BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Masalah yang dibahas dalam penelitian ini adalah cara mengamankan script html pada alamat laman. Masalah utama yang diambil penulis pada penelitian ini adalah munculnya permasalahan dalam mengamankan informasi yang ada pada script html pada alaman laman sebuat halaman. Oleh karena itu penulis menawarkan solusi berupa aplikasi keamanan script html pada alamat laman menggunakan algoritma AES dan RSA.

III.2. Strategi Pemecahan Masalah

Beberapa strategi pemecahan masalah dalam rancang bangun aplikasi keamanan script html pada alamat laman ini adalah sebagai berikut:

1. Menghasilkan suatu aplikasi yang dapat mengamankan script html pada alamat laman.
2. Menghasilkan suatu aplikasi yang meminimalisir tindak penyalahgunaan script html pada alamat laman.

III.3. Analisa Kebutuhan Sistem

Pembuatan aplikasi ini membutuhkan serangkaian peralatan yang dapat mendukung kelancaran proses rancang bangun aplikasi keamanan script html untuk mengamankan informasi yang ada pada alamat laman, Berikut ini aspek-aspek yang di butuhkan.

III.3.1. Perangkat Keras (*Hardware*)

Hardware merupakan komponen yang terlihat secara fisik, yang saling bekerjasama dalam pengolahan data. Spesifikasi *minimum hardware* yang digunakan adalah sebagai berikut :

a. Laptop : *Core i3 Processor*
b. *Hard disk* : 500 GB

III.3.2. Perangkat Lunak (*Software*)

Software adalah intruksi atau program-program komputer yang dapat digunakan oleh komputer dengan memberikan fungsi serta penampilan yang diinginkan. Dalam hal ini *software* yang digunakan dalam perancangan aplikasi adalah:

a. Sistem operasi *windows* 8
b. *Visual Basic* Net. 2010
c. Sublime Text 3
d. XAMPP

III.4. Penerapan Algoritma AES

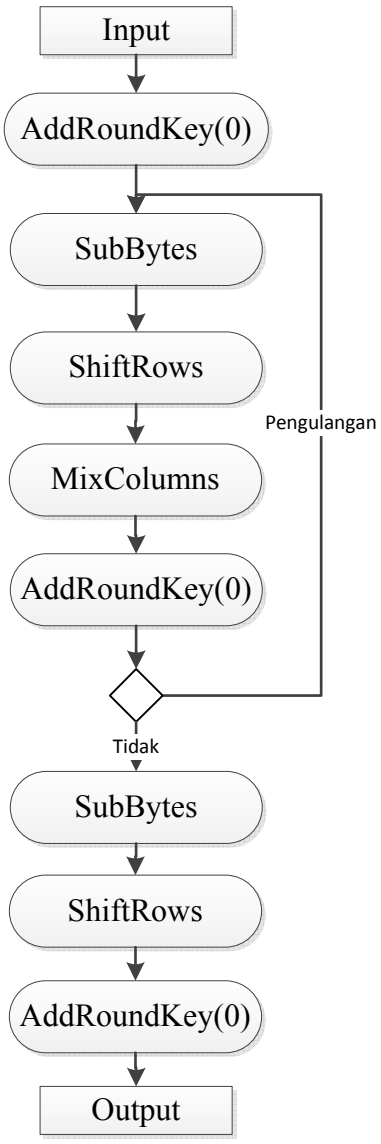
Proses Enkripsi AES 128 Bit : Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran *ciphertext* dinamakan dengan state. Setiap state akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, *AddRoundKey*, *ShiftRows*, dan *MixColumns*. Kecuali tahap ketiga

tahap lainnya akan diulang pada setiap proses sedangkan tahap *MixColumns* tidak akan dilakukan pada tahap terakhir. Berikut ini adalah langkah-langkah penggunaan algoritma AES dalam mengamankan *file* :

1. Siapkan array berukuran 4x4 bernama Kunci
2. Siapkan array berukuran 4x4 bernama State
3. Cetak : “Masukkan 16 bilangan heksadesimal sebagai kunci : “
4. Simpan enam belas nilai tersebut sebagai nilai dari masing-masing elemen array Kunci.
5. Cetak : “Masukkan *file* yang akan dienkripsi : “
6. Konversikan *file* tersebut ke dalam bentuk bit menggunakan kode ASCII.
7. Konversikan kode ASCII tersebut ke dalam heksadesimal
8. Kelompokkan bit-bit teks tersebut menjadi 128 bit tiap bagiannya.
9. Ambil 128 bit pertama untuk diproses.
10. Kelompokkan bit teks tersebut menjadi 16 bagian dengan 8 bit tiap bagiannya.
11. Masukkan tiap-tiap bagian teks tersebut ke dalam tiap-tiap sel pada matriks berukuran 4x4.
12. Konversikan bit ke dalam heksadesimal.
13. Lakukan langkah *AddRoundKey*
14. Lakukan langkah *SubBytes*
15. Lakukan langkah *ShiftRows*
16. Lakukan langkah *MixColumns*
17. Lakukan langkah *AddRoundKey*
18. Ulangi langkah 13-16 sebanyak 9x.
19. Jika langkah 17 sudah dilakukan, maka lakukan langkah *SubByte*
20. Lakukan langkah *ShiftRows*

21. Lakukan langkah *AddRoundKey*

Untuk lebih jelasnya dapat dilihat pada *flowchart* algoritma AES pada gambar III.1.



Gambar III.1. *Flowchart* Algoritma AES

III.4.1. Rumus Penerapan Algoritma AES

Berikut ini adalah contoh kasus penggunaan algoritma AES, untuk lebih jelasnya dapat dilihat sebagai berikut :

Misal, sebuah *file* html dengan nama file sebagai berikut :

File html : **TwoOneNineTwo.html**

Kunci : **ThatsMyKungFu**

Langkah selanjutnya adalah mengubah nama file dari video dan juga kunci kedalam bentuk hexadesimal.

