BAB 4 PERANCANGAN

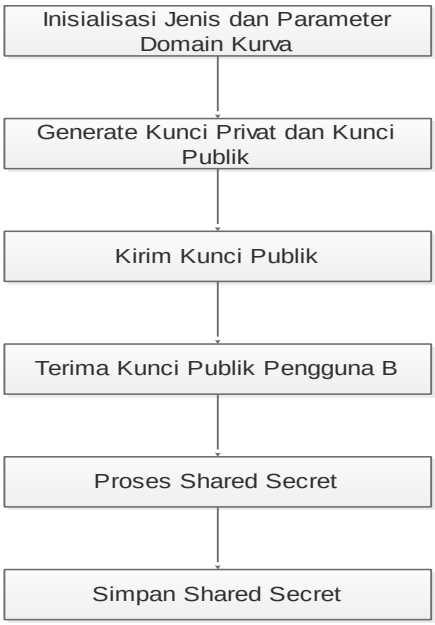
Bab ini akan membahas mengenai perancangan proses algoritma serta perancangan sistem yang akan dibuat. Pada perancangan proses algoritma bertujuan untuk merancang penggunaan algoritma beserta komponen penyusun dalam algoritma tersebut yang akan digunakan dalam membangun sistem enkripsi SMS pada telepon selular berbasis android dengan menggunakan metode ECDH dan AES. Pada perancangan sistem digunakan untuk memberi gambaran tahapan pembuatan sistem enkripsi SMS pada telepon selular berbasis android dengan menggunakan metode ECDH dan AES. Perancangan lingkungan pengujian dilakukan guna mengetahui lingkungan pengujian yang akan digunakan dalam menguji sistem.

4.1 Perancangan Proses Algoritma

Perancangan proses algoritma bertujuan untuk merancang penggunaan algoritma beserta komponen penyusun dalam algoritma tersebut. Tahap ini terdiri dari tiga skema proses bisnis algoritma, yaitu skema proses tukar kunci, proses enkripsi dan dekripsi. Contoh perhitungan manual juga disertakan pada tahap perancangan proses algoritma guna untuk mengetahui bagaimana proses perhitungan yang ada pada algoritma AES.

4.1.1 Proses Tukar Kunci

Proses pertukaran kunci dilakukan dengan menggunakan metode Elliptic Curve Diffie-Hellman. Parameter domain elliptic curves yang digunakan adalah secp256k1 untuk pembuatan *shared secret* yang kemudian digunakan untuk kunci enkripsi dan dekripsi.



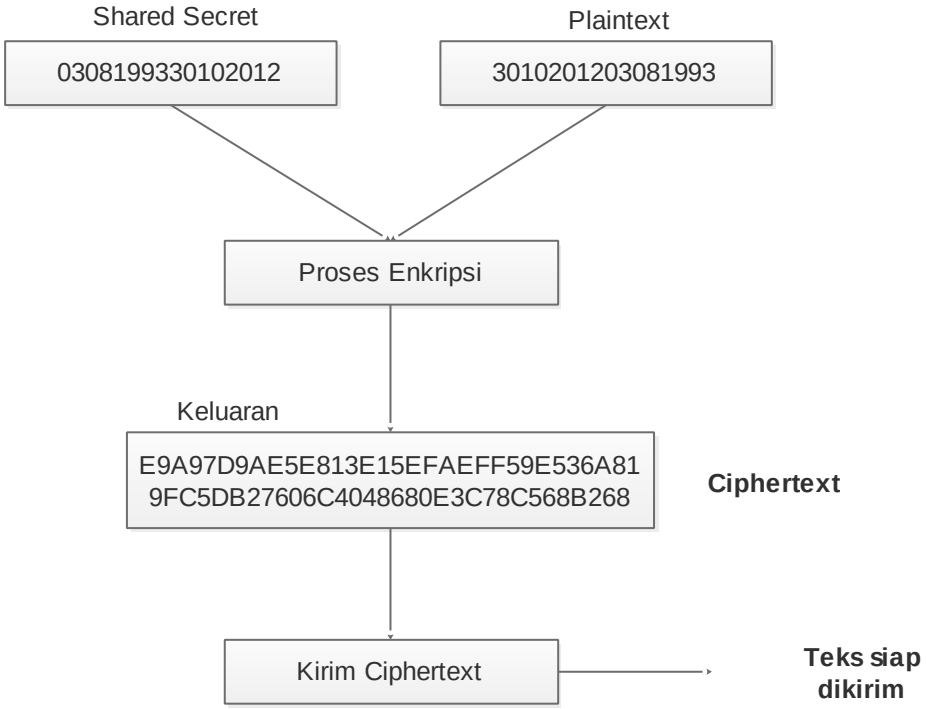
Gambar 4.1 Proses Tukar Kunci

Proses tukar kunci yang sebagaimana terlihat pada gambar 4.1 mempunyai skema proses sebagai berikut:

a. Sistem melakukan inisialisasi jenis kurva serta parameter domain kurva yang akan digunakan.
b. Sistem melakukan generate kunci privat dan kunci publik.
c. Kunci publik yang telah dilakukan proses generate dikirim kepada host lain.
d. Sistem menerima kunci publik dari host lain.
e. Sistem melakukan penghitungan guna mendapatkan kunci *shared secret*

4.1.2 Proses Enkripsi

Proses enkripsi data *plaintext* dilakukan ketika telah tersedia kunci rahasia bersama (*shared secret*) serta data *plaintext* yang dibutuhkan. Enkripsi data *plaintext* dilakukan setelah proses tukar kunci.



Gambar 4.2 Proses Enkripsi

Proses enkripsi yang sebagaimana terlihat pada gambar 4.2 mempunyai proses sebagai berikut:

a. Sistem mendapatkan komponen penyusun proses enkripsi yaitu *plaintext* serta *shared secret* yang dibutuhkan.
b. Sistem melakukan proses enkripsi dengan menggunakan komponen penyusun *plaintext* dan kunci yang sesuai.

c. Proses enkripsi tersebut akan menghasilkan keluaran berupa *ciphertext* yang siap untuk ditampilkan pada *textview* di dikirimkan.

Berikut merupakan contoh perhitungan manual enkripsi AES, yaitu:

- Dimisalkan data yang diperoleh adalah sebagai berikut:
  - *Plaintext* : 54 77 6f 20 4f 6e 65 20 4e 69 6e 65 20 54 77 6f (Hex)
  - *Key(shared Secret)*: 54 68 61 74 73 20 6d 79 20 4b 75 6e 67 20 46 75 (Hex)

1. Proses inisialisasi kunci

   Pertama-tama pengguna memberikan inputan pada perangkat lunak pengujian berupa *external key* (kunci eksternal) sebesar 16 karakter (128 bit). Kunci eksternal kemudian dikonversi ke dalam bentuk heksadesimalnya sebesar 32 heksadesimal dan dibagi menjadi 16 bagian, dan diisikan pada *state array* dimulai dari kotak kiri atas kemudian kebawah. Setelah pengisian *state array* maka dilakukan *round key* 1 seperti lanngkah berikut:

   a. Transformasi *RotCol* terhadap kolom ke-4 (paling kanan) dari kunci eksternal

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

W'[0]	W'[1]	W'[2]	W'[3]
54	73	20	20
68	20	4B	46
61	6D	75	75
74	79	6E	67

Gambar 4.3 Tahapan *RotCol* pada *Round key* 1

   b. Transformasi *SubBytes* terhadap kolom hasil dari *RotCol* yang di dapa pada tahap a (w'[3])

w[3]	S-Box	W'[3]
67	>	B7
20	>	5A
46	>	9D
75	>	85

Gambar 4.4 Tahapan *SubBytes* pada *Round key* 1

   c. Lakukan XOR dari kolom hasil transformasi *SubBytes* (tahap b) dengan *Round Constant* dan kolom kunci eksternal (kolom paling kiri) sehingga meghasilkan kolom 1 dari *round key* 1.

w[3]		*Round Constant*		g(w[3])
B7		01		B6
5A	⊕	00	=	5a
9D		00		9D
85		00		85

w[0]		g(w[3])		w[4]
54	⊕	B6	=	E2
68		5a		32
61		9D		FC
74		85		F1

Gambar 4.5 Tahapan XOR dengan Round Constant dan kunci Eksternal

d. Untuk mendapatkan hasil kolom lainya, lakukan XOR kolom sebelumnya dengan kolom yang sama pada *round key* sebelumnya.

w[4]		w[1]		w[5]
E2	⊕	73	=	91
32		20		12
FC		6D		91
F1		79		88

w[5]		w[2]		w[6]
91	⊕	20	=	B1
12		4B		59
91		75		E4
88		6E		E6

W[6]		W[3]		W[7]
B1	⊕	67	=	D6
59		20		79
E4		46		A2
E6		75		93

Gambar 4.6 Tahapan XOR Round Constant dengan Kunci Eksternal

e. Dengan diketahui nilai w[4] w[5], w[6] dan w[7], maka digabungkan menjadi satu dalam *state array* dan menghasilkan *round key* pertama.

