BAB III

ANALISIS MASALAH DAN RANCANGAN PROGRAM

III.1. Analisis

Penelitian bertujuan untuk merancang sebuah sistem yang dapat melakukan penyampaian sebuah pesan multi *chatting* kedalam media LAN . Ada beberapa persyaratan yang harus dipenuhi. Adapun persyaratan kebutuhan sistem yang dirancang ini adalah sebagai berikut :

1. Perangkat keras (*Hardware*)

   Komputer *Pentium* IV dengan *Processor* 1,5 GHz, *Harddisk* 80 *Gbyte* dan RAM 512 Mbyte.

2. Perangkat Lunak (*Software*)

   Bahasa pemrograman yang digunakan adalah *NetBeansIDE*6.8 dan berjalan pada *sistem operasi windows.*

III.2. Strategi Pemecahan Masalah

Adapun langkah-langkah yang penulis lakukan dalam menyelesaikan masalah perancangan aplikasi *multi chatting* ini, terdiri dari beberapa tahapan sebagai berikut :

1. Mengumpulkan Teori dan Contoh-Contoh Kasus

   Dalam tahap ini penulis mengumpulkan teori-teori yang berhubungan dengan masalah teknik *chatting* menggunakan *java.* Teori-teori ini penulis kumpulkan dari beberapa sumber seperti buku-buku di perpustakaan, artikel-

artikel di internet serta referensi dari tugass akhir mahasiswa lain yang berhubungan dengan masalah yang dihadapi.

2. Merancang Program

Setelah teori-teori dan contoh-contoh kasus penunjang penulis rasakan cukup, langkah selanjutnya penulis melakukan perancangan terhadap program. Program penulis rancang untuk dapat melakukan teknik menyampaikan aplikasi *multi chatting*.

Langkah pertama dalam perancangan program ini adalah merancang proses kerja sistem. Proses kerja sistem penulis rancang menggunakan sebuah bagan alir *(flowchart*) yang menjelaskan secara rinci proses-proses yang akan dilakukan program dalam menyampaikan sebuah pesan teks pada aplikasi *multi chatting*.

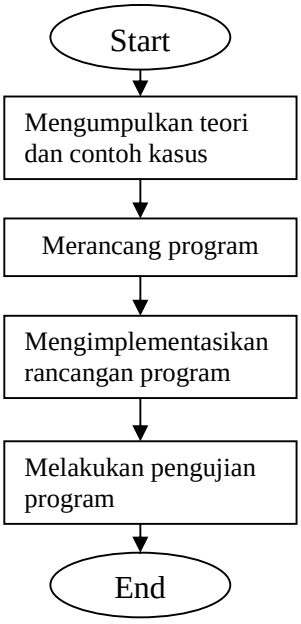
Langkah berikutnya adalah merancang bentuk tampilan program. Bentuk tampilan program yang penulis rancang adalah sbuah form dengan tombol-tombol yang dapat diguanakan user untuk berinteraksi dengan program yang dirancang. Dalam langkah ini penulis juga merancang pemrograman yang akan penulis gunakan dalam implementasi rancangan program dalam bahasa pemrograman yang digunakan.

3. Mengimplementasikan Rancangan Program

Bahasa pemrograman yang penulis pilih dalam implementasi rancangan program adalah *NetBeans IDE*.68. Bahasa pemrograman ini penulis pilih karena lebih familiar dan Opensource dibandingkan bahasa pemrograman lain dan sering penulis gunakan pada saat perkuliahan.

Pada tahapan ini, penulis mengimplementasikan rancangan tampilan program serta melakukan *coding* sesuai dengan bahasa pemrograman yang digunakan. Tahapan implementasi program yang penulis lakukan adalah membuat tampilan form, membuat module-module yang dibutuhkan serta membuat sintax-sintax terhadap tombol-tombol dan menu-menu pada form. Melakukan pengujian program.

Pada tahapan akhir ini, penulis melakukan serangkaian pengujian terhadap program yang dihasilkan. Pengujian-pengujian ini dilakukan untuk mencari kesalahan-kesalahan (*error*) pada program dan melakukan perbaikan-perbaikan yang dibutuhkan. Adapun skema metode penyelesaian masalah yang penulis lakukan dapat dilihat pada Gambar III.1.



