BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisa Masalah

Proses Analisa sistem merupakan langkah kedua pada pengembangan sistem. Analisa sistem dilakukan untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Sistem perbandingan belum begitu di ketahui oleh banyak orang. Seseorang yang ingin melakukan pengamanan data yang ada pada aplikasi atau suatu program yang digunakan lebih aman dari gangguan yang ingin merusak data tersebut. Berkembangnya teknelogi informasi secara otomatis akan menambah jumlah data pribadi. Hal ini secara otomatis dapat lupa terhadap keamanan data tersebut.

Untuk itu, sistem yang penulis rancang yaitu mengenai perbandingan mengamankan sebuah data teks dalam *platform* android dengan menggunakan algoritma kriptografi AES128 dan DES untuk mengetahui tentang keamanan yang dilakukan oleh kedua algoritma tersebut mana yang lebih aman dan akurat. Dalam tahap pengembangan sistem perbandingan ini, analisa sistem merupakan hal yang harus dilakukan sebelum proses perancangan sistem.

III.2. Strategi Pemecahan Masalah

Adapun strategi pemecahan masalah dari sistem perbandingan yang dirancang yaitu sebagai berikut:

1. Suatu data dapat memiliki nilai kerahasiaan, oleh karena itu data tersebut harus diamaankan dengan memberikan hak keamanan dengan mengubah data asli ke data yang tidak dimengerti oleh seseorang dengan algoritma AES128 dan DES.
2. Data yang sangat penting harus mempunyai keamanan yang baik juga dengan demikian pada perancangan ini kita akan membandingkan dua algoritma yaitu AES128 dan DES untuk mengetahui mana yang lebih aman dan akurat dalam mengamankan data.

III.3. Perbandingan Kemanan Data

III.3.1. DES

DES adalah blok cipher yang beroperasi pada 64-bit. Sebuah masukan blok 64-bit dari *plaintext* akan dienkripsi menjadi 64-bit *output* blok teks *cipher*. Ini adalah sebuah Algoritma simentris, yang berarti algoritma dan kunci yang sama digunakan untuk enkripsi dan dekripsi.

Keamanan DES terletak di kunci 56-bit. blok *plaintext* diambil dan dimasukan melalui permutasi awal. Kuncinya juga diambil pada waktu yang sama. Kuncinya disajikan dalam blok 64-bit dengan setiap bit 8 menjadi cekparitas. Kunci 56-bit kemudian diekstraksi dan siap digunakan. 64-bit blok *plaintext* dibagi menjadi dua bagian 32 bit, bernama kanan setengah setngah kiri. Dua bagian *plaintext* kemudian digabungkan dengan data dari kunci dalam operasi yang disebut Fungsi *f*. Ada 16 putaran Fungsi *f*, setelah itu dua bagian yang digabungkan menjadi satu 64-bit blok, yang kemudian dimasukkan melalui final permutasi untuk menyelesaikan operasi algoritma dan 64-bit teks *cipher* blok dikeluarkan.

Cara kerja enkripsi dan dekripsi algoritma DES dengan menggunakan operasi dasar untuk proses adalah sebagai berikut:

III.3.1.1. Enkripsi data 64-bit

Langkah-langkah yang dilakukan dalam proses enkripsi 64-bit adalah sebagai berikut:

1. Kita harus mengambil blok 64 bit. Jika mengambil blok kurang dari 64 bit perlu ada penamabahan supaya dalam penggunaannya ada kesesuaian dengan jumlah datanya.
2. Dibentuk ip pada blok data 64 bit dengan memperhatikan tabel permutasi.

Tabel III.1. *Initial Permutation* (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

3. Kemudian bagilah data blok tersebut menjadi 2 bagian. 32 pertama disebut L[0] dan 32 bit kedua disebut R[0].
4. Ke-enam belas *subkey* di operasikan dengan blok data, dimulai dari j=1. Caranya adalah sebagai berikut:
   a. R[j-1] di kembangkan menjadi 48-bit menurut pemilihan fungsi ekspansi.

Tabel III.2. Expantion Permutation (E)

32	**1**	**2**	**3**	**4**	**5**
4	**5**	**6**	**7**	**8**	**9**
12	**13**	**14**	**15**	**16**	**17**
16	**17**	**18**	**19**	**20**	**21**
20	**21**	**22**	**23**	**24**	**25**
24	**25**	**26**	**27**	**28**	**29**
28	**29**	**30**	**31**	**32**	**1**

   b. Kemudian XOR kan E(R[j-1]) dengan K[j].

c. Hasil E(R[j-1]) XOR K[j] dipecah menjadi 8 blok 6-bit. Kelompok bit 1-6 disebut B[1], bit 7-12 disebut B[2], dan seterusnya hingga bit 32-48 disebut B[8].

d. Jumlah bit dikurangi dengan penukaran nilai-nilai yang ada dalam tabel S untuk setiap B[j]. Dimulai dengan j=1, setiap nilai dalam tabel S memiliki 4 bit, seperti salah satu contoh tabel S-box berikut ini:

Tabel III.3. Subsitution Box-1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
2	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
3	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
4	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

e. Permutasi kembali dilakukan kembali pada kombinasi hasil subsitusi.

Tabel III.4. Permutation (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	4	9
19	13	30	6	22	11	3	25

f. Kemudian hasil subsitusi di XOR kan dengan L[j-1]. Selanjutnya hasil tersebut menjadi R[j].

g. Ulangi langkah tersebut diatas hingga K[16].

h. Kemudian permutasi akhir dilakukan kembali dengan tabel permutasi yang merupakan invers dari permutasi awal

Tabel III.5. Final Permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31

38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	50	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

III.3.1.2. Dekripsi Data 64-Bit

Dekripsi dilakukan dengan proses yang sama, hanya saja kunci yang digunakan dalah K[j] dalam urutan yang berlawan, yaitu memasukkan K[16] terlebih dahulu, kemudian K[15], dan seterusnya hingga K[1]. Algoritmanya adalah sebagai berikut :

1. Mengambil blok *ciphertext* 6 bit. Apabila pengambilan blok ciphertext kurang dari 64 bit, perlu adanya penambahan supaya dalam penggunaannya ada kesesuaian dengan jumlah datanya.
2. Bentu IP pada blok enkripsi 64 bit dengan memperhatikan tabel permutasi.
3. Bagilah blok enkripsi tersebut menjadi 2 bagian. 32 bit pertama disebut L[0] dan 32 bit ke dua disebut R[0].
4. Keenam belas subkey dioperasikan dengan blok ciphertext, dimualai dengan j=1 dan terbagi menjadi cara sebagai berikut:
    a. R[j-1] dikembangkan menjadi 48 bit menurut fungsi pemilihan ekspansi.
    b. Langkah berikutnya adalah meng-XOR-kan E(R[j-1]) dengan K[j].
    c. Hasil E(R[j-1]) XOR K[j] dipecah menjadi 8 blok 6 bit. Kelompok bit 1-6 disebut B[1], 7-12 disebut B[2], dan seterusnya hingga 43-48 disebut B[8].
    d. Jumlah dikurangi dengan pertukaran nilai-nilai yang ada dalam tabel S untuk setiap B[j] yang dimulai dengan j=1. Setiap nilai dalam S memiliki 4 bit.
    e. Permutasi dilakukan kembali pada kombinasi hasil subsitusi, kemudian di XOR kan,

f. Permutasi akhir dilakukan kembali dengan tabel permutasi yang merupakan invers dari permutasi awal.

III.3.1.3. Pemrosesan Kunci

Algoritma pemrosesan kunci sangat berguna saat implementasi program. Adapun algoritmtmanya adalah sebagai berikut :

1. Pengguna memasukkan kunci sebesar 64 bit atau 8 karakter, dimana nantinya dimana setiap bit pada kunci akan digunakan sebagai paritas. Misalkan PASSWORD.

   01010000 01000001 01010011 01010011 01010111 01001111 01010010 01000100.
2. Langkah selanjutnya adalah proses permutasi. Prmutasi dilakukan pada kunci 64 bit. Tahapan ini bit-bit paritas tidak dilibatkan sehingga bit kunci berkurang menjadi 56 bit. Contoh hasil dari permutasi sebagai berikut :

Tabel III.6. Initial Permutation One (P-1)

57	49	41	33	25	17	19
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	45	48	30	2
14	6	61	53	45	37	29
21	13	5	28	20	12	4

   Menjadi 0000000 0111111 1100000 0000101 0111110 0101100 0000100 0001101.
3. Setelah itu hasilnya dibagi menjadi dua bagian. 28 bit pertama disebut C[0] dan 28 bit kedua disebut D[0].

   C_0 = 0000000 0111111 1100000 0000101

D_0 = 0111110 0101100 0000100 0001101

4. Langkah selanjutnya adalah melakukan penggeseran kedua bagian dari kiri kenana bergantung pada setiap putaran.
5. Pada setiap hasil penggeseran, kemudian digabungkan lalu dilakukan permutasi kembali C_j, D_j . permutasi tersebut dikenal sebagai choice-2 (PC-2) seperti berikut ini:

Tabel III.7. Initial Permutation Two (P-2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Dengan melakukan permutasi PC-2 maka akan didapatkan nilai K_i, dan itu dilakukan terus menerus hingga ke-16 kunci berhasil disusun.

