BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisis Masalah

Pengiriman data *file* dapat dilakukan melalui perangkat *android* dengan mudah, telah banyak pengembangan aplikasi yang dapat mengirimkan data *file* dengan penggunaan yang cukup mudah. Namun pengembangan aplikasi tersebut masih banyak yang belum mendukung keamanan data teks untuk menjaga kerahasiaan dari penyalahgunaan. Dengan perkembangan teknologi khususnya dibidang IT, semakin berkembang juga tingkat kejahatan yang dapat membuat data penting menjadi tidak aman karena dapat diakses oleh siapa saja Pada analisa sistem ini membahas tentang perancangan aplikasi keamanan transfer data teks dengan menggunakan algoritma RC4 pada penggunaan perangkat *mobile phone Android*. Kemanan yang menggunakan algoritma RC4 ini menggunkan konsep enkripsi dan dekripsi data.

Algoritma kriptografi RC4 memproses enkripsi deskripsi dengan proses yang sama sehingga hanya ada satu fungsi yang dijalankan untuk menjalankan kedua proses tersebut. RC4 mempunyai sebuah S-Box dan key dalam bentuk array 256 byte yaitu : S0, S1, ..., S255 yang berisi permutasi dari bilangan 0 sampai 255, K0, K1, …, K255 dengan kategori stream simetrik. Sedangkan untuk inisialisasi S-Box yaitu dengan mengisikan nilai 1 sampai dengan 255 dimulai dari S0 sampai dengan S255, isi S-Box secara berurutan, yaitu S0=0, S1=1, ... , S255=255 (Scheiner, 2001). Untuk inisialisasi *key* yaitu dengan mengisikan *array*

K255 byte dengan kunci yang diulangi sampai seluruh array K0,K1, ... K255 terisi seluruhnya. Pseudocode yang terbentuk untuk menciptakan initialisasi key adalah sebagai berikut.

For I = 0 to 255

Ki = I mod length (key)

Pseudorandom adalah nilai yang dibangkitkan dari nilai S-Box dan Key yang telah diinisialisasi. Caranya set indeks j dengan nol, dan melakukan penukaran nilai S-Box yang sudah diinisialisasi sebelumnya dengan nilai perulangan ditambah dengan S-Box awal ditambah dengan nilai dari array K yang dapat digambarkan dan dijelaskan dalam *Pseudocode* sebagai berikut.

j = 0

for i = 0 to 255

j = (j + Si + Ki) mod 256

swap Si dan Sj

Fungsi *swap* merupakan fungsi yang menukarkan nilai S ke-i dengan nilai S ke-j. Kemudian membangkitkan nilai pseudorandom key berdasarkan indeks dan nilai S-Box. Terdapat 2 indeks yaitu i dan j, yang diinisialisasi dengan bilangan nol. Untuk menghasilkan random byte langkahnya adalah sebagai berikut.

i = 0

j = 0

for idx = 0 to len-1

i = (i + 1) mod 256

j = (j + Si) mod 256

swap Si dan Sj

t = (Si + Sj) mod 256

k = St

buffidx = k XOR buffidx

Rumus diatas merupakan penerapan algoritma yang dibahas, Adapun keterangan dari rumus diatas sebagai berikut :

1. buff merupakan pesan yang akan dienkripsi atau dekripsi
2. len merupakan panjang dari buff

Nilai *pseudorandom key* inilah yang akan di XOR dengan plainteks untuk menghasilkan cipherteks atau XOR dengan cihperteks untuk menghasilkan plainteks. Untuk menghasilkancipherteks yaitu dengan rumus ”Cipherteks = Plainteks XOR K” sedangkan untuk menghasilkan plainteks yaitu dengan rumus “Plainteks = Cipherteks XOR K”.

Contoh lain dari implementasi algoritma RC4 adalah saat proses *key scheduling algorithm* didapatkan S-Box terakhir 54, 157, 62, 162, 25, 135, 195, 103, 208, 8, 188, 42, 165, 13, 141, 253, 35, 231, 108, 134, 93, 82, 49, 9, 83, 139, 147, 38, 87, 193, 22, 219, 113, 248, 155, 117, 64, 123, 154, 67, 53, 94, 46, 102, 133, 170, 106, 194, 24, 246, dan 45.

III.1.1. Perancangan Perangkat (*Software* / *Hardware*)

Ada beberapa aplikasi pendukung atau *software* maupun *hardware* yang yang digunakan dalam perancangan, agar perancangan berjalan dengan baik yang terdiri dari.