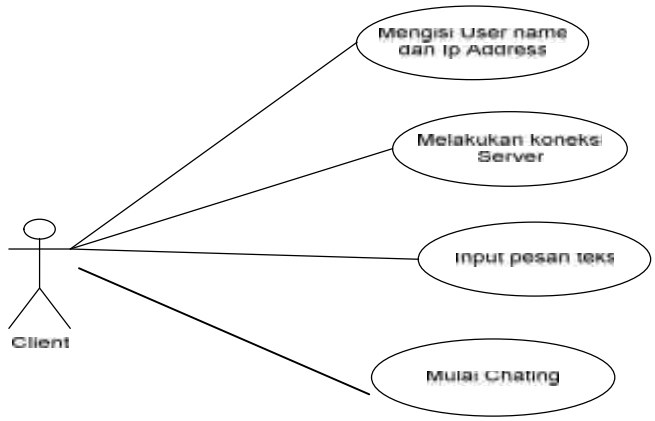
Gambar III.1 Skema Metode Penyelesaian Masalah

III.3. Struktur Data yang Digunakan

Struktur data yang digunakan penulis dalam perancangan perangkat lunak adalah *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML yang digunakan meliputi perancangan *Diagram Use Case*, *Activity Diagram* dan *Squence Diagram*.

III.3.1. *Diagram Use Case*

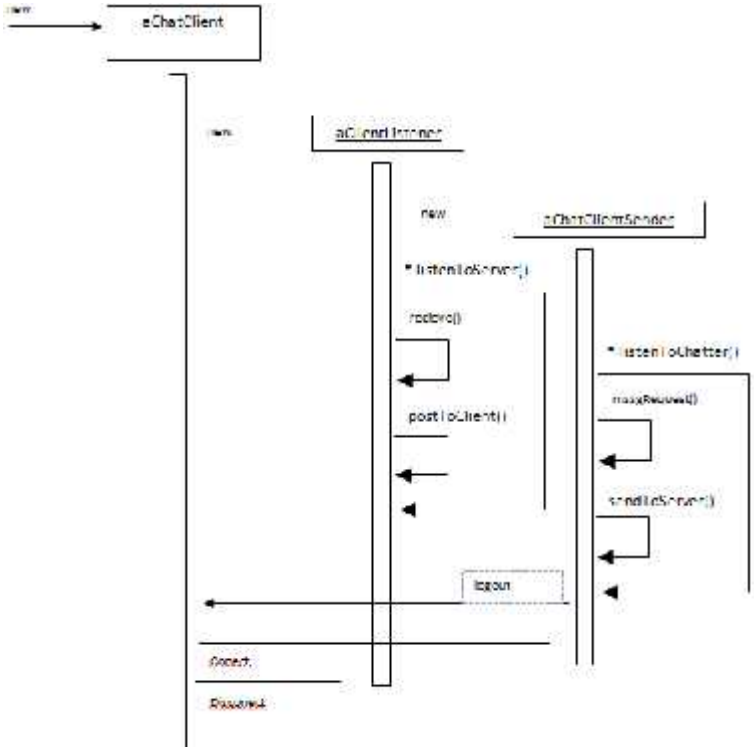
Diagram use case digunakan untuk memberikan gambaran kebutuhan perangkat lunak secara visual. *Use case* diagram yang memiliki *use case* proses sisip teks dan *use case* proses ekstrak teks. Proses sisip teks memiliki 3 *include* didalamnya, sedangkan proses ekstrak memiliki 2 include dan 1 extend. Pengirim merupakan pengguna yang melakukan penyisipan teks. sedangkan penerima adalah pengguna yang melakukan ekstraksi pesan. *Use case* memilih pesan digunakan oleh pengirim untuk memilih pesan yang akan disisipkan kedalam citra digital, kemudian use case memilih citra digital digunakan oleh pengirim untuk memilih berkas citra digital yang akan digunakan sebagai media penyisipan. *Use case* memilih berkas citra digital oleh penerima digunakan untuk memilih berkas citra digital yang akan diekstraksi untuk mendapatkan pesan yang disisipkan. Kita bisa lihat pada Gambar III.2.



Gambar III.2. *Use case diagram* aplikasi *chatting*.

