Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

PERANCANGAN SISTEM PEMBAYARAN JALAN TOL BERBASIS *ALGORITMA RIJNDAEL*

Design Of Toll Road Payment System Based On Rijndael Algorithm

Arief Suyono [1], Ngarap Imanuel Manik[2*]
Computer Science–Applied Mathematics, Bina Nusantara University[1, 2]
Jln. Kebon Jeruk Raya No.27, Jakarta 11480, Indonesia

Corresponding author : Tel: 0812-1104453
Corresponding author e-mail: manik@binus.ac.id

Received: June 06, 2025**. Revised:** June 28, 2025**. Accepted:** July 2, 2025 **. Issue Period:** Vol.9 No.3 (2025), Pp. 1109-1117

Abstrak: Untuk meningkatkan pelayanan pembayaran karcis jalan tol, diusulkan pembayaran karcis menggunakan program komputer, sehingga dapat mempercepat dan mengefisienkan proses pembayaran. Sehubungan dengan hal di atas maka dirancang sebuah program simulasi pembayaran jalan tol dengan algoritma kriptografi–Rijndael. Algoritma ini dipilih karena mempunyai sistem keamanan yang cukup baik dan tidak rumit untuk diterapkan. Makalah ini membahas tentang penerapan algoritma Rijndael pada sistem pembayaran jalan tol. Pada aplikasi yang dirancang akan mencatat transaksi-transaksi yang dilakukan oleh pengguna jalan tol ke dalam database. Dengan adanya program aplikasi ini pembayaran biaya jalan tol menjadi lebih efisien.

Kata kunci: : *Algoritma rijndael, pembayaran jalan tol, program simulasi*

Abstract: *To improve toll road ticket payment services, it is proposed to use a computer program to pay for tickets, so as to speed up and streamline the payment process. In connection with the above, a toll road payment simulation program was designed using the cryptographic algorithm – Rijndael. This algorithm was chosen because it has a fairly good security system and is not complicated to implement. This paper discusses the application of the Rijndael algorithm to the toll road payment system. The designed application will record transactions made by toll road users into a database. With this application program, the payment of toll road fees becomes more efficient.*

Keywords: *Rijndael algorithm, toll road payment, simulation program*

I. PENDAHULUAN

Sarana transportasi adalah salah satu urat nadi negara. Adanya sarana transportasi yang baik akan turut mendukung perekonomian negara. Karena itu sarana transportasi harus diperhatikan dalam pembangunan. Jalan tol merupakan salah satu sarana transportasi yang terdapat di Indonesia. Jalan tol ini sudah merupakan kebutuhan untuk melayani jaringan transportasi yang berkembang pesat, terutama di kota-kota besar. Tidak hanya itu, jalan tol juga merupakan faktor penting dalam hubungan antar kota ataupun wilayah di Indonesia. Bukan hanya sebagai sarana transportasi melainkan juga sebagai sumbangan devisa terhadap Negara maupun Pemerintah Daerah.

DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Dalam pengelolaannya, pelayanan jalan tol ini harus diperhatikan karena hal itu merupakan faktor yang signifikan. Pelayanan jalan tol yang umum ditemui di Indonesia adalah seperti derek gratis dalam tol sampai bengkel terdekat, wc umum, informasi kepadatan lalu lintas, dan hal-hal semacam itu. Pengguna jalan tol sendiri dibagi ke dalam tiga kategori yaitu kendaraan pribadi, kendaraan umum, serta kendaraan yang digunakan sebagai alat angkut. Pembagian kelas pengguna ini dimaksudkan untuk pembedaan dari besarnya penarikan uang tol.

Dalam hal ini, pelayanan oleh pihak jalan tol dilakukan secara manual. Yaitu dengan menggunakan kasir-kasir di pintu keluar-masuk tol. Dengan semakin majunya teknologi, cara ini semakin tidak efisien. Beberapa faktor yang cukup menghambat saat pembayaran jalan tol yaitu penyediaan uang kecil oleh kasir tol, belum lagi antrian mobil untuk menunggu giliran membayar tol. Faktor ini tidak cukup signifikan, karena hanya pada pintu-pintu tol tertentu terjadi antrian yang cukup panjang.