File video :

T	**w**	**O**	**O**	**n**	**E**	**N**	**i**	**N**	**e**	**T**	**w**	**O**
54	**77**	**6F**	**4F**	**6E**	**65**	**4E**	**69**	**6E**	**65**	**54**	**77**	**6F**

Kunci :

T	**H**	**A**	**t**	**s**	**M**	**y**	**K**	**U**	**n**	**G**	**F**	**U**
54	**68**	**61**	**74**	**73**	**4D**	**79**	**4B**	**75**	**6E**	**67**	**46**	**75**

Langkah selanjutnya yang dilakukan adalah mencari nilai Roundkey pertama, untuk mencari roundkey pertama dapat dilakukan sebagai berikut :

w[0] = (54, 68, 61, 74)

w[1] = (73, 20, 6D, 79)

w[2] = (20, 4B, 75, 6E)

w[3] = (67, 20, 46, 75)

g(w[3]) :

- byte kiri bergeser dari w[3] : (20, 46, 75, 67)
- Substitusi Byte (S-Box): (B7, 5A, 9D, 85)
- Menambahkan putaran konstan (01, 00, 00, 00)
- Memberi : g(w[3]) = (B6, 5A, 9D, 85)

w[4] = w[0] g(w[3]) = (E2, 32, FC, F1) :

0101 0100 1011 0110	0110 1000 0101 1010	0110 0001 1001 1101	0111 0100 1000 0101
1110 0010 E2	0011 0010 32	1111 1100 FC	1111 0001 F1

w[5] = w[4] XOR w[1] = (91, 12, 91, 88)

w[6] = w[5] w[2] = (B1, 59, E4, E6)

w[7] = w[6] w[3] = (D6, 79, A2, 93)

Roundkey pertama : E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93

Round0 : 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Round1 : E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93

Round2 : 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA

Round3 : D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB

Round4 : A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B

Round5 : B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69

Round6 : BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E

Round7 : CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A

Round8 : 8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C

Round9 : BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8

Round10 :28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 26

State Matrix dan Round key No.0 Matrix :

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \qquad \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix}$$

XOR entri yang sesuai, misalnya, 69 XOR 4B = 22

$$\begin{array}{r} 0110\ 1001 \\ 0100\ 1011 \\ \hline 0010\ 0010 \end{array}$$

State Matrix baru adalah

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

Gantikan setiap entri (byte) dari matriks keadaan saat ini dengan entri yang sesuai di AES S-Box. Misalnya : byte 6E diganti dengan masuknya S-Box di baris 6 dan kolom E, yaitu, oleh 9F. Ini mengarah ke Matriks Status baru :

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

Lapisan non-linear ini untuk ketahanan terhadap serangan kriptanalisis yang berbeda dan linier. empat baris digeser secara siklis ke kiri dengan offset 0,1, 2, dan 3. State Matrix baru adalah :

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

Langkah pencampuran linier ini menyebabkan difusi bit pada beberapa putaran. Campur Kolom mengalikan matriks tetap terhadap Matriks Status saat ini :

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

Entri BA adalah hasil dari (02x63) XOR (03x2F) XOR (01xAF) XOR (01xA2) :

02 x 63 = 00000010 x 01100011 = 11000110

03 x 2F = (02 x 2F) XOR 2F = (00000010 x 00101111) XOR 00101111 = 01110001

01 x AF = AF = 10101111 dan 01 x A2 = A2 = 10100010

Karenanya :

$$\begin{array}{r} 11000110 \\ 01110001 \\ 10101111 \\ 10100010 \\ \hline 10111010 \end{array}$$

State Matrix dan Round key No.1 Matrix:

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \qquad \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix}$$

XOR menghasilkan Matriks Status baru

$$\begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

Output AES setelah Putaran 1: 58 47 08 8B 15 B6 1C BA 59 D4 E2 E8 CD 39 DF CE

Putaran 2 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} 6A & 59 & CB & BD \\ A0 & 4E & 48 & 12 \\ 30 & 9C & 98 & 9E \\ 3D & F4 & 9B & 8B \end{pmatrix} \qquad \begin{pmatrix} 6A & 59 & CB & BD \\ 4E & 48 & 12 & A0 \\ 98 & 9E & 30 & 9B \\ 8B & 3D & F4 & 9B \end{pmatrix}$$

Putaran 2 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} 15 & C9 & 7F & 9D \\ CE & 4D & 4B & C2 \\ 89 & 71 & BE & 88 \\ 65 & 47 & 97 & CD \end{pmatrix} \qquad \begin{pmatrix} 43 & 0E & 09 & 3D \\ C6 & 57 & 08 & F8 \\ A9 & C0 & EB & 7F \\ 62 & C8 & FE & 37 \end{pmatrix}$$

Putaran 3 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} 1A & AB & 01 & 27 \\ B4 & 5B & 30 & 41 \\ D3 & BA & E9 & D2 \\ AA & E8 & BB & 9A \end{pmatrix} \qquad \begin{pmatrix} 1A & AB & 01 & 27 \\ 5B & 30 & 41 & B4 \\ E9 & D2 & D3 & BA \\ A9 & AA & E8 & BB \end{pmatrix}$$

Putaran 3 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} AA & 65 & FA & 88 \\ 16 & 0C & 05 & 3A \\ 3D & C1 & DE & 2A \\ B3 & 4B & 5A & 0A \end{pmatrix} \qquad \begin{pmatrix} 78 & 70 & 99 & 4B \\ 76 & 76 & 3C & 39 \\ 30 & 7D & 37 & 34 \\ 54 & 23 & 5B & F1 \end{pmatrix}$$

