

PEMBUATAN PROTOTIPE GAME MULTIPLAYER ONLINE

Ibnu Gunawan, Rudy Adipranata

Universitas Kristen Petra, Jl. Siwalankerto 121-131, Surabaya

E-mail: ibnu@petra.ac.id, rudya@petra.ac.id

ABSTRAKSI

Game merupakan salah satu industri besar di dunia saat ini, khususnya game yang memiliki fitur online. Online game dibuat dengan tujuan agar user dapat berinteraksi dengan user yang lain, sehingga akan mengurangi tingkat kebosanan dalam memainkan sebuah game.

Tujuan penelitian ini adalah untuk mempelajari dan membuat prototipe dari game multiplayer online. Pemrograman dilakukan dengan menggunakan bahasa pemrograman C++, kompiler Microsoft Visual C++.NET, dan dibantu dengan library dari OpenGL Utility Toolkit (GLUT) ver. 3.7.6 untuk grafik serta SXML ver. 0.8 dalam membuat user interface-nya.

Hasil pengujian sistem memperlihatkan bahwa lebih dari satu client dapat terhubung ke server yang sudah listen pada port tertentu. Client sudah dapat berinteraksi satu sama lain, seperti chat, whisper, dan bergerak dalam map sesuai dengan tujuan dari tugas akhir ini.

Kata kunci: Multiplayer, Game Online, Prototipe Game Multiplayer Online

1. PENDAHULUAN

Banyak hal yang dapat mempengaruhi laku atau tidaknya suatu game di pasaran. Hal-hal ini pun berkembang sesuai dengan perkembangan di bidang hardware yang semakin baik. Salah satunya ialah grafik, dengan semakin baiknya kualitas VGA card saat ini grafik 2D yang dulu sudah dinilai cukup dan memuaskan untuk sebuah game sekarang telah ditinggalkan dan banyak yang beralih pada grafik 3D yang dapat menampilkan sebuah objek dari segala sisi. Hal lain yang mempengaruhi ialah *gameplay* dari suatu game, *gameplay* suatu game dapat membuat seorang user memainkan game tersebut terus menerus. Masih banyak juga hal-hal lain yang mempengaruhi seperti musik, jalan cerita (*story line*), dan lain lain.

Tapi jika game yang dibuat tidak memiliki ide-ide yang baru maka memainkan game akan menjadi sangat membosankan, melihat banyaknya game yang beredar saat ini, dengan tipe-tipe yang hampir sama. Banyak ide-ide baru yang dimunculkan dalam membuat game agar user yang memainkan tidak cepat bosan, salah satunya ialah interaksi antar user sehingga menimbulkan suatu komunitas dalam game. Semakin banyak orang yang bermain dalam suatu game tentunya akan mengurangi tingkat kebosanan dari seorang user yang memainkan game tersebut. Saat ini di Indonesia sendiri sedang merebak banyak game online yang memiliki banyak komunitas.

Dari penjelasan di atas dapat dipahami bahwa suatu keunikan tersendiri pada game akan menjadi daya tarik para gamer untuk memainkannya. Baik itu dari segi cerita, *sound*, *style*, *effect*, *gameplay* ataupun grafik yang berbeda. Yang ingin ditekankan dalam penelitian ini adalah mengenai komunitas. Karena suatu komunitas yang terjadi dalam suatu game akan membuat user semakin sering memainkan game tersebut dan tentu saja game itu bertahan di pasaran. Itulah sebabnya prototipe game multiplayer online ini dipelajari dan dibuat.

1.1 Ruang Lingkup penelitian

Agar penelitian bisa terfokus maka penelitian harus dibatasi pada:

- Dari sisi *graphic*, fungsi-fungsi yang akan dibuat hanya difokuskan pada:
 - Fungsi *rendering texture*, untuk dapat membuka *file texture* yang diperlukan seperti: *bmp* (*bitmap*), *MD2* (model karakter)
 - Kamera *full 3D* yang Membuat pandangan kamera yang dapat diputar 360 derajat, sehingga user dapat leluasa melihat karakternya dari berbagai sudut pandang
 - *Zoom*, dapat memperbesar layar dengan titik pusat berdasarkan karakter sehingga dapat diatur sesuai dengan keinginan user
- Dari sisi koneksi *online*, fungsi akan dibuat dengan memiliki fungsi-fungsi dasar untuk berinteraksi antar user dalam game, seperti:
 - Fungsi manajemen, untuk dapat membuat user baru atau menghapus user yang ada
 - *Chatting*, untuk berkomunikasi antar user baik secara umum atau *whisper*
 - Mengirim dan menerima *packet* dengan dasar *message-oriented*
 - Membuat *socket* pada server dan koneksi ke *socket* oleh client
- Dari sisi *gameplay*, Secara singkat permainan dimulai dengan mengisi *username* dan *password* pada *form login*, setelah terhubung dengan server, maka user dapat menjelajahi daerah tempat karakter tersebut berada, bertemu dengan user-user lainnya yang berada pada daerah yang sama dan saling berinteraksi. Tampilan yang akan dihasilkan dari program ini hanyalah tampilan karakter kita dan karakter lainnya yang berada pada server yang sama dapat berjalan bebas pada suatu map yang telah ada, dan masing-masing user dapat melakukan interaksi, baik itu *chatting* umum (terbuka), *whisper* (langsung kepada user yang dituju), *chatting grup* (hanya di dalam grup yang dapat membaca).
- Fungsi-fungsi umum yang diperlukan dalam suatu program seperti:
 - *Database*, Berisi status dari karakter yang dimainkan dan yang telah disimpan, menggunakan *Microsoft Access*

- o Manajemen *memory* yang berguna untuk mengatur penggunaan *memory* agar tidak terjadi kebocoran *memory*
- Teknologi yang dipergunakan dalam pembuatan *game*:
 - o *Software* yang dibuat menggunakan *Microsoft Visual C++ .NET*
 - o Koneksi *client - server* menggunakan *Winsocket ver 2.2*
 - o *Graphic* menggunakan *OpenGL*
 - o *Operating System* menggunakan *Windows XP Professional Edition*

2. DASAR TEORI

2.1 Winsock

Winsock atau *Windows Socket* merupakan salah satu *network programming interface* yang berdasar pada *socket* pada *Microsoft Windows*. *Winsock* digunakan agar sebuah aplikasi dalam *Windows* dapat berkomunikasi dengan aplikasi yang lain baik dalam komputer yang sama atau komputer yang berbeda. *Winsock* telah menjadi standar *socket* yang di sisipkan pada semua versi *Microsoft Windows* sejak *Windows 95*. Versi terakhir dari *winsock* saat ini ialah *Winsock 2.2*.

2.2 Socket

Socket diartikan sebagai suatu cara untuk berkomunikasi antar program dalam suatu jaringan. *Socket* diidentifikasi dengan menggunakan suatu penanda yang unik (*unique identification*) yang biasa disebut sebagai *socket descriptor*. Ada beberapa macam *socket* yaitu: *stream sockets*, *datagram sockets* dan *raw sockets*. Dalam program ini menggunakan *stream sockets* karena *stream sockets* merupakan *connection-oriented*, artinya hubungan baru dapat dilakukan jika *socket* pengirim dan *socket* penerima telah terhubung. *Stream socket* menggunakan *TCP (Transmission Control Protocol)*, *TCP* memastikan bahwa paket yang kita kirimkan sesuai dengan urutan dan bebas *error*, itulah sebabnya *stream socket* bersifat *reliable* dan diterima sesuai dengan urutan pada saat dikirim, misalnya : jika kita mengirim 2 buah *item* dengan urutan "1,2", maka pada saat diterima akan diterima sesuai urutan "1,2".