1. Perangkat Lunak (*Software*)
    a. *Operating System*, OS yang digunakan dalam perancangan dan tes untuk program aplikasi yang dirancang yaitu menggunakan *windows 7* untuk merancang aplikasi, dan OS *Android* untuk menjalankan aplikasi.
    b. JDK *Java 1.7*, sebagai bahasa program dan *compiler Java* serta SDK *Android* untuk perancangan aplikasi pada perangkat *mobile phone Android*.
2. Perangkat Keras (*Hardware*)
    a. Komputer yang setara dengan *processor dual core*.
    b. *Smartphone Android* Jelly Bean Versi 4.2.2.
    c. *Mouse, keyboard*, dan *Monitor.*

III.1.2. Strategi Pemecahan Masalah

Beberapa strategi pemecahan masalah dalam perancangan ini penulis simpulkan sebagai berikut :

1 Membatasi akses aplikasi bagi pengguna dengan hanya memiliki kemampuan untuk transfer data yang telah disandikan, hasil penyandian bertipe data teks atau dengan ekstensi *.*txt*, agar dapat dengan mudah dibaca oleh sistem.

2 Tidak menggunakan media *database,* karena hasil penyandian menghasilkan sebuah data teks yang memiliki penyimpanan yang besar sehingga memberatkan sumber daya penyimpanan pada perangkat *mobile phone* yang digunakan, perancangan difokuskan mampu menyandikan dengan instan yang disimpan pada memory sementara untuk menciptakan *file-file* hasil proses..

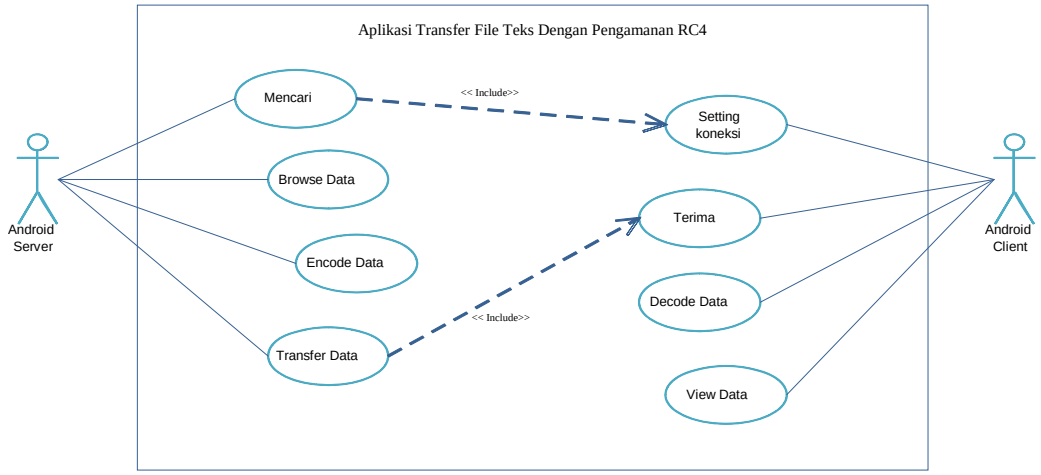
3 Aplikasi yang dihasilkan menyediakan algoritma RC4 digunakan sebagai penyandian data teks (*string*).

III.2. Perancangan Sistem

Pada tahapan ini perancangan menggambarkan sistem yang akan dibangun dan sesuai dengan objek penelitian. Hal ini bertujuan untuk mengetahui bagaimanakah hasil akhir dari sistem yang dilakukan. Aplikasi ini dibangun dengan desain layar-layar *form.* berikut merupakan gambar perancangan dari tampilan sistem yang direncanakan.

III.3.1. *Use Case* Diagram

Diagram *use case* ini menggambarkan *user* (aktor) yang menggunakan sistem dan perilaku *user* terhadap sistem, dapat pada gambar III.1 berikut.



