BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Kemajuan yang pesat juga terjadi di bidang jaringan komputer dengan konsep *open system*-nya, sehingga memudahkan seseorang untuk masuk ke dalam jaringan tersebut. Hal tersebut dapat mengakibatkan proses pengiriman data menjadi tidak aman karena dapat dimanfaatkan oleh pihak lain yang tidak bertanggung jawab untuk mengambil data maupun informasi di tengah jalan. Pesan rahasia merupakan hal penting yang butuh untuk dilindungi dan dijaga kerahasiaannya. Oleh karena itu maka tidak jarang muncul kejahatan-kejahatan yang dengan sengaja dilakukan oleh orang yang tidak bertanggung jawab, dengan semakin banyaknya orang yang melakukan tindakan kriminal yang dengan sengaja melakukan pencurian data rahasia dan merusak data rahasia sehingga bisa merugikan pihak tertentu.

Permasalahan yang dihadapi oleh beberapa orang dalam pengiriman data dan informasi sangat sulit dikarenakan mengirim pesan rahasia kepada seseorang yang jaraknya jauh dari kita lebih tinggi tingkat kebocorannya dari pada tingkat keamanannya dan pesan-pesan tersebut tidak terjamin kerahasiaannya. Pengamanan data dapat dilakukan pada berbagai jenis data, contohnya pada teks, citra, suara maupun video. Pengamanan data dilakukan jika seseorang menginginkan kerahasiaan data yang dimiliki tetap terjaga atau tidak semua orang boleh mengetahui isi dari data tersebut.

III.2. Strategi Pemecahan Masalah

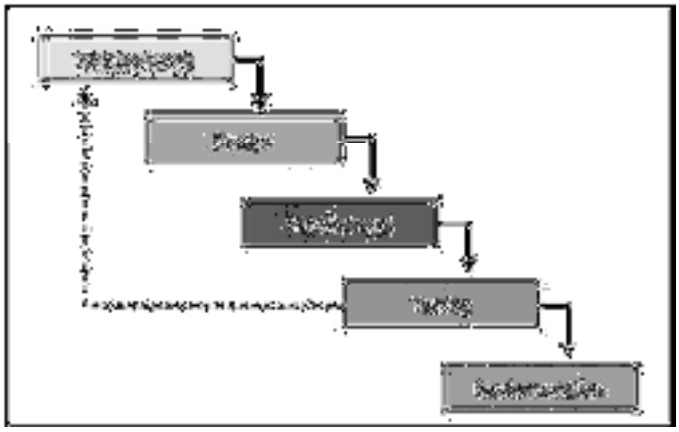
Dengan adanya analisis masalah yang dijelaskan sebelumnya, maka strategi pemacahan masalah yang dapat dilakukan adalah mencegah terjadinya hal-hal yang seperti itu. Maka peneliti membangun suatu aplikasi konversi dari data gambar ke bentuk *file* yang lain. Pada penelitian ini untuk mengamankan data gambar dengan menerapkan metode kriptografi dan mengubah data gambar menjadi *file* teks yang akan disimpan menjadi *file .syn*. Hal tersebut dilakukan untuk memperkecil ukuran data dan mengelabuhi orang bahwa sebenarnya data asli tersebut berupa data gambar. Dalam penelitian ini, akan dilakukan pengamanan data yang lebih mendalam mengenai kombinasi antara konversi data gambar menjadi *file* teks, lalu *file* teks hasil konversi tersebut dienkrip menggunakan metode AES yang diterapkan pada keamanan *file* gambar untuk mencapai tingkat keamanan yang tinggi.

Aplikasi yang akan dirancang dikembangkan dengan *tool developer Microsoft Visual Studio 2010*, aplikasi ini hanya bisa dijalankan pada PC bukan pada *mobile device*. Perancangan aplikasi keamanan yang dirancang ini menggunakan bahasa pemograman *Visual Basic* yang mempunyai sekumpulan *class - class* yang digunakan untuk mengembangkan perangkat lunak berbasis GUI (*Graphical User Interface*). Selain itu juga mempunyai *class–class* yang digunakan untuk menambahkan fungsi dan kemampuan interaksi yang variatif dari pemrograman *visual basic*.

Dalam pembuatan aplikasi ini, penulis merancang 4 desain yaitu *title bar*, *button*, *toolbar* dan *background* aplikasi. *Title Bar* digunakan untuk menempatkan nama aplikasi yang dirancang. *Button* digunakan untuk menempatkan perintah–

perintah dasar seperti mencari *file* gambar, konversi gambar ke teks (*file .syn*), enkrip dan dekrip, konversi teks (*file .syn*) ke gambar, mendapatkan informasi cara menggunakan aplikasi serta informasi pembuat aplikasi dan keluar dari aplikasi.