Hal lain yang lebih mempengaruhi ialah kejujuran kasir dalam memberikan karcis tol. Sering kali sewaktu melewati pintu tol, karcis tol ini tidak diberikan oleh kasir. Mungkin dengan anggapan bahwa banyak orang yang tidak peduli, tetapi untuk pengelola jalan tol hal ini sangat merugikan karena karcis tersebut dapat digunakan untuk kendaraan lain yang melewati pintu tol tersebut.

Dengan semakin majunya teknologi, hal-hal seperti diatas diharapkan tidak lagi terjadi. Hal ini dapat dimungkinkan dengan adanya smartcard. Contoh smartcard yang paling umum yaitu kartu kredit ataupun kartu telepon. Smartcard ini dapat menggantikan tugas kasir untuk menarik biaya penggunaan jalan tol kepada pengemudi. Aplikasi yang dibangun untuk membaca smartcard ini biasanya menggunakan algoritma enkripsi. Pada teknologi ini penggunaan enkripsi mutlak dibutuhkan. Hal ini dikarenakan smartcard dapat menyimpan informasi mengenai pemilik smartcard tersebut dan bila sampai digunakan oleh orang lain, maka pemilik smartcard dapat dirugikan.

Ada banyak algoritma dalam enkripsi. Pada tulisan ini algoritma yang akan digunakan ialah Advanced Encryption Standard (AES). Algoritma ini merupakan penyempurnaan dari algoritma-algoritma yang sudah ada sebelumnya. Dibandingkan dengan algoritma pendahulunya, DES, AES lebih cepat dan mudah diterapkan baik pada software maupun hardware, dan hanya membutuhkan memory yang kecil [1][2]. Untuk memberikan pelayanan yang semakin baik dari hari ke hari, perusahaan pengelola jalan tol harus terus melakukan inovasi dalam hal pelayanan. Inovasi ini dapat berupa kecepatan, maupun kenyamanan dalam bertransaksi. Tetapi selain kecepatan dan kenyamanan dibutuhkan pula keamanan dalam bertransaksi.[3]

Diharapkan dengan adanya program aplikasi yang telah dibuat dapat membantu pihak pengelola jalan tol untuk dapat melayani pengguna jalan tol dengan lebih cepat, baik, sekaligus aman. Program aplikasi yang dibangun ini akan menggunakan input data yang di-enkripsi menggunakan algoritma Advanced Encryption Standard (AES), yang disebut juga dengan Rijndael. Input data inilah yang nantinya akan digunakan pada program aplikasi pembayaran jalan tol secara otomatis.[4]

II. METODE DAN MATERI

II.1 ALGORITMA AES(*Advanced Encryption Standard*) RIJNDAEL

Algoritma ini merupakan *block cipher* yang banyak digunakan sebagai standar enkripsi. Algoritma Rijndael menggunakan *substitution-permutation network* yang pengimplementasiannya cukup mudah dan hanya membutuhkan memori yang kecil. Sebagai suatu standar enkripsi, AES sekarang sedang dikembangkan dalam skala yang luas.

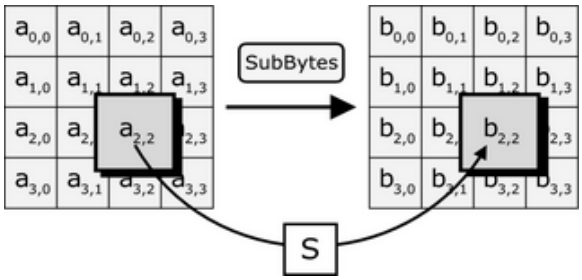
AES ini menggunakan ukuran *block* 128-bit dan mempunyai ukuran *key* 128, 192, dan 256-bit. AES beroperasi menggunakan *array* 4×4 byte yang disebut *state*. [5]

DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

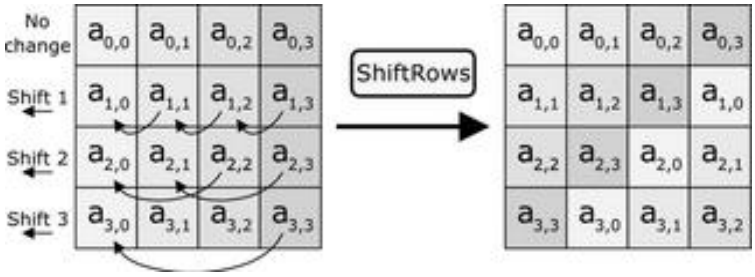
Untuk enkripsi, setiap *round* dari AES (kecuali *round* terakhir) terdiri dari empat tahap:

1. ***SubBytes*** : Pada langkah ini, setiap *byte* pada *array* ditukar menggunakan *S-box* 8 bit. Operasi ini mengakibatkan enkripsi ini tidak linear. *S-box* yang digunakan diturunkan oleh fungsi *invers* dari fungsi polinomial 2^8 yang dikenal tidak linear. Untuk menghindari serangan yang berdasarkan fungsi aljabar sederhana, *S-box* ini dibentuk dengan mengkombinasikan fungsi invers dengan transformasi *affine* yang tidak dapat diubah kembali. *S-box* juga dipilih untuk menghindari ditemukannya titik tertentu dan juga kebalikan dari titik tersebut, seperti pada gambar 1.



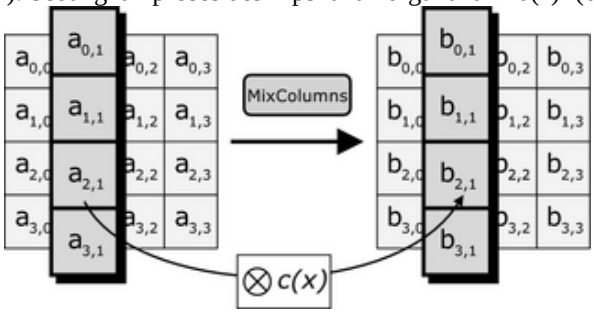
Gambar 1. langkah *SubBytes*

2. ***ShiftRows*** : Langkah ini dilakukan pada baris dari tiap *state*, langkah ini menggeser tiap *byte* pada tiap baris dengan *offset* tertentu. Pada AES, baris awal tidak di ubah. Setiap *byte* dari baris kedua dipindah ke kiri satu. Begitu pula dengan baris ketiga dan keempat, digeser pula sesuai dengan *offset* kedua dan ketiganya. Dengan cara ini, setiap kolom dari state output terdiri dari byte yang berasal dari kolom state input yang ditunjukkan dalam gambar 2.[6]



Gambar 2. langkah *ShiftRows*

3. ***MixColumns*** : Pada langkah ini, keempat *byte* pada tiap kolom dari *state* dikombinasikan menggunakan transformasi linear yang *invertible*. Fungsi ini menggunakan masukan empat *byte* dan keluaran empat *byte*, dimana tiap masukan *byte* mempengaruhi keempat *byte* keluaran. *MixColumns* yang digunakan bersama *ShiftRows* menghasilkan keacakan pada cipher. Tiap kolom diperlakukan sebagai suatu polinomial 2^8 dan kemudian dioperasikan dengan *modulo* $x^4 + 1$ dengan suatu fungsi polinomial tertentu $c(x)$. Langkah ini dapat dilihat pula sebagai suatu matrix multiplikatif pada *Rijndael's finite field*. $c(x)$ yang digunakan untuk enkripsi ialah $c(x)=(03)x^3+(01)x^2+(01)x+(02)$. Sedangkan proses deskripsi akan digunakan $c(x)=(0B)x^3+(0D)x^2+(09)x+(0E)$.