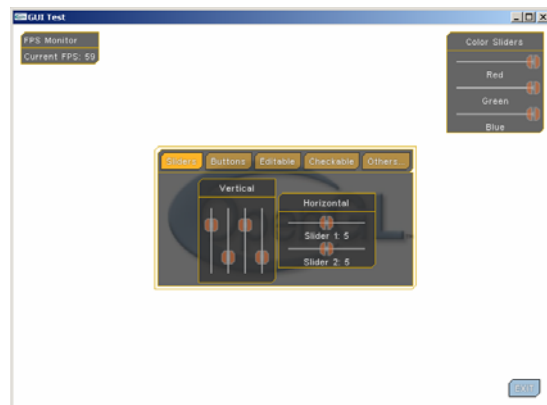
2.3 OpenGL

OpenGL adalah salah satu interface untuk 3D grafik *programming* yang dikembangkan oleh *Silicon Graphics Inc* (www.opengl.org). Dalam *OpenGL* terdapat fungsi-fungsi yang sering digunakan *programmer* untuk membuat suatu program yang menggunakan grafik baik 2D atau 3D, seperti *matrix*, *texture mapping*, *lighting effect*, transformasi, menggambar titik, garis dan *polygon*, dan lain sebagainya.

2.4 SXML

SXML merupakan salah satu *GUI* untuk pengguna bahasa *C++*, yang memungkinkan *user* untuk menambahkan *control* seperti *button*, *checkbox*, *radio button*, *edit box*, *scroll bar* pada *OpenGL* program. *SXML* dapat menggunakan *xml* dalam membuat *GUI*, Sehingga untuk mengubah *GUI*, tidak perlu merubah *source code*, tapi hanya mengubah

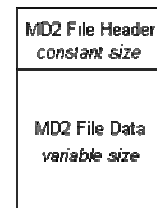
xml-nya saja. Versi terakhir dari *SXML* adalah versi 0.85. Gambar 1 merupakan contoh penggunaan *SXML*. (<http://www.realityflux.com/abba/>)



Gambar 1. Contoh tampilan *SXML*

2.5 Model MD2

MD2 file format adalah sebuah 3D model file format yang digunakan oleh *Id Software's Quake II engine*. File *MD2* terbagi menjadi dua bagian yaitu *header* dan data seperti yang terlihat pada gambar 2.



Gambar 2. *MD2 File Format*

Listing 2.1 berikut struktur *header* dari *MD2 file format*:

```

/* md2 header */
typedef struct
{
    int ident;           /* magic number:
"IDP2" */
    int version;        /* version: must be
8 */
    int skinwidth;      /* texture width */
    int skinheight;     /* texture height */
    int framesize;      /* size in bytes of
a frame */
    int num_skins;      /* number of skins
*/
    int num_vertices;   /* number of
vertices per frame */
    int num_st;         /* number of texture
coordinates */
    int num_tris;       /* number of
triangles */
    int num_glcmds;     /* number of opengl
commands */
    int num_frames;     /* number of frames
*/
    int offset_skins;   /* offset skin data
*/
    int offset_st;      /* offset texture
coordinate data */
    int offset_tris;    /* offset triangle
data */
    int offset_frames; /* offset frame data
*/
    int offset_glcmds; /* offset OpenGL
command data */
    int offset_end;     /* offset end of file*/
} md2_header_t;
    
```

Mengingat program ini lebih difokuskan pada sistem jaringannya, pada program ini tidak lagi dibuat fungsi untuk dapat me-load MD2 file format, tapi menggunakan fungsi yang sudah tersedia di internet.

2.6 Collision Detection

Collision detection adalah fungsi yang digunakan untuk memeriksa tabrakan yang bisa terjadi saat karakter digerakkan. Dalam project ini fungsi collision detection dibuat sederhana hanya untuk memeriksa apakah karakter utama bertabrakan dengan karakter lain. Pengecekan dilakukan berdasar posisi karakter lain ditambah/dikurangi dengan range tertentu.

3. DESKRIPSI SISTEM

Pada bab ini akan dibahas deskripsi kerja dari prototipe game online. Secara garis besar, sistem terdiri dari dua bagian utama, yaitu: client dan server. Server dibagi menjadi tiga bagian, salah satu alasannya ialah untuk meringankan beban kerja dari masing-masing server jika suatu saat banyak client yang login. Artinya dalam pengembangannya nanti, bisa saja login-server, char-server dan map-server ditempatkan di server yang berbeda satu sama lain. Di satu sisi memecah menjadi tiga bagian juga agar mudah untuk membuatnya, mengecek error dan bug yang ada.