Uraian tahapan perancangan dalam pembuatan aplikasi keamanan ini dijelaskan pada gambar diagram *waterfall* di bawah ini.



Gambar III.1. Tahapan Pembuatan Aplikasi

Adapun keterangan dari gambar di atas adalah sebagai berikut :

1. *Requirement* (Analisa Kebutuhan Sistem).

   Dalam tahap ini dilakukan proses pencarian bahan-bahan yang berkaitan dengan perancangan aplikasi keamanan data gambar menggunakan algoritma AES. Pengutipan yang dilakukan dapat berupa teori ataupun beberapa pendapat dari beberapa buku bacaan ataupun jurnal – jurnal yang terkait dengan hal tersebut untuk menunjang perancangan program.

2. *Design* (Perancangan Sistem)

   Berdasarkan analisa yang telah dilakukan, selanjutnya dilakukan proses perancangan aplikasi keamanan data gambar yang berhubungan dengan

perancangan arsitektur sistem, perancangan antarmuka, perancangan modul-modul yang berintegrasi dalam suatu sistem.

3. *Development* (Pembangunan Sistem)

   Pada tahap ini akan dilakukan proses pembangunan aplikasi keamanan data gambar menggunakan algoritma AES, sehingga pengguna dapat menggunakan aplikasi ini sesuai dengan hasil perancangan yang dibuat dengan menggunakaan interface Visual Studio 2010. Implementasi dilakukan dengan menggunakan perangkat yang sudah dieksplorasi sebelumnya.

4. *Testing* (Uji Coba Sistem)

   Pada tahap ini dilakukan beberapa tes terhadap sistem yang telah diimplementasikan. *Testing* dilakukan dengan percobaan menjalankan aplikasi ini dan melihat hasil dari proses konversi gambar dan mengamankan hasil konversi tersebut dengan algoritma AES. Adapun pendekatan yang dilakukan penulisan dalam melakukan pengujian sistem yang dibuat adalah *Black Box Testing* dimana pengujian ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah proses enkrip dan dekrip berjalan dengan benar dan aplikasinya dapat digunakan oleh pengguna.

5. *Implementation* (Implementasi Sistem)

   Pada tahap ini aplikasi keamanan data gambar melewati tahap pengujian dan siap untuk digunakan oleh pengguna. Tidak menutup kemungkinan sistem ini mengalami perubahan ketika sudah digunakan oleh pengguna. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian. Tahap pendukung atau pemeliharaan dapat mengulangi proses

pengembangan mulai dari analisis spesifikasi untuk perubahan sistem informasi yang sudah ada, tapi tidak untuk membuat sistem informasi baru.

III.2.1. Konversi Gambar (Citra) ke *File* Teks (*File .syn*)

Citra juga dapat dikonversi menjadi bentuk teks dengan memanfaatkan proses pengenalan teks dari citra. Pengenalan teks adalah metode yang dikembangkan untuk membuat sistem yang mampu memberi pengertian atau deskripsi mengenai objek teks pada citra. Pengenalan teks merupakan bagian dari teknik analisa citra. Elemen dasar citra yang dianalisa pada teknik pengolahan teks adalah elemen bentuk. Proses pengenalan teks atau pendeteksian teks pada media gambar berarti sistem dapat mengenali/membaca suatu objek teks pada media gambar dan menulisnya ke bentuk *string*. String merupakan defenisi karakter berdasarkan kode ASCII.

String sering kita temui pada piranti lunak pengolah kata atau pada program penyunting teks. Teks pada media citra adalah bukan merupakan *string*, karena teks tersebut bukan merupakan perwakilan dari kode – kode ASCII, tapi merupakan objek yang terbentuk dari susunan piksel. Metode dasar yang digunakan untuk mengenali objek tersebut sebagai teks adalah dengan mencirikan bentuk dari karakter tersebut masing – masing.

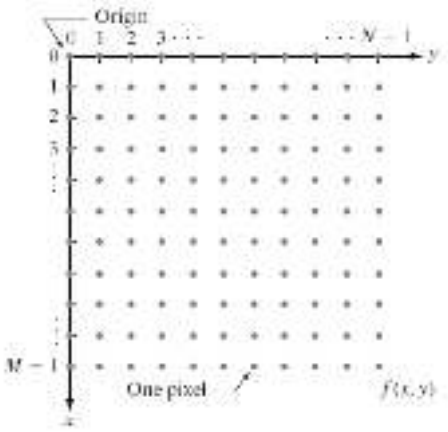
Sistem yang akan dirancang ini terdiri dari dua sistem utama, yaitu :

1. Sistem konversi data citra menjadi data teks (*file .syn*).
2. Sistem konversi data teks (*file .syn*) menjadi data teks menjadi data citra.