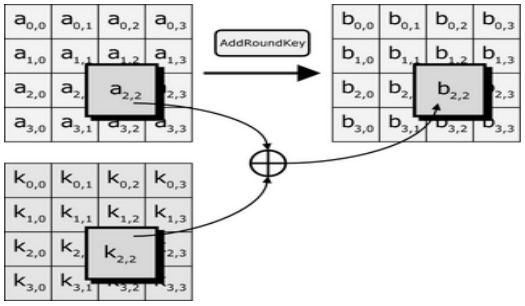


DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Gambar 3. langkah *MixColumns*

4. ***AddRoundKey*** : Pada langkah ini, *subkey*-nya dikombinasikan dengan tiap state. Untuk tiap giliran, satu *subkey* diturunkan dari *key* utama menggunakan *key schedule*. Tiap *subkey* mepunyai ukuran yang sama dengan *state*. Tiap *subkey* kemudian ditambahkan dengan cara menggabungkan tiap *byte* pada *state* dengan *byte* yang berhubungan pada *subkey* menggunakan operasi XOR.



Gambar 4. langkah *AddRoundKey Round* terakhir melakukan proses *MixColumns*.

***S-Box* Rijndael**

Nilai S-box yang digunakan pada algoritma Rijndael ditunjukkan dengan nilai hexadesimal berikut ini:

Tabel 1. *S-Box* Rijndael

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Di sini kolom akan ditentukan berdasarkan digit yang paling tidak penting, dan barisnya ditentukan dengan digit utama. Contohnya nilai `0x9a` diubah menjadi `0xb8` dengan menggunakan *S-box* Rijndael.
Invers *S-box* adalah *S-box* yang dijalankan secara terbalik. Contohnya, invers *S-box* dari `0xdb` adalah `0x9f`. Tabel 2. berikut merepresentasikan invers dari *S-box* Rijndael:

Tabel 2. Invers *S-Box* Rijndael

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb

DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

10 |7c e3 39 82 9b 2f ff 87 34 8e 43 44 c4 de e9 cb
20 |54 7b 94 32 a6 c2 23 3d ee 4c 95 0b 42 fa c3 4e
30 |08 2e a1 66 28 d9 24 b2 76 5b a2 49 6d 8b d1 25
40 |72 f8 f6 64 86 68 98 16 d4 a4 5c cc 5d 65 b6 92
50 |6c 70 48 50 fd ed b9 da 5e 15 46 57 a7 8d 9d 84
60 |90 d8 ab 00 8c bc d3 0a f7 e4 58 05 b8 b3 45 06
70 |d0 2c 1e 8f ca 3f 0f 02 c1 af bd 03 01 13 8a 6b
80 |3a 91 11 41 4f 67 dc ea 97 f2 cf ce f0 b4 e6 73
90 |96 ac 74 22 e7 ad 35 85 e2 f9 37 e8 1c 75 df 6e
a0 |47 f1 1a 71 1d 29 c5 89 6f b7 62 0e aa 18 be 1b
b0 |fc 56 3e 4b c6 d2 79 20 9a db c0 fe 78 cd 5a f4
c0 |1f dd a8 33 88 07 c7 31 b1 12 10 59 27 80 ec 5f
d0 |60 51 7f a9 19 b5 4a 0d 2d e5 7a 9f 93 c9 9c ef
e0 |a0 e0 3b 4d ae 2a f5 b0 c8 eb bb 3c 83 53 99 61
f0 |17 2b 04 7e ba 77 d6 26 e1 69 14 63 55 21 0c 7d

II.2 Finite Field Arithmetic

Aritmetika pada *finite field* mempunyai perbedaan dengan aritmetika yang biasa. Semua operasi pada *finite field* mempunyai hasil perhitungan yang berada dalam *finite field* tersebut. Walaupun elemen-elemen pada *finite field* dapat dinyatakan dalam bentuk numerik (desimal, heksadesimal, ataupun biner), seringkali lebih mudah dinyatakan dalam bentuk polinomial, dengan setiap variabel pada polinomialnya mewakili bit-bit pada nilai binernya. [7]

Contoh dibawah ini merepresentasikan nilai yang sama pada *finite field* dengan karakteristik 2:

Heksadesimal : {53}
Biner : {01010011}
Polinomial : $x^6 + x^4 + x + 1$

Eksponen pada polinomial digunakan sebagai penanda yang memungkinkan untuk mengetahui nilai setiap bit pada saat terjadi operasi aritmetika. Jika yang digunakan adalah nilai heksadesimal atau biner maka digunakan tanda kurung (’{‘ dan ‘}’) yang menunjukkan bahwa nilai tersebut adalah elemen dari *finite field.*

Penambahan dan pengurangan dilakukan menggunakan dua polinomial. Setiap variabel pada polinomial hanya boleh memiliki nilai satu atau nol. Pada *finite field* dengan karakteristik 2, penambahan dan pengurangan sama, dan keduanya dilakukan dengan menggunakan operator **XOR**. Contohnya:

Heksadesimal : {53} + {CA} = {99}
Biner : {01010011} + {11001010} = {10011001}
Polinomial : $(x^6 + x^4 + x + 1) + (x^7 + x^6 + x^3 + x) = x^7 + x^4 + x^3 + 1$

Pada kedua nilai diatas terdapat x^6, dan $x^6 + x^6$ menjadi $2x^6$. Tetapi karena setiap koefisien harus di*modulo* 2 maka hasilnya akan menjadi $0x^6$ sehingga nilai tersebut dibuang.

Perkalian pada *finite field* ialah perkalian yang di*modulo* dengan *irreducible polynomial* yang digunakan untuk mendefinisikan *finite field* tersebut. Maksudnya, perkalian tersebut diikuti dengan pembagian yang menggunakan *irreducible polynomial* sebagai pembagi, sisa hasil bagi tersebut ialah hasilnya. Contoh:

jika *irreducible polynomial* yang digunakan
$f(x) = x^8 + x^4 + x^3 + x + 1$
(*irreducable polynomial* yang digunakan pada algoritma Rijndael),
maka
{53} • {CA} = {01}
$(x^6 + x^4 + x + 1)(x^7 + x^6 + x^3 + x)$
$= x^{13} + x^{12} + x^9 + \cancel{x^7} + x^{11} + x^{10} + \cancel{x^7} + x^5 + x^8 + \cancel{x^7} + x^4 + x^2 + \cancel{x^7} + x^6 + x^3 + x$
$= x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$

DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

dan

$x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$ *modulo*
$x^8 + x^4 + x^3 + x + 1 =$ (11111101111110 mod 100011011) = 1,

yang ditunjukkan dengan pembagian bersusun dibawah ini (menggunakan notasi biner untuk mempermudah):

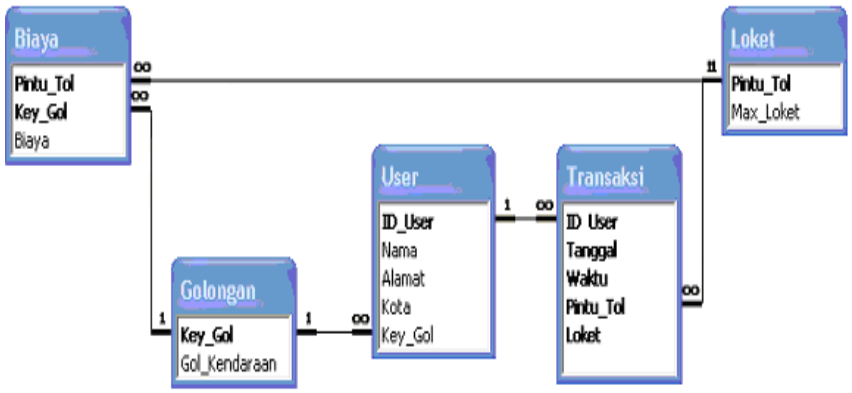
```
       111101
100011011)11111101111110
       100011011
        1110000011110
        100011011
         110110101110
         100011011
          10101110110
          100011011
           0100011010
           000000000
            100011010
            100011011
             00000001
```

III. PEMBAHASAN DAN HASIL

Program aplikasi yang dirancang ini akan menggunakan *database* sebagai sumber data yang dibutuhkan. *Database* tersebut akan berisi data pintu tol yang ada, data *user* yang terdaftar, dan data transaksi yang terjadi. Sedangkan untuk program aplikasinya sendiri akan terbagi menjadi dua, yaitu *server* dan *client*.