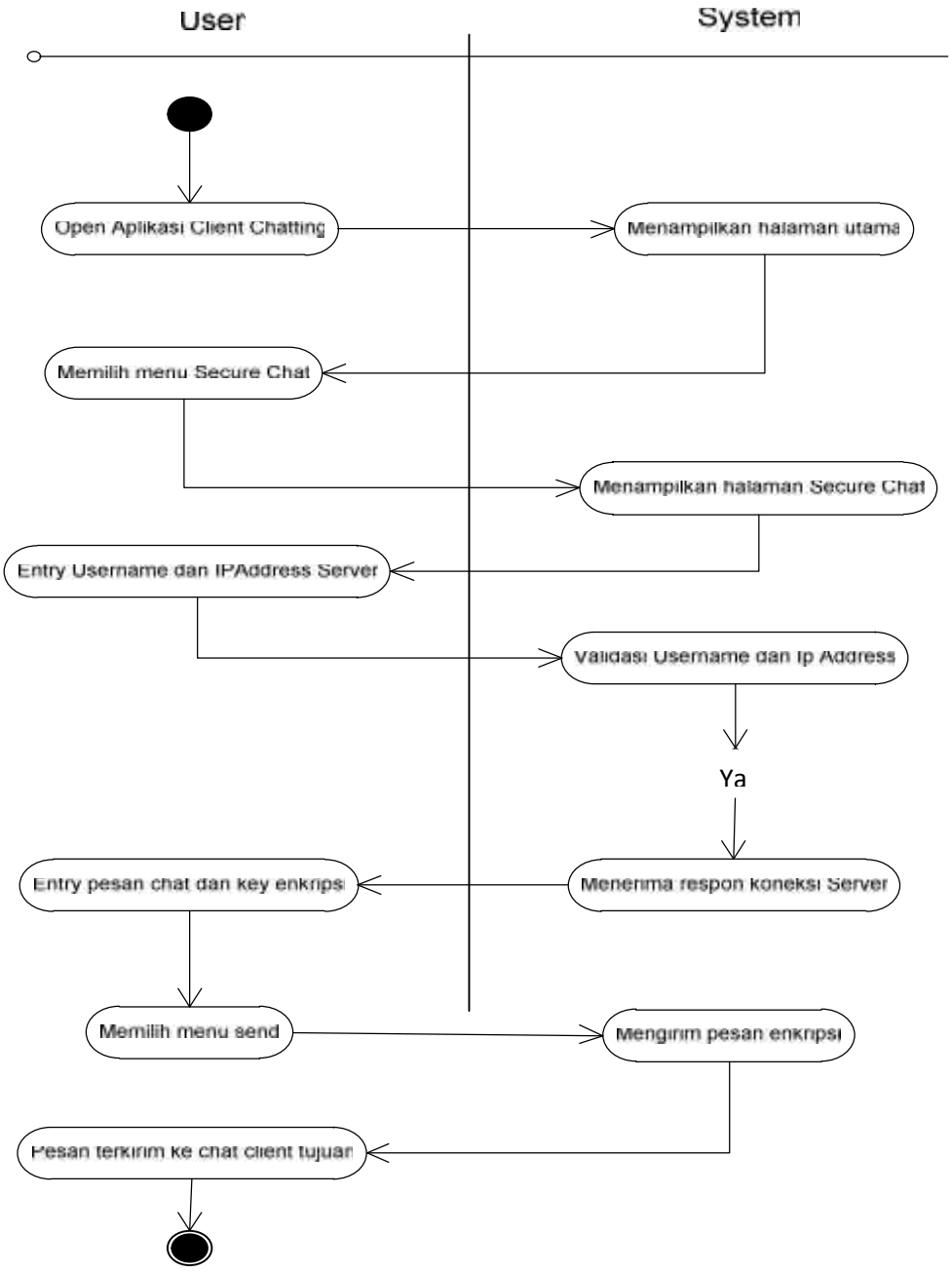
(Sumber : Miftakhul Huda, 2005)

1. *Squence Diagram* aplikasi *Chatting*



Gambar III.3 *Squence Diagram* aplikasi *Chatting*

(Sumber : Miftakhul Huda, 2005)