Pada sistem konversi data citra menjadi data teks (*file .syn*) membutuhkan data masukkan berupa data citra. Suatu citra dapat didefinisikan sebagai fungsi

f(x,y) berukuran M baris dan N kolom, dengan *x* dan *y* adalah koordinat spasial, dan amplitudo *f* di titik koordinat *(x,y)* dinamakan intensitas atau tingkat keabuan dari citra pada tersebut.

Apabila nilai *x* dan *y* dan nilai amplitudo *f* secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar III.2. menunjukkan posisi koordinat citra digital.



Gambar III.2. Koordinat Citra Digital

Elemen-elemen pada citra digital (berarti elemen matriks) disebut sebagai *picture element* atau piksel (*pixel*). Jadi citra yang berukuran M x N mempunyai MN buah *pixel*. Misalkan sebuah citra digital berukuran 256 x 256 *pixel* dengan derajat keabuan 256 level dan direpresentasikan secara numerik dengan matriks terdiri 256 baris (di indeks 0 sampai 255) dan 256 kolom.

Nilai RGB tiap piksel akan diubah menjadi karakter tertentu. Misal diketahui nilai piksel pada suatu citra adalah nilai R = 176, G = 22 dan B = 95. Untuk nilai R= 176 langkah-langkah untuk menghasilkan karakter adalah sebagai berikut :

1. Operasi *div* → 176 *div* 92 = 1

Karakter pertama yang dihasilkan adalah : *get* karakter ASCII ke 1+33=34 → ” (ditambah dengan nilai 33 karena karakter ASCII yang digunakan dimulai dari nilai desimal 33).

2. Operasi *mod* → 176 *mod* 92 = 84

   Karakter kedua yang dihasilkan adalah : *get* karakter ASCII ke 84+33=117 → u

3. Jadi nilai R = 176 diubah menjadi karakter → “u
4. Lakukan langkah 1 hingga 3 untuk mengubah nilai G dan B, sehingga untuk setiap piksel terdiri dari 6 karakter.

III.2.2. Penerapan Metode *Advance Encryption Standard* (AES)

Alur sistem aplikasi pada penelitian ini adalah hasil gabungan teknik pengamanan data yang memanfaatkan kriptografi menggunakan metode *Advanced Encryption Standard* untuk melakukan enkripsi pada *file* teks (*plaintext*) dengan kunci (*key*) yang yang hanya diketahui oleh *user* tanpa ada pihak lain yang mengetahuinya sehingga informasi yang terkandung dalam *file* teks (*plaintext*) tidak dapat diketahui oleh pihak manapun yang tidak diinginkan, kemudian hasilnya yaitu *ciphertext* disimpan menjadi *file* enkrip berupa *file* **.encrypt*.

III.2.2.1. Proses Enkripsi Algoritma AES

Proses enkripsi pada algoritma AES umumnya digunakan untuk melindungi data. Adapun spesifikasi sistem Enkripsi sebagai berikut:

1. Input

   Input data file yang dapat di proses adalah format file yang memiliki karakter yang mendukung ASCII seperti format jpg/jpeg. Pemilihan jenis data file tersebut disesuaikan dengan kebutuhan user dan file yang dipilih dapat di pastikan dapat diproses dengan Algoritma AES.

2. Proses

   Dalam sistem ini Enkripsi file masukkan tidak lebih dari lima mega bytes (5MB) hal tersebut dilakukan untuk menjaga kinerja kecepatan proses enkripsi data file. Langkah proses yang direncanakan :

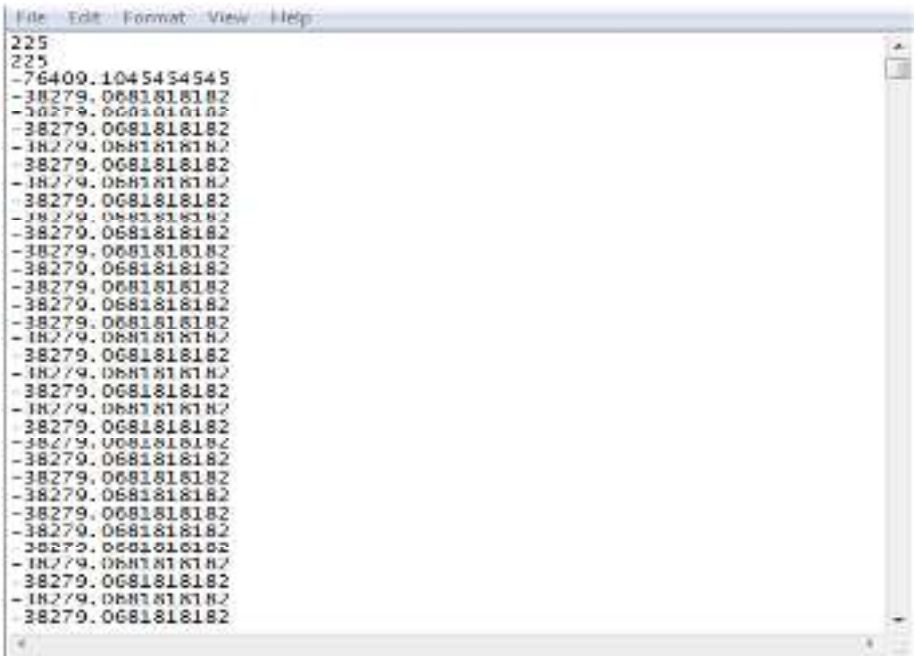
   1. Menginput data file
   2. Mengkonversi data file
   3. Proses enkripsi file hasil konversi
   4. Menyimpan hasil enkripsi

