BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Dalam perkembangan teknologi komputer dan jaringan komputer saat ini, khususnya komunikasi pesan teks lewat *chatting*. *Chatting* merupakan kegiatan yang menyenangkan. Namun demikian pesan yang dikirim dalam pesan singkat berbentuk pesan asli (*plaintext*) pesan yang dikirimkan bersifat pribadi dan orang tertentu yang boleh membaca pesan tersebut sehingga berbahaya jika adanya kemungkinan komputer digunakan oleh pihak lain baik sengaja dipinjamkan kepada pihak lain maupun tidak disengaja komputer hilang atau diambil oleh pihak lain sehingga pihak lain tersebut dapat dengan mudah dan bebas untuk membuka data penting. Oleh karena itu timbul suatu gagasan yang mengacu pada permasalahan-permasalahan tersebut, yakni untuk membuat suatu aplikasi keamanan pesan *chatting* yang dapat melindungi pesan yang dianggap penting dengan cara mengenkripsi pesan tersebut sehingga sulit untuk dibaca oleh pihak-pihak yang tidak berhak atas pesan tersebut. Dalam penggunaan aplikasi antara *server* dengan *client* dihubungkan dengan menggunakan koneksi *wireless* yang terdapat pada satu area yang sama sehingga dalam menggunakan aplikasi ini tidak dibutuhkan biaya tambahan seperti biaya data ataupun biaya SMS. Pada ruangan terbuka koneksi *wireless* dapat digunakan hingga jarak 100 meter sehingga untuk melakukan *chat* antara *server* dengan *client* tidak harus berada pada jarak yang dekat.

III.2. Kriptografi Algoritma RC5

Algoritma enkripsi RC5 didesain oleh Profesor Ronald Rivest dan pertama kali dipublikasikan pada Desember 1994. Sejak publikasinya, RC5 telah menarik perhatian banyak peneliti dalam bidang kriptografi dalam rangka menguji tingkat keamanan yang ditawarkan oleh algoritma RC5 (RSA Laboratory Technical Report TR-602). Pada dasarnya RC5 di desain dengan sedemikian rupa untuk memenuhi syarat-syarat sebagai :

a. RC5 harus merupakan blok chiper simetris. Kunci rahasia yang sama dalam kriptografi digunakan dalam enkripsi dan dekripsi. Teks awal dan teks terenkripsi memiliki panjang yang sudah ditentukan dalam blok.
b. RC5 harus cocok untuk hardware ataupun software. Hal tersebut berarti RC5 hanya akan menggunakan operasi perhitungan primitif yang sering kali ditemukan pada mikroprosessor.
c. RC5 harus cepat, hal tersebut lebih kurang dikarenakan RC5 merupakan algoritma word-oriented dengan kata lain pada operasi komputasi dasar yang digunakan harus dapat memproses data word secara penuh dalam satu waktu.
d. RC5 harus dapat beradaptasi pada berbagai panjang data word. Semisal pada prosessor 64 bit, panjang data word yang digunakan lebih panjang daripada prosesor 32 bit. RC5 harus dapat memanfaatkan hal tersebut, oleh karenanya RC5 memiliki parameter w yang yang menandakan panjang word.
e. RC5 harus dapat beroperasi dalam berbagai jumlah round. Jumlah round yang bervariasi memungkinkan pengguna untuk memanipulasi RC5 untuk menjadi lebih cepat dan aman.

f. RC5 harus dapat beroperasi dalam berbagai panjang kunci. Hal tersebut mengakibatkan panjang kunci b menjadi parameter dalam algoritma RC5.
g. RC5 haruslah berstruktur sederhana. Struktur yang sederhana tersebut belum tentu menghasilkan kemanan yang rendah. Namun, struktur sederhana akan memugkinkan analisis dan eveluasi yang cepat untuk menentukan kekuatan algoritma.
h. RC5 harus hemat dalam penggunaan memori. Hal tersebut memungkinkan implemantasi RC5 kedalam smart-card atau perangkat lain yang memiliki keterbatasan memori.
i. RC5 harus mengimplementasikan metode data-dependent rotations. Metode RC5 merupakan kriptografi primitif yang merupakan sasaran pengkajian RC5. Data-dependent rotations merupakan suatu teknik yang merotasi data secara sirkuler sebanyak N rotasi. (Suryawan dan Hamdani ; 2013 : 44-49)