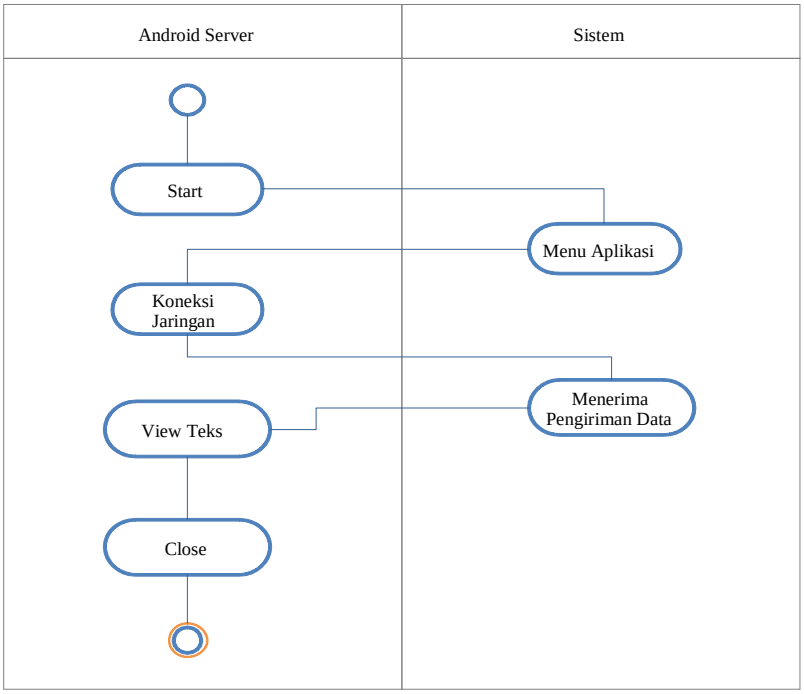
Gambar III.1. *Use Case* Diagram Sistem Transfer Data Teks

Pada gambar *use case* tersebut memiliki dua aktor yaitu android client sebagai pengirim dan android server sebagai penerima. Sebelum melakukan pengiriman data android server terlebih dahulu mencari koneksi *wifi* dan

kemudian android client yang menkonfigurasi jaringan yang telah digunakan. Untuk kemanan data teks, *android server* melakukan proses enkripsi yang kemudaian ditransfer dan setelah itu *android client* menerimanya dan dapat langsung melakukan proses dekripsi data sehingga data dapat dilihat seperti semula.

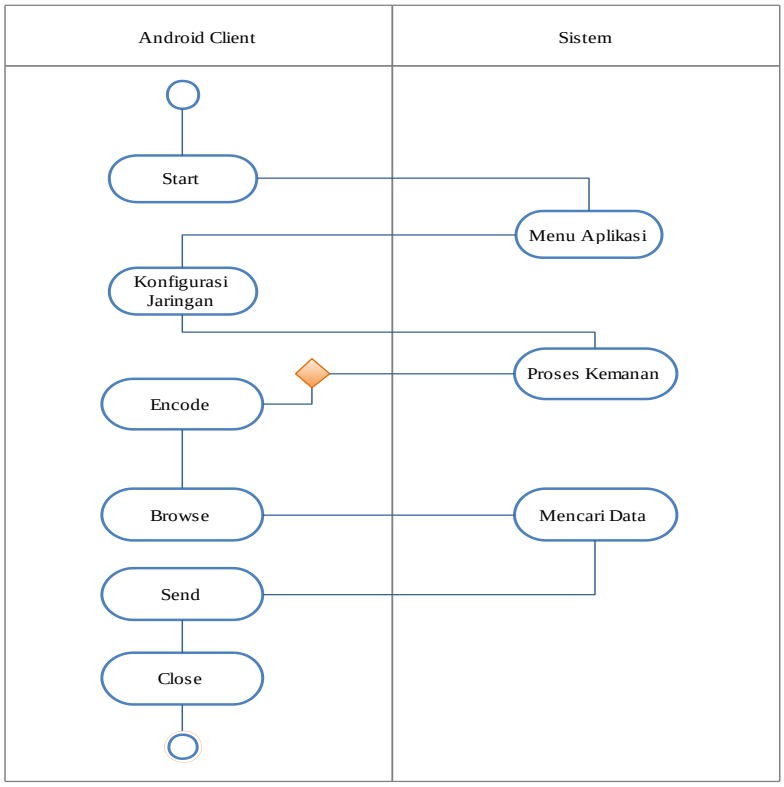
III.3.2. *Activity* Diagram Android Server

Activity diagram adalah teknik untuk mendiskusikan logika *prosedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* diagram banyak mempunyai peran seperti halnya *flowchart*. *Activity diagram* bisa mendukung perilaku pararel sedangkan *flowchart* tidak bisa. Berikut ini adalah *activity* diagram aplikasi yang dirancang dapat dilihat pada gambar III.2 dan gambar III.3:



Gambar III.2. *Activity* Diagram Aplikasi *Android Client*

Pada gambar diatas dapat dilihat proses aliran data yang terjadi, setelah pengguna menjalankan aplikasi proses selanjutnya pengiriman data dapat dilakukan jika terhubung pada jaringan. Berikut dibawah ini dapat dilihat *activity* diagram untuk perangkat *client*. Yang dapat dilihat pada gambar III.3 dibawah ini.