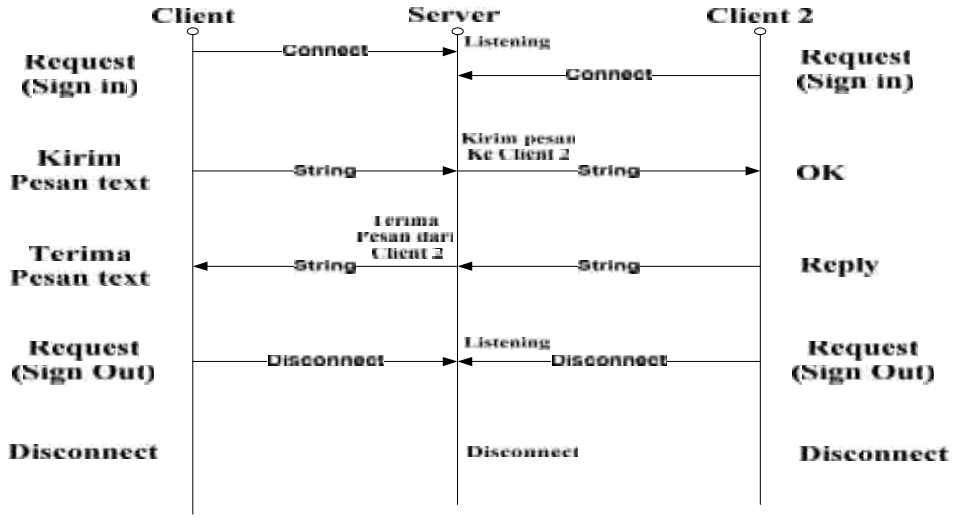
Activity Diagram Aplikasi *Chatting*

Gambar III.4 *Activity Diagram* Aplikasi *Chatting*

(Sumber : Miftakhul Huda, 2005)

2. Desain koneksi antara *Client Chatting* dan *Server Chatting.*

Pembangunan koneksi antara *Client Chatting* dengan *Server Chatting* dilakukan dengan cara *Client Chatting* mendaftarkan *channel* yang digunakan dalam melakukan komunikasi menggunakan sebuah *port. Server Chatting* akan melakukan hal yang sama namun *Server Chatting* akan berperan untuk melakukan proses *listening*. Jika ada *Client Chatting* yang melukukan koneksi, *Server Chatting* akan menyetujui setiap *request* yang masuk ke *Server Chatting*, sehingga koneksi antar *Client Chatting* dan *Server Chatting* dapat terjadi. Setiap *Client Chatting* harus mengetahui *IP Address* dari *Server Chatting* untuk dapat melakukan koneksi. *Port* atau terminal yang akan digunakan untuk melakukan koneksi haruslah merupakan *Port* yang kosong. *Port* yang digunakan dalam aplikasi *chatting* ini adalah *port 2124. Protocol setup* koneksi antara *Client Chatting* dengan *Server Chatting* dapat dilihat pada gambar III.5



Gambar III.5. *Squence Diagram* koneksi antara *Client* dengan *Server*.

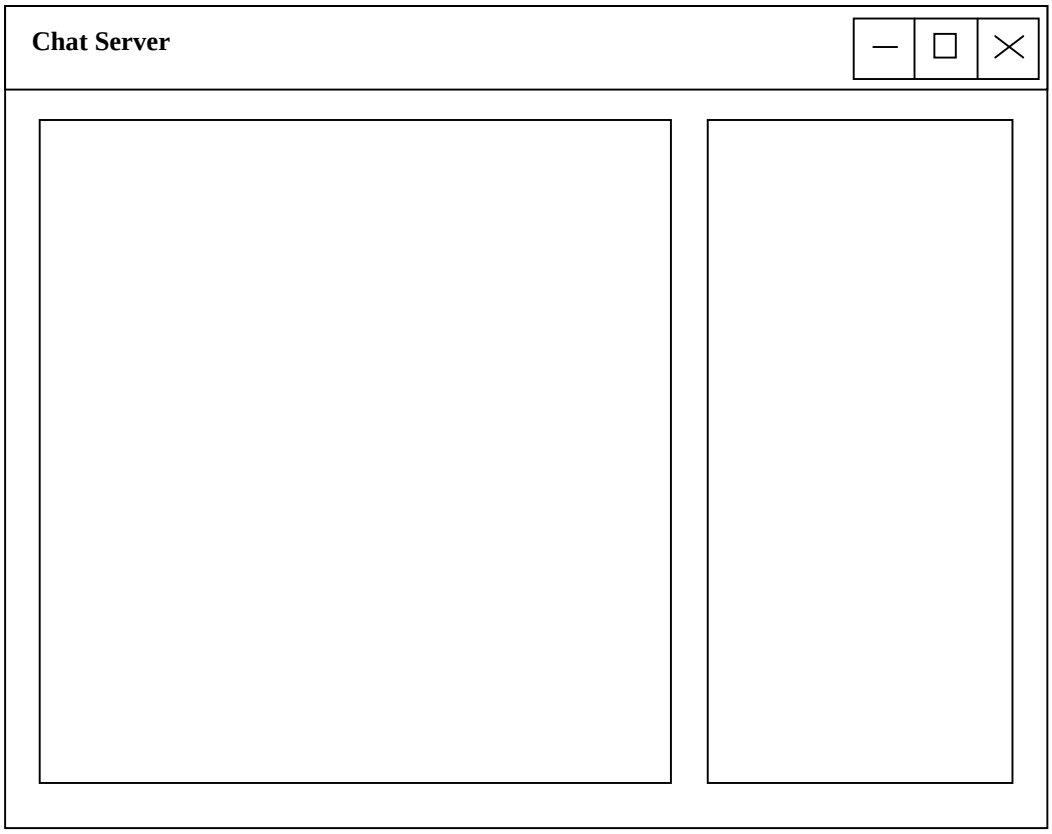
(Sumber : Miftakhul Huda, Membuat Aplikasi Database dengan Java, 2005)