Seperti yang dijelaskan sebelumnya, algoritma RC5 merupakan metode enkripsi menggunakan metode simetrik dan pengolahan dalam bentuk blok *chiper*, jadi kata kunci yang sama digunakan untuk proses enkripsi dan dekripsi. Parameter-parameter yang digunakan dalam RC-5 adalah sebagai berikut:

d. Round atau jumlah putaran disimbolkan dengan r yang memiki nilai antara 1, 2, 3, 4, ..., 225.
e. Jumlah *word* dalam *bit* disimbolkan dengan w. Jumlah yang disupport adalah 16 *bit*, 32 *bit*, dan 64 *bit*.
f. Kata kunci (*key word*) disimbolkan dengan b dengan range 1, 2, 3, 4,..., 225.

Ada 3 proses utama dalam RC5, yaitu perluasan kunci, enkripsi dan dekripsi. Perluasan kunci merupakan proses membangkitkan kunci internal dengan memanfaatkan komputasi rotasi *left regular shift* (<<<) dan *right regular shift* (>>>), dengan panjang kunci tergantung dari jumlah putaran. Kunci internal kemudian digunakan dalam proses enkripsi dan dekripsi.

Proses enkripsi dibagi menjadi tiga, yaitu: penjumlahan integer, XOR dan rotasi. Untuk lebih jelasnya, adalah sebagai berikut:

4. Enkripsi

Diasumsikan terdapat dua buah blok *input* sebesar w *bit*, A dan B. Dan diasumsikan juga bahwa pembentukan kunci internal telah dilakukan, sehingga array S[0...t-1] telah dihitung. Sehingga *pseudecode* untuk proses enkripsi seperti di bawah:

A← A+ KI[0]

B← B + KI[1]

for i ←1 to r do

A← ((A⊕B) <<< B) + KI[2i]

B← ((B⊕A) <<< A) + KI[2i+1]

Endfor

5. Dekripsi

Algoritma pada proses dekripsi merupakan kebalikan dari proses enkripsi. Jika tadinya digeser ke kiri, maka pada proses dekripsi dilakukan pergeseran ke kanan (*right regular shift*).

for i ←r downto 1 do

B← ((B – KI[2i+1])>>>A) ⊕A

A← ((A – KI[2i])>>>B) ⊕B

endfor

B ←B - KI[1]

A ←A - KI[0]

Untuk mendekripsi cipherteks, diperlukan KI yang sama dengan KI saat mengenkripsi. Proses pembangkitan KI pada kedua proses tersebut juga sama.

6. Pembentukan kunci internal

K[0-1]...K[b] disalin ke tabel L[0-1]...L[b] dengan aturan di-padding dengan karakter 0 hingga ukuran L[i]menjadi w/2 bit. Sebagai contoh:

K[0] = k L[0] = k000

K[1] = r L[1] = r000

K[2] = i L[2] = i000

K[3] = p L[3] = p000

K[4] = t L[4] = t000

K[5] = o L[5] = o000

Kemudian, inisialisasi tabel kunci internal KI dengan ukuran t = 2r + 2 seperti berikut:

KI[0] ←P

for i ←1 to t – 1 do

KI[i] ←KI[i – 1]

Endfor

Algoritma pembentukan kunci internal menggunakan konstanta P dan Q yang didapatkan dari fungsi yang melibatkan bilangan irasional sebagai berikut:

$P = Odd[(e - 2)2^w]$

$Q = Odd[(f - 1)2^w]$

Keterangan:

e = 2.718281828459.....

F = 1.618033988749.....

Kemudian L dan S digabungkan dengan algoritma berikut:

```
i ←0
j ←0
X ←0
Y ←0
n ←3*max(r,c)
for k ←1 to n do
KI[i] ← (KI[i] + X + Y)<<< 3
X ←KI[i]
i ← (i + 1) mod t
L[j] ← (L[j] + X + Y)<<< 3
Y ←L[j]
j← (j + 1) mod c
endfor
```

Keterangan:

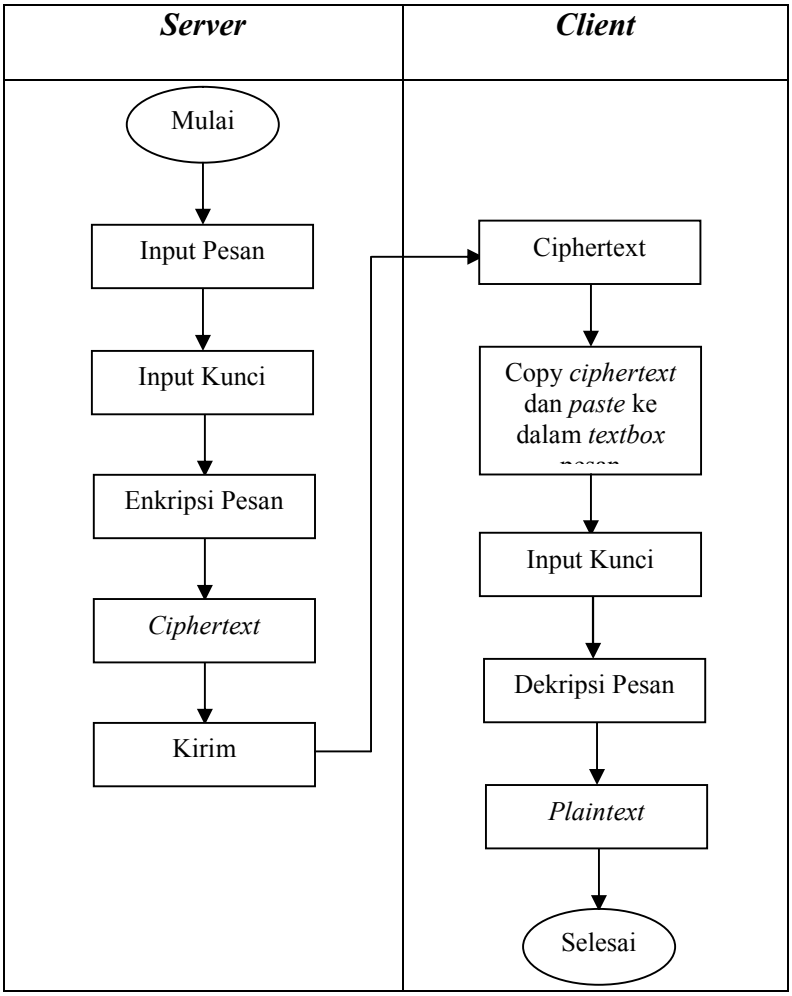
Max(r,c) adalah fungsi menentukan bilangan terbesar antara r dan c. c adalah nilai maksimal dari panjang kunci b dibagi 4. (Suryawan dan Hamdani ; 2013 : 44-49)

III.3. Desain Sistem

Implementasi algoritma RC5 untuk aplikasi *chat* dengan *client server* berbasis android dirancang dengan menggunakan perangkat lunak *Eclipse* Juno. Perancangan sistem yang dirancang terdiri dari *use case*, *flowchart*, *activity diagram* serta desain dan penjelasan dari sistem yang dirancang. Berikut adalah perancangannya :

III.3.1. *Flow Of Document* (FOD)

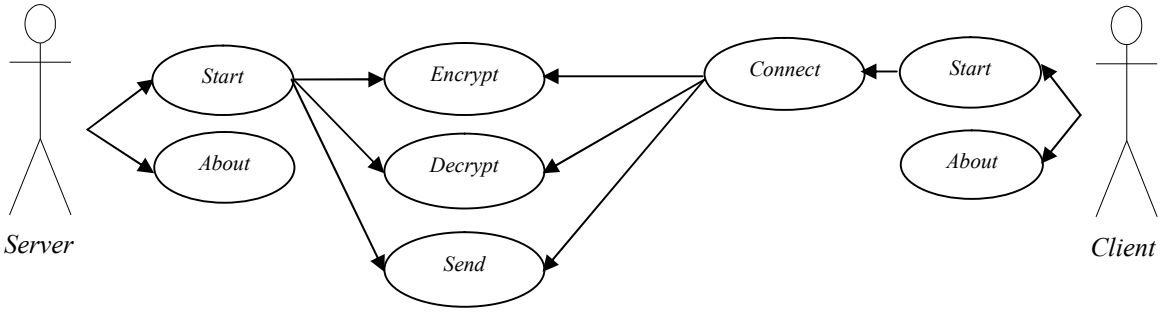
Berikut ini merupakan FOD dari aplikasi implementasi algoritma RC5 untuk aplikasi *chat* dengan *client server* berbasis android pada saat melakukan enkripsi, dekripsi dan mengirimkan pesan.