3. Output

   Terdapat dua jenis output pada saat menggunakan proses enkripsi yang digunakan, yaitu output pada proses enkripsi dan output pada proses dekripsi. File output pada proses enkripsi tidak akan dapat dibaca, sedangkan File output proses dekripsi hasilnya harus sama dengan data file sebelum digunakan pada aplikasi.

Berikut contoh gambar yang digunakan sebagai data masukkan dapat dilihat pada gambar III.3, dan data keluaran dari hasil enkrip hasil konversi dapat dilihat pada gambar III.4.

Gambar III.3. Data Gambar



Gambar III.4. Gambar Hasil Konversi

Pada tahap ini akan disimulasikan proses enkripsi Algoritma AES, yang dimana plainteks dan kunci yang digunakan adalah masing – masing sebagai berikut :

Plaintext : Two One Nine Two

Key : Thats my Kung Fu

Langkah pertama yang dilakukan adalah mengubah plainteks diatas menjadi bentuk hexadecimal dan seterusnya, kemudian untuk kunci nya kita ubah kedalam bentuk karakter pada ASCII.

Adapun langkah – langkahnya adalah sebagai berikut :

1. Ubah *key* ke dalam bilangan *Hex*.

   Thats My Kung Fu

   Key (Hex) : 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

2. Ubah *plaintext* ke dalam bilangan *Hex*.

   Two One Nine Two

   Plaintext (Hex) : 54 77 6F 20 4F 6E 65 20 4E 69 6E 65 20 54 77 6F

3. w[0] = (54; 68; 61; 74) ; w[1] = (73; 20; 6D; 79) ; w[2] = (20; 4B; 75; 6E) ; w[3] = (67; 20; 46; 75)

4. g(w[3]) :

   a. geser ke kiri *byte* dari w[3] : (20; 46; 75; 67)

   b. Subtitusi *Byte* (S-Box) : (B7; 5A; 9D; 85)

   c. Tambahkan konstanta putaran (01; 00; 00; 00) ke persamaan g(w[3]) = (B6; 5A; 9D; 85)

5. w[4] = w[0] $\oplus$ 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] ⊕ 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)

6. *First roundkey* : E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
7. Round 0 : 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
8. Round 1 : E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
9. Round 2 : 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA
10. Round 3 : D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB
11. Round 4 : A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B
12. Round 5 : B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69
13. Round 6 : BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E
14. Round 7 : CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A
15. Round 8 : 8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C
16. Round 9 : BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8
17. Round 10 : 28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 26
18. State Matrix dan Roundkey No.0 Matrix :

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

19. XOR entri yang sesuai, contoh, 69 ⊕ 4B = 22

$$\begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array} \qquad \begin{array}{cccc} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \end{array}$$

20. Matriks baru terbentuk :

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

21. Subtitusi setiap entri (*byte*) dari matriks dengan entri yang sesuai di *AES S-Box*.

22. Misalnya : *byte 6E* diganti dengan *S-Box* di baris 6 dan kolom E, oleh 9F.

23. Maka terbentuk matriks baru :

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

24. Geser ke kiri empat baris dengan *offset* sebesar 0,1,2, dan 3

25. Matrix baru terbentuk :

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

26. Langkah pencampuran linier ini menyebabkan difusi *bit* pada beberapa putaran, lalu lakukan perkalian matriks.

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

27. Cari hasil dari $(02 \bullet 63) \oplus (03 \bullet 2F) \oplus (01 \bullet AF) \oplus (01 \bullet A2)$:

    a. $02 \bullet 63 = 00000010 \bullet 01100011 = 11000110$

b. $03 \bullet 2F = (02 \bullet 2F) \oplus 2F = (00000010 \bullet 00101111) \oplus 00101111$ = 01110001

