

3. PERANCANGAN SISTEM

Pada bab ini akan membahas mengenai perancangan program aplikasi *chatting*. Perancangan program aplikasi *chatting* ini dibagi menjadi 2 bagian, yaitu perancangan *direct chatting* dan perancangan *Indirect chatting*. Dalam *indirect chatting*, *user* dapat mengirimkan pesan ke *user* lain didalam *group*, dengan cara cek *IP address* yang dituju, lalu kata-kata atau pesan yang akan dikirimkan ditulis dalam *database*, setelah itu baru dikirimkan pada *IP address* yang dituju.

Di *indirect chatting* ini, *user* juga dapat mengirimkan *file* apapun ke semua *user* lainnya dalam satu *group* yang dimasukinya. Di dalam program ini juga *user* juga dapat membunyikan suara yang telah disediakan program ini sebagai peringatan untuk *user* lain, yang terjadi komputer yang terhubung satu *group* dengan *user* yang membunyikan suara, maka semua komputer akan mengeluarkan suara secara bersama-sama.

3.1. Cara Kerja Sistem

Pertama, gambar ini akan menggambarkan cara kerja sistem aplikasi *chatting* yang akan dirancang :

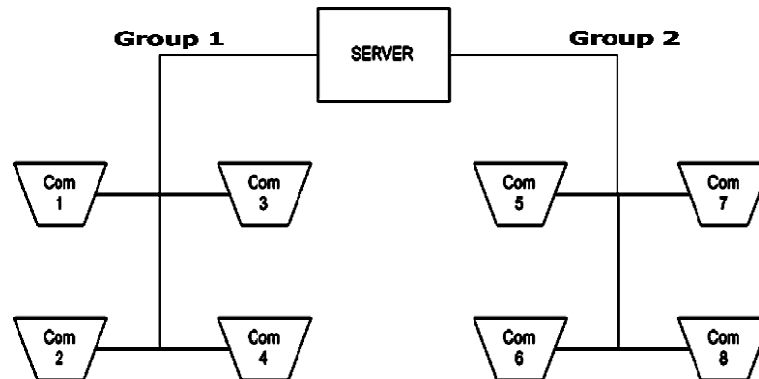


Diagram 3.1 Skema Cara Kerja *Chatting*

Di dalam Gambar 3.1. dapat dilihat bahwa Komp 1, komp 2, komp 3, komp 4 merupakan *group* satu, sedangkan komp 5, komp 6, komp 7, komp 8 merupakan *group* dua. Didalam aplikasi ini jika menginginkan kedua *group* berkomunikasi secara bersama–sama, maka harus mendapat ijin dari *server* yang membuat *group* tersebut.

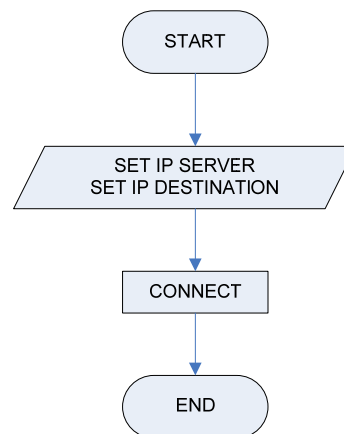
3.2. Aplikasi *Chatting*

3.2.1. *Direct Chatting*

Direct chatting adalah *chatting* secara langsung tanpa menggunakan *database*, sehingga fasilitasnya sangat sedikit.

3.2.1.1. *Flowchart Connect Direct Chatting*

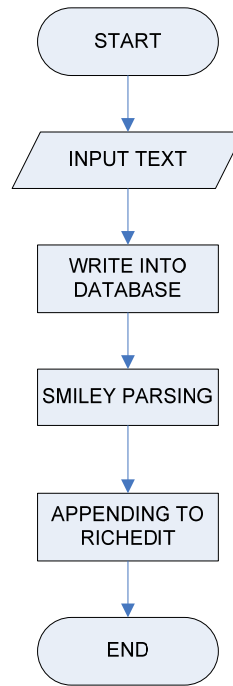
Proses *connect direct chatting* adalah proses pertama kali sebelum seorang *user* dapat melakukan *chatting* dengan *user* lain yang sedang aktif. Proses *connect direct chatting* dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Flowchart Connect Direct Chatting*