Putaran 4 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} BC & 51 & EE & B3 \\ 38 & 38 & EB & 12 \\ 04 & FF & 9A & 18 \\ 20 & 26 & 39 & A1 \end{pmatrix} \qquad \begin{pmatrix} BC & 51 & EE & B3 \\ 38 & EB & 12 & 38 \\ 9A & 18 & 04 & FF \\ A1 & 20 & 26 & 39 \end{pmatrix}$$

Putaran 4 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} 10 & BC & D3 & F3 \\ D8 & 94 & E0 & E0 \\ 53 & EA & 9E & 25 \\ 24 & 40 & 73 & 7B \end{pmatrix} \qquad \begin{pmatrix} B1 & 08 & 04 & E7 \\ CA & FC & B1 & B2 \\ 51 & 54 & C9 & 6C \\ ED & E1 & D3 & 20 \end{pmatrix}$$

Putaran 5 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} C8 & 30 & F2 & 94 \\ 74 & B0 & C8 & 37 \\ D1 & 20 & DD & 50 \\ 55 & F8 & 66 & B7 \end{pmatrix} \qquad \begin{pmatrix} C8 & 30 & F2 & 94 \\ B0 & C8 & 37 & 74 \\ DD & 50 & D1 & 20 \\ B7 & 55 & F8 & 66 \end{pmatrix}$$

Putaran 5 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} 2A & 26 & 8F & E9 \\ 78 & 1E & 0C & 7A \\ 1B & A7 & 6F & 0A \\ 5B & 62 & 00 & 3F \end{pmatrix} \qquad \begin{pmatrix} 9B & 23 & 5D & 2F \\ 51 & 5F & 1C & 38 \\ 20 & 22 & BD & 91 \\ 68 & F0 & 32 & 56 \end{pmatrix}$$

Putaran 6 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} 14 & 26 & 4C & 15 \\ D1 & CF & 9C & 07 \\ B7 & 93 & 7A & 81 \\ 45 & 8C & 23 & B1 \end{pmatrix} \qquad \begin{pmatrix} 14 & 26 & 4C & 15 \\ CF & 9C & 07 & D1 \\ 7A & 81 & B7 & 93 \\ B1 & 45 & 8C & 23 \end{pmatrix}$$

Putaran 6 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} A9 & 37 & AA & F2 \\ AE & D8 & 0C & 21 \\ E7 & 6C & B1 & 9C \\ F0 & FD & 67 & 3B \end{pmatrix} \qquad \begin{pmatrix} 14 & 8F & C0 & 5E \\ 93 & A4 & 60 & 0F \\ 25 & 2B & 24 & 92 \\ 77 & E8 & 40 & 75 \end{pmatrix}$$

Putaran 7 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} FA & 73 & BA & 58 \\ DC & 49 & D0 & 76 \\ 3F & F1 & 36 & 4F \\ F5 & 9B & 09 & 9D \end{pmatrix} \qquad \begin{pmatrix} FA & 73 & BA & 58 \\ 49 & D0 & 76 & DC \\ 36 & 4F & 3F & F1 \\ 9D & F5 & 9B & 09 \end{pmatrix}$$

Putaran 7 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} 9F & 37 & 51 & 37 \\ AF & EC & 8C & FA \\ 63 & 39 & 04 & 66 \\ 4B & FB & B1 & D7 \end{pmatrix} \qquad \begin{pmatrix} 53 & 43 & 4F & 85 \\ 39 & 06 & 0A & 52 \\ 8E & 93 & 3B & 57 \\ 5D & F8 & 95 & BD \end{pmatrix}$$

Putaran 8 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} ED & 1A & 84 & 97 \\ 12 & 6F & 67 & 00 \\ 19 & DC & E2 & 5B \\ 4C & 41 & 2A & 7A \end{pmatrix} \qquad \begin{pmatrix} ED & 1A & 84 & 97 \\ 6F & 67 & 00 & 12 \\ E2 & 5B & 19 & DC \\ 7A & 4C & 41 & 2A \end{pmatrix}$$

Putaran 8 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} E8 & 8A & 4B & F5 \\ 74 & 75 & EE & E6 \\ D3 & 1F & 75 & 58 \\ 55 & 8A & 0C & 38 \end{pmatrix} \qquad \begin{pmatrix} 66 & 70 & AF & A3 \\ 25 & CE & D3 & 73 \\ 3C & 5A & 0F & 13 \\ 74 & A8 & 0A & 54 \end{pmatrix}$$

Putaran 9 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} 33 & 51 & 79 & 0A \\ 3F & 8B & 66 & 8F \\ EB & BE & 76 & 7D \\ 92 & C2 & 67 & 20 \end{pmatrix} \qquad \begin{pmatrix} 33 & 51 & 79 & 0A \\ 8B & 66 & 8F & 3F \\ 76 & 7D & EB & BE \\ 20 & 92 & C2 & 67 \end{pmatrix}$$

Putaran 9 setelah kolom Mix dan setelah Roundkey:

$$\begin{pmatrix} B6 & E7 & 51 & 8C \\ 84 & 88 & 98 & CA \\ 34 & 60 & 66 & FB \\ E8 & D7 & 70 & 51 \end{pmatrix} \qquad \begin{pmatrix} 09 & A2 & F0 & 7B \\ 66 & D1 & FC & 3B \\ 8B & 9A & E6 & 30 \\ 78 & 65 & C4 & 89 \end{pmatrix}$$

Putaran 10 setelah Pengganti Byte dan setelah Shift Rows:

$$\begin{pmatrix} 01 & 3A & 8C & 21 \\ 33 & 3E & B0 & E2 \\ 3D & B8 & 8E & 04 \\ BC & 4D & 1C & A7 \end{pmatrix} \qquad \begin{pmatrix} 01 & 3A & 8C & 21 \\ 3E & B0 & E2 & 33 \\ 8E & 04 & 3D & B8 \\ A7 & BC & 4D & 1C \end{pmatrix}$$