c. $01 \bullet AF = AF = 10101111$ and $01 \bullet A2 = A2 = 10100010$

$$\begin{array}{cccccccc} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

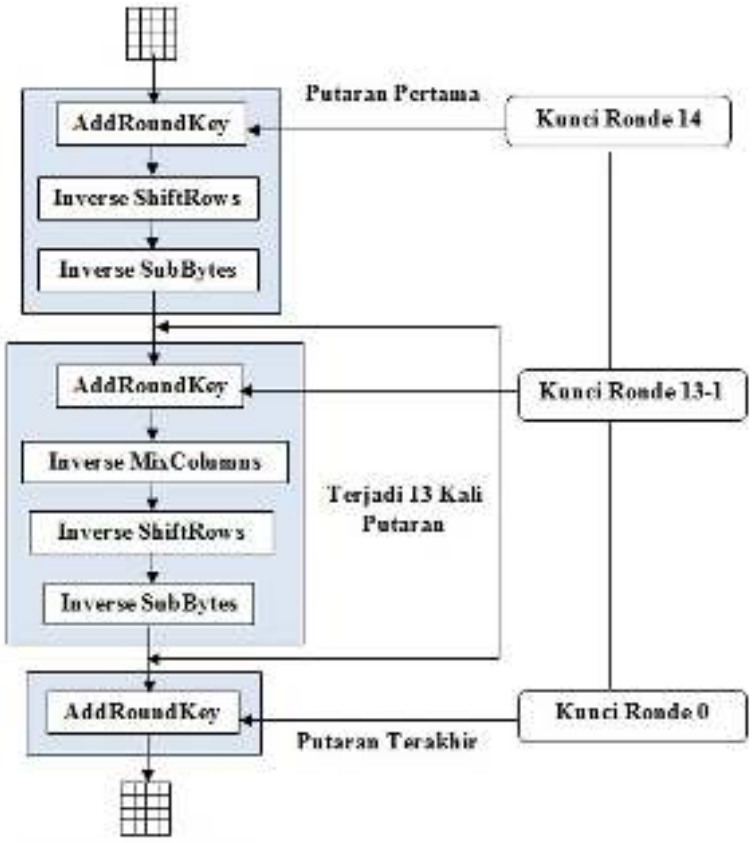
28. XOR matriks berikut :

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \text{ XOR } \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix} = \begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

29. Output AES Round 1: 58 47 08 8B 15 B6 1C BA 59 D4 E2 E8 CD 39 DF CE
30. Lakukan *Substitute Byte*, *Shift Rows*, *Mixcolumns* dan *Roundkey* seperti langkah di atas sampai 16x putaran.
31. Maka akan di dapat *ciphertext* : 29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A

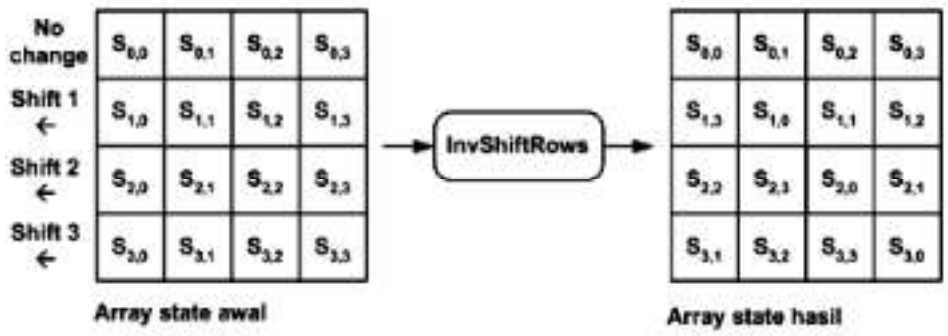
III.2.2.2. Proses Dekripsi Algoritma AES

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada diagram alir dalam gambar berikut.



Gambar III.5. Diagram Alir Proses Dekripsi Algoritma AES

InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi *InvShiftRows* terdapat pada gambar berikut.



Gambar III.6. Transformasi *InvShiftRows*

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada *state* dipetakan dengan menggunakan table *Inverse S-Box*. Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan sebagai berikut :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Hasil dari perkalian matriks tersebut, setiap *byte* dalam kolom *array state* akan digantikan dengan nilai baru. Persamaan matematis untuk setiap *byte* tersebut pada persamaan 4.

$$s'0,c = (\{0E\} \bullet s0,c) \oplus (\{0B\} \bullet s1,c) \oplus (\{0D\} \bullet s2,c) \oplus (\{09\} \bullet s3,c)$$

$$s'1,c = (\{09\} \bullet s0,c) \oplus (\{0E\} \bullet s1,c) \oplus (\{0B\} \bullet s2,c) \oplus (\{0D\} \bullet s3,c)$$

$$s'2,c = (\{0D\} \bullet s0,c) \oplus (\{09\} \bullet s1,c) \oplus (\{0E\} \bullet s2,c) \oplus (\{0B\} \bullet s3,c)$$

$$s'3,c = (\{0B\} \bullet s0,c) \oplus (\{0D\} \bullet s1,c) \oplus (\{09\} \bullet s2,c) \oplus (*0E\} \bullet s3,c)$$