3.2.1.2. *Flowchart* Mengirim Pesan

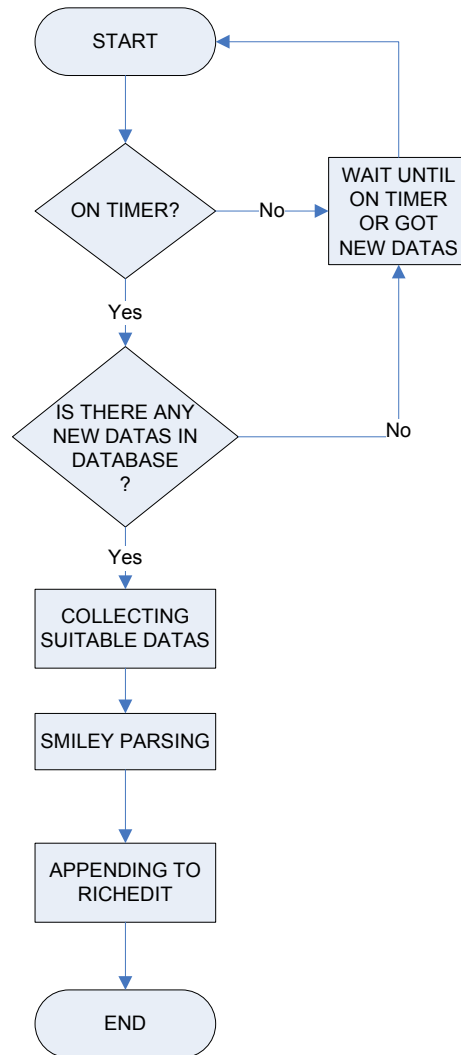
Proses mengirim pesan dilakukan ketika seorang *user* ingin melakukan *chatting* dengan *user* lain yang sedang aktif. *User* berhak memilih *user* yang lain untuk dikirim pesan. Proses mengetik pesan dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Flowchart* Mengirim Pesan

3.2.1.3. Flowchart Menerima Pesan

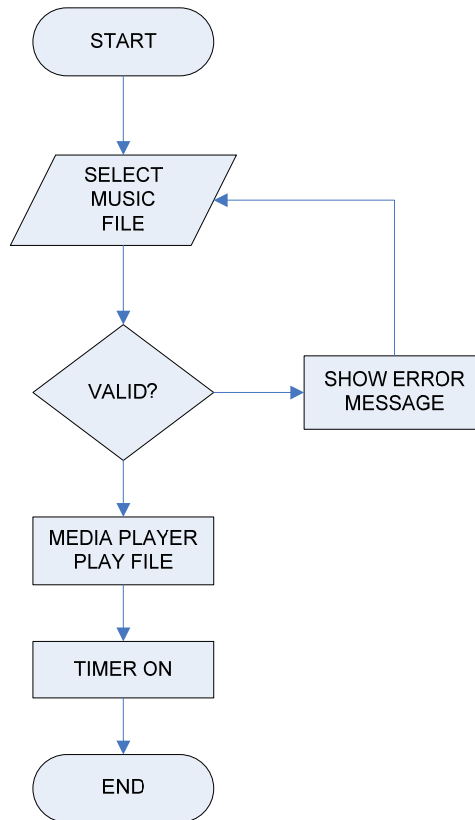
Proses menerima pesan dilakukan ketika seorang *user* telah mengirimkan pesan dengan *user* lain yang sedang aktif. Proses menerima pesan dapat dilihat pada Gambar 3.3.



Gambar 3.3. Flowchart Menerima Pesan

3.2.1.4. *Flowchart* Mendengarkan Musik

Semua *user* bisa mendengarkan musik secara langsung dengan memilih terlebih dahulu file yang ingin di putar lewat *media player*. Proses mendengarkan music dapat dilihat pada Gambar 3.4.