III.3.2. Algoritma AES *(Advanced Encryption Standard)*

Pada algoritma AES, jumlah blok input, blok output, dan *state* adalah 128 bit. Dengan besar data 128 bit, berarti *Nb* = 4 yang menunjukkan panjang data tiap baris adalah 4 byte. Dengan panjang kunci 128-bit, maka terdapat sebanyak 3,4 × 1038 = 2128 kemungkinan kunci. Jika komputer tercepat dapat mencoba 1 juta kunci setiap detik, maka akan dibutuhkan waktu 5,4 × 1024 tahun untuk mencoba seluruh kunci. Jika tercepat yang dapat mencoba 1 juta kunci setiap milidetik, maka dibutuhkan waktu 5,4 × 1018 tahun untuk mencoba seluruh kunci

Dengan blok input atau blok data sebesar 128 bit, *key* yang digunakan pada algoritma AES tidak harus mempunyai besar yang sama dengan blok input. *Cipher key* pada algoritma

AES bisa menggunakan kunci dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini. Di bawah ini adalah Tabel yang memperlihatkan jumlah *round* (*Nr*) yang harus diimplementasikan pada masing-masing panjang kunci *(Tabel 2.6: Perbandingan jumlah Round dan Key).*[5] Tidak seperti *DES* yang berorientasi bit, *Rijndael* beroperasi dalam orientasi *byte.* Setiap putaran mengunakan kunci internal yang berbeda (disebut *round key*). *Enciphering* melibatkan operasi substitusi dan permutasi.

Tabel III.8. *Perbandingan jumlah Round dan Key*

	Jumlah (*Nk words*)	Ukuran Blok (*Nb words*)	Jumlah Putaran (*Nr*)
AES-128	16	16	10
AES-192	24	16	12
AES-256	32	16	14

Untuk memahami cara kerja AES128, dapat dimulai dengan melihat konsep dasar bagaimana algoritma AES128 beroperasi pada blok 128-bit dan kunci 128-bit.

III.3.2.1. Enkripsi AES128-Bit

1. *AddroundKey* : melakukan XOR antara state awal *(plain text)* dengan *chiper key*. Tahap ini disebut juga *initial round*
2. Round : putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah sebagai berikut :
   a. *SubBytes* : subsitusi byte dengan tabel subsitusi *S-Box.*

Tabel III.9. S-Box

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08

b. *ShiftRows* : pergeseran baris-baris *array state* secara *wrapping.*

c. *MixColumns* : pengacakan data di masing-masing kolom *array state.*

d. *AddroundKey* : peng-XOR-an antara *state* sekarang *round key*.

3. Final round : proses untuk putaran terakhir.

a. *SubBytes*

b. *ShiftRows*

c. *AddroundKey*

III.3.2.2. Dekripsi AES128-Bit

Proses dekripsi dilakukan pada arah yang berlawanan untuk menghasilkan inverse chiper yang mudah dipahami untuk algoritma AES. Urutan kerja deskripsi AES bukan merupakan kebalikan dari enkripsi. Ada proses yang dipertukarkan urutannya, walau pun pengguanan kuncinya sama.sebagai berikut:

1. Transformasi byte di berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran bit ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran bit ke kiri. Pada baris kedua, pergeseran bit dilakukan sebanyak 3 kali, sedangkan pada baris ketiga dan baris keempat, dilakukan pergeseran bit sebanyak dua kali dan satu kali.

2. *InvSubBytes* juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada *state* dipetakan dengan menggunakan Tabel *inverse S-Box*. Tabel ini berbeda dengan tabel S-Box karna hasil yang didapat dari tabel ini adalah hasil dari dua proses yang bebeda urutannya.
3. Pada *InvMixColumns*, kolom-kolom pada tiap *state* (*word*) akan dipandang sebagai polinom atas GF(2^8) dan mengalikan modulo $x^4 + 1$ dengan polinom tetap.
4. Transformasi *Inverse AddRoundKey* tidak mempunyai perbedaan dengan transformasi *AddRoundKey* karena pada transformasi ini hanya dilakukan operasi penambahan sederhana dengan menggunakan operasi *bitwise* XOR

   Pada tahap akhir dilakukan pengurangan terhadap masing-masing register dengan *key word* untuk mendapatkan plainteks.

III.3.2.3. Ekspansi Kunci

Pada pengguna unkuran kunci 128 bit perlu menyiapkan 10 buah kunci yang dihasilkan dari ekspansi kunci awal. Berikut adalah contoh kunci awal yang telah di input kan ke dalam matriks sebagai berikut :

$$\begin{pmatrix} 41 & 45 & 49 & 4D \\ 42 & 46 & 4A & 4E \\ 43 & 47 & 4B & 4F \\ 44 & 48 & 4C & 50 \end{pmatrix}$$

Kemudian ambil 4 byte terakhir yaitu 4D 4E 4F 50, lalu geser byte pertama menjadi byte terakhir.hasilnya 4E 4F 50 4D. Kemudian subsitusikan dengan menggunakan tabel S-Box, hasilnya adalah 2F 84 53 E3.

Langkah selanjutnya XOR kan dengan konstanta nilai tertentu dari pengguna.

2F XOR 01 = 00101111 XOR 00000001 = 00101110 =2E
84 XOR 00 = 00000100 XOR 00000000 = 10000100 =84
53 XOR 00 = 01010011 XOR 00000000 = 01010011 =53
E3 XOR 00 = 11100011 XOR 00000000 = 11100011 =E3

Kemudian XOR kan 2E 84 53 E3 dengan 4 *byte* baris pertama kunci awal yaitu 41 42 43 44.

2E XOR 41 = 00101110 XOR 01000001 = 01101111 =6F
84 XOR 42 = 10000100 XOR 01000010 = 11000110 =C6
53 XOR 43 = 01010011 XOR 01000011 = 00010000 =10
E3 XOR 44 = 11100011 XOR 01000100 = 10100111 =A7

Hasil proses XOR diatas adalah 6F C6 10 A7 yang merupakan 4 *byte* pertama dari kunci yang baru untuk byte.

45 XOR 6F = 01000101 XOR 01101111 = 00101010 =6F
46 XOR C6 = 01000110 XOR 11000110 = 10000000 =C6
47 XOR 10 = 01000111 XOR 00010000 = 01010111 =10
48 XOR A7 = 01001000 XOR 10100111 = 11101111 =A7

Demikian seterusnya hingga didapatkan 16 *byte* set kunci yang baru.

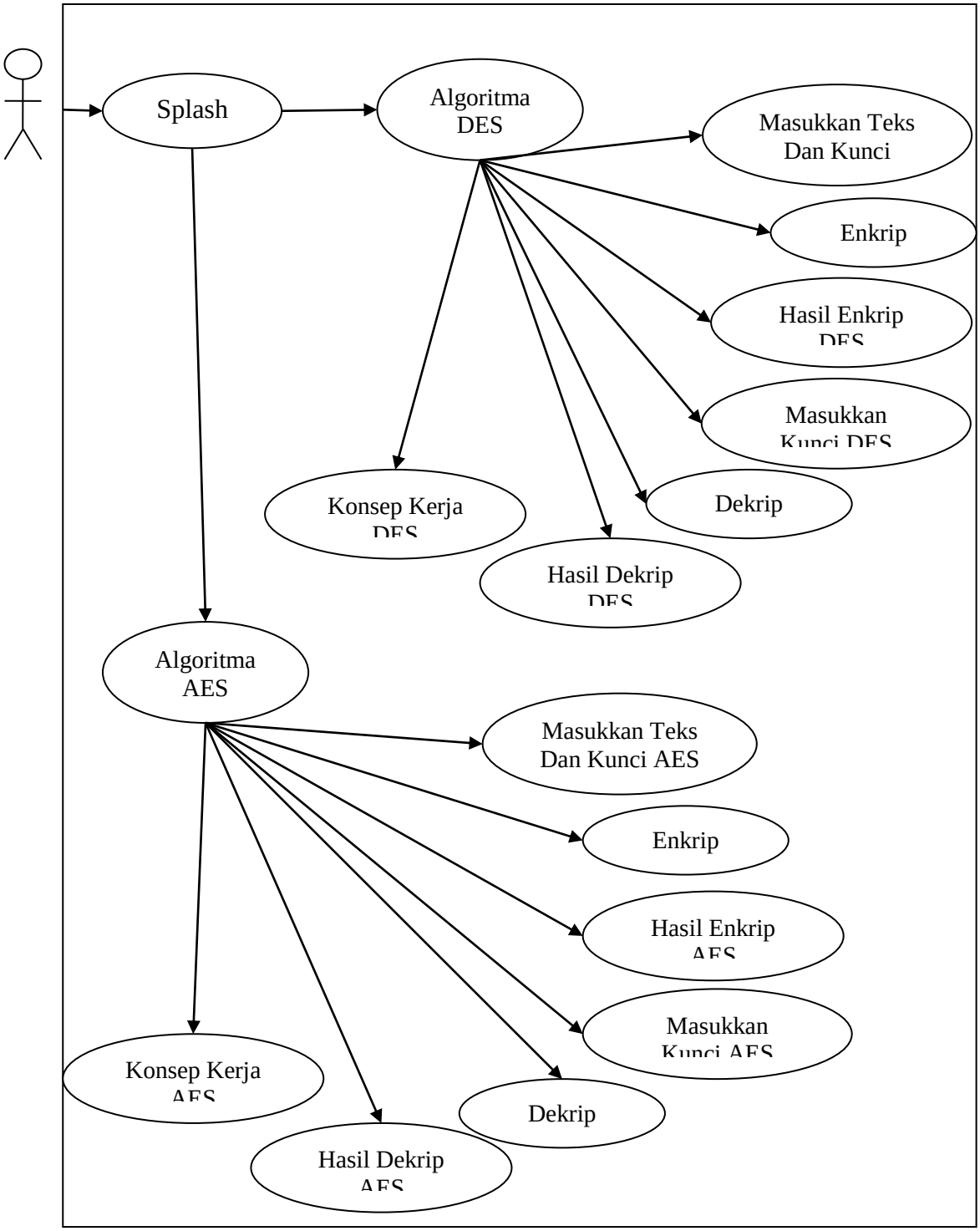
III.4. UML

Penggambaran UML menggunakan diagram *use-case* yang selanjutnya setiap proses bisnis yang terjadi akan diperjelas dengan diagram *activity* lalu diilustrasikan secara detail

menggunakan diagram *sequence*. Aktor dan pelaku yang terlibat dalam sistem adalah sebagai berikut :