III.4. Rancangan Layar Aplikasi Chatting

Rancangan layar dari aplikasi *chatting* sebagai pesan pada jaringan *Local Area Network* berbasis *Java* adalah sebagai berikut :

1. Rancangan layar *Server Chatting*.

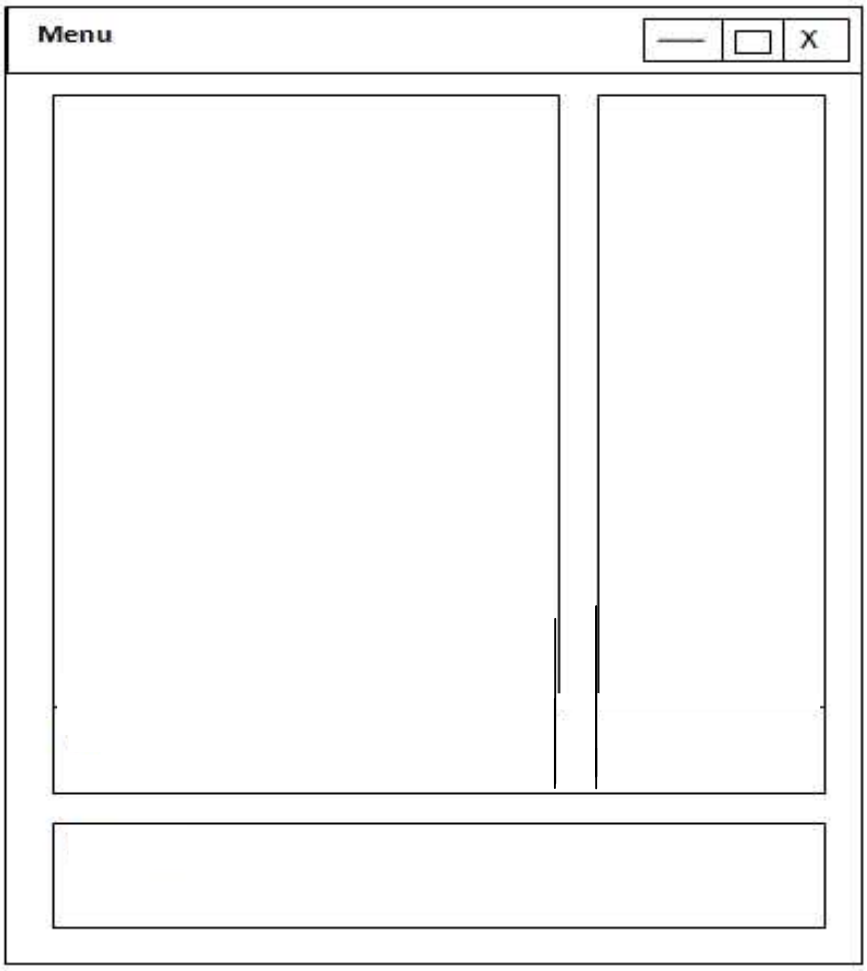
Layar *Server Chatting* pada aplikasi *Chatting* ini berfungsi untuk menampilkan nama *user* yang masuk (*log in*) dan keluar (*log out*) dari *chatroom* yang berjalan pada jaringan *Local Area Network*. Bentuk rancangan layar Server Chatting dapat dilihat pada gambar III.6 :



Gambar III.6. Rancangan layar *Server Chatting*.

2. Rancangan layar *Client Chatting*.

Layar *Client Chatting* pada aplikasi *Chatting* ini berfungsi untuk menampilkan pesan yang dikirim dan diterima oleh *user* yang tergabung dalam *chatroom*. Bentuk Layar *Client Chatting* dapat dilihat pada gambar III.7 .