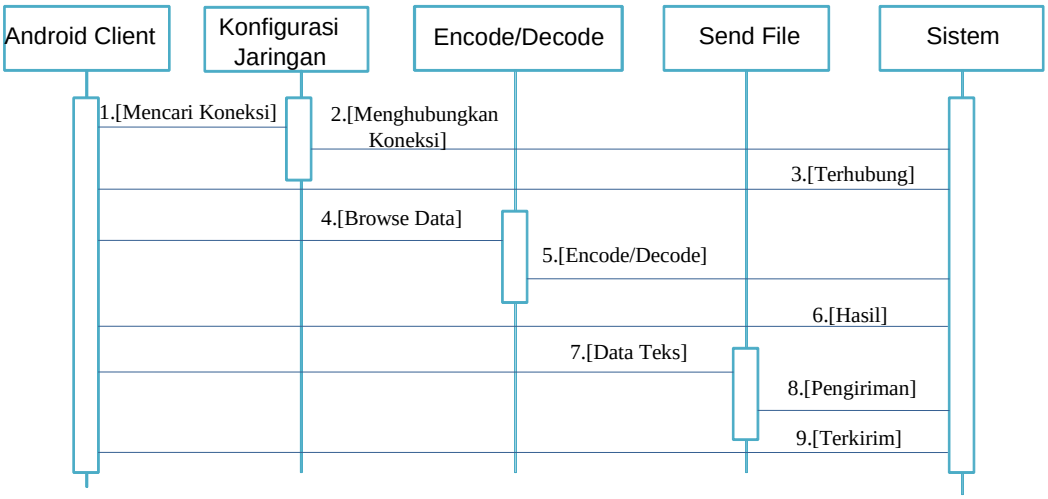


Gambar III.3. *Activity* Diagram Aplikasi *Android Server*

Gambar III.3 diatas menjelaskan proses yang berangsung pada perangkat *client*. Dimana proses enkripsi data pada proses transfer *file* dapat berjalan setelah masing-masing perangkat terhubung.

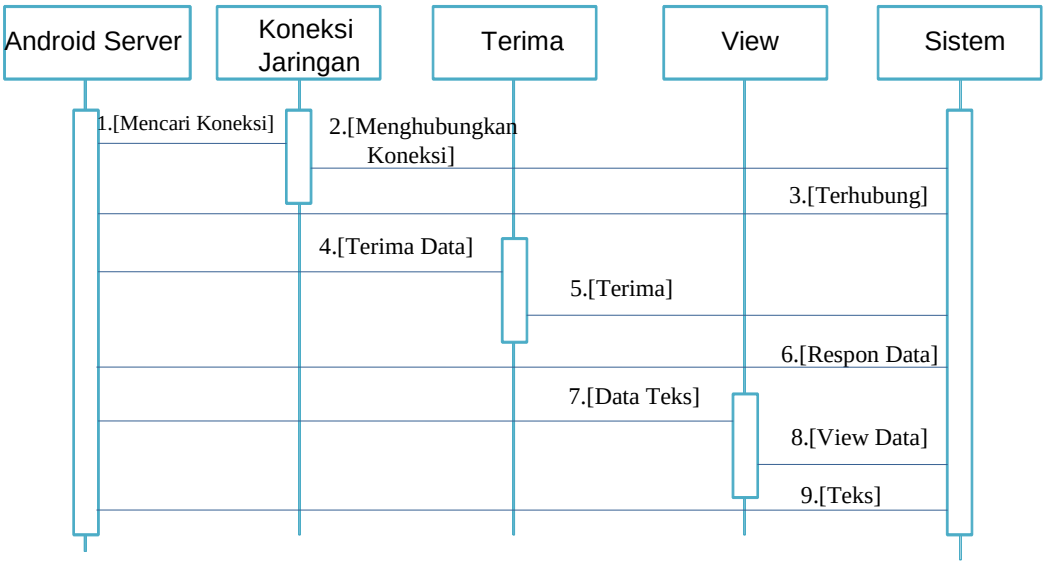
III.3.3. *Squence* Diagram

Sequence diagram yang digunakan untuk menggambarkan sebuah adegan yang menggambarkan sistem untuk proses penggunaan aplikasi., dapat dilihat pada gambar III.4 dan III.5.