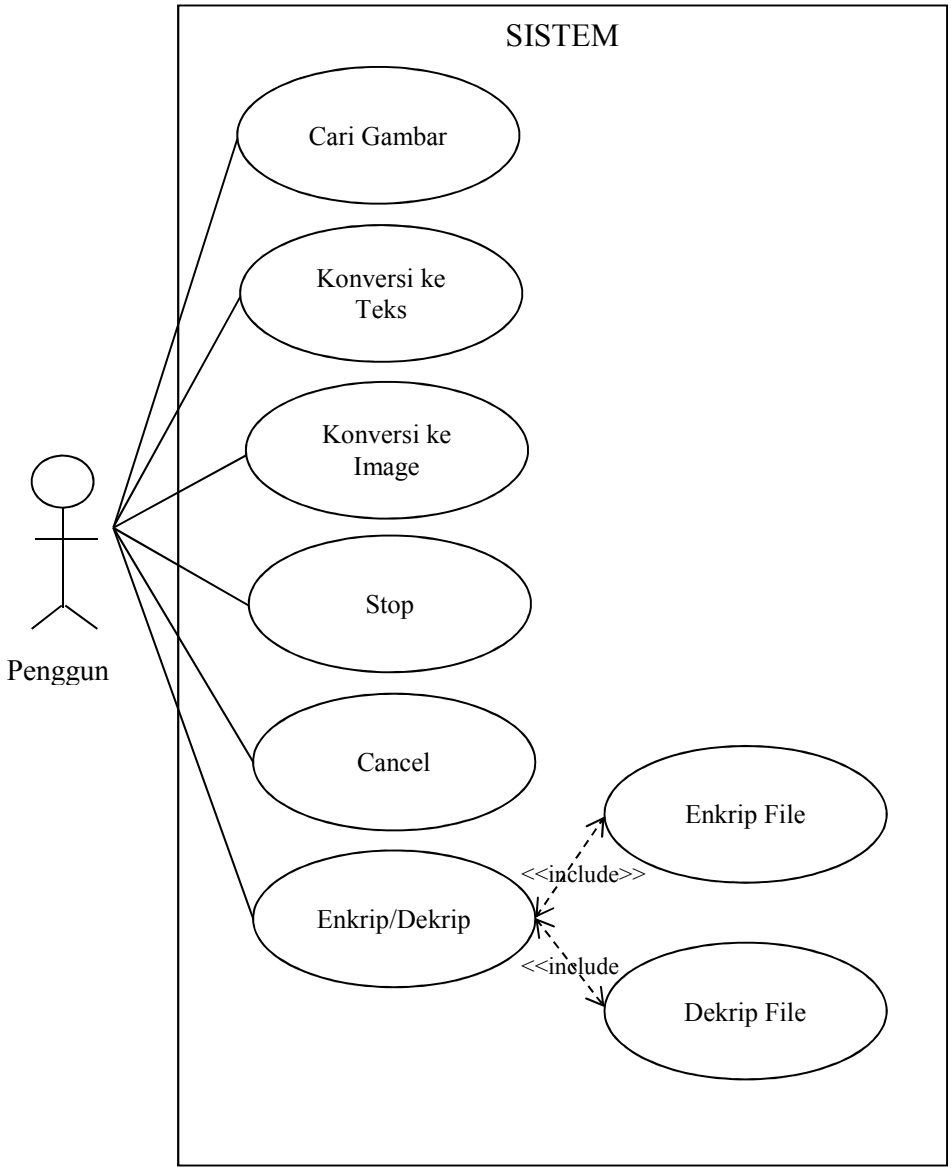
III.3. Desain Sistem

Bentuk rancangan sistem yang akan dibuat menggunakan beberapa bentuk diagram dari *Unified Modeling Language* (UML) yaitu *Use Cace Diagram*, *Sequence Diagram* dan *Activity Diagram*.

III.3.1. Use Case Diagram

Use Case Diagram adalah gambaran *graphical* dari beberapa atau semua *actor*, *use case*, dan interaksi diantara komponen-komponen tersebut yang

memperkenalkan suatu sistem yang akan dibangun digunakan untuk menjelaskan bagaimana langkah-langkah yang seharusnya dikerjakan oleh sistem. Adapun *Use Case Diagram* dari aplikasi yang dirancang dapat dilihat pada gambar III.7.