III.4.1. *Use Case* Diagram

Adapun *use-case* diagram dapat dilihat pada gambar III.9 berikut ini



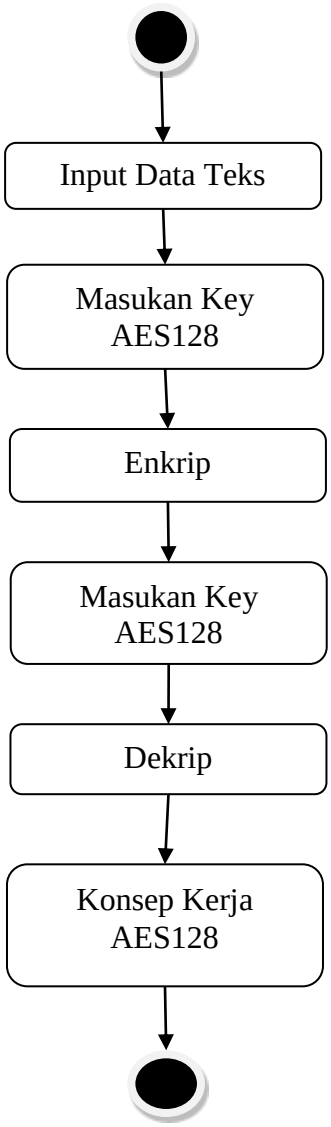
Gambar III.1. ***Use Case Diagram***

III.4.2. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut adalah gambar *activity diagram* dari sitem yang dirancang yaitu :

1. Activity Diagram Enkrip dan dekrip AES128

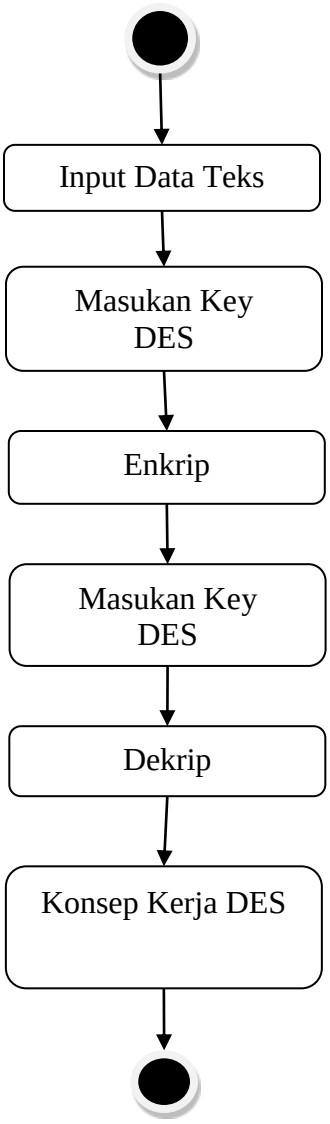
Activity diagram Enkrip dan dekrip AES128, dapat dilihat pada gambar dibawah ini :



Gambar III.2. *ActivityDiagram* Enkrip dan Dekrip AES128

2. *Activity Diagram Enkrip* dan *dekrip DES*

Activity diagram Enkrip dan dekrip DES dapat dilihat pada gambar dibawah ini :



Gambar III.3. ***Activity Diagram Enkrip dan Dekrip DES***

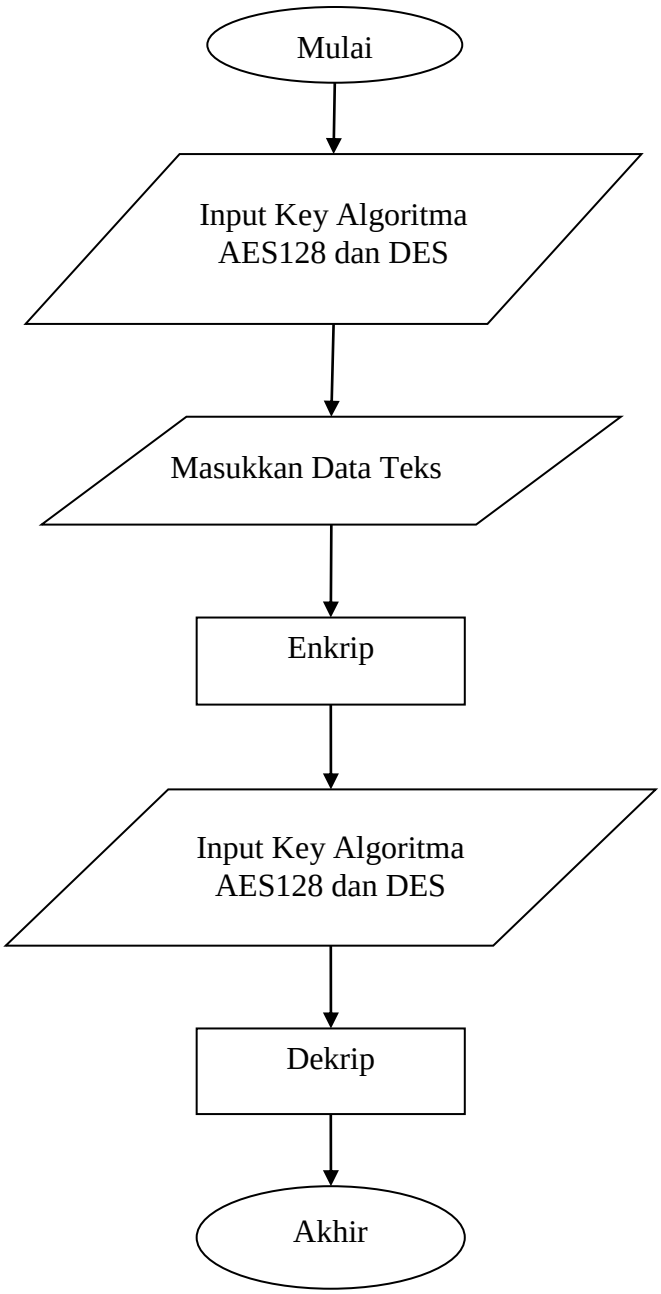
III.4.3. *Flowchart*

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran inin dinyatakan dengan simbol. Dengan demikian setiap symbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan

adanya *flowchart* urutan proses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah.

III.4.3.1. *Flowchart* Enkripsi AES128 dan DES

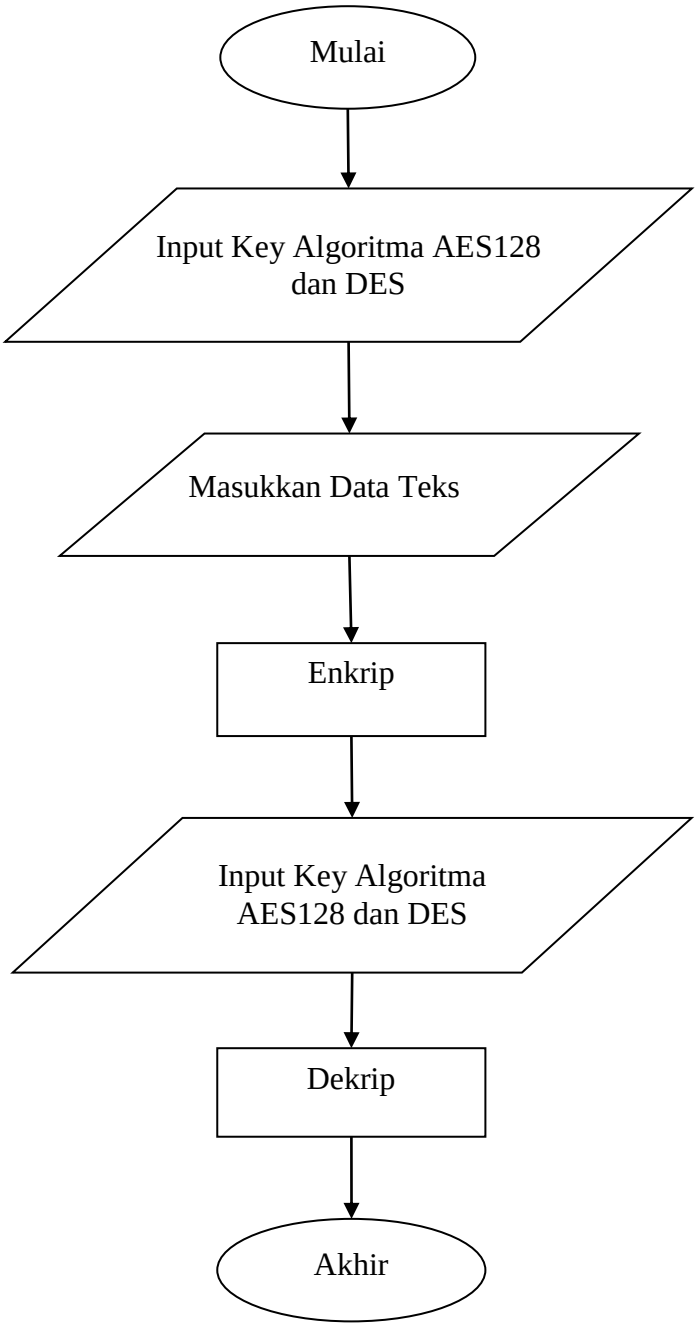
Flowchart ini dibuat untuk menjelaskan proses jalannya enkripsi dan dekripsi DES pada program, seperti pada gambar III.5 berikut ini.



Gambar III.4. Flowchart Proses Enkripsi AES128 dan DES

III.4.3.2. *Flowchart* Enkrip Dan Dekrip AES128 dan DES

Flowchart ini dibuat untuk menjelaskan proses jalannya enkripsi dan deekripsi AES128 pada program, seperti pada gambar III.5 berikut ini.