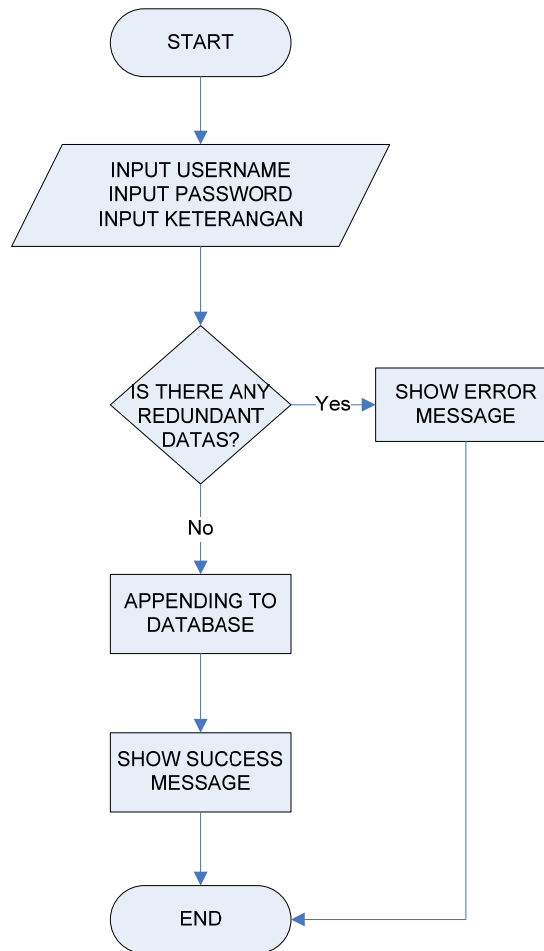
Gambar 3.4. *Flowchart* Mendengarkan Musik

3.2.2. *Indirect Chatting*

Indirect Chatting adalah *chatting* secara terstruktur atau *chatting* yang didalamnya terdapat manajemen *chatting* untuk mengetahui siapa saja yang terhubung dalam aplikasi ini dan harus menggunakan *database*.

3.2.2.1. *Flowchart* Registrasi

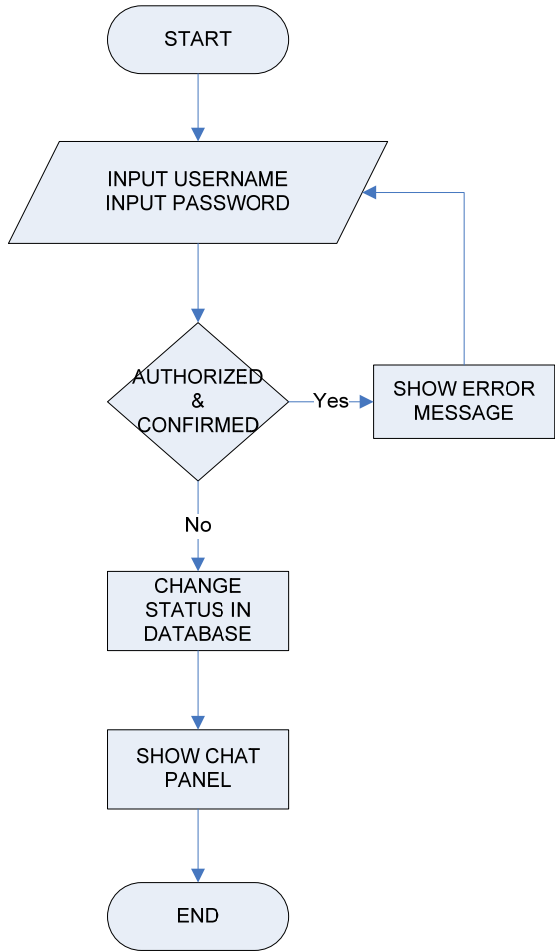
Sebelum melakukan *chatting*, terlebih dahulu pengguna yang belum memiliki akun harus melakukan registrasi terlebih dahulu, adapun proses registrasinya dapat dilihat pada Gambar 3.5.