Gambar III.7. *Use Case Diagram* Aplikasi Keamanan Konversi Gambar

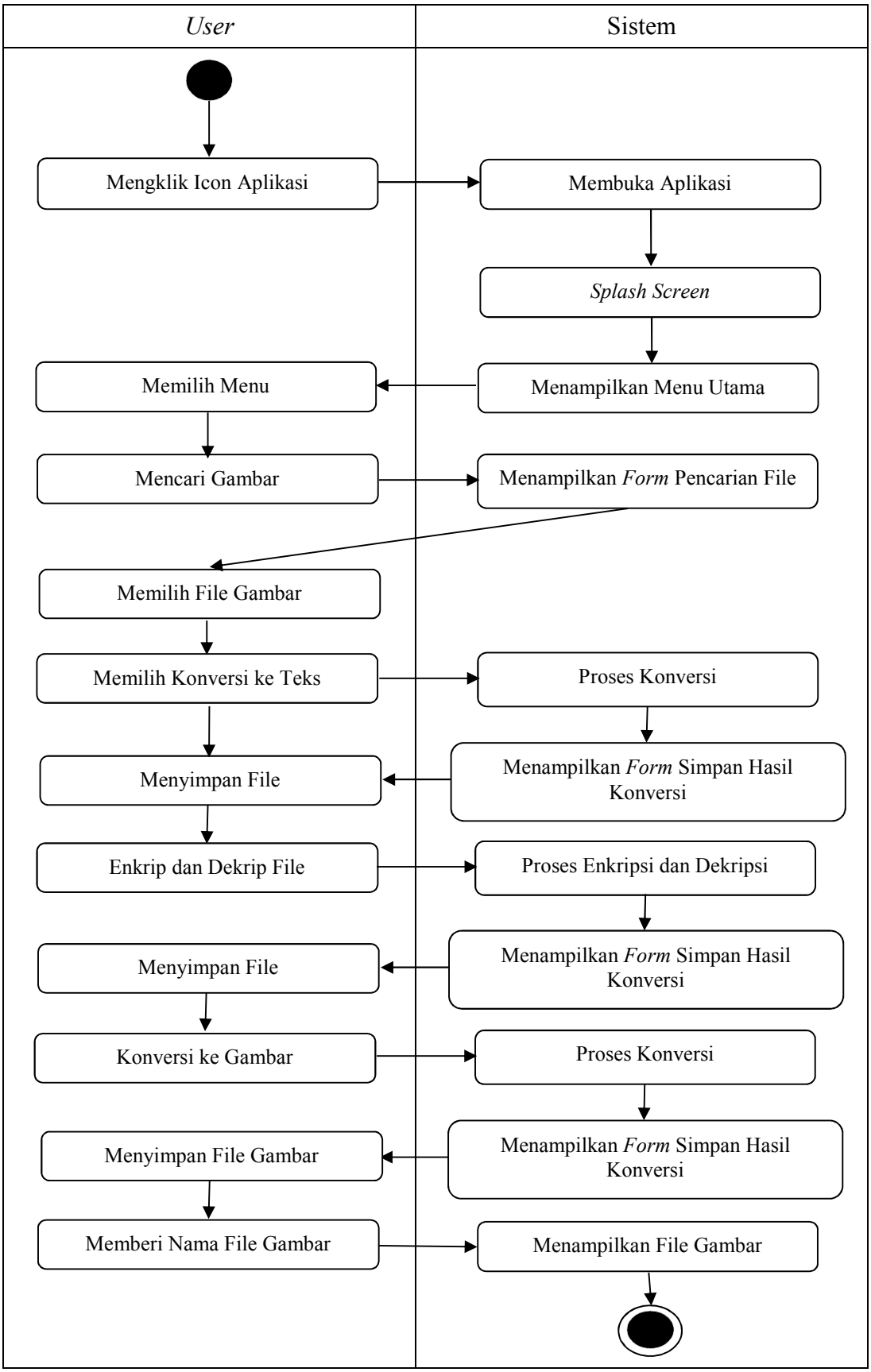
Dari gambar tersebut dapat dilihat bahwa pengguna pada tampilan awal aplikasi akan melihat beberapa menu yaitu menu Pengamanan *File Image*. Pada

menu ini terdapat tombol Cari Gambar, Konversi ke Teks, Konversi ke *Image*, *Stop*, *Cancel* dan Enkrip/Dekrip. Adapun fungsi dari tombol tersebut adalah :

1. Cari Gambar, berfungsi untuk mencari atau memilih *file* gambar yang akan dikonversi.
2. Konversi ke Teks, berfungsi untuk mengkonversi *file* gambar ke *file* teks *(file .syn)*.
3. Konversi ke *Image*, berfungsi untuk mengubah atau mengkonversi *file* teks *(file .syn)* ke *file* gambar.
4. Stop, berfungsi untuk menghentikan proses konversi
5. Cancel, berfungsi untuk membatalkan proses konversi.
6. Enkrip/Dekrip, berfungsi untuk mengenkrip dan mendekrip *file* teks *(file .syn)*.