Dalam project ini server dibagi menjadi tiga bagian, yaitu:

- Login-server, yang bertugas untuk menangani client saat login kedalam game
- Char-server, yang bertugas untuk menangani data-data karakter dari client yang telah login dengan sukses
- Map-server, yang bertugas untuk menangani semua action antar client dan server pada saat berada di dalam game.

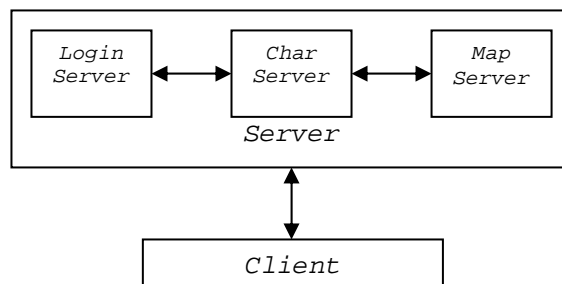
Server harus dijalankan berurutan dari login-server, char-server, lalu map-server. Ini dilakukan agar char-server dapat mengirim informasi yang dimiliki ke login-server, begitu juga dengan map-server yang harus mengirim informasi yang dimiliki ke char-server.

Setelah server dijalankan, maka client dapat berkomunikasi dengan server. Secara singkat client akan melakukan koneksi awal dengan login-server untuk melakukan proses login. Setelah login sukses, client akan memutuskan hubungan dengan login-server dan menghubungkan dengan char-server. dalam char-server, client akan memperoleh data dari semua karakter yang dimiliki oleh username yang digunakan saat login. Setelah selesai mengambil data, client akan memutuskan hubungan dengan char-server, dan akan menghubungkan dengan map-server. Dalam map-server, client akan melakukan proses rendering sesuai dengan data yang telah diperoleh dari char-server.

Client yang dibuat menggunakan OpenGL sebagai engine grafiknya, dan SXML ver. 0.8 sebagai interface (GUI) dari program. Semua yang berhubungan dengan grafik, camera, map, model karakter, texture, interface (GUI) diprogram pada client. Secara lebih mendetail, sistem dibagi menjadi enam bagian besar, yaitu:

- Fungsi-fungsi dasar untuk client dan server.

- sistem login-server.
- sistem char-server.
- sistem map-server.
- database.
- client.



Gambar 3. Garis Besar Sistem Kerja Program

4. IMPLEMENTASI SISTEM

Secara lengkapnya, bab ini akan dibagi menjadi enam bagian besar yang masing-masing bagian akan menjelaskan implementasi sistem. Bagian-bagian itu adalah:

- Implementasi fungsi-fungsi dasar
- Implementasi login-server
- Implementasi char-server
- Implementasi map-server
- Implementasi database
- Implementasi client

4.1 Implementasi Fungsi-fungsi Dasar

Fungsi-fungsi dasar yang diimplementasi disini disiapkan untuk digunakan baik pada client maupun server. Beberapa fungsi dasar yang diimplementasi dalam project ini ialah: pembuatan class CSocketManager (socket), CSimpleClientSocket (client), CSimpleServerSocket (server), dan pembuatan class Database yang berfungsi untuk menangani file database berekstensi .mdb yaitu file database yang berasal dari Microsoft Access.

4.2 Implementasi Login-Server

Login-server digunakan untuk menangani client yang melakukan proses login untuk masuk ke dalam game. Login-server menggunakan database logindata yang terdapat satu direktori dengan login-server. Login-server menggunakan class CSocketManager, class CSimpleServerSocket dan class Database. Sedangkan untuk struktur data, login-server menggunakan dua buah yaitu: struktur data ClientData dan ServerData. Listing 4.1 dan 4.2 berikut merupakan struktur data dari ClientData dan ServerData:

```

struct ClientData
{
    string    _Date;
    string    _Time;
};
    
```

Listing 4.1 Struktur Data ClientData

```
struct ServerData
{
    string      _Name;
    sockaddr_in _Address;
    u_int      _Port;
    int        _Socket;
};
```

Listing 4.2 Struktur Data *ServerData*

ClientData digunakan sebagai untuk menyimpan informasi tambahan dari *client* yang terhubung ke *server*. Pada *login-server*, *ClientData* berisi variabel *_Date* dan *_Time* yang berfungsi untuk menyimpan waktu dan tanggal saat *client* berhasil *login* ke *server*. Informasi ini nantinya akan di simpan dalam tabel login dalam *database* logindata.