Gambar III.5. Flowchart Proses Dekripsi AES128 dan DES

III.5. Spesifikasi Perangkat

Dalam perancangan aplikasi untuk perangkat *Android Mobile Phone* ini penulis menggunakan beberapa perangkat agar aplikasi ini dapat berjalan lancar dan sesuai dengan yang diharapkan, yaitu sebagai berikut ini :

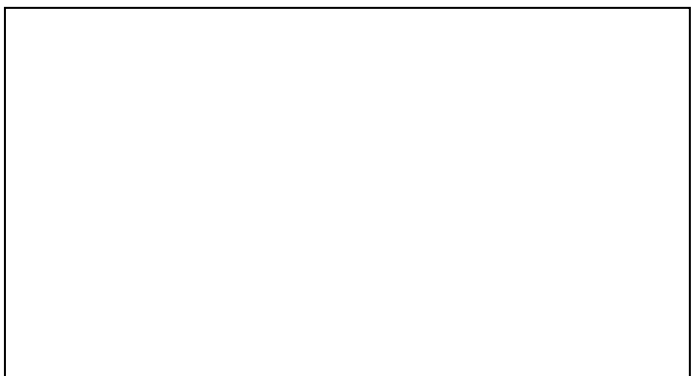
1. Perangkat Keras (*Hardware*)
   a. Komputer yang setara *AMD*
   b. *Smartphone Android* dengan OS 4.2.1 atau diatasnya
   c. *Mouse, Keyboard* dan Monitor
2. Perangkat Lunak (*Software)*
   a. *Operating System,* OS yang dipergunakan dalam perancangan adalah *Windows* 7 dan untuk pengujian adalah OS Android pada perangkat *mobile.*
   b. *Eclipse ADT (Android Development Tools),* sebagai editor *source code Java.*
   c. JDK *Java 7.0,* sebagai bahasa program.

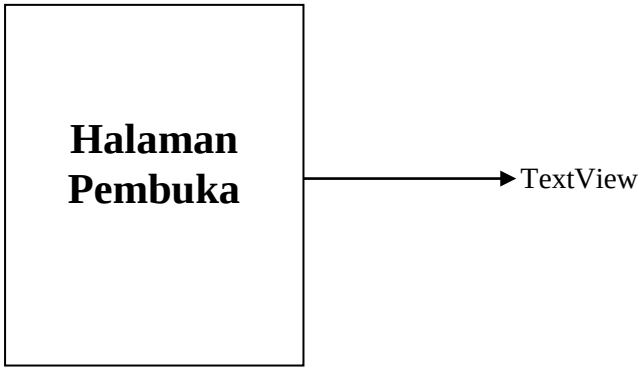
III.6. Desain Sistem

Dalam proses perancangan ini akan dijelaskan beberapa rancangan aplikasi yang akan dibangun yaitu sebagai berikut :

III.6.1. Rancangan Awal Pembukaan Program

Gambar III.6 ini dibuat untuk menampilkan rancangan awal ketika program pertama kali dibuka.





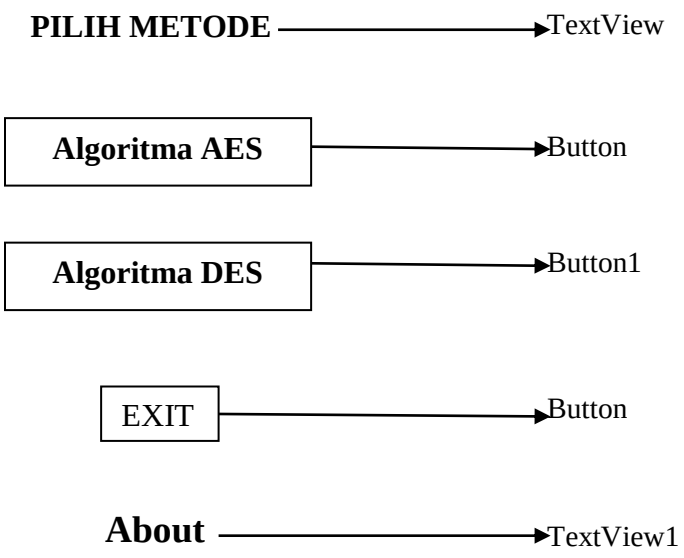
Gambar III.6. Form Halaman Pembuka

Penjelasan sebagai berikut :

1. Textview : untuk menampilkan halaman pembuka

III.6.2. Rancangan Menu Pilihan

Berikut ini adalah rancangan form menu utama yang dapat lihat pada gambar III.7 di bawah ini.



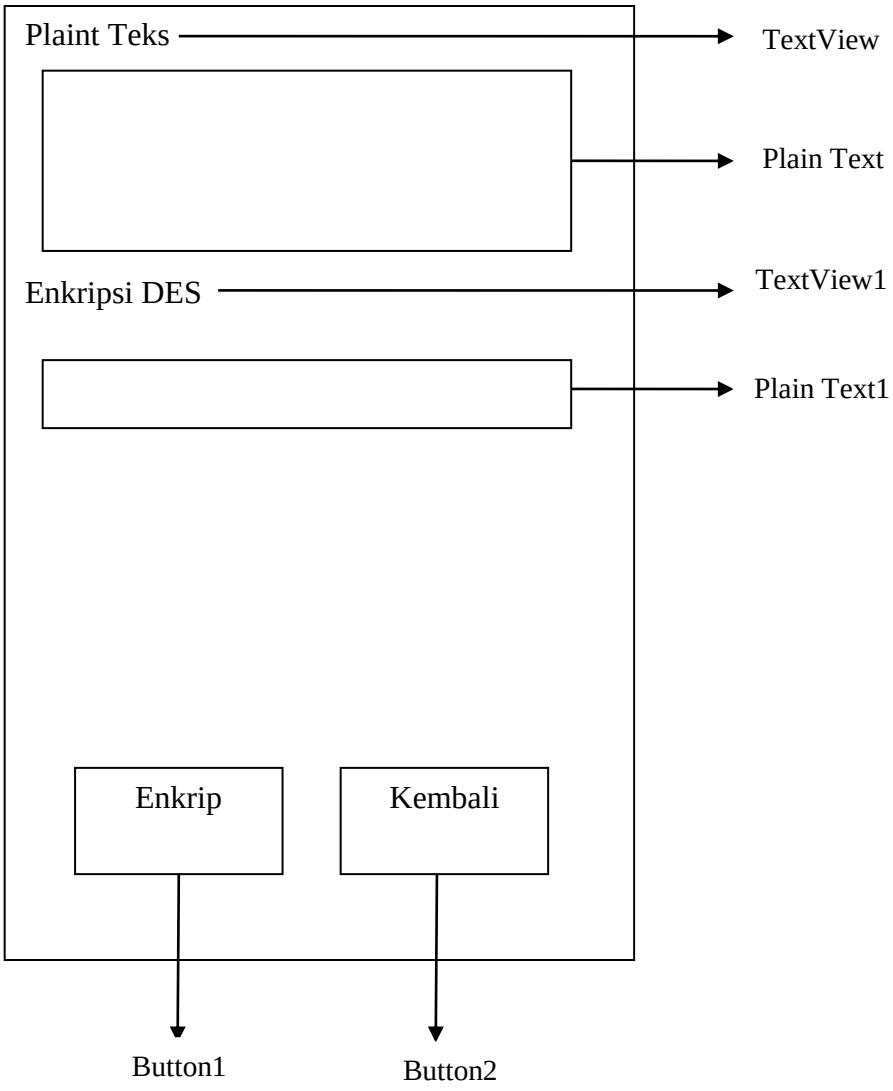
Gambar III.7 Form Menu

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan tulisan pilih metode
2. Button : untuk memilih algoritma AES
3. Button1 : untuk memilih algoritma DES
4. Button2 : untuk keluar dari aplikasi
5. TextView1: untuk menampilkan tentang aplikasi

III.6.3. Rancangan Form Enkrip DES

Berikut ini adalah rancangan form Enkrip DES yang dapat lihat pada gambar III.8 di bawah ini.



Gambar III.8. Form Enkrip DES

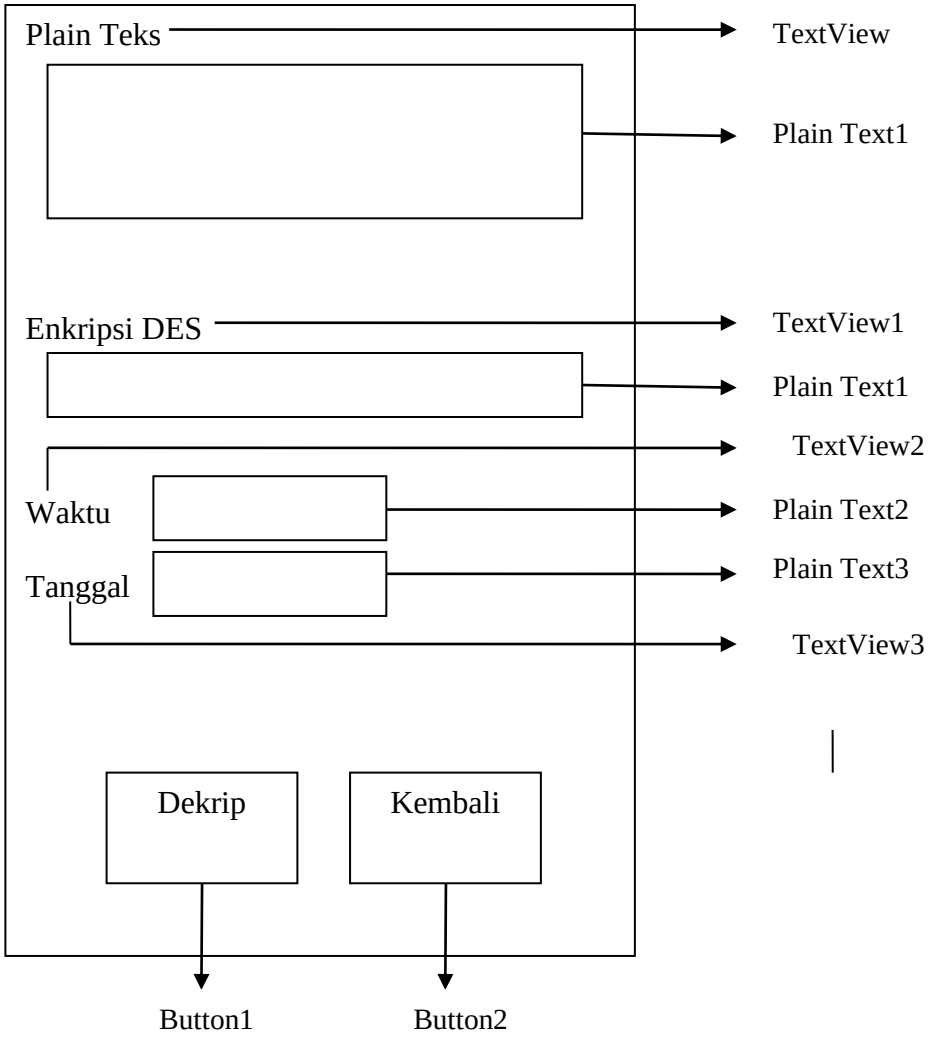
Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks dimasukkan
2. TextView1: untuk menunjukkan tempat kunci
3. Button1 : untuk memilih proses enkrip

4. Button2 : untuk memilih prose dekrip
5. PlainText : untuk memasukkan kunci DES
6. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.4. Rancangan Form Hasil Enkripsi DES

Berikut ini adalah rancangan form hasil enkripsi yang dapat lihat pada gambar III.9 di bawah ini.