Gambar 3.5. *Flowchart Registrasi*

3.2.2.2. *Flowchart Login*

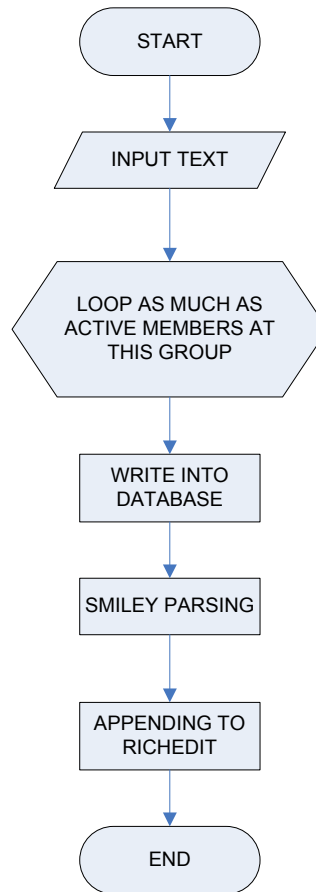
Sebelum melakukan *chatting*, terlebih dahulu pengguna yang memiliki akun harus melakukan terlebih dahulu, adapun proses *login* tersebut dapat dilihat pada Gambar 3.6.



Gambar 3.6. *Flowchart Login*

3.2.2.3. Flowchart Mengirim Pesan

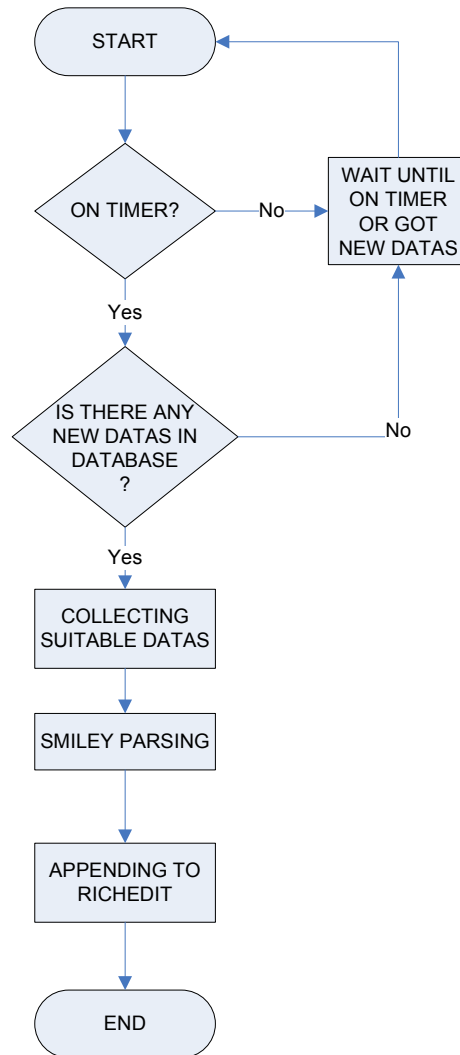
Apabila pengguna akan melakukan *chatting*, terlebih dahulu pengguna harus melakukan pengetikan dan pengiriman pesan kepada pengguna lain yang ingin diajak *chatting*, adapun proses pengiriman pesan dapat dilihat pada Gambar 3.7.



Gambar 3.7. Flowchart Mengirim Pesan

3.2.2.4. Flowchart Menerima Pesan

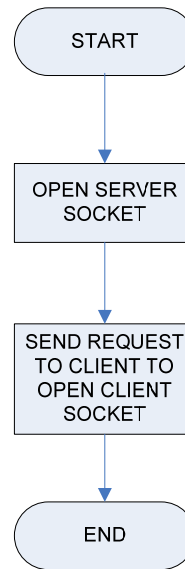
Apabila pengguna sudah mengetik pesan dan mengirimkannya kepada pengguna lain yang akan diajak *chatting*, maka proses menerima pesan dapat dilihat pada Gambar 3.8.