Putaran 10 setelah Roundkey (Perhatian: tidak ada kolom Mix di round terakhir):

$$\begin{pmatrix} 29 & 57 & 40 & 1A \\ C3 & 14 & 22 & 02 \\ 50 & 20 & 99 & D7 \\ 5F & F6 & B3 & 3A \end{pmatrix}$$

File video *cipher* yang dihasilkan adalah sebagai berikut :

29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A

III.5. Penerapan Algoritma RSA

RSA menggunakan dua buah bilangan bulat prima untuk mendapatkan *public key* dan *private key* yang akan digunakan dalam proses enkripsi dan dekripsi pesan. RSA digunakan pada aplikasi ini untuk mengenkripsi pesan rahasia yang berupa *file* agar keamanan dari pesan rahasia tadi semakin kuat. Sepasang kunci yang dipakai pada kedua proses ini adalah kunci publik (*e,n*) sebagai kunci enkripsi dan kunci privat *d* sebagai sebagai kunci dekripsi dimana *e*, *d* dan *n* adalah bilangan bulat positif. Algoritma RSA adalah sebuah *block cipher algorithm* (algoritma yang bekerja per blok data) yang mengelompokkan plaintext menjadi blok-blok terlebih dahulu sebelum dilakukan enkripsi hingga menjadi ciphertext.

III.5.1. Rumus Penerapan Algoritma RSA

Enkripsi dan dekripsi dari suatu blok plaintext M dan blok *ciphertext* C , dengan menggunakan persamaan II.3 dan persamaan II.4 maka dihasilkan persamaan baru yaitu :

$M = Cd \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$ (Persamaan III.1)

Untuk mendapatkan hal di atas syarat-syarat yang harus dipenuhi adalah sebagai berikut:

1. Nilai e, d, n dapat dicari, sehingga di dapat $M^{ed} = M \bmod n$ untuk setiap $M < n$.
2. Relatif lebih gampang untuk menghitung M^e dan C^d untuk setiap nilai dari $M < n$.
3. Susah dalam praktek untuk mencari d dengan diberikan e dan n.
4. $ed = k\theta(n) + 1$

Persamaan ini menjadi :

- $e\,d = 1 \bmod \theta n)$
- $d = \frac{1+k\theta(n)}{e}$

Ringkasan dari algoritma RSA adalah sebagai berikut :

Key Generator

- Pilih p,q p dan q prima, $p \neq q$
- Hitung $n = p * q$
- Hitung $\theta(n) = (p - 1)(q - 1)$

- Pilih *integer* e (θ (n),e) = 1; 1 < e < $\theta(n)$
- Hitung d d = $\frac{1+k\theta\ (n)}{e}$
- *Public-key* KU = {e,n}
- *Private-key* KR = {d,n}

Enkripsi :

- *Plaintext* M < n
- *Ciphertext* C = M^e (mod n) Dekripsi
- *Ciphertext* C

Plaintext M = C^d (mod n)

III.6. Desain Sistem

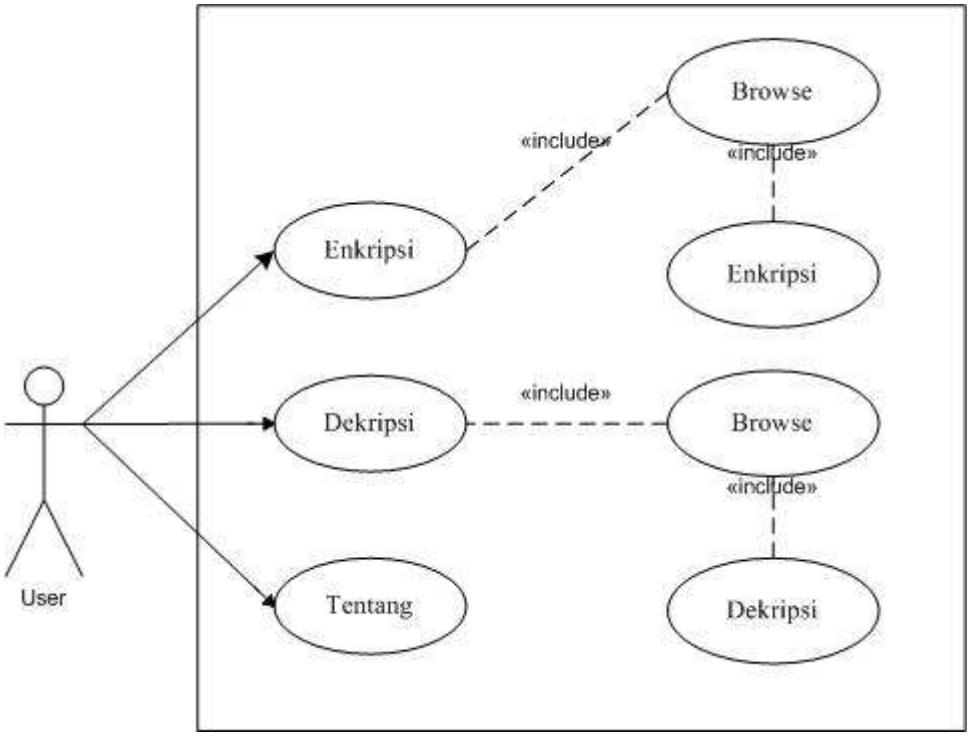
Pada tahap ini dirancang sebuah desain dari perancangan sistem rekomendasi keamanan. Bagaimana desain yang akan digunakan pada antarmuka perangkat berbasis *desktop*. Perancangan sistem yang dirancang terdiri dari *use case*, *activity diagram* serta desain dan penjelasan dari sistem yang dirancang. Berikut adalah perancangannya :