Gambar III.4. *Squence* Diagram Aplikasi *Android Client*

Adapun penjelasan dari sequence diagram diatas ditujukan pada android client, yaitu terdapat 3 aktivitas yang terhubung ke system diantaranya konfigurasi jaringan yang menyesuaikan jaringan koneksi yang telah dipilih oleh server. proses enkripsi dan dekripsi digunakan untuk merubah data yang akan dikirim dalam hasil data yang telah enkripsi ataupun dekripsi. Selanjutnya data yang telah diproses dapat langsung dikirim ataupun transfer ke android server.



Gambar III.5. *Squence* Diagram Aplikasi *Android Server*

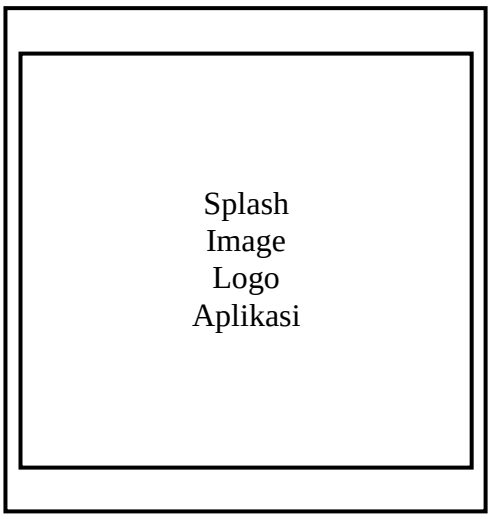
Sedangkan gambar diatas ditujukan kepada aplikasi android *server* yaitu koneksi jaringan yang merupakan *server* yang mencari koneksi *wifi* untuk melakukan proses transfer data. Setelah dikirim data dari *client*, *server* menerima ataupun mengkonfirmasi pengiriman tersebut dan dapat langsung dibuka oleh *android server*.

III.3. Desain *Interface*

Pada tahap ini menggambarkan desain ataupun rancangan tampilan pada aplikasi yang direncanakan, adapun desain *interface* yang dibuat terdiri dari beberapa tampilan yang dapat dilihat pada penjelasan berikut.

1. *Interface Splash*

Rancangan *interface splash* merupakan rancangan awal pembuka aplikasi, Yang dapat dilihat pada gambar III.6.

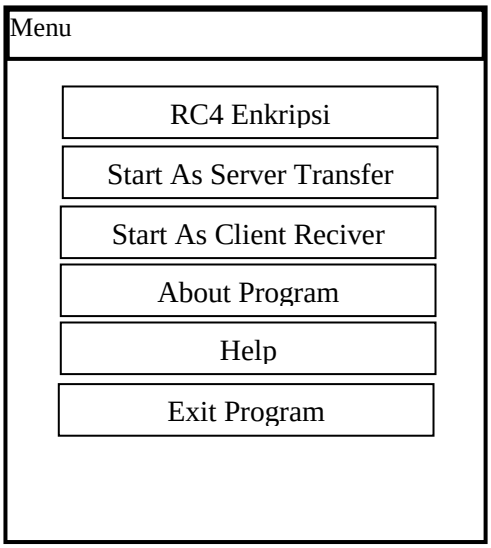


Gambar III.6. Rancangan *Interface Splash*

Pada rancangan diatas terdapat logo ataupun gambar, pada rancangan *interface* ini difungsikan sebagai tampilan awal pembuka setelah pengguna menjalankan aplikasi.

2. *Interface Menu*

Pada rancangan layar menu ini merupakan layar yang didalamnya terdapat beberapa menu yang dapat digunakan oleh pengguna, dapat dilihat pada gambar III.7 berikut.



Gambar III.7. Rancangan *Interface* Menu

Pada gambar diatas terdapat beberapa menu untuk yang memiliki fungsi yang berbeda-beda yaitu seperti:

a. Menu RC4 Enkripsi, merupakan menu yang digunakan untuk mengenkripsi dan dekripsi *file*.
b. *Start As Server Transfer*, merupakan menu bagi pengguna yang ingin menjalankan aplikasi sebgai *server* yang ingin mentransfer *file*.
c. *Start As Client Receiver*, merupakan menu bagi pengguna yang ingin

menjalankan aplikasi sebagai *client* yang ingin menerima *file* yang dikirim dari *server*.

