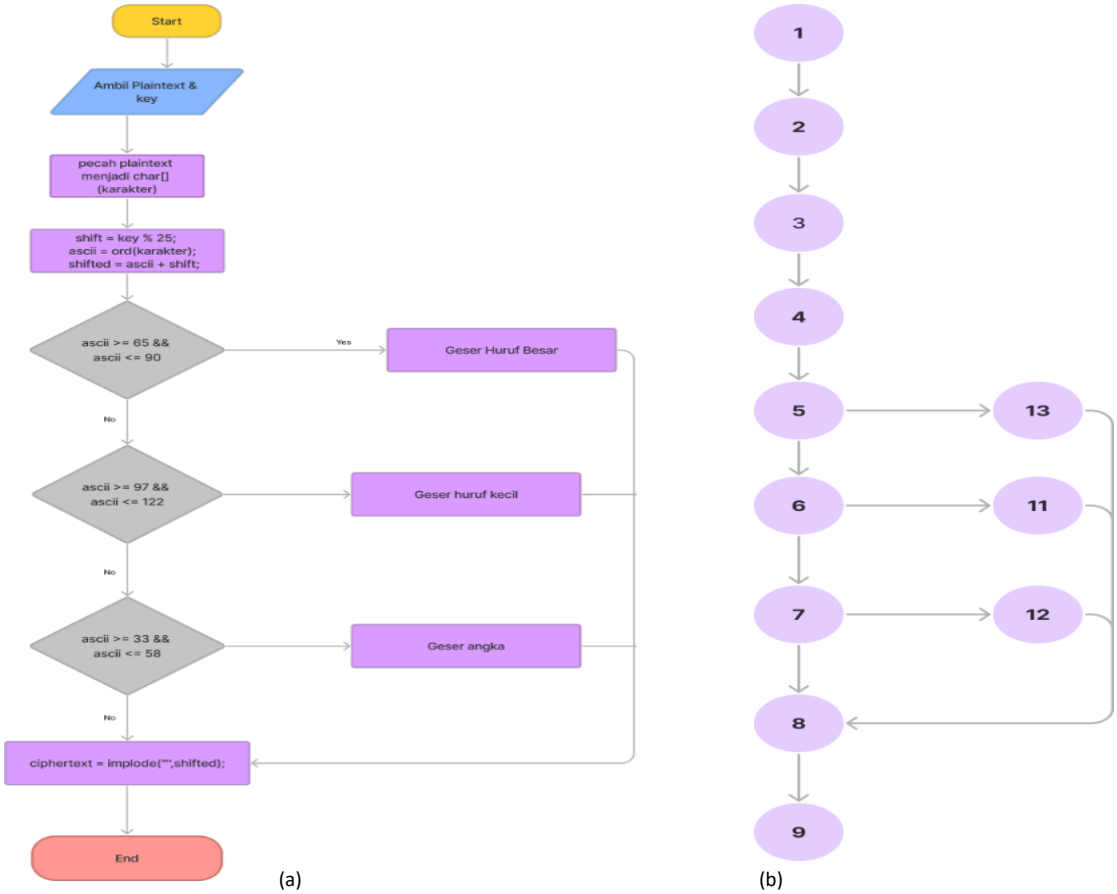
3.6. Pengujian Algoritma

Pengujian dimulai dengan melakukan pemeriksaan pada *source code* kemudian memetakan *flowchart* ke dalam *flowgraph*, selanjutnya menghitung besarnya jumlah *edge* dan *node* untuk menentukan besarnya *cy-lomatic complexity*. Berikut ini adalah *pseudocode* dari proses enkripsi *Caesar Cipher* yang dijalankan oleh sistem seperti pada gambar 10.

```
Deklarasi
    key : integer;
    ciphertext : string;
Dekripsi
    input key,plaintext;
    karakter [i] ← plaintext;
    shift = key % 25;
    ascii = ord(karakter);
    shifted = ascii + shift;
    if (ascii >= 65 && ascii <= 90) {
        if (shifted < 65) {
            shifted = 91 - (65 - shifted);
        }

        if (shifted > 90) {
            shifted = (shifted - 90) + 64;
        }
    }
    if (ascii >= 97 && ascii <= 122) {
        if (shifted < 97) {
            shifted = 123 - (97 - shifted);
        }

        if (shifted > 122) {
            shifted = (shifted - 122) + 96;
        }
    }
    if (shifted >= 33 && shifted <= 58) {
        shifted = shifted;
    }

    return chr($shifted);
    ciphertext = implode("",ascii);

write (ciphertext)
```

Gambar 10. *Pseudocode* proses enkripsi *Caesar Cipher*

Untuk tahap selanjutnya adalah mengubah *pseudocode* menjadi *flowchart* seperti pada gambar 11(a).