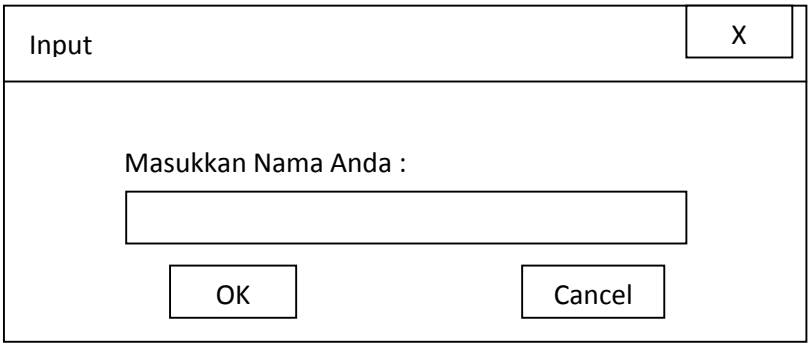


Gambar III.7. Rancangan layar *Client Chatting.*

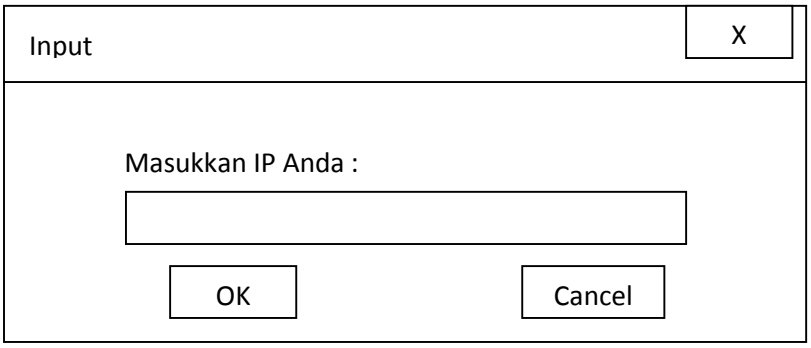
3. Rancangan layar jendela koneksi

Layar jendela koneksi menampilkan *form* untuk nama *user* dan *IP Server* yang ada pada *Client Chatting*. Form ini berfungsi untuk melakukan permintaan (*request*) koneksi kepada *Server Chatting. IP Address* yang dimasukkan harus

sesuai dengan *IPAddress* dimana *Server* berjalan. Bentuk dari layar koneksi *User Name* dapat dilihat pada gambar III.8, dan III.9



Gambar III.8. Layar Koneksi *User Name*



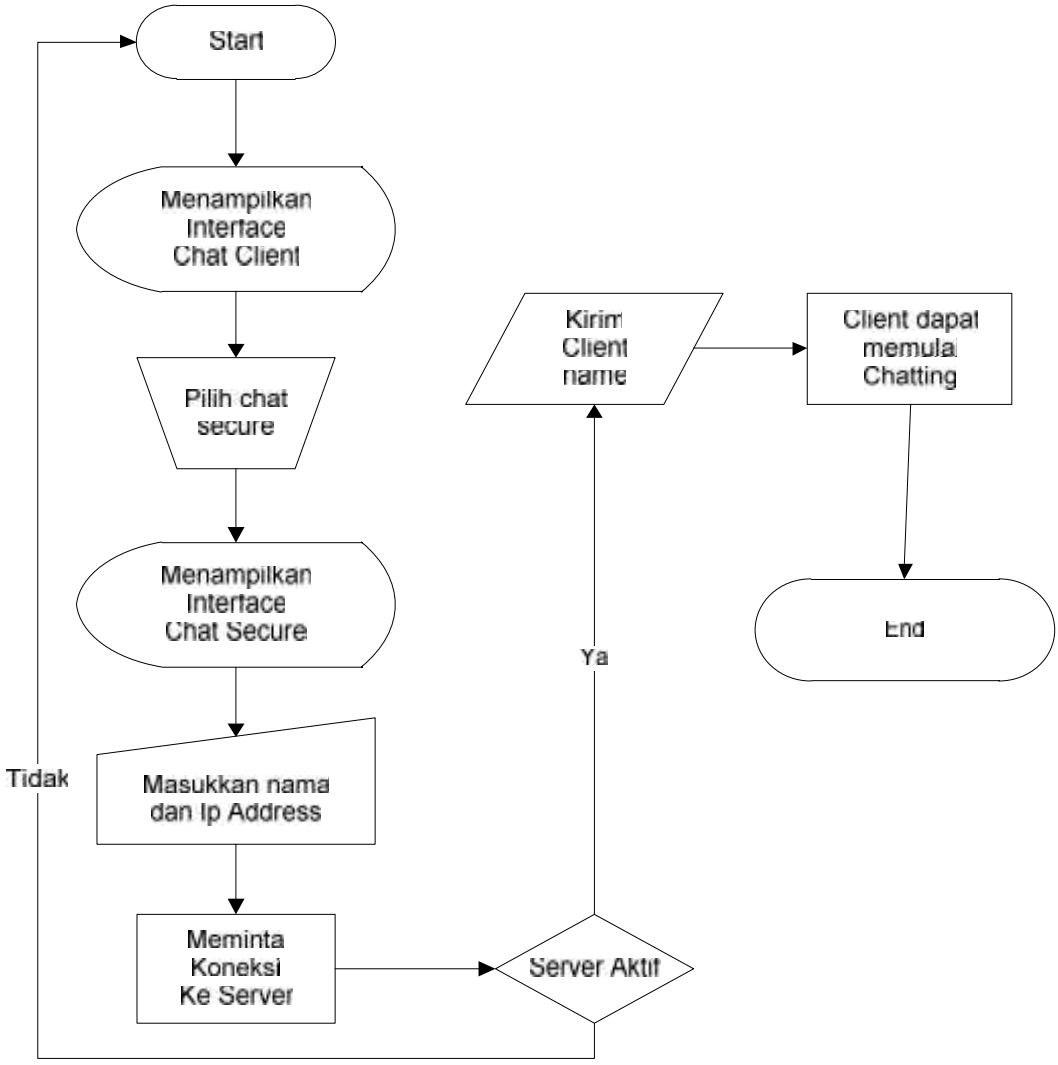
Gambar III.9. Layar Koneksi *IP Address*