Gambar III.1. *Flow Of Document* (FOD) Enkripsi, Dekripsi dan Kirim Pesan

III.3.2. *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 :

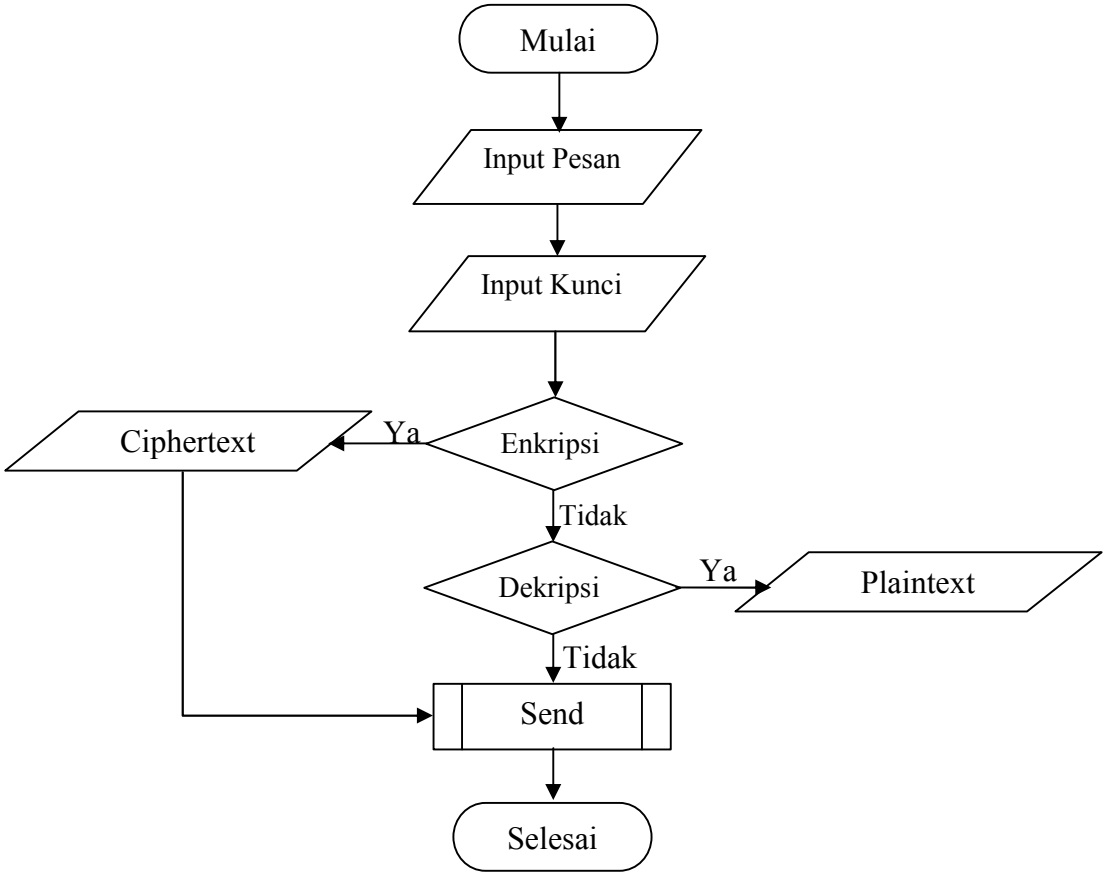


Gambar III.2. *Use Case Diagram Server Dan Client*

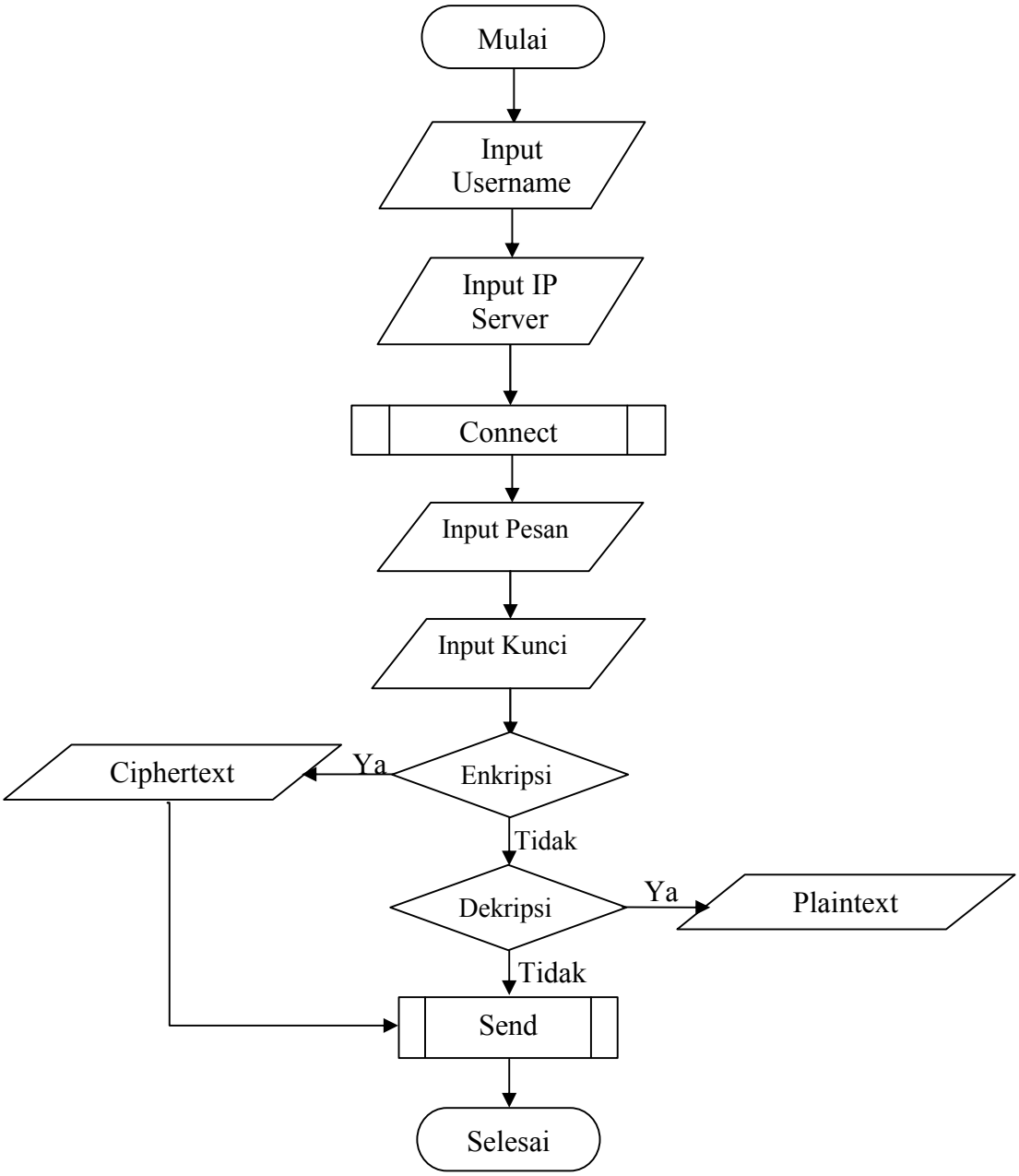
III.3.3. *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program.

Tujuan utama dari penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Dalam perancangan aplikasi ini digunakan bagan alir (*flowchart*) untuk menjelaskan proses kerja dari perangkat lunak yang dirancang.