Aplikasi *Server* akan bertugas untuk me-*routing* data yang diterima untuk dicatat ke dalam *database* ataupun mengecek data *user*. Aplikasi ini hanya akan mempunyai dua tombol utama yaitu untuk mengaktifkan dan menon-aktifkan kondisi *server*.

Aplikasi *Client* bertugas untuk mengambil data dari *user*, mengolahnya dan kemudian mengirimkan data tersebut ke aplikasi *Server*. Aplikasi ini harus dikenali dahulu oleh *server* untuk dapat menjalankan fungsinya. Untuk itu disediakan dua tombol, yaitu untuk menjalin koneksi ke *server* dan memutuskan koneksi tersebut. Selain itu terdapat pula satu tombol lain yang berguna untuk mengolah data *user* dan mengirimkan data tersebut ke *server*. Gambar5 berikut menunjukkan Entity Relation Diagram Database Tol [8][9]



DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Gambar 5. Entity Relation Diagram Database Tol

Untuk proses pengiriman data user pada aplikasi *client*, diperlukan data aktual yang terdapat pada *database* telah dienkripsi. Data tersebut dibentuk dengan format:

AAAAAAAAAAAAAAA#HHHHHHHHHHHHH?A@

Keterangan:

A mewakili angka (‘0’-‘9’) dan

H mewakili huruf ('A' - 'Z' dan 'a' - 'z'), ‘ ‘, dan ‘$’

Lima belas digit pertama digunakan untuk tempat nomor ID_user diikuti dengan ‘#’ pada digit keenam belas. Tiga belas digit berikutnya adalah huruf yang berisi nama useryaitu pada digit ke-17 hingga 29. Bila nama user tidak sampai 13 digit, maka diisi dengan ‘$’ untuk menggenapinya menjadi tiga belas digit, bila nama *user* lebih dari tiga belas digit, hanya akan diambil tiga belas digit awalnya saja. Digit digit ke-30 diisi dengan ‘?’ diikuti angka yang berisi tipe golongan kendaraan pada digit ke-31 dan ditutup dengan ‘@’ pada digit ke-32.

A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	#	H	H	H	H	H
0									1										2	
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

H	H	H	H	H	H	H	H	?	A	@
								3		
2	3	4	5	6	7	8	9	0	1	2

contoh: 000000000000001#Arif Suryono$?1@

Setelah data awal disusun dengan format tersebut maka data tersebut dienkripsi. Data yang telah dienkripsi inilah yang nantinya digunakan pada kolom ‘Input Data dari Kartu’ pada form client. Data ini kemudian akan didekripsi sebelum dikirimkan ke aplikasi server untuk dicatat sebagai suatu transaksi.
Untuk menambah *user* yang dapat melakukan transaksi harus terlebih dahulu menambahkan data kedalam *database* dan kemudian membuat data ‘input data dari kartu’ untuk *user* yang bersangkutan.

Cara Pengoperasian Program

Program aplikasi ini dibagi ke dalam dua bagian utama, yaitu *server* dan *client*. Cara mengoperasikannya adalah sebagai berikut:

- Langkah 1 – jalankan aplikasi *server*.

Tekan tombol [Activate] pada aplikasi server untuk mengaktifkan *server*. Kemudian akan ditampilkan tampilan layar seperti gambar 6.