III.3.2. Activity Diagram

Activity Diagram menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. *Activity Diagram* menggambarkan aktivitas yang dilakukan oleh pengguna dalam menjalankan aplikasi ini. Adapun *Activity diagram* dapat dilihat pada gambar III.8.

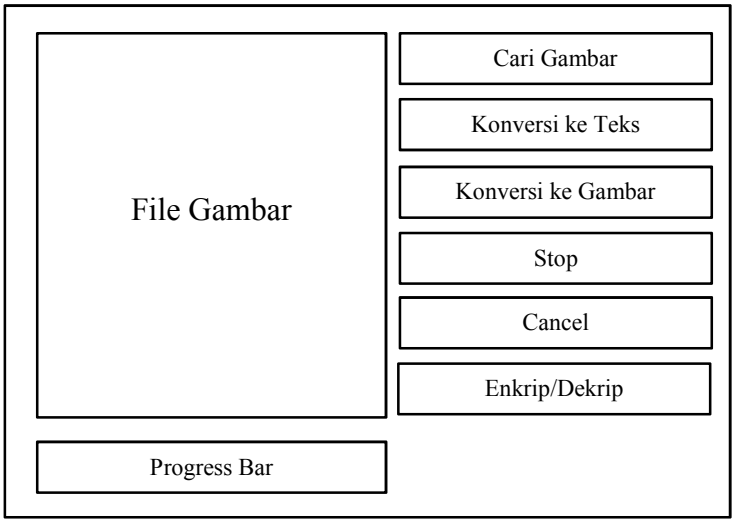


Gambar III.8. *Activity Diagram* Aplikasi *Game* Mencocokkan Gambar

III.4. Desain Sistem Secara Detail

III.4.1. Rancangan *Form* Menu Utama

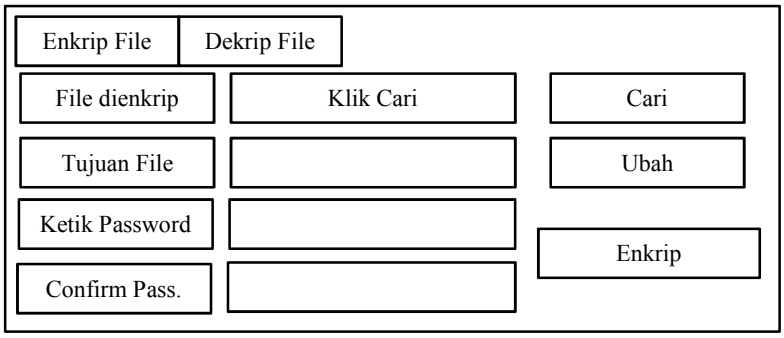
Rancangan Menu Utama menampilkan menu Pengamanan *File Image*. Pada menu ini terdapat tombol Cari Gambar, Konversi ke Teks, Konversi ke *Image*, *Stop*, *Cancel* dan Enkrip/Dekrip *File*.. Rancangan *Form* Menu Utama dapat dilihat pada Gambar III.9.



Gambar III.9. Rancangan *Form* Menu Utama

III.4.2. Rancangan *Form* Enkrip *File*

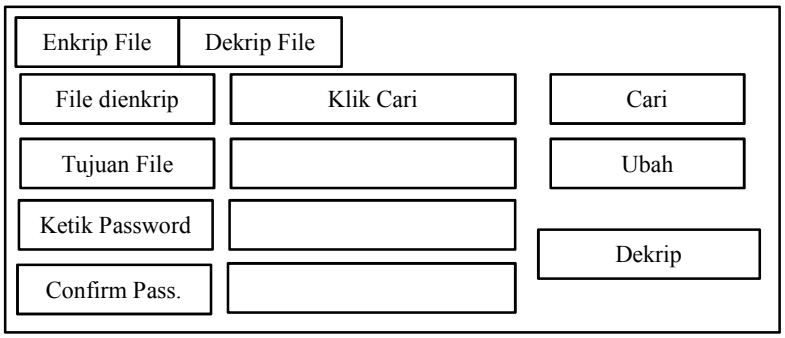
Rancangan *form* Enkrip File berisi proses enkrip dan dekrip file teks *(file .syn)*. Rancangan *form* Enkrip *File* dapat dilihat pada Gambar III.10.



Gambar III.10. Rancangan *Form* Enkrip *File*

III.4.3. Rancangan *Form* Dekrip *File*

Rancangan *form* Dekrip File berisi proses enkrip dan dekrip file teks *(file .syn)*. Rancangan *form* Dekrip *File* dapat dilihat pada Gambar III.11.



Gambar III.11. Rancangan *Form* Dekrip *File*