Gambar III.4. *Flowchart Server*



Gambar III.5. ***Flowchart Client***

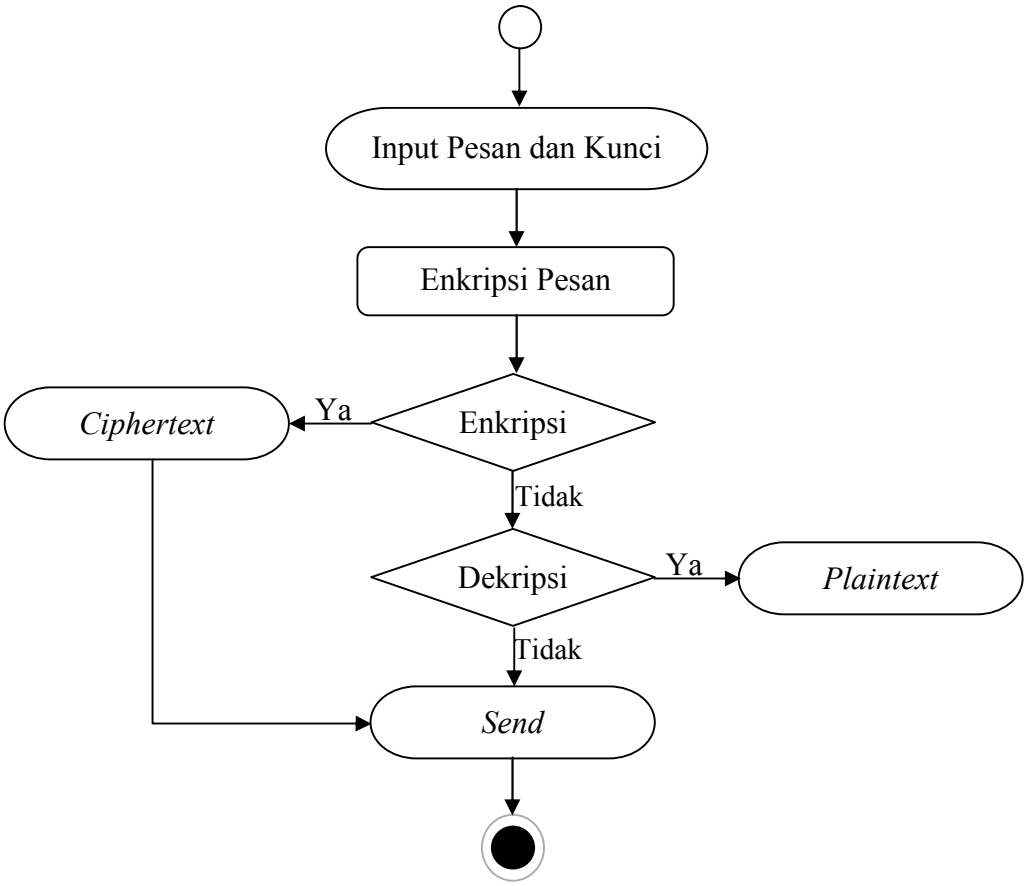
1. Proses *Flowchart Server*
    a. Mulai
    b. *Input* pesan
    c. *Input* kunci
    d. Enkripsi/Dekripsi/Kirim pesan
    e. Selesai
2. Proses *Flowchart Client*
    a. Mulai
    b. *Input Username*
    c. *Input* IP *Server*
    d. *Input* pesan
    e. *Input* kunci
    f. Enkripsi/Dekripsi/Kirim pesan
    g. Selesai

III.3.4. *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* yang terdapat pada aplikasi yaitu sebagai berikut :