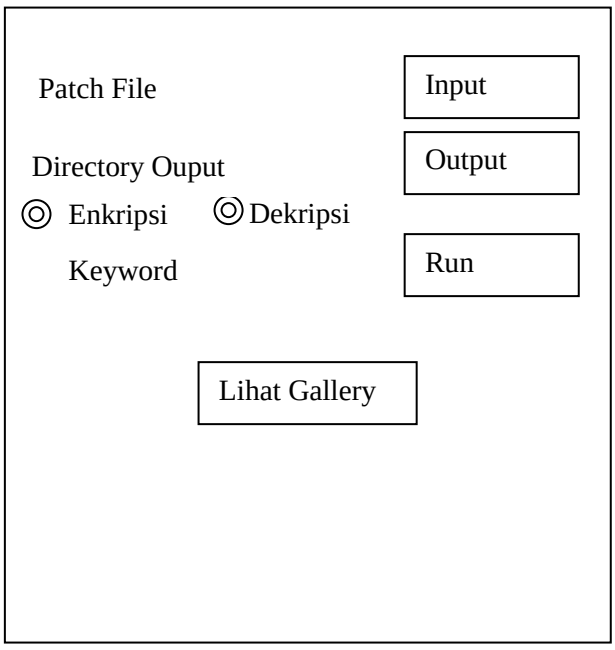
d. *About* Program, merupakan menu yang menampilkan *form* yang berisikan informasi tentang perancang aplikasi.

e. *Help*, merupakan menu yang menampilkan tentang cara penggunaan aplikasi.

f. *Exit* Program, merupakan menu yang difungsikan untuk pengguna dengan mudah menutup aplikasi.

3. Menu RC4 Enkripsi

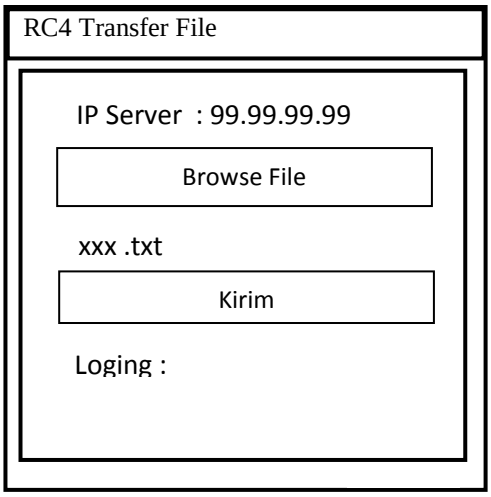
Pada menu RC4 Enkripsi pengguna aplikasi dapat melakukan proses enkripsi dan dekripsi *file*, untuk rancangan menu RC4 Enkripsi dapat dilihat pada gambar III.8 berikut ini :



Gambar III.8. Rancangan *Interface* Menu RC4 Enkripsi

4. *Interface* Kirim *File* (*Server*)

Pada rancangan *interface* ini, layar yang ditampilkan merupakan pilihan menu pengguna untuk mengirim *file*, dapat dilihat pada gambar III.9 berikut.



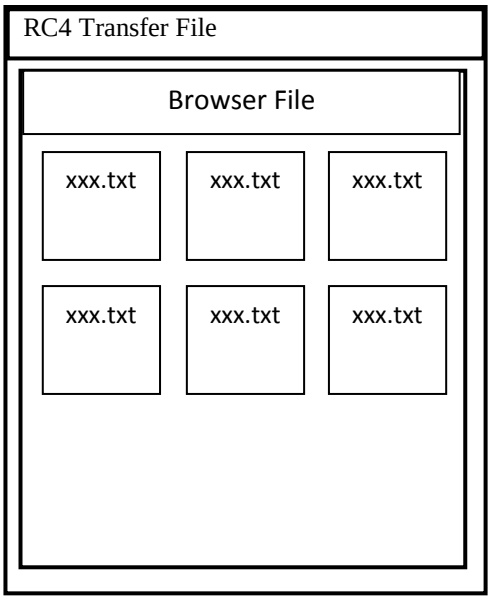
Gambar III.9. Rancangan *Interface* Kirim *File* (*Server*)