Sedangkan *ServerData* merupakan struktur data untuk menyimpan data-data dari *char-server* yang terhubung dengan *login-server*. variabel *_Name* berisi nama dari *char-server* yang terhubung, *_Address* merupakan *Ip Address* dari *char-server*, *_Port* merupakan informasi yang dikirim dari *char-server* mengenai *port* berapa *char-server* *listen*, dan *_Socket* berisi *socket descriptor* dari *char-server*.

4.3 Implementasi Char-Server

Char-server digunakan untuk menangani *client* yang telah berhasil melakukan proses *login* pada *login-server*. *Char-server* menggunakan *database* *chardata* yang terdapat dalam satu direktori yang sama dengan *char-server*. *Char-server* menggunakan *class CSocketManager*, *class CSimpleServerSocket*, *class CSimpleClientSocket* dan *class Database*.

Sedangkan untuk struktur data, *char-server* hanya menggunakan struktur data *ServerData*. Struktur dari *ServerData* sama dengan struktur *ServerData* yang terdapat pada *login-server*.

4.4 Implementasi Map-Server

Map-server digunakan untuk menangani *client* yang telah berhasil mengambil data karakter dan memilihnya pada *char-server*.

Map-server menggunakan *class CSocketManager* dan *class CSimpleServerSocket*. Sedangkan untuk struktur data, *char-server* hanya menggunakan struktur data *ClientData*. Struktur dari *ClientData* berisi informasi tambahan dari *client* yang terhubung ke *map-server*. Listing 4.3 berikut merupakan struktur data dari *ClientData*:

```
struct ClientData
{
    int          _AccId;
    int          _Socket;
    int          _CharId;
    string      _CharName;
    string      _Job;
    int          _Level;
    int          _PartyId;
    string      _Map;
    float       _PosX;
    float       _PosY;
    float       _PosZ;
};
```

Listing 4.3 struktur data dari *ClientData*

4.5 Implementasi Database

Project ini membutuhkan *database* untuk menyimpan data-data *client* seperti *username* dan

password hingga data-data dari karakter. Project ini memiliki dua buah *database*, yaitu *database* logindata dan *chardata*.

Database logindata terdapat dalam direktori yang sama dengan *login-server*, sedangkan *database* *char-data* terdapat pada direktori yang sama dengan *char-server*. Dalam *database* logindata terdapat tabel login yang berisi data-data *client* yang digunakan saat login. sedangkan pada *database* *chardata* terdapat tabel *chardata* yang berisi data-data karakter dari sebuah *username*.

4.6 Implementasi Client

Client dibuat menggunakan *GLUT* (*OpenGL Utility Toolkit*) ver. 3.7.6 sebagai *engine* grafiknya dan *Winsock* ver. 2.0 sebagai *engine* jaringannya. Untuk *interface*, *project* ini memakai *XML* ver. 0.8.

Client dibuat untuk dapat berkomunikasi dengan *login-server*, *char-server* dan *map-server*, sekaligus dapat menampilkan data-data yang diterima dari *map-server* kedalam *object 3D* sesuai dengan informasi yang didapat.

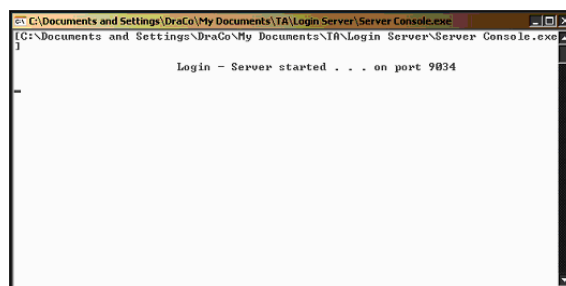
5. PENGUJIAN SISTEM

Pada bab ini akan dibahas pengujian sistem pada prototipe *game online* yang telah diimplementasi pada bab 4. Implementasi sistem. Secara garis besar pengujian sistem akan dibagi menjadi empat bagian besar. Bagian-bagian itu adalah:

- Pengujian terhadap *login-server*
- Pengujian terhadap *char-server*
- Pengujian terhadap *map-server*
- Pengujian terhadap *client*

5.1 Pengujian Terhadap Login-Server

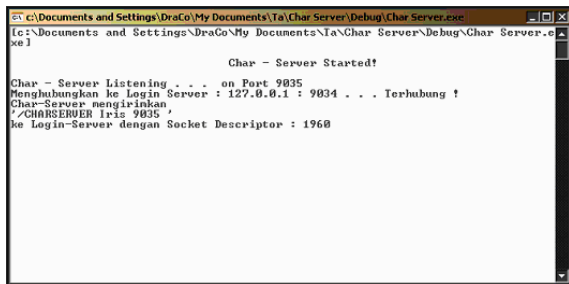
Login-server adalah *server* utama dalam program ini, saat dijalankan *login-server* akan *listen* sekaligus menunggu message yang masuk pada *port* 9034.



Gambar 4. *Login-server* Saat *Listen* Pada *Port* 9034

5.2 Pengujian Terhadap Char-Server

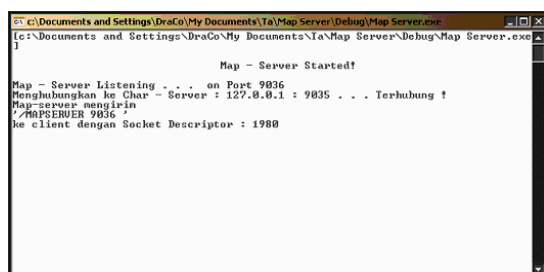
Setelah *login-server* dijalankan, *server* selanjutnya yang dijalankan adalah *char-server*. *Char-server* akan *listen* pada *port* 9035, dan setelah *listen* *char-server* akan melakukan hubungan ke *login-server* sebagai *client*. Jika *char-server* tidak dapat melakukan hubungan ke *login-server*, maka *char-server* akan *exit* secara otomatis.



Gambar 5. Char-Server Saat Listen Pada Port 9035 dan melakukan hubungan ke Login-Server

5.3 Pengujian Terhadap Map-Server

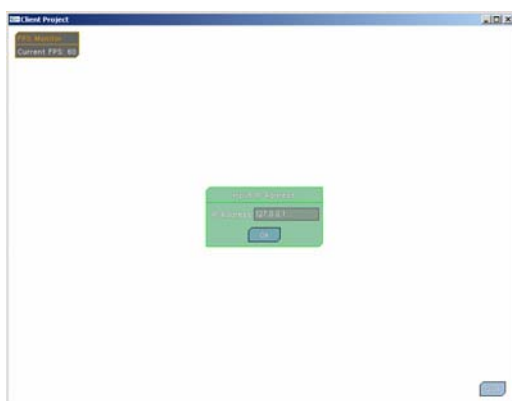
Setelah *char-server* dijalankan, *server* selanjutnya yang dijalankan adalah *map-server*. *Map-server* akan *listen* pada *port* 9036, dan setelah *listen* *map-server* akan melakukan hubungan ke *char-server* sebagai *client*. Jika *map-server* tidak dapat melakukan hubungan ke *char-server*, maka *map-server* akan *exit* secara otomatis.



Gambar 6. Map-Server Saat Listen dan Telah Terhubung ke Char-Server

5.4 Pengujian Client

Saat *client* dijalankan, *interface* dari input *IP address* akan langsung ditampilkan, seperti pada gambar 5.19. Pada gambar 5.19. diperlihatkan *client* meminta inputan berupa *IP address* dari *login-server*, dalam pengembangannya nanti kalau *IP address* dari *login-server* sudah tidak berubah, maka *interface* ini dapat diabaikan.