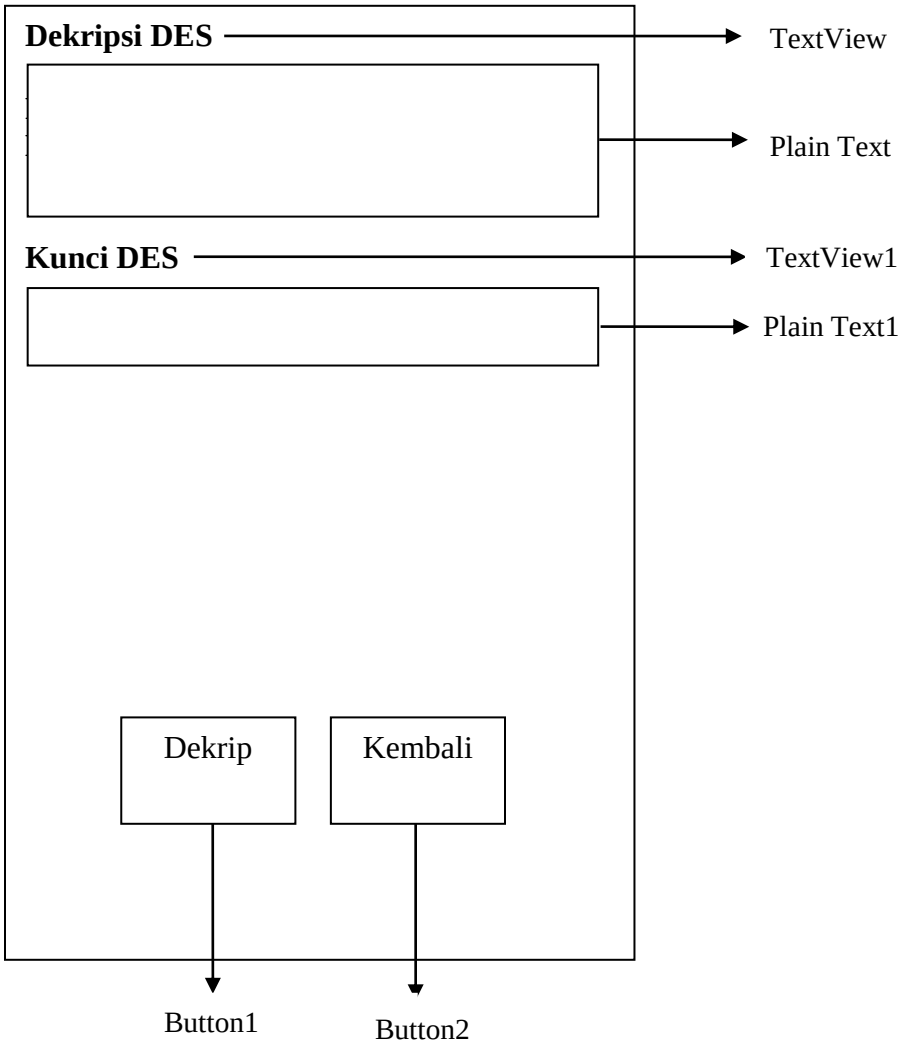
Gambar III.9. Form Hasil Enkripsi DES

Penje

1. TextView : untuk menunjukkan tempat teks
2. TextView1 : untuk menunjukkan tempat kunci
3. TextView2 : untuk menunjukkan waktu
4. TextView3 : untuk menunjukkan Tanggal
5. Button1 : untuk memilih proses dekrip
6. Button2 : untuk kembali ke menu
7. PlainText1 : untuk menampilkan hasil enkrip teks
8. PlainText2 : untuk menampilkan hasil enkrip kunci
9. PlainText3 : untuk menampilkan pengukuran waktu enkrip
10. PlainText4 : untuk menampilkan tanggal pengengkripan

III.6.5. Rancangan Form Dekrip DES

Berikut ini adalah rancangan form hasil Dekripsi yang dapat lihat pada gambar III.10 di bawah ini.



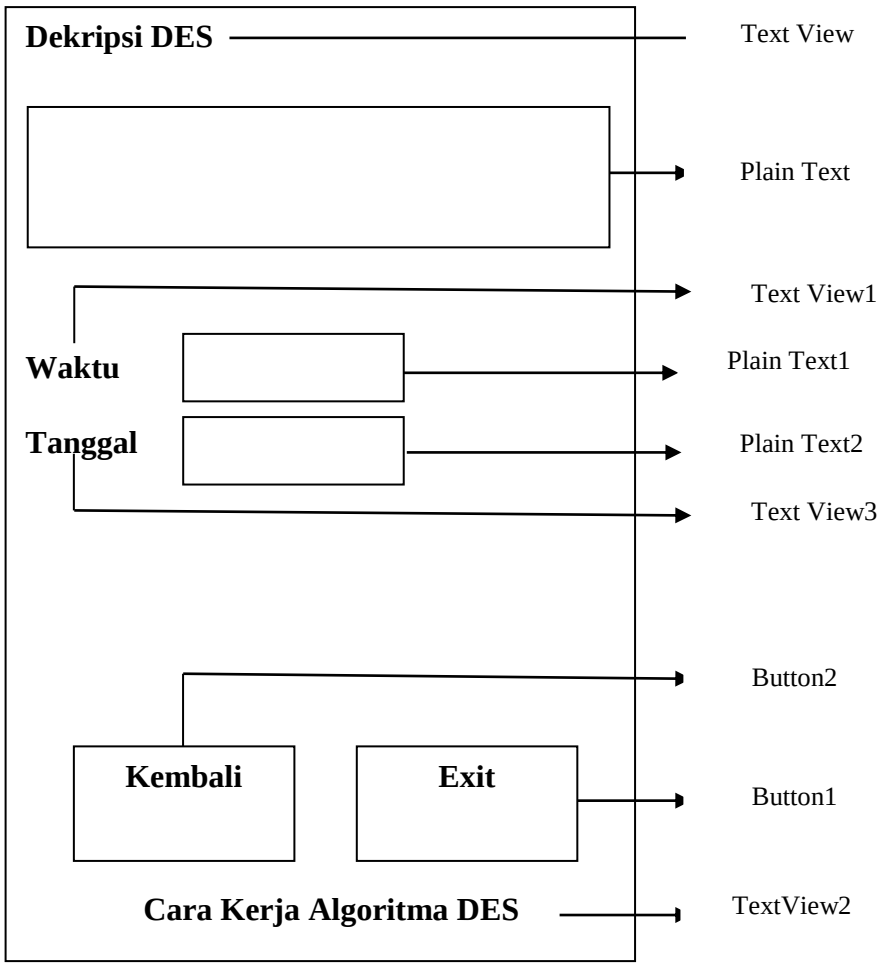
Gambar III.10. Form Dekrip DES

Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks dimasukkan
2. TextView1: untuk menunjukkan tempat kunci
3. Button1 : untuk memilih proses enkrip
4. Button2 : untuk memilih prose dekrip
5. PlainText : untuk memasukkan kunci DES
6. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.6. Rancangan Form Hasil Dekrip DES

Berikut ini adalah rancangan form hasil Dekripsi yang dapat lihat pada gambar III.11 di bawah ini:



Gambar III.11. Form Hasil Dekripsi DES

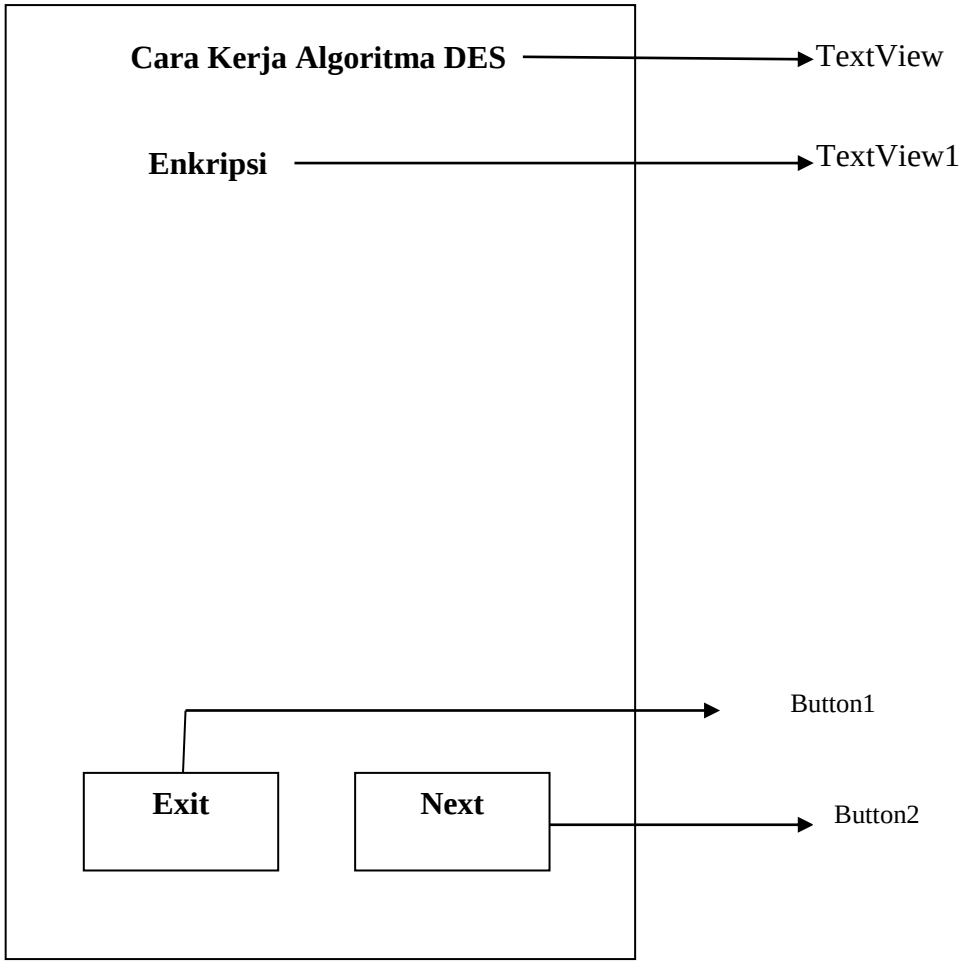
Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks
2. TextView1 : untuk menunjukkan tempat kunci
3. TextView2 : untuk masuk layout cara kerja DES
4. TextView3 : untuk menunjukkan tanggal

5. Button1 : untuk keluar dari aplikasi
6. Button2 : untuk kembali ke menu
7. PlainText1 : untuk menampilkan hasil enkrip teks
8. PlainText2 : untuk menampilkan pengukuran waktu enkrip
9. PlainText3 : untuk menampilkan tanggal pendekripan

III.6.7. Rancangan Form Cara Kerja Ke-1 Algoritma DES

Berikut ini adalah rancangan form Cara Kerja Algoritma DES yang dapat lihat pada gambar III.12 di bawah ini.



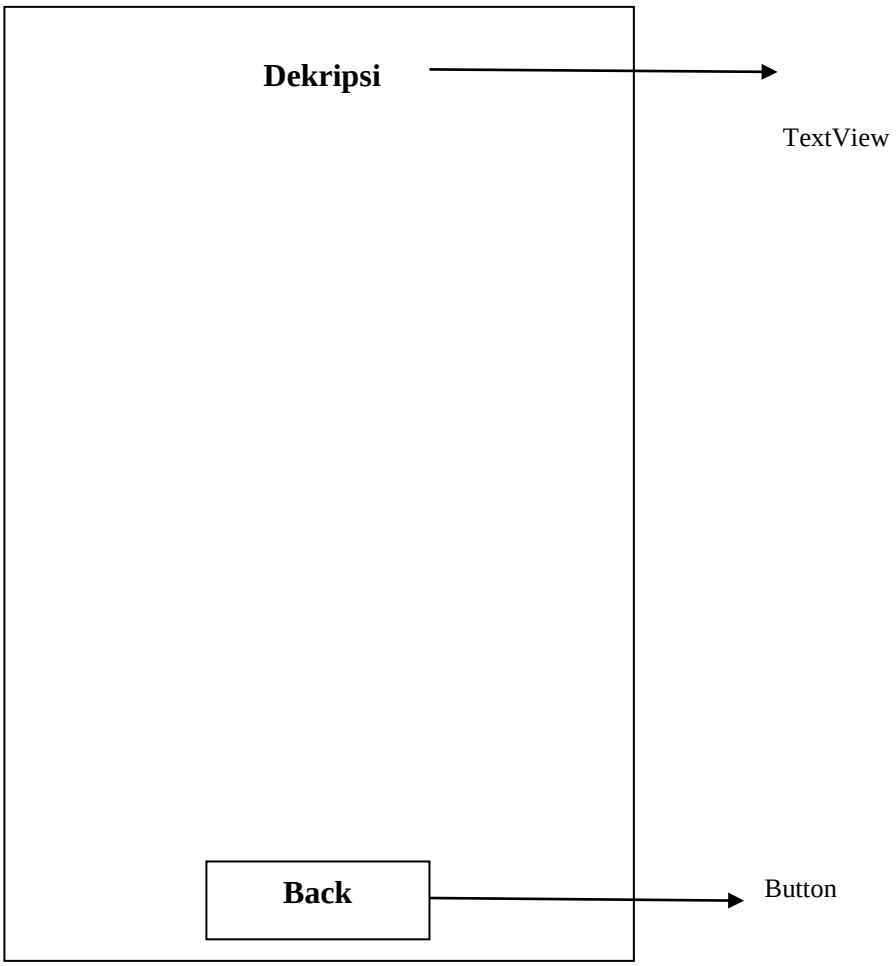
Gambar III.12. Form Cara Kerja Ke-1 Algoritma DES

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan judul cara kerja algoritma DES
2. TextView1 : untuk menampilkan judul
3. Button1 : untuk keluar dari aplikasi
4. Button2 : untuk lanjut ke layout selanjutnya
5. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.8. Rancangan Form Cara Kerja Ke-2 Algoritma DES

Berikut ini adalah rancangan form Cara Kerja Algoritma DES yang dapat lihat pada gambar III.13 di bawah ini.



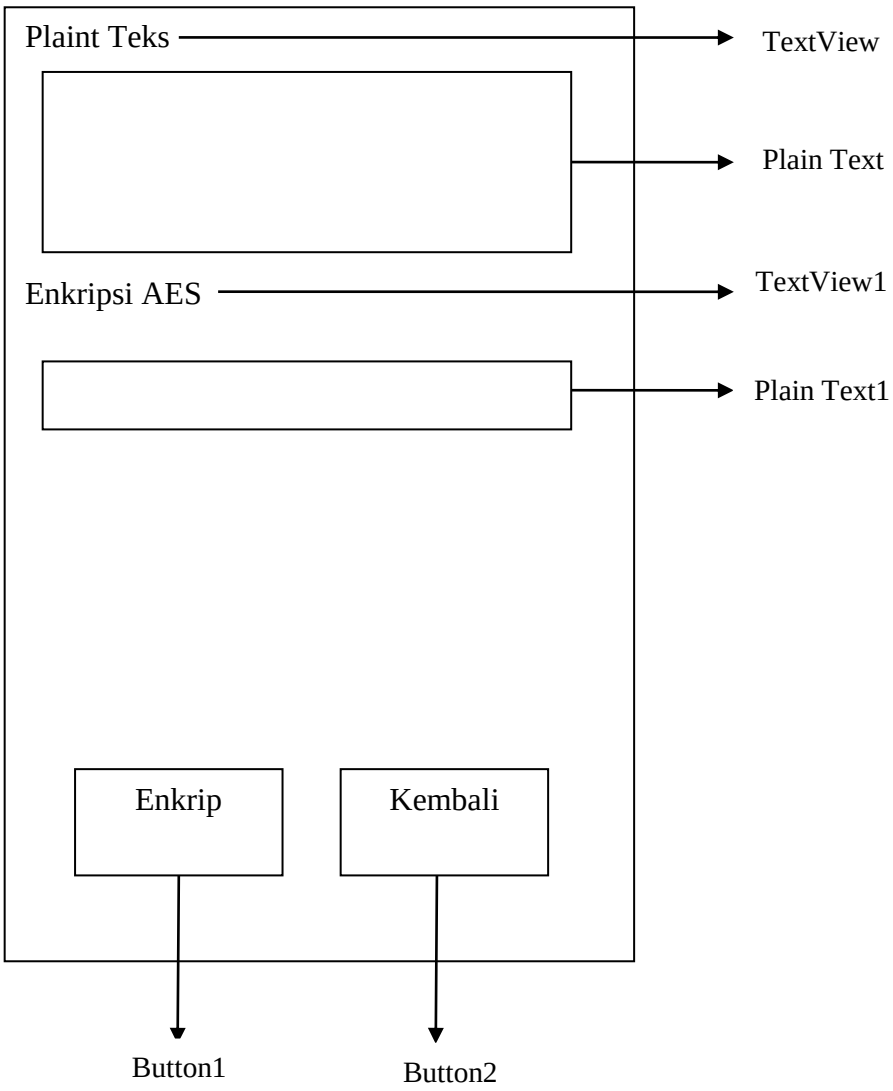
Gambar III.13. Form Cara Kerja Ke-2 Algoritma DES

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan judul
2. Button : untuk kembali ke penjelasan Enkripsi DES

III.6.9. Rancangan Form Enkrip AES

Berikut ini adalah rancangan form Enkrip AES yang dapat lihat pada gambar III.14 di bawah ini.