Pada gambar diatas terdapat beberapa fitur atau menu yang dapat dijelaskan yang diantaranya adalah

a. Terdapat beberapa label pada layar diatas seperti label untuk menampilkan IP pada android *server* dan pada bagian bawah merupakan info tentang *loging* proses transfer *file* dan juga proses pengiriman, apabila telah terhubung maka terdapat pemberitauan pada label tersebut.
b. Menu *Browse File* yang merupakan menu untuk mencari *file* teks yang terdapat pada *memory android* yang akan dikirimkan pada perangkat android *client*.
c. Menu Kirim yang merupakan menu untuk mengirimkan *file* teks yang telah terhubung antara *server* dan *client*.

5. *Interface Browse File*

Pada rancangan *interface* menu *browser file* yang apabila pengguna memilih menu tersebut dapat memilih *file* yang terdapat pada memori pada perangkat *Android*, dapat dilihat pada gambar III.10 berikut.

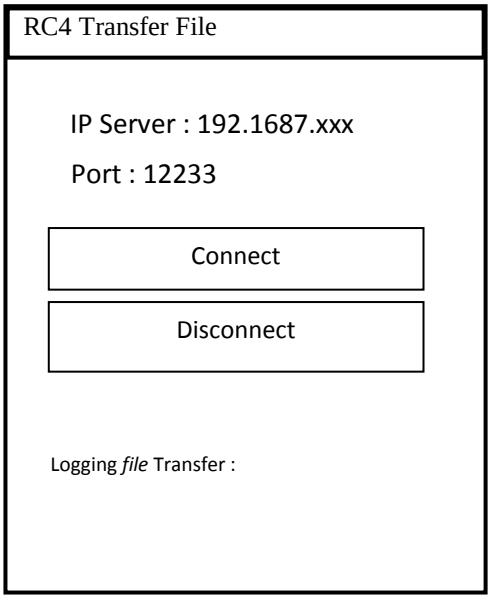


Gambar III.10. Rancangan *Interface Browser File*

Pada gambar diatas merupakan tampilan untuk penguna mencari *file* yang ingin dikirim. Pencarian *file* terdapat pada penyimpanan pada memori perangkat *android* yang berisikan *folder* ataupun *file* yang telah ada pada memori yang disimpan.

6. *Interface* Terima *File* (*Client*)

Pada rancangan *interface* ini merupakan layar yang ditampilkan jika pengguna memilih untuk melakukan menerima *file*, dapat dilihat pada gambar III.11 berikut.



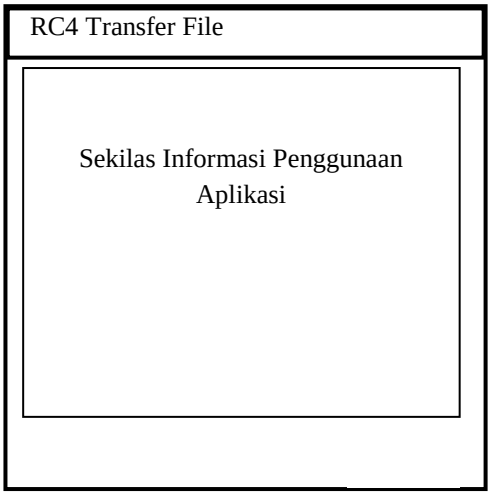
Gambar III.11. Rancangan *Interface* Terima *File* (Client)

Pada gambar diatas terdapat beberapa yang memiliki fungsi yang berbeda-beda diantranya sebagai berikut:

a. *IP Server* yang merupakan *form input*an yang harus diisikan oleh pengguna *client* yang disesuaikan dengan *ip* yang dimiliki oleh perangkat server.
b. *Port* yang merupakan penghubung dua perangkat agar dapat melakukan komunikasi antar dua perangkat.
c. Tombol *connect* dan *disconnect* yang merupakan perintah untuk menghubungkan pada perangkat *server*.

7. *Interface* Bantuan Penggunaan

Pada rancangan *interface* ini berisikan tentang informasi cara penggunaan aplikasi yang dibangun, dapat dilihat pada gambar III.12 berikut.



Gambar III.12. Rancangan *Interface* Bantuan Penggunaan

8. *Interface* Tentang Aplikasi

Pada rancangan ini berisikan tentang informasi mengenai perancang, dapat dilihat pada gambar III.13 berikut.

Gambar III.13. Rancangan *Interface* Tentang Aplikasi