III.6.1. *Use Case Diagram*

Use case mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi yang ada didalam sistem informasi tersebut. Berikut adalah *use case diagram* dari sistem yang dirancang :

III.6.1.1. *Use Case Diagram* Sistem Berbasis *Desktop*

Berikut adalah *use case diagram* dari sistem berbasis *desktop* untuk pengguna :



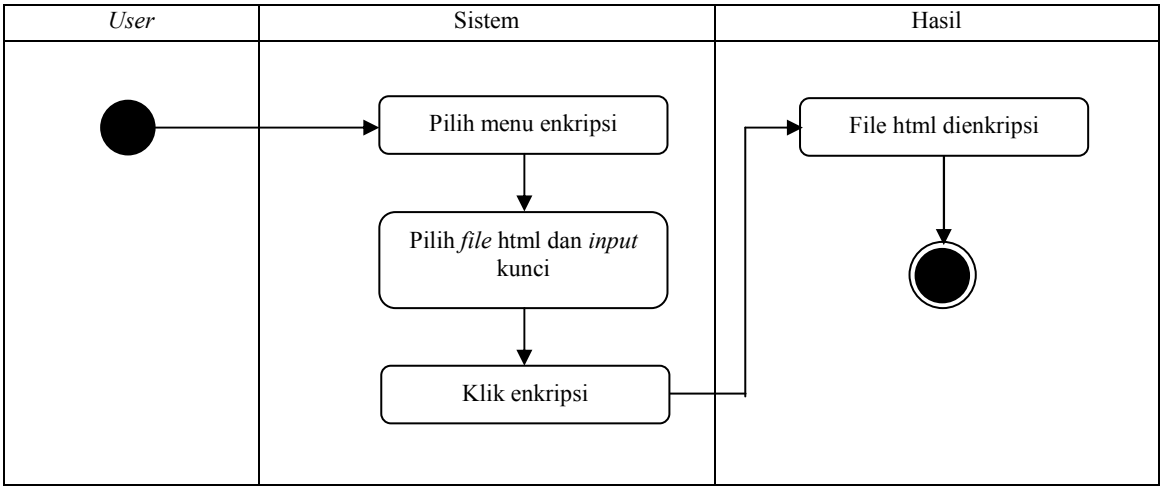
Gambar III.2. *Use Case Diagram* Aplikasi Keamanan Script HTML Berbasis *Desktop*

III.6.2. *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* yang terdapat pada aplikasi yaitu sebagai berikut :

III.6.2.1. *Activity Diagram* Enkripsi

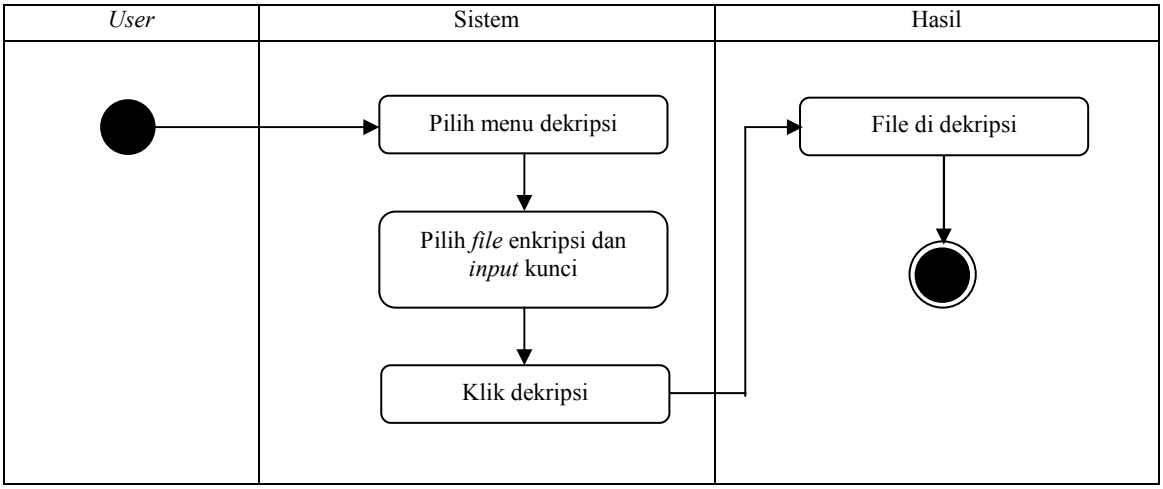
Activity diagram enkripsi menggambarkan alir aktifitas pengamanan *file* html yang dilakukan oleh pengguna dan diproses di dalam sistem. Proses enkripsi *file* html dapat dilihat pada gambar III.4.



Gambar III.3. *Activity Diagram* Enkripsi

III.6.2.2. *Activity Diagram* Dekripsi

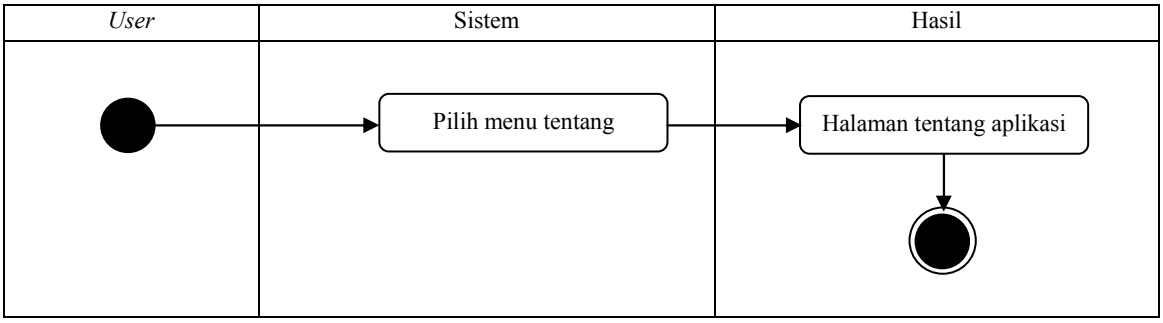
Activity diagram dekripsi menggambarkan alir aktifitas dalam melakukan proses dekripsi *file* html yang telah di enkripsi. *Activity diagram* dekripsi dapat dilihat pada gambar III.4.