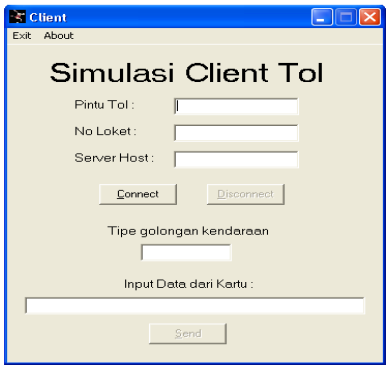
Gambar 6. Tampilan layar langkah 1

- Langkah 2 – jalankan aplikasi *client.*

DOI: 10.52362/jisamar.v9i3.1950

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Masukkan nama pintu tol, nomor loket tol dan *serverhost* pada *box* yang ada. Tekan tombol [connect] untuk berinteraksi dengan aplikasi *server*. Kemudian akan terlihat tampilan seperti gambar 7.



Gambar 7. Tampilan layar langkah 2

- Langkah 3 – mulai transaksi

User dapat mulai melakukan transaksi dengan cara memasukkan 'tipe golongan kendaraan' dan 'input data dari kartu' (data terenkripsi). Untuk bertransaksi dengan aplikasi *server*, tekan tombol [Send]. Data akan didekripsi pada aplikasi *client*, dan kemudian dikirimkan ke aplikasi *server*. Transaksi akan dicatat pada *database* seperti pada tampilan gambar 8 berikut.

Gambar 8. Tampilan layar langkah 3

IV. KESIMPULAN

Berdasarkan hasil rancangan program server-client, analisis dan evaluasi terhadap aplikasi program maka dapat disimpulkan : Program ini mampu melakukan dekripsi algoritma rijndael sesuai dengan batasan-batasan yang telah ditentukan sebelumnya. Demikian juga dari beberapa contoh-contoh kasus yang diberikan, program ini cukup mampu mendeteksi adanya kesalahan, baik pada saat memasukkan input maupun saat validasi ke absahan input data yang terhubung ke database. Untuk pengoperasiannya program ini harus terhubung antara aplikasi server-client secara real-time untuk dapat melakukan tansaksi. Transaksi yang terjadi akan dicatat pula secara langsung ke database. Program ini dapat membantu dalam melakukan transaksi pembayaran jalan tol yang dilakukan secara prabayar (membayar sebelum masuk jalan tol, contohnya tol dalam kota).

REFERENSI

[1] Bauer, F. L. (2022), *Decrypted Secrets: Methods and Maxims of Cryptology*, 3rd ed. Springer-

DOI: 10.52362/jisamar.v9i3.1950

Verlag, New York.

[2] C. C. Lu and S. Y. Tseng. (2020) . "Integration of AES (Advanced Encryption Standard) encrypter and decrypter," *Proceeding, Application-Specific Systems, Architecture and Processor*.

[3] Kurniawan, Yusuf (2014) *Kriptografi: Keamanan Internet dan Jaringan Komunikasi*.Informatika, Bandung.

[4] M. A. Hasan, M. Z. Wang and V. K. Bhargava. (2018) . *A Modified Massey-Omura Parallel Multiplier for a Class of Finite Fields.* IEEE Trans. Computers, Vol. 42, No. 10.

[5] National Institute of Standards and Technology.(2020) *AES Algorithm (Rijndael) Information*. http://csrc.nist.gov/CryptoToolkit/aes/

[6] Page, M. - Jones. (2018). *The Practical Guide to Structured Systems Design.* Yourdon Press, New York.

[7] Parker, SP. (2007) *McGraw-Hill Dictionary of Mathematics.* McGraw-Hill Companies, Inc., New York.

[8] Pressman, RS. (2021). *Software Engineering : A Practioner's Approach. (9th ed.)* McGraw-Hill Companies, Inc., Singapore.

[9] Sommerville, I. (2022). *Software Engineering, (10th ed.)* Addison Wesley Publishing Limited, Harlow.

[10] Stallings, William. (2015) . *Network and Internetwork Security Priciples and Practice,*Prentice Hall, Englewood Cliffs, New Jersey 07632.

[11] Trape, Wade & Lawrence C. Washington (2022), *Introduction to Cryptography with Coding Theory*, Prentice Hall Inc., New Jersey 07458.

DOI: 10.52362/jisamar.v9i3.1950