Gambar 11. *flowchart* algoritma *Caesar Cipher*

Berdasarkan *flow graph* pada gambar 11(b) dapat diketahui bahwa jumlah *edge* (E) = 13 yang merupakan garis yang menghubungkan *node*, jumlah *node* (N)=11 yang merupakan lingkaran yang menggambarkan suatu aktivitas, jumlah *predicate* (P) = 3 yang merupakan *node* bercabang, dan jumlah *region* (R) = 4 yang menandakan area dalam *flow graph*. Jika dimasukkan kedalam rumus perhitungan *cyclomatic complexity* dari jumlah *region*, *cyclomatic complexity* dari *Edge* dan *Node*, dan *cyclomatic complexity* dari *predicate code* (P) maka akan menghasilkan sebagai berikut:

A. *Cyclomatic complexity* dari jumlah *region*
   V(G) = R
   V(G) = 4
B. *Cyclomatic complexity* dari *Edge* dan *Node*
   V(G) = E-N+2
   V(G) = 13-11+2
   V(G) = 4
C. *Cyclomatic complexity* dari *predicate code*
   V(G) = P+1
   V(G) = 3+1
   V(G) = 4

Berdasarkan hasil perhitungan *cyclomatic complexity* didapatkan humlah hasil *independent path* adalah 4:

Path 1: 1-2-3-4-5-6-7-8-9
Path 2 : 1-2-3-4-5-13-8-9
Path 3 : 1-2-3-4-5-11-8-9
Path 4 : 1-2-3-4-5-12-8-9

Berdasarkan pengujian *white box* pada algoritma enkripsi data dengan *Caesar chipper* dihasilkan nilai *cyclomatic complexity* yang sama yaitu 4. Sehingga dapat disimpulkan bahwa sistem berjalan dengan baik karena pengujian menghasilkan nilai yang sama.

4. KESIMPULAN

Berdasarkan pengujian yang sudah dilakukan dengan menggunakan *whitebox* didapatkan hasil bahwa sistem berjalan dengan baik karena pengujian *cyclomatic complexity* menghasilkan nilai yang sama yaitu 4. Penggunaan algoritma dapat membantu mengamankan data yang berkaitan dengan data keuangan sehingga tidak dapat dimanfaatkan oleh orang yang tidak bertanggung jawab. Dengan aplikasi pencatatan pembayaran dapat membantu dalam mengelola dan menginformasikan info tagihan kepada pelanggan dan juga dapat mengamankan data pembayaran.

DAFTAR PUSTAKA

[1] H. T. S. Alrikabi and H. T. Hazim, "Enhanced Data Security of Communication System Using Combined Encryption and Steganography," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 16, pp. 144–157, 2021, doi: 10.3991/ijim.v15i16.24557.

[2] J. Zhu *et al.*, "Enabling rack-scale confidential computing using heterogeneous trusted execution environment," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2020-May, pp. 1450–1465, 2020, doi: 10.1109/SP40000.2020.00054.

[3] C. Sharma, S. . Jain, and A. K. Sharma, "Explorative Study of SQL Injection Attacks and Mechanisms to Secure Web Application Database- A Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 3, pp. 79–87, 2016, doi: 10.14569/ijacsa.2016.070312.

[4] R. Stofberg, C. M. Mabaso, and M. H. R. Bussin, "Employee responses to pay transparency," *SA J. Ind. Psychol.*, vol. 48, pp. 1–12, 2022, doi: 10.4102/sajip.v48i0.1906.

[5] W. Przychodzen and F. Gómez-Bezares, "CEO–Employee Pay Gap, Productivity and Value Creation," *J. Risk Financ. Manag.*, vol. 14, no. 5, 2021, doi: 10.3390/jrfm14050196.

[6] R. Aulia, A. Zakir, and D. A. Purwanto, "Penerapan Kombinasi Algoritma Base64 Dan Rot47 Untuk Enkripsi Database Pasien Rumah Sakit Jiwa Prof. Dr. Muhammad Ildrem," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 2, no. 2, pp. 146–151, 2018, doi: 10.30743/infotekjar.v2i2.300.

[7] C. Wadisman, I. Nozomi, and S. Rahmawati, "PENGAMANAN DATABASE MENGGUNAKAN KOMBINASI ALGORITMA (CEST CRYPTOGRAPHY) DAN ALGORiTMA BASE64," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 7, no. 1, pp. 33–38, 2020, doi: 10.33330/jurteksi.v7i1.896.

[8] E. Setyaningsih, "Keamanan file dokumen menggunakan algoritme Advanced Encryption Standard pada aplikasi berbasis Android," *Jnanaloka*, pp. 11–23, 2020, doi: 10.36802/jnanaloka.2020.v1-no1-13.

[9] P. Studi Teknik, K. Kunci, and P. Korespondensi, "Aplikasi Perpaduan Enkripsi Base64 Dengan Metode Steganografi Distrete Cosine Transform (Dct)," *J. Sintaks Log.*, vol. 2, no. 2, pp. 37–45, 2022.