Gambar III.4. *Activity Diagram* Dekripsi

III.6.2.3. *Activity Diagram* Tentang

Activity diagram tentang menggambarkan alir aktifitas dalam melakukan proses pemilihan menu tentang pada aplikasi. *Activity diagram* tentang dapat dilihat pada gambar III.5.



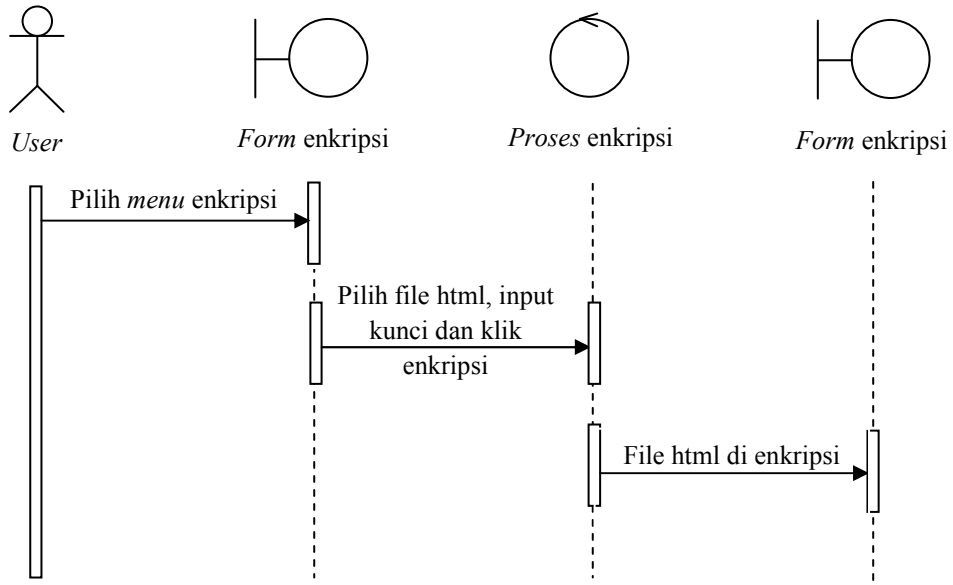
Gambar III.5. *Activity Diagram* Tentang

III.6.3. *Sequence Diagram*

Sequence diagram pada aplikasi yang akan dibuat yaitu : *Sequence diagram* enkripsi, dekripsi dan tentang.

III.6.3.1. *Sequence Diagram* Enkripsi

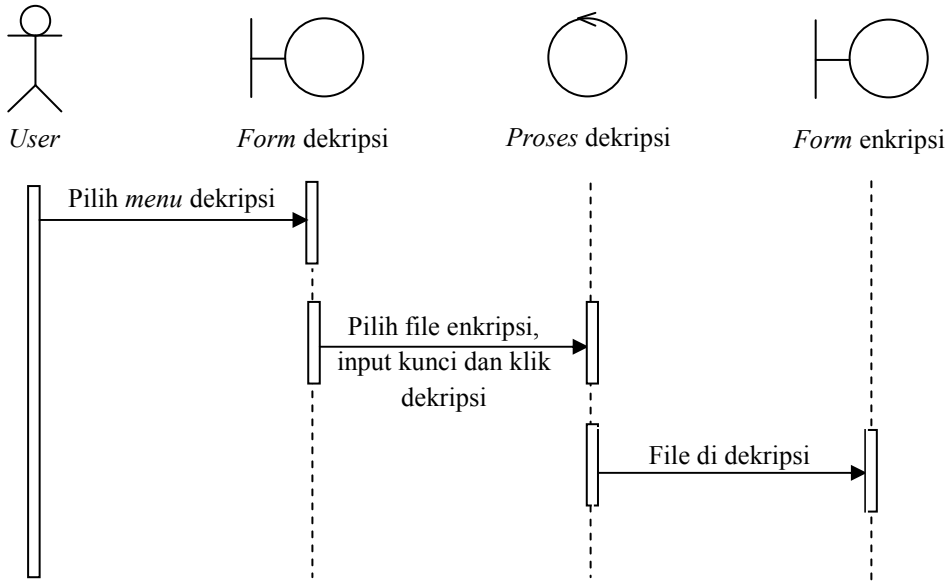
Sequence diagram enkripsi menggambarkan interaksi yang terjadi pada saat melakukan proses enkripsi *file* html. *Sequence diagram* enkripsi ditunjukkan pada gambar III.6.



Gambar III.6. *Sequence Diagram* Enkripsi

III.6.3.2. *Sequence Diagram* Dekripsi

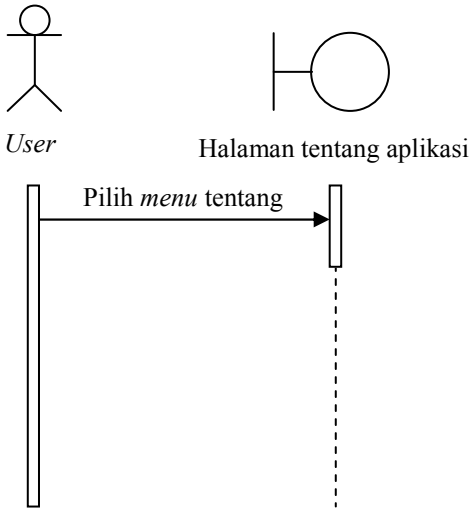
Sequence diagram dekripsi menggambarkan interaksi yang terjadi pada saat melakukan proses dekripsi *file* html. *Sequence diagram* dekripsi ditunjukkan pada gambar III.7.