Gambar 3.8. Flowchart Menerima Pesan

3.2.2.5. *Flowchart Server Meminta untuk Mengirim Pesan*

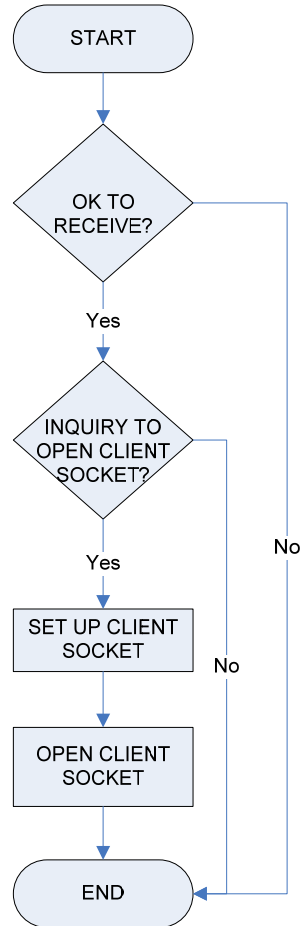
Adapun proses *server* meminta untuk mengirim pesan dapat dilihat pada Gambar 3.9.



Gambar 3.9. *Flowchart Server Meminta untuk Mengirim Pesan*

3.2.2.6. Flowchart Client menerima pesan

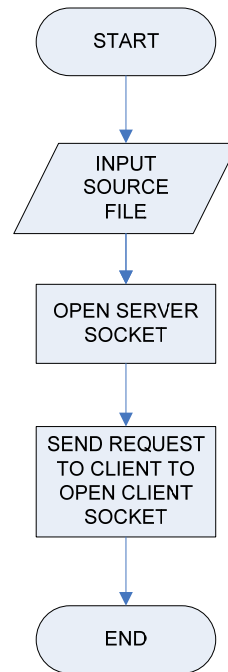
Adapun proses *client* menerima pesan dapat dilihat pada Gambar 3.10.



Gambar 3.10. Flowchart Client Receive Request

3.2.2.7. Flowchart File Transfer

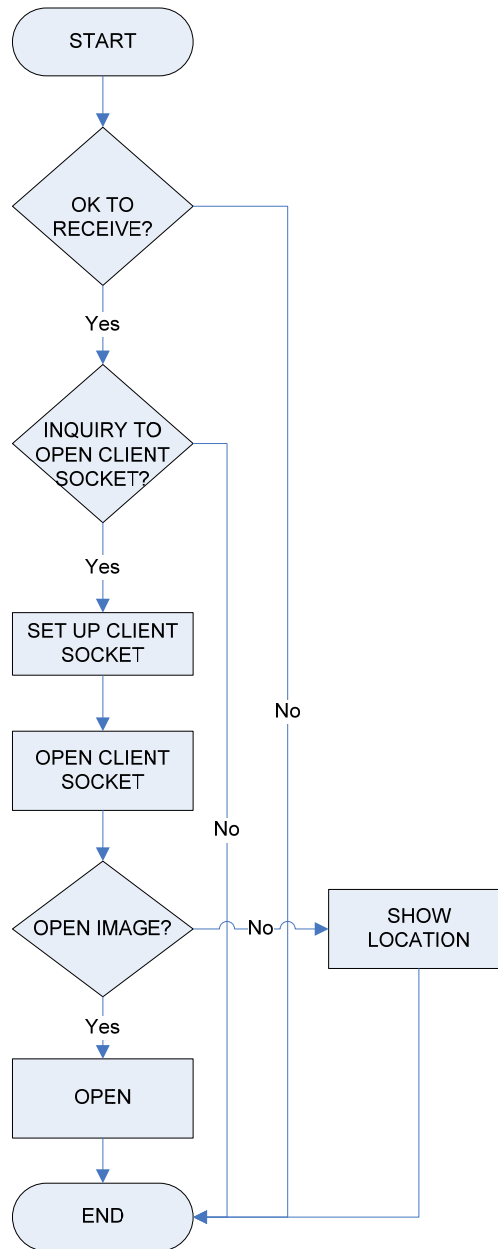
Apabila pengguna akan melakukan proses *file transfer* kepada pengguna yang lain, misalnya berupa pengumuman, adapun prosesnya dapat dilihat pada Gambar 3.11.