1. *Activity Diagram* Enkripsi Pesan

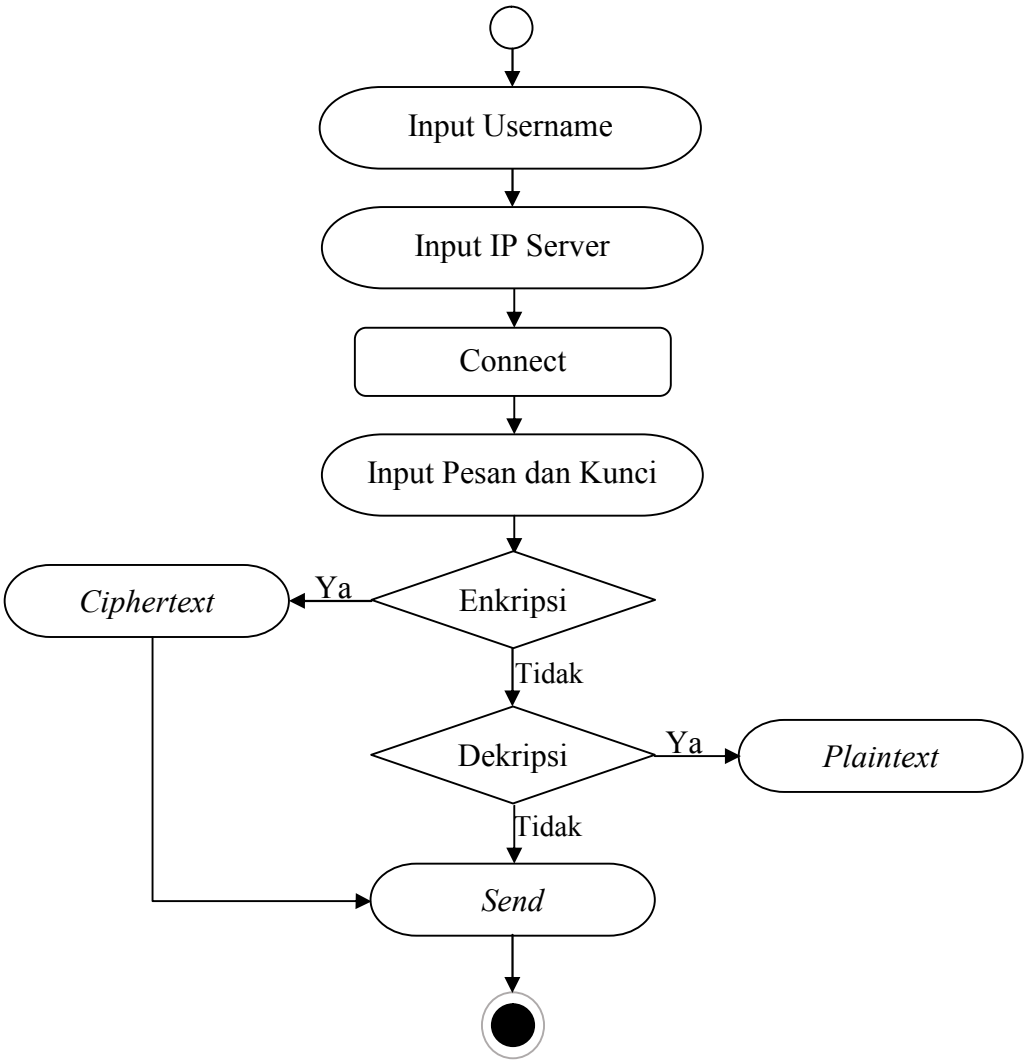
Activity diagram enkripsi pesan menggambarkan alir aktifitas pengenkripsian pesan yang dilakukan oleh pengguna dan diproses didalam sistem.



Gambar III.6. ***Activity Diagram Server***

2. *Activity Diagram* Dekripsi Pesan

Activity diagram dekripsi pesan menggambarkan alir aktifitas pendeskripsian pesan yang dilakukan oleh pengguna dan diproses didalam sistem.

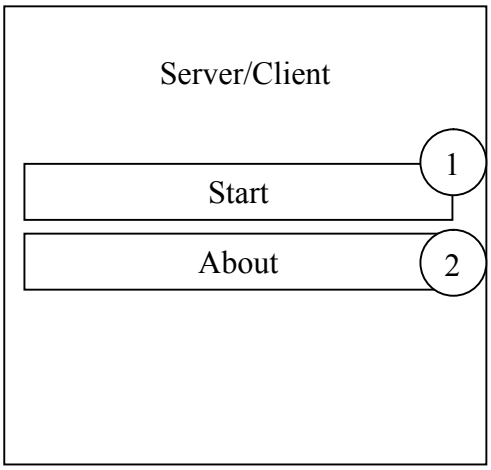


Gambar III.7. *Activity Diagram* Dekripsi Pesan

III.5. 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 aplikasi kriptografi yang dirancang dengan antarmuka pada perangkat *mobile*, yaitu :

1. Desain *Form Menu* Utama



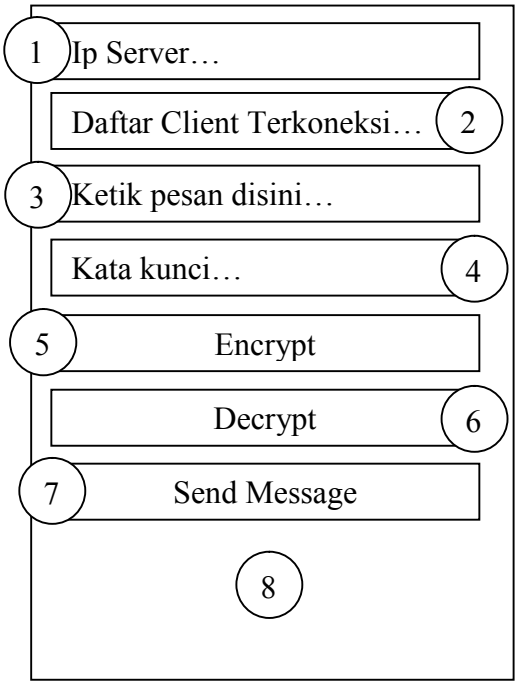
Gambar III.8. *Form Menu* Utama

Keterangan tampilan menu utama, yaitu :

1) Tombol untuk melakukan memulai aplikasi.