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

Gambar 4.7 *Round key* 1

Setelah mendapatkan hasil dari *round key* 1, maka dilanjutkan untuk mendapat *round key* 2 sampai *round key* 10 dengan mengulangi langkah seperti pada *round key* 1.

2. Proses Enkripsi

Pada awal proses enkripsi, 16 byte data masukan $in_0, in_1, \dots, in_{15}$, disalin ke dalam array state. Byte input berisi *plaintext* sebesar 16 karakter (32 heksadesimal). Proses awal yang dilakukan adlah *initial round*. *Initial round* merupakan proses XOR antara *state array* dengan external key ($Initial\ Round = state\ array \oplus external\ key$) dan menghasilkan *state array* round 1 seperti pada gambar 4.8

State Matriks

54	4F	4E	20
77	6E	69	54
6F	65	6E	77
20	20	65	6F

$\oplus$

Roundkey 0

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

00	3C	6E	47
1F	4E	22	74
0E	8	1B	31
54	59	0B	1A

Gambar 4.8 *Initial round*

Setelah dilakukan *initial round*, langkah selanjutnya adalah proses enkripsi round 1 dilakukan dalam tahap sebagai berikut:

a. Setelah mendapatkan hasil *state array* round 1, subtitusikan setiap byte dari *state array* tersebut dengan byte pada AES S-Box, sehingga akan didapat array state yang baru, yaitu:

00	3C	6E	47
1F	4E	22	74
0E	8	1B	31
54	59	0B	1A

>	>	>	>
>	>	>	>
>	>	>	>
>	>	>	>

63	EB	9F	A0
C0	2F	93	92
AB	30	AF	C7
20	CB	2B	A2

Gambar 4.9 Proses *SubBytes* terhadap *State array* Round 1

b. Setelah melakukan subtitusi, lakukan shift rows pada state matriks. Proses Shift Row merupakan pergeseran secara *wrapping* (siklik) pada 3 baris terakhir dari *array state*.

63	EB	9F	A0
C0	2F	93	92
AB	30	AF	C7
20	CB	2B	A2

			[1]	[2]	[3]	[4]
		[1]	[2]	[3]	[4]	
	[1]	[2]	[3]	[4]		
[1]	[2]	[3]	[4]			

63	EB	9F	A0
2F	93	92	C0
AF	C7	AB	30
A2	20	CB	2B

Gambar 4.10 Transformasi *ShiftRows* pada *State array* Round 1

c. Setelah melakukan proses *ShiftRows*, lakukan operasi *MixColumn* pada *State array*

Multiplication Array

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

State array

63	EB	9F	A0
2F	93	92	C0
AF	C7	AB	30
A2	20	CB	2B

=

BA	84	E8	1B
75	A4	8D	40
F4	8D	6	7D
7A	32	0E	5D

Gambar 4.11 Transformasi *MixColumn* pada *State array* Round 1

d. Setelah proses *MixColumn*, lakukan transformasi *AddRoundKey* pada *state array* round 1. Transformasi *AddRoundKey* merupakan XOR antara *state array* round ke 1 dengan *round key* ke 1.

BA	84	E8	1B
75	A4	8D	40
F4	8D	6	7D
7A	32	0E	5D

⊕

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

58	15	59	CD
47	B6	D4	39
8	1C	E2	DF
8B	BA	E8	CE

Gambar 4.12 Transformasi *AddRoundKey* pada *State array* Round 1

e. Untuk *Round key* berikutnya, ulangi proses Subtitusi, *ShiftRow*, *MixColumn*, serta *AddRoundKey* hingga didapat Round key 9. Pada final round (round 10), hilangkan proses *MixColumn*, hal tersebut dikarenakan memperberat proses sehingga menjadi tidak efektif. Setelah final round, maka akan didapatkan hasil *ciphertext*, yaitu:

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

=

)	W	@	->
Ã		“	
P			×
_	Ö	³	:

Gambar 4.13 *Ciphertext* Hasi dari Proses Enkripsi

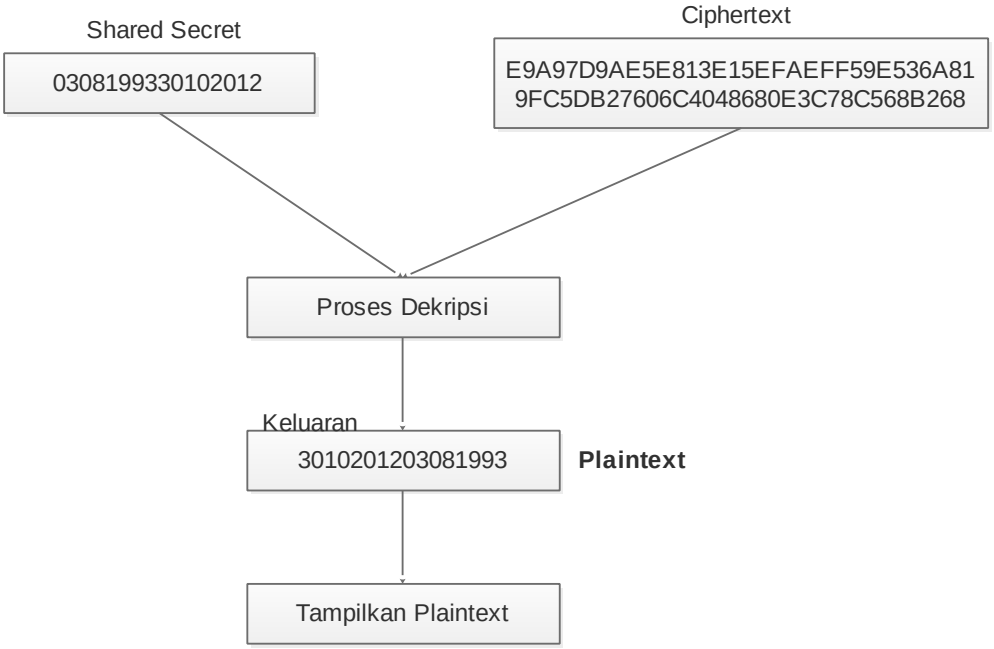
4.1.3 Proses Dekripsi

Proses dekripsi dilakukan setelah proses tukar kunci dan setelah sistem menerima *ciphertext* yang dikirim dari host lain. Hal tersebut dikarenakan komponen penyusun yang di perlukan dalam proses dekripsi adalah *ciphertext* dan kunci yang sesuai. Proses dekripsi sebagaimana terlihat pada gambar 4.14 mempunyai proses sebagai berikut:

a. Sistem mendapatkan komponen penyusunnya yaitu *ciphertext* serta *shared secret* yang dibutuhkan.

b. Setelah *ciphertext* dan *shared secret* didapat, maka akan dilakukan proses dekripsi.

c. Setelah proses dekripsi, akan dihasilkan keluaran berupa *plaintext* yang siap untuk ditampilkan pada *textview*



Gambar 4.14 Proses Dekripsi

4.2 Perancangan Sistem

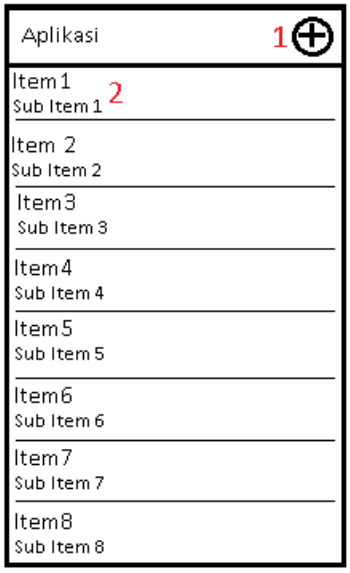
Pada subbab ini akan dijabarkan mengenai analisis kebutuhan serta perancangan antar muka sistem yang akan digunakan dalam analisis sistem enkripsi SMS berbasis android dengan menggunakan metode ECDH dan AES. Analisis kebutuhan digunakan untuk mengetahui kebutuhan-kebutuhan yang diperlukan sistem. Perancangan antar muka dilakukan untuk memberi gambaran antar muka sistem yang akan dibangun.