Gambar III.7. *Sequence Diagram* Dekripsi

III.6.3.4. *Sequence Diagram* Tentang

Sequence diagram tentang berisikan informasi tentang aplikasi yang dibuat. *Sequence diagram* tentang ditunjukkan pada gambar III.8.

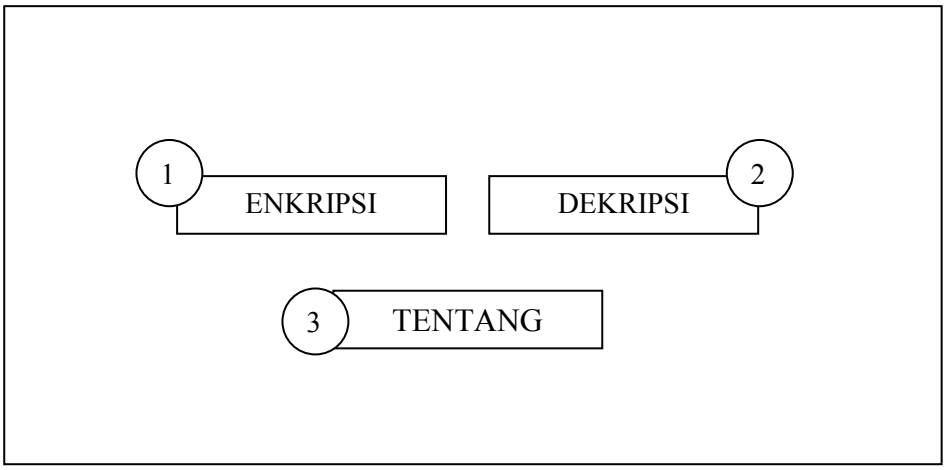


Gambar III.8. *Sequence Diagram* Tentang

III.7. Desain *User Interface*

Antarmuka peamakai (*user interface*) adalah tampilan program yang dapat dilihat, didengar atau dipersepsikan oleh pengguna dan perintah-perintah atau mekanisme yang digunakan pemakai untuk mengendalikan operasi dan memasukkan data. Berikut ini merupakan perancangan antarmuka dari Aplikasi keamanan Script HTML pada alamat laman menggunakan algoritma AES dan RSA, yaitu :

1. Desain Halaman Utama

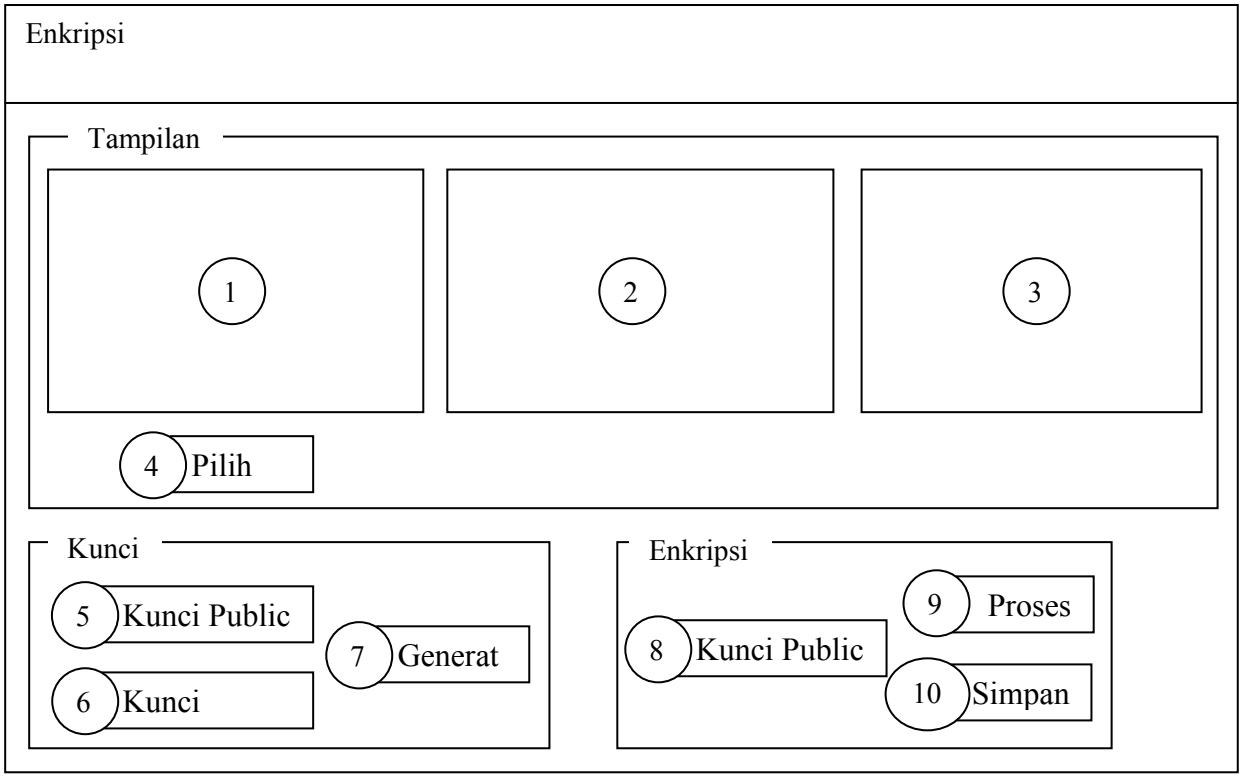


Gambar III.9. Desain Halaman Utama

Merupakan tampilan rancangan halaman utama saat dijalankan. Adapun keterangannya sebagai berikut :

1) *Menu* untuk menampilkan *form* enkripsi.
2) *Menu* untuk menampilkan *form* dekripsi.
3) *Menu* untuk menampilkan *form* tentang aplikasi.