Gambar 7. Interface Input IP Address

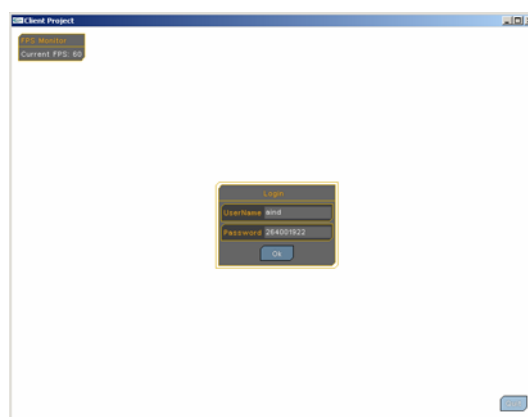
Setelah *user* mengisi *IP address* yang ingin dihubungi pada *textbox* *interface* input *IP address*, *client* akan berusaha melakukan hubungan dengan *login-server* sesuai dengan *IP address* yang telah diinputkan.

Jika tidak dapat dihubungi, *client* akan menampilkan *interface error box*. Untuk tampilan

error box akan ditampilkan pada bagian akhir dari bab ini. Sebaliknya jika *login-server* dapat dihubungi, maka *client* akan menampilkan *interface login*, dan menutup *interface* input *IP address*.

Pada gambar 8. diperlihatkan *client* yang telah terhubung ke *login-server*. Pada *interface login*, ditampilkan dua buah *textbox* untuk *username* dan *password*, yang nantinya akan diinputkan oleh *client* yang ingin masuk kedalam game.

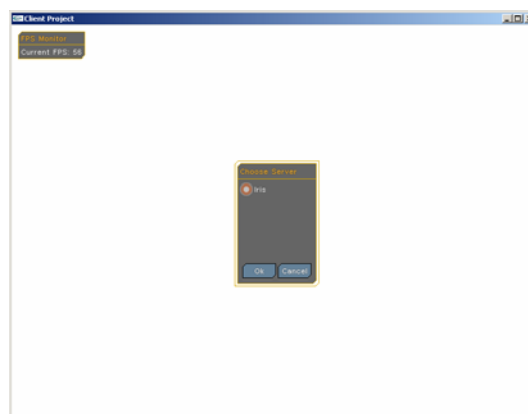
Selanjutnya, *client* akan menampilkan data server yang diterima pada *interface* pilih server, seperti yang ditampilkan pada gambar 9.



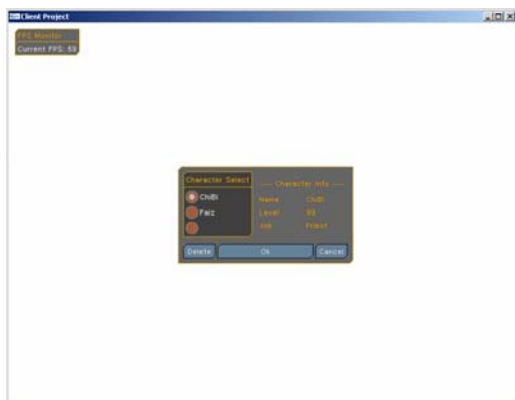
Gambar 8. Interface Login

Setelah memilih *server* pada *interface* pilih server, *client* akan berusaha untuk melakukan koneksi ke *char-server* yang dipilih. Ini dilakukan dengan memilih server yang ada dan mengklik button "Ok". Saat terhubung dengan *char-server*, *interface* pilih karakter akan ditampilkan seperti pada gambar 10.

Pada *interface* ini, *client* dapat memilih karakter yang ingin digunakan dalam game. Setiap *username* memiliki tiga buah karakter maksimal. *Interface* ini juga menampilkan informasi dari karakter yang dipilih, seperti terlihat pada gambar 10 karakter yang dipilih bernama "ChiBi" dengan level "99" dan job-nya adalah "Priest".