III.5 Flowchart Aplikasi *Chatting.*

Flowchart dari aplikasi *Client chatting* sebagai pesan pada jaringan *Local Area network* berbasis *Java* dapat dilihat pada gambar III.10.

1. *Flowchart* aplikasi *Client Chatting.*

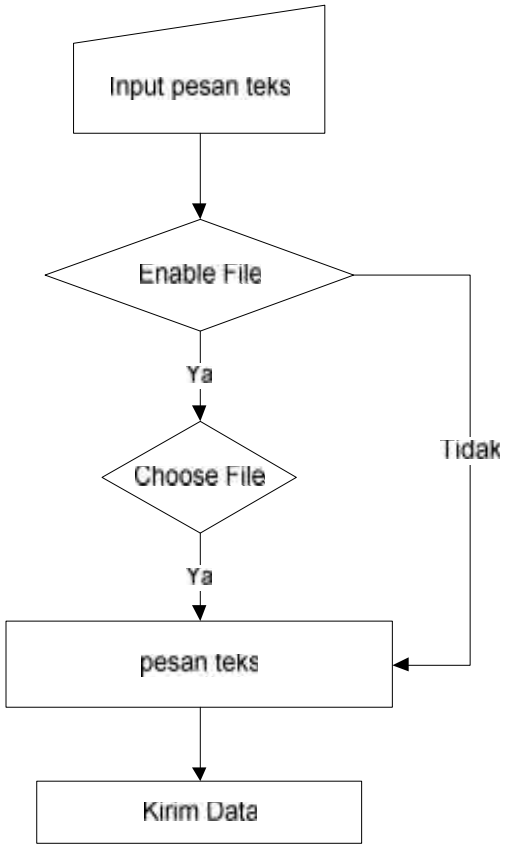
Adapun rancangan aplikasi *client chatting* yang penulis rancang dapat kitalihat pada gambar III.10.



Gambar III.10. *Flowchart* aplikasi *Client Chatting.*

2. *Flowchart* kirim data aplikasi *Chatting*

Berikut merupakan rancangan flowchart kirim data aplikasi chatting dapat dilihat pada gambar III.11