4.2.1 Perancangan Antar Muka Perangkat Lunak Pengujian

Pada bagian ini akan dijelaskan mengenai perancangan antar muka perangkat lunak enkripsi SMS pada android dengan menggunakan metode AES dan ECDH. Berikut merupakan gambar perancangan antarmuka perangkat lunak pengujian yang akan dibuat.

a. Halaman Awal

Halaman awal merupakan tampilan awal ketika pengguna menjalankan perangkat lunak pengujian. Pada halaman awal terdapat *image button* serta *list view* yang bertujuan untuk melihat pesan yang masuk pada sistem.



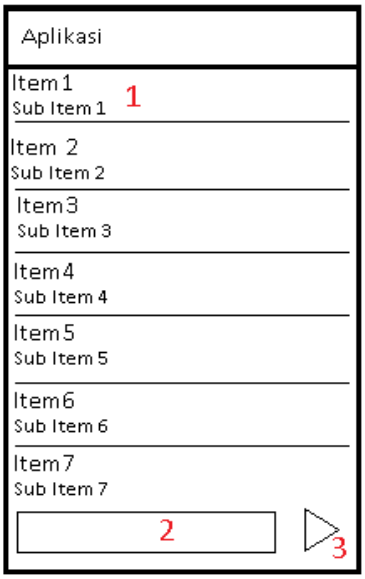
Keterangan:

1. Merupakan *image button* untuk menambah kontak
2. Merupakan *list view* dari kontak dan pesan

Gambar 4.15 Tampilah Halaman Awal

b. Halaman Buat Pesan

Halaman Buat Pesan merupakan halaman yang muncul jika pengguna memilih menu Buat Pesan. Pada halaman Buat Pesan terdapat list percakapan yang dilakukan antar kontak, *text view* yang digunakan untuk menulis pesan serta *button image* untuk perintah mengirim pesan terenkripsi



Keterangan:

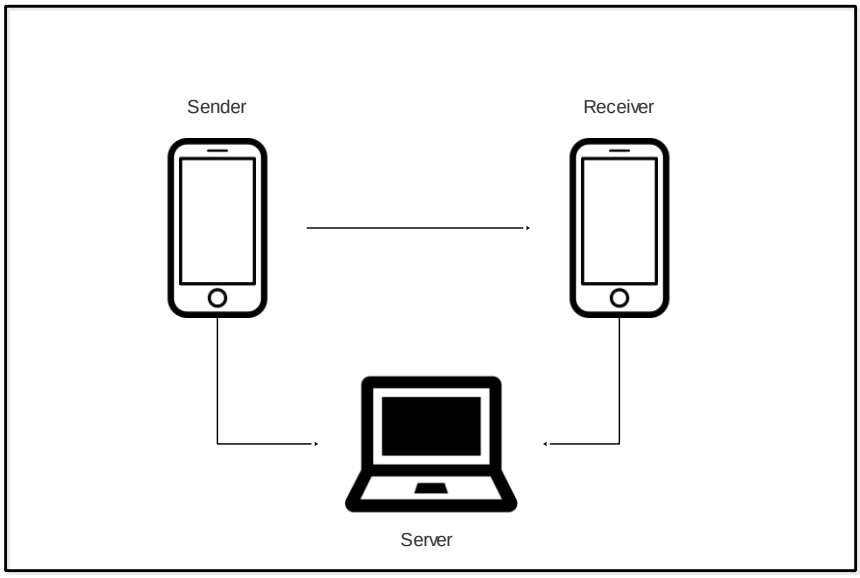
1. Merupakan *list view* dari obrolan pesan SMS pengguna dengan kontak
2. Merupakan *text view* dari input pesan SMS
3. Merupakan *Image button* untuk melakukan enkripsi serta mengirim pesan SMS

Gambar 4.16 Tampilan Halaman Buat Pesan

4.3 Perancangan Lingkungan Pengujian

Perancangan lingkungan pengujian dilakukan untuk memberi gambaran lingkungan pengujian yang akan digunakan pada sistem. Penelitian ini menggunakan 3 pengujian yaitu pengujian kriptografi, performansi, serta *avalanche effect*.

Pada gambar 4.17 menunjukkan rancangan lingkungan pengujian sistem. Pada rancangan lingkungan pengujian tersebut terdapat dua host berupa piranti bergerak yang terhubung pada satu server. Perangkat lunak diinstal pada kedua host. Server menerima data yang dihasilkan dari proses pengujian. Pengujian ini dilakukan untuk mengetahui hasil proses kriptografi pada sistem, performansi enkripsi dan dekripsi, serta *avalanche effect*.



Gambar 4.17 Rancangan lingkungan pengujian