Gambar 9. Interface pilih server

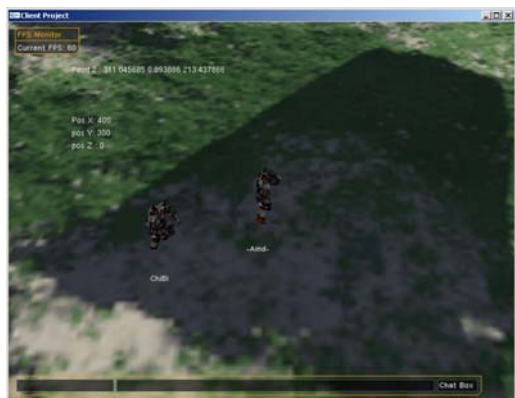


Gambar 10. Interface Pilih Karakter

Pemilihan karakter dilakukan dengan memilih karakter yang ada dan selanjutnya diakhiri dengan mengklik button "Ok". Setelah karakter yang diinginkan dipilih pada interface pilih karakter, client akan memutuskan hubungan dengan char-server, dan berusaha menghubungkan dengan map-server, sesuai dengan informasi yang didapat dari char-server.

Setelah terhubung ke map-server, client akan mulai me-render dunia dari game menggunakan OpenGL. Setelah map selesai di-render, client akan mulai me-render karakter yang berada pada map itu, termasuk karakter-karakter lain yang data-datanya telah diterima dari map-server dan disimpan pada client. Proses ini seperti terlihat pada gambar 11.

Setelah masuk dalam game, akan ditampilkan interface chat box seperti yang terlihat pada bagian bawah dari gambar 5.8. Chat box ini berguna sebagai alat komunikasi antara karakter dengan karakter lainnya. Melalui chat box ini user dapat mengirimkan pesan baik itu secara umum yang dapat didengarkan oleh semua orang yang berada pada satu map, atau pesan secara khusus yang ditujukan hanya untuk satu karakter lain.



Gambar 11. Keadaan Game Saat Client Terhubung Pada Map-Server

6. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

- Server dibagi menjadi tiga bagian untuk meringankan beban kerja server. Server dibagi berdasar tugasnya yaitu: login-server yang menangani proses login, char-server yang menangani proses pemberian data karakter, dan

map-server yang menangani segala action dalam game.

- Jika ada salah satu dari server yang tidak listen, maka client tidak akan dapat terhubung kedalam game.
- Proses terpenting dalam program client-server menggunakan winsock adalah pada saat pengecekan message yang masuk. Pada program ini digunakan asynchronous socket, karena server perlu untuk memeriksa message yang masuk sekaligus dapat mengirim message jika diperlukan, sedangkan digunakan pada client karena client akan selalu mengecek message yang masuk, sekaligus me-render objek OpenGL.
- Client yang terhubung sudah dapat melakukan action seperti: chat, whisper, dan bergerak pada map yang ada, sekaligus sudah dilakukan pengecekan collision detection.
- Pengiriman data posisi karakter dari client ke server tidak dilakukan secara real time, ini dikarenakan akan membebaskan server. Pada program ini client hanya mengirimkan data dari posisi yang ingin dituju oleh karakter pada saat penekanan klik kiri dilakukan. Sehingga client yang lain hanya menerima posisi akhir dan men-generate sendiri pergerakan posisi dari posisi terakhir karakter ke posisi yang ingin dituju dengan menggunakan rumus interpolasi linear.

DAFTAR PUSTAKA

- Angel, Edward. *Interactive Computer Graphics*. Addison Wesley Longman, Inc. 2000.
- Beej's. *Beej's Guide to Network Programming*. 2005. <<http://jcatki.no-ip.org/beej/intro.htm>>
- Byrne, Greg. *Security Issues of Online Gaming*. *Online Game*. 2005. <<http://www.gamedev.net/reference/articles/article2062.asp>>
- Game and Networking Tutorials*. 2003. <<http://www.gametutorials.com>>
- Game Tutorials*. 2004. <<http://www.ultimategametutorials.com>>
- Hajnoczi, Stefan. *Winsock 2 For Games*. 2005. <<http://www.gamedev.net/reference/programming/features/ws2games/default.asp>>
- Harrington, Steven. *Computer Graphics a Programming Approach*. McGraw-Hill Book Company. 1987.
- MMORPG Tutorial*. 2004. <<http://www.onrpg.com>>
- OpenGL Information*. 2005. <<http://www.opengl.org>>
- OpenGL Tutorials*. 2005. <<http://nehe.gamedev.net>>
- SXML Website*. 2005. <<http://www.realityflux.com/abba>>