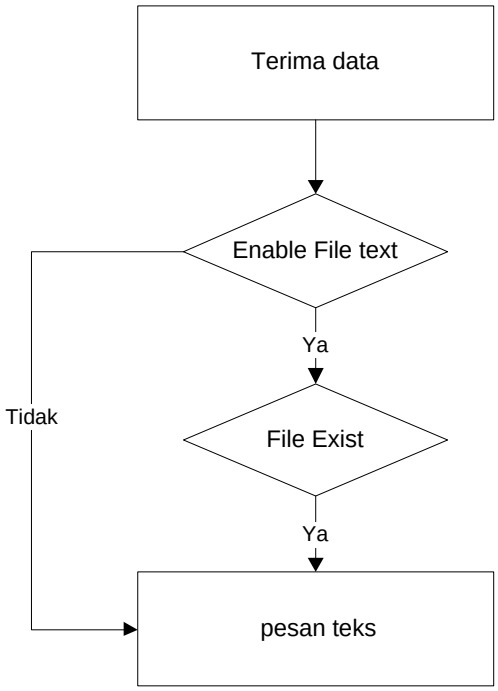


Gambar III.11 *Flowchart* kirim data aplikasi *Chatting*

Pada *Flowchart* tersebut menjelaskan pengiriman data yang berjalan di *dekstop* (*peer to peer*) yang digunakan *client chatting.* Cara kerja pengiriman data adalah *user* mengetikkan pesan teks yang akan dikirim kepada *user* yang lain.

3. *Flowchart* terima data aplikasi *Chatting*

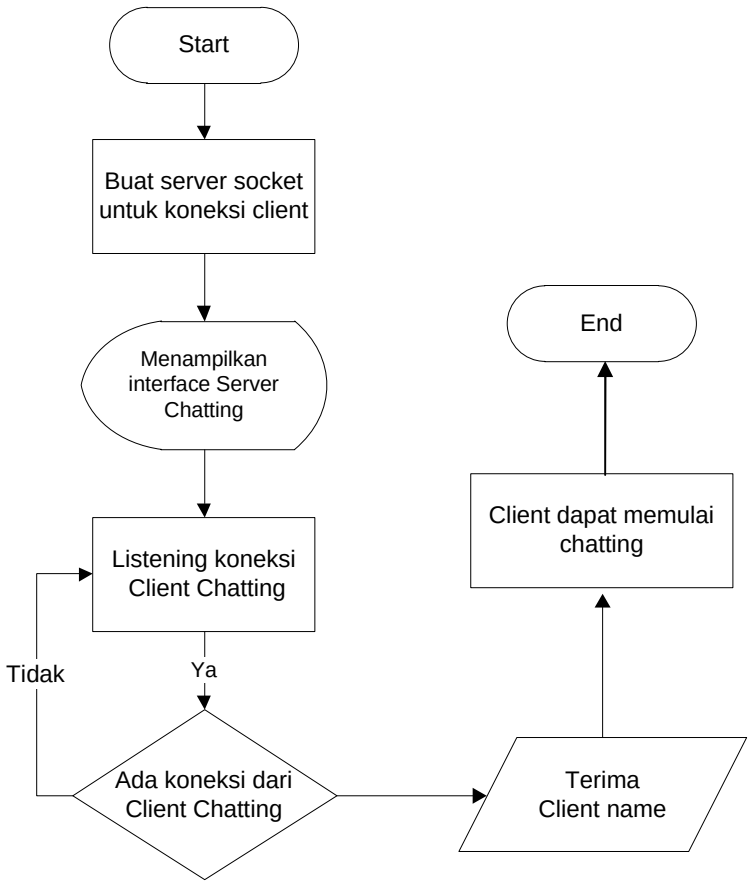
Berikut ini merupakan rancangan terima data pada *aplikasi chatting* dapat dilihat pada gambar III.12.



Gambar III.12 *Flowchart* terima data aplikasi *Chatting*

Adapun sebaliknya data/pesan yang diterima akan di *verifikasi* apakah mengandung file teks atau tidak, lalu kemudian pesan tersebut akan menggunakan

4. *Flowchart* aplikasi *Server Chatting*



Gambar III.13. *Flowchart* aplikasi *Server Chatting*

Berdasarkan gambar *Flowchart* tersebut, cara kerja dari sistem *chatting* adalah sebagai berikut : Pertama kali *client* harus menginputkan *client name* dan *IP Address* kemudian *client* akan meminta koneksi kepada *server*. Jika *server* tidak aktif maka akan muncul pesan *error*, jika *server* aktif atau ditemukan maka koneksi *client* akan diterima. Selanjutnya *server* akan menerima *client name*, kemudian *server* melakukan *validasi* apakah jumlah *client* masih dibawah batas maksimal *client* yang dapat di *handle server*, jika kondisi tersebut tidak terpenuhi maka muncul pesan *error*. Jika kondisi tersebut terpenuhi maka *client* dapat melakukan komunikasi dengan *user* lain. *Client* mengirimkan pesan teks. *Server*

menerima pesan teks. *Server* meneruskan pesan tersebut kepada penerima pesan. Penerima pesan atau *Client* kedua akan menerima pesan tersebut .