2. Desain *Form* Enkripsi



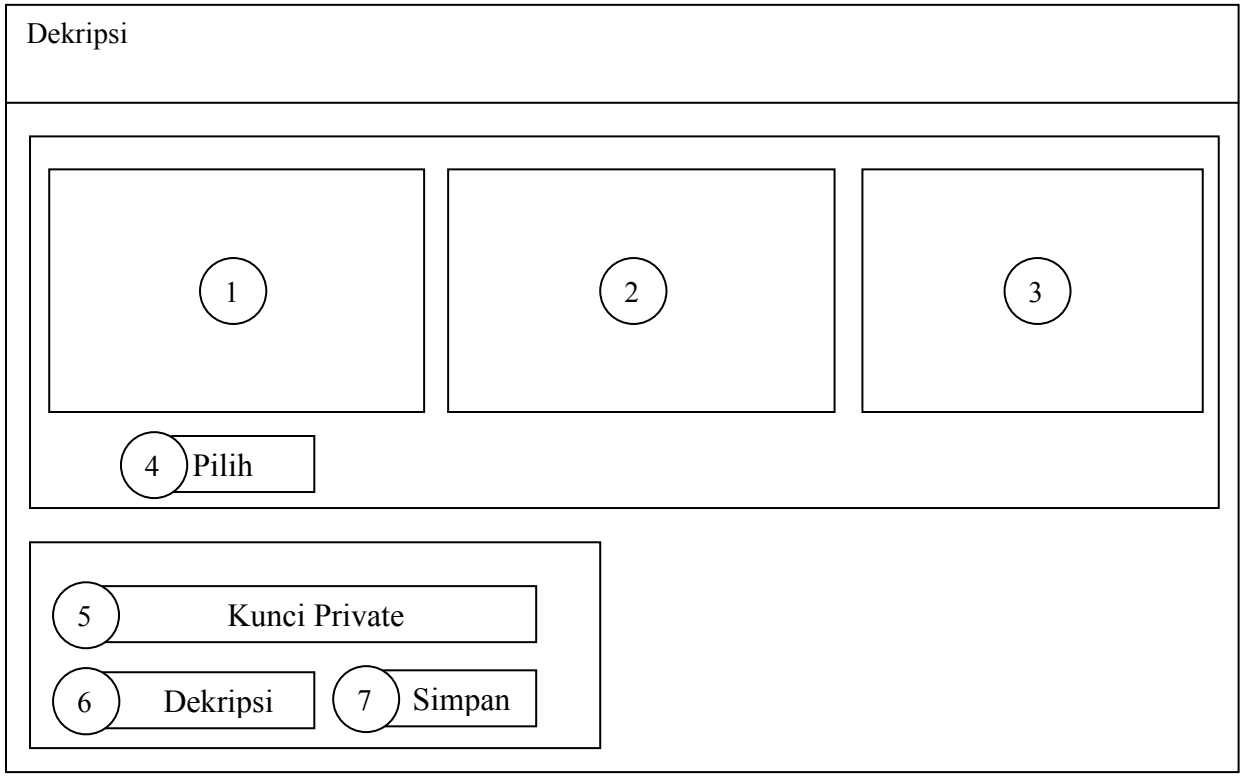
Gambar III.10. Desain *Form* Enkripsi

Keterangan tampilan *form* enkripsi, yaitu :

1) Tampilan *Plaintext*.
2) Tampilan *Ciphertext* pertama.
3) Tampilan *Ciphertext* kedua.
4) Tombol untuk memilih lokasi *file* yang akan dienkripsi.
5) *Textbox* untuk kata kunci *public*.
6) *Textbox* untuk kata kunci *private*.
7) Tombol untuk melakukan *generate* kunci.

8) *Textbox* untuk konfirmasi kunci *public* yang digunakan untuk proses enkripsi.
9) Tombol untuk proses enkripsi.
10) Tombol untuk penyimpanan *file* hasil enkripsi

3. Desain *Form* Dekripsi



Gambar III.11. Desain *Form* Dekripsi

Adapun keterangannya sebagai berikut :

1) Tampilan *Ciphertext* pertama.
2) Tampilan *Ciphertext* kedua.
3) Tampilan *Plaintext* hasil dekripsi.

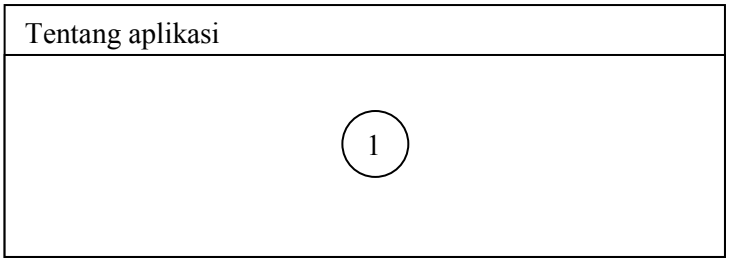
4) Tombol untuk memilih *file* yang akan didekripsi.

5) *Textbox* untuk menginputkan kata kunci *private*.

6) Tombol untuk melakukan dekripsi.

7) Tombol untuk melakukan penyimpanan *file* hasil dekripsi.

4. Desain *Form* Tentang



Gambar III.12. Desain *Form* Tentang

Adapun keterangannya sebagai berikut :

1) Menampilkan informasi dari aplikasi yang dibangun.