2) Tombol untuk membuka halaman tentang *programer*.

2. Desain Halaman *Chatting Server*

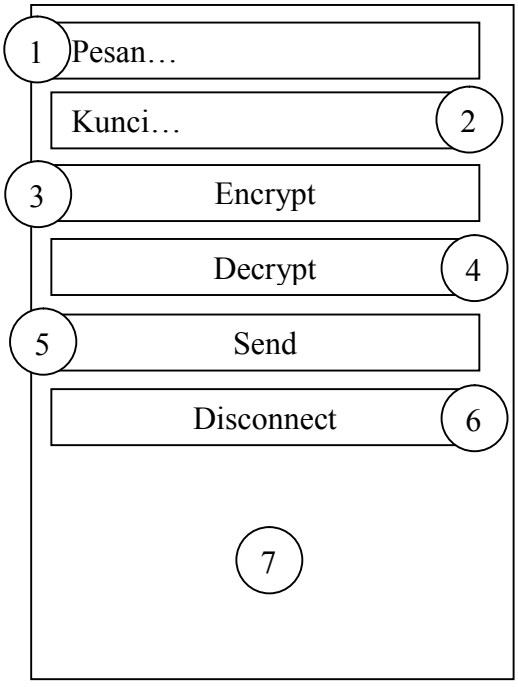


Gambar III.9. Halaman *Chatting Server*

Merupakan tampilan rancangan halaman chatting. Adapun keterangannya sebagai berikut :

1) Menampilkan alamat IP dari *server*.
2) Menampilkan daftar *client* yang terkoneksi ke *server*.
3) *Textbox* yang digunakan untuk memasukkan pesan *chatting*.
4) *Textbox* yang digunakan untuk memasukkan kunci untuk enkripsi dan dekripsi.
5) Tombol yang digunakan untuk melakukan enkripsi pesan.
6) Tombol yang digunakan untuk melakukan dekripsi pesan.
7) Tombol yang digunakan untuk mengirim pesan.
8) Area untuk menampilkan pesan *chatting*.

3. Desain Halaman *Chatting Client*

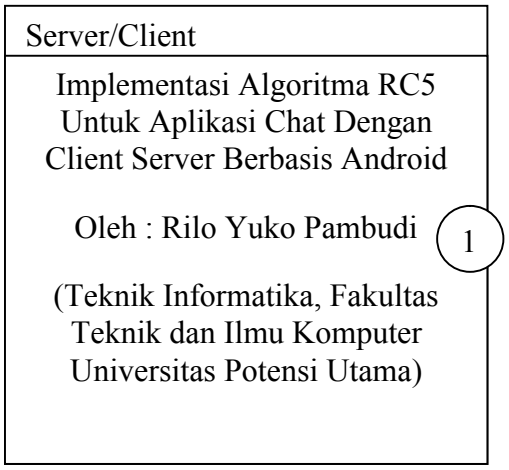


Gambar III.10. Halaman *Chatting Client*

Merupakan tampilan rancangan halaman chatting. Adapun keterangannya sebagai berikut :

1) *Textbox* yang digunakan untuk memasukkan pesan *chatting*.
2) *Textbox* yang digunakan untuk memasukkan kunci untuk enkripsi dan dekripsi.
3) Tombol yang digunakan untuk melakukan enkripsi pesan.
4) Tombol yang digunakan untuk melakukan dekripsi pesan.
5) Tombol yang digunakan untuk mengirim pesan.
6) Tombol untuk memutuskan koneksi.
7) Area untuk menampilkan pesan *chatting*.

5. Desain *Form About*



Gambar III.11. *Form About*

Adapun keterangannya sebagai berikut :

1) Menampilkan data dari *programer*.