Gambar 3.11. Flowchart File Transfer

3.2.2.8. Flowchart Image Transfer

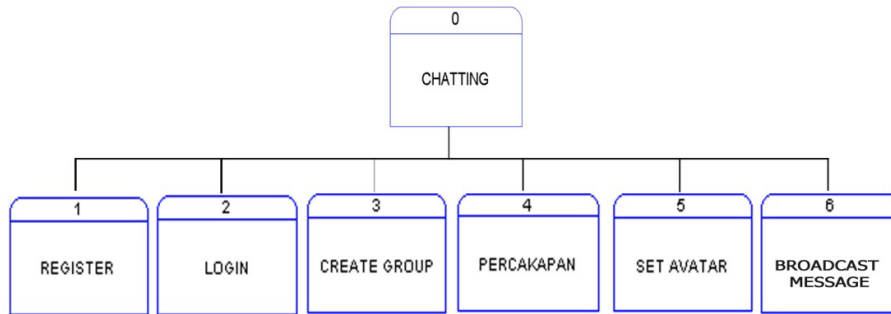
Apabila pengguna akan melakukan proses *image transfer* kepada pengguna yang lain, misalnya berupa bagan, adapun prosesnya dapat dilihat pada Gambar 3.12.



Gambar 3.12. Flowchart Image Transfer

3.2.3. Diagram Berjenjang

Diagram Berjenjang adalah diagram yang berisi hirarki proses apa saja yang terjadi dalam suatu sistem diagram berjenjang dari sistem aplikasi *chatting* ini dapat dilihat sebagai berikut :



Gambar 3.13. Diagram Berjenjang Sistem Aplikasi *Chatting*

3.2.4. Proses-Proses yang ada Berdasarkan Diagram Berjenjang

Seperti yang terlihat pada gambar 3.15., maka kita dapat melihat proses-prosesnya sebagai berikut:

1. Proses Registrasi mengacu pada bagan 1.

Dalam proses ini, pertama – pertama *user* harus melakukan registrasi setelah itu akan disimpan dalam tabel *user*, lalu admin akan menerima request dari *user* dan admin merubah status register menjadi aktif, lalu dikembalikan kepada *user* itu sendiri.

2. Proses *Login* mengacu pada bagan 2.

Di dalam proses ini pengguna dapat memasukkan *username* dan password yang telah didaftarkan oleh pengguna itu sendiri dan diaktifkan oleh admin pada proses sebelumnya. Dalam proses ini *login* mengambil data dari tabel *user* untuk mengetahui apakah data pengguna sudah aktif atau belum.

3. Proses *Create Group* mengacu pada bagan 3

Di dalam proses setelah *login*, pengguna dapat membuat *group* baru untuk berbicara dengan para pengguna lainnya. Setelah pengguna membuat *group* baru, maka data *group* akan disimpan didalam *grouplist*.

4. Proses Percakapan mengacu pada bagan 4

Di dalam proses ini setelah membuat *group* pada proses sebelumnya, dalam proses ini pengguna sudah dapat menggunakan program *chatting* ini, berbicara dengan pengguna lain yang masuk dalam *group* tersebut, dan pembicaraan dapat disimpan ke dalam bentuk teks.