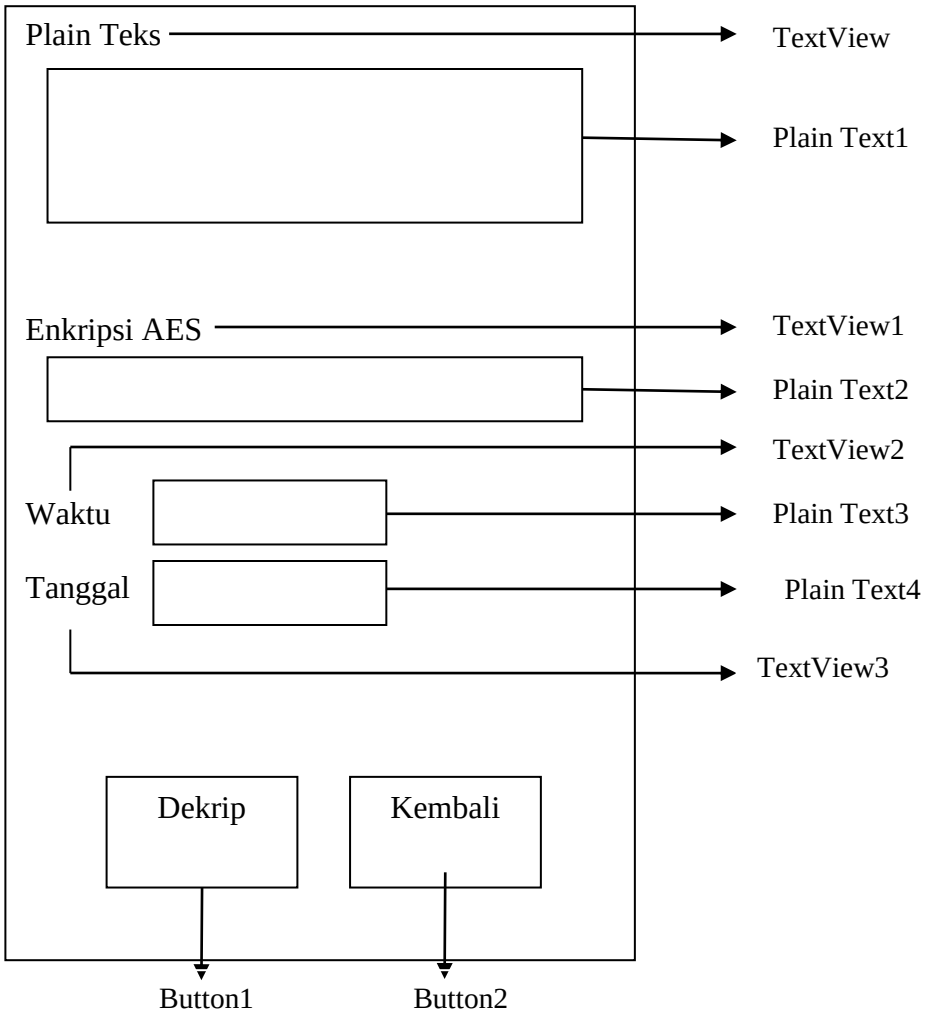
Gambar III.14. Form Enkrip AES

Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks dimasukkan
2. TextView1 : untuk menunjukkan tempat kunci
3. Button1 : untuk memilih proses enkrip
4. Button2 : untuk memilih prose dekrip
5. PlainText : untuk memasukkan kunci AES
6. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.10. Rancangan Form Hasil Enkripsi AES

Berikut ini adalah rancangan form hasil enkripsi yang dapat lihat pada gambar III.15 di bawah ini.



Gambar III.15. Form Hasil Enkripsi AES

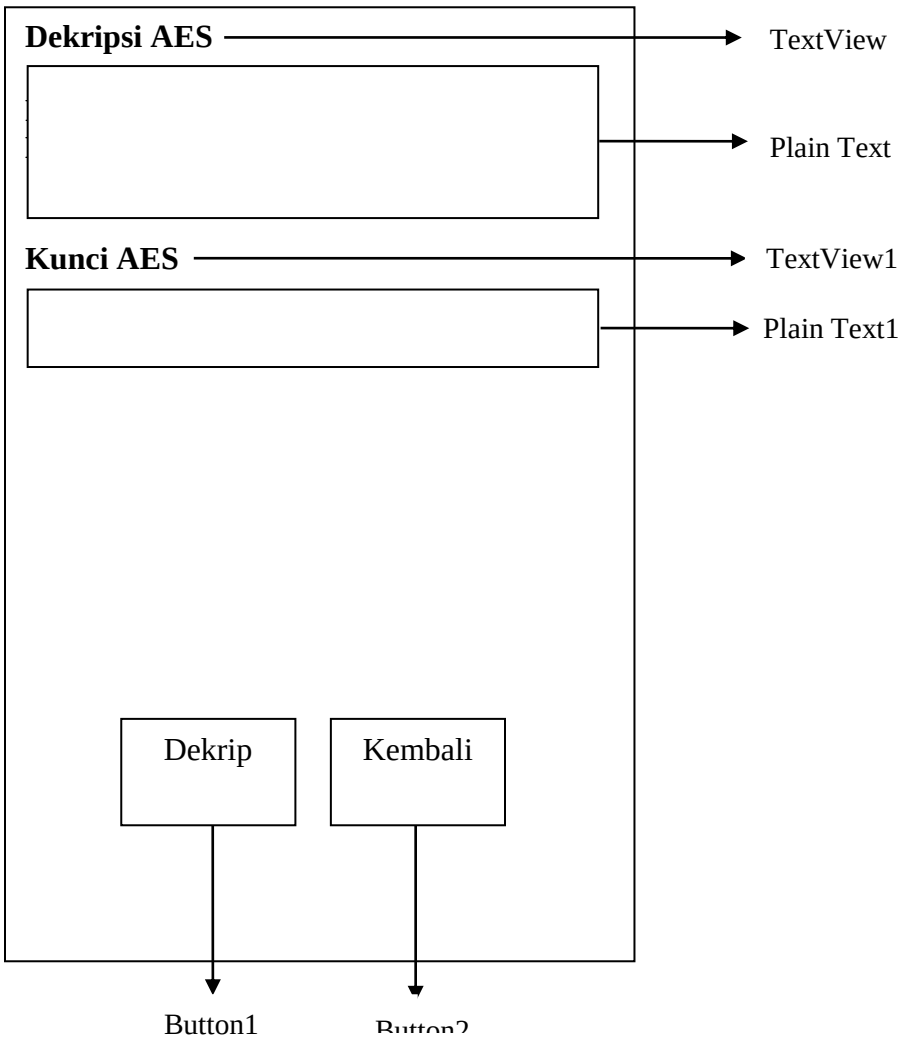
Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks
2. TextView1 : untuk menunjukkan tempat kunci
3. TextView2 : untuk menunjukkan waktu
4. TextView3 : untuk menunjukkan tanggal
5. Button1 : untuk memilih proses dekrip
6. Button2 : untuk kembali ke menu
7. PlainText1 : untuk menampilkan hasil enkrip teks

8. PlainText2 : untuk menampilkan hasil enkrip kunci
9. PlainText3 : untuk menampilkan pengukuran waktu enkrip
10. PlainText4 : untuk menampilkan tanggal pengengkripan

III.6.11. Rancangan Form Dekrip AES

Berikut ini adalah rancangan form hasil Dekripsi yang dapat lihat pada gambar III.16 di bawah ini:



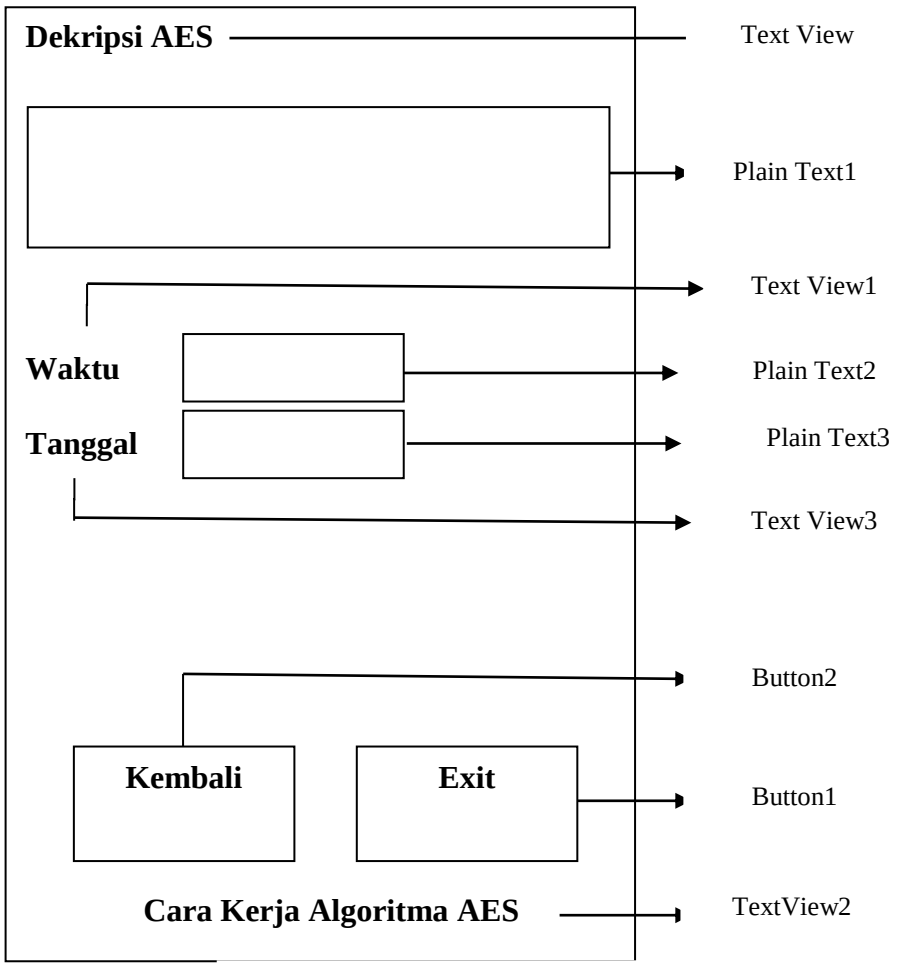
Gambar III.16. Form Dekrip AES

Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks dimasukkan
2. TextView1 : untuk menunjukkan tempat kunci
3. Button1 : untuk memilih proses enkrip
4. Button2 : untuk memilih prose dekrip
5. PlainText : untuk memasukkan kunci AES
6. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.12. Rancangan Form Hasil Dekrip AES

Berikut ini adalah rancangan form hasil Dekripsi yang dapat lihat pada gambar III.17 berikut ini:



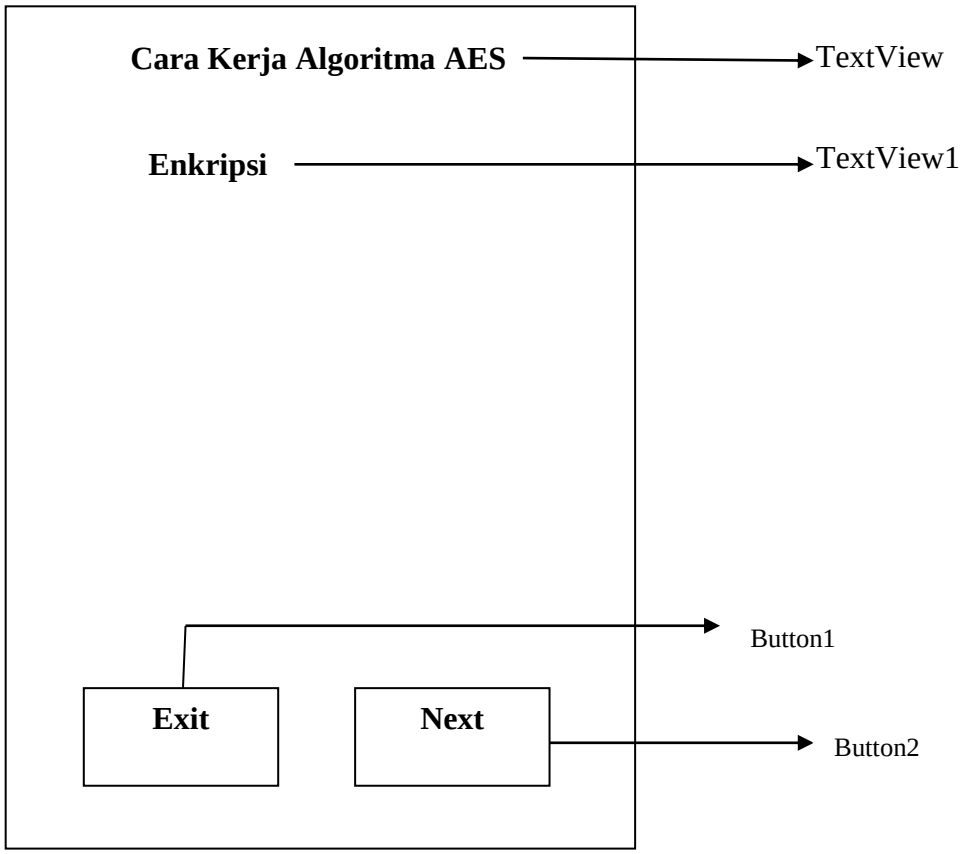
Gambar III.17. Form Hasil Dekripsi

Penjelasan sebagai berikut :

1. TextView : untuk menunjukkan tempat teks
2. TextView1 : untuk menunjukkan waktu
3. TextView2 : untuk masuk layout cara kerja AES
4. TextView3 : untuk menunjukkan tanggal
5. Button1 : untuk keluar dari aplikasi
6. Button2 : untuk kembali ke menu
7. PlainText1 : untuk menampilkan hasil enkrip teks
8. PlainText2 : untuk menampilkan pengukuran waktu enkrip
9. PlainText3 : untuk menampilkan tanggal pendekripan

III.6.13. Rancangan Form Cara Kerja Ke-1 Algoritma AES

Berikut ini adalah rancangan form Cara Kerja Algoritma AES yang dapat lihat pada gambar III.18 di bawah ini.



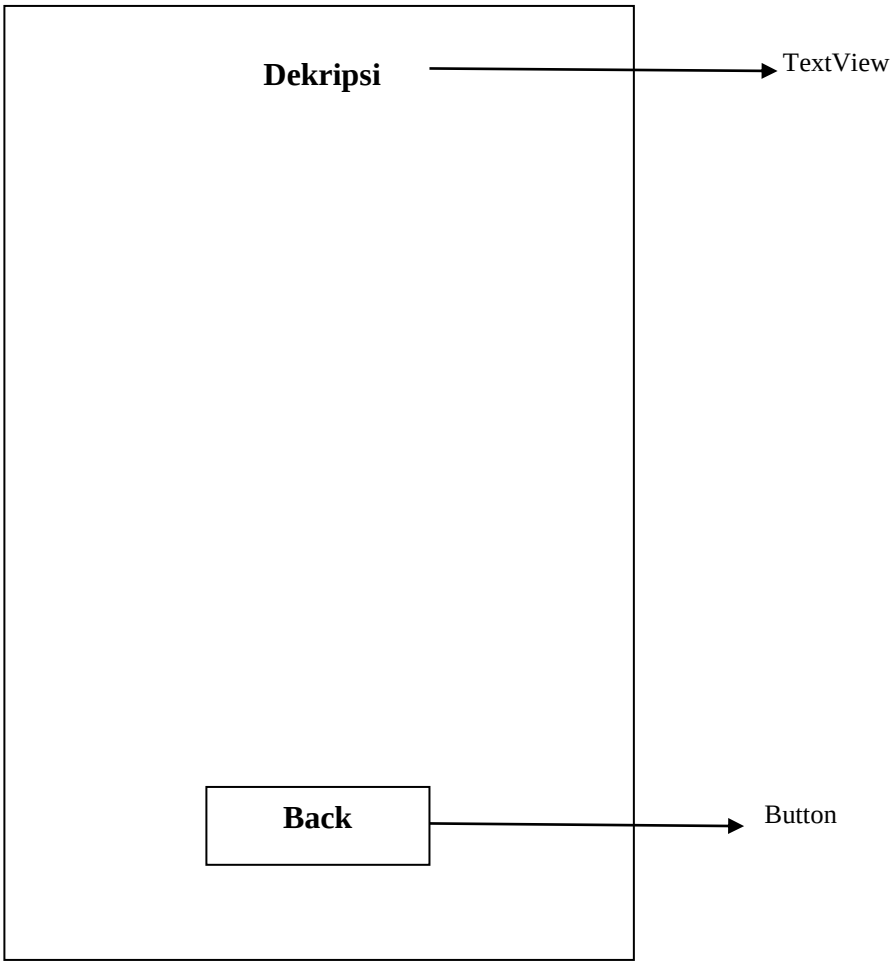
Gambar III.18. Form Cara Kerja Ke-1 Algoritma AES

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan judul cara kerja algoritma AES
2. TextView1 : untuk menampilkan judul
3. Button1 : untuk keluar dari aplikasi
4. Button2 : untuk lanjut ke layout selanjutnya
5. PlainText : untuk memasukkan teks yang akan di enkrip

III.6.14. Rancangan Form Cara Kerja Ke-2 Algoritma AES

Berikut ini adalah rancangan form Cara Kerja Algoritma AES yang dapat lihat pada gambar III.19 di bawah ini :



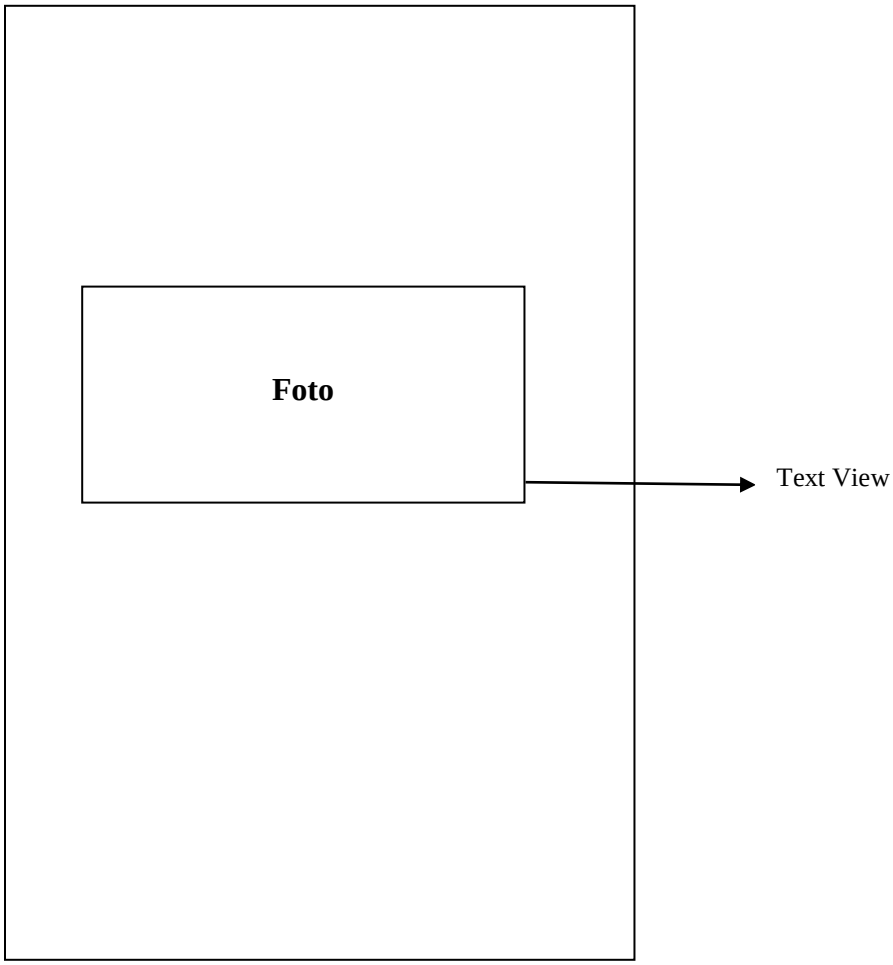
Gambar III.19. Form Cara Kerja Ke-2 Algoritma AES

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan judul
2. Button : untuk kembali ke penjelasan Enkripsi AES

III.6.14. Rancangan Form Tentang Aplikasi

Berikut ini adalah rancangan form pengenalan tentang aplikasi yang dapat lihat pada gambar III.20 berikut ini :



Gambar III.20. Form Tentang Aplikasi

Penjelasan sebagai berikut :

1. TextView : untuk menampilkan foto, judul skripsi dan nama