5. Proses *Set Avatar* mengacu pada bagan 5

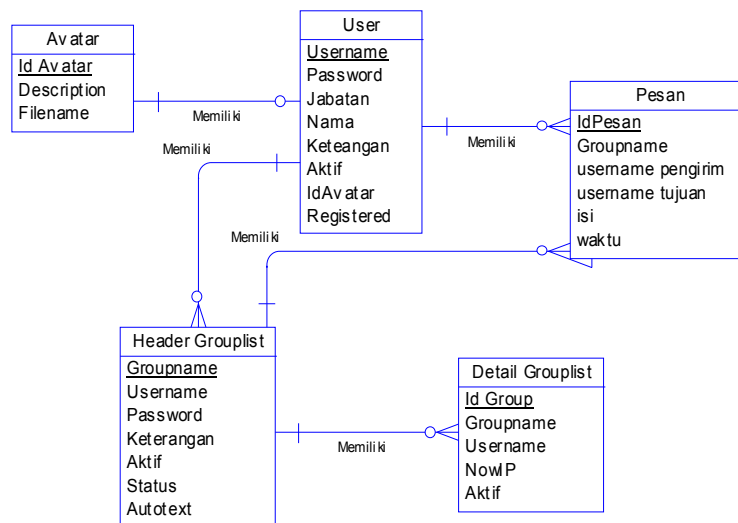
Dalam proses ini, pengguna juga dapat menggunakan fasilitas yang disediakan oleh program *chatting* ini, yaitu dapat memilih gambar yang sebagai *profile* pengguna yang telah disediakan dalam *database*.

6. Proses Broadcast message mengacu pada gambar proses 6

Dalam proses ini, pengguna juga dapat menggunakan fasilitas yang disediakan oleh program *chatting* ini, yaitu dapat mengirimkan *file* teks dengan format *.rtf* kepada pengguna lain

3.2.5. Entity Relational Diagram

Sebelum terbentuknya database yang baik maka harus dimulai perancangan *ER Diagram*. *ER Diagram* ini menampilkan Desain konseptual dari sistem yang akan dibuat seperti gambar dibawah ini. Untuk Pembuatan *ER – Diagram* seperti yang terlihat pada gambar dibawah ini, penulis menggunakan *Software Power Designer*.



Gambar 3.14. *Entity Relationship (ER) Diagram Sistem Aplikasi Chatting*

3.3. Desain Tabel

Dari sistem–sistem yang penulis gunakan, maka tabel–tabel yang penulis gunakan adalah :

3.3.1. Tabel User

Tabel *user* berisikan data-data *user* yang diinputkan oleh pengguna pada saat pendaftaran *user*.

Tabel 3.1. Tabel *User*

Nama	Tipe Data	Ukuran	Keterangan
<i>Username</i> *	char	20	Primary key
password	char	20	
jabatan	char	20	
nama	char	40	
keterangan	char	255	
aktif	integer	6	
idavatar	integer	11	
registered	integer	6	

3.3.2. Tabel Pesan

Tabel pesan berisikan data-data percakapan para pengguna yang disimpan pada saat *chatting*, tetapi hanya bersifat sementara.

Tabel 3.2. Tabel Pesan

Nama	Tipe Data	Ukuran	Keterangan
IDPesan *	integer	10	Primary key
groupname	char	20	
<i>username</i> pengirim	char	20	
<i>username</i> tujuan	char	20	
isi	char	255	
waktu	date/time	19	

3.3.3. *Tabel Header Grouplist*

Tabel *Header Grouplist* berisikan data-data pengguna yang membuat *group* untuk *chatting*.

Tabel 3.3. Tabel Header Grouplist

Nama	Tipe Data	Ukuran	Keterangan
Groupname *	char	20	Primary key
username	char	20	
password	char	20	
Keterangan	char	255	
Aktif	integer	6	
Status	integer	5	
Autotext	char	255	

3.3.4. *Tabel Detail Grouplist*

Tabel *detail grouplist* berisikan data-data pengguna yang membuat *group* untuk *chatting*.

Tabel 3.4. Tabel Detail Grouplist

Nama	Tipe Data	Ukuran	Keterangan
Idgroup *	integer	11	Primary key
groupname	char	20	
username	char	20	
nowip	char	100	
aktif	integer	6	

3.3.5. *Tabel Avatar*

Tabel *avatar* berisikan data gambar yang ada dalam program *chatting*.

Tabel 3.5. Tabel Avatar

Nama	Tipe Data	Ukuran	Keterangan
idavatar *	integer	11	Primary key
filename	char	255	
description	char	255	