

PROTOTYPE GAME 2D MENGGUNAKAN ISOMETRIC VIEW DAN DIRECT X

Djoni Haryadi Setiabudi, Ibnu Gunawan

Fakultas Teknologi Industri, Jurusan Teknik Informatika - Universitas Kristen Petra

e-mail : djonihs@petra.ac.id, e-mail : ibnu@petra.ac.id

Aristotales Lengkong

Alumni Fakultas Teknologi Industri Jurusan Teknik Informatika - Universitas Kristen Petra

ABSTRAK: Pembuatan game dua dimensi lebih mudah dari pembuatan game tiga dimensi. Selain itu tuntutan *hardware* yang diperlukan untuk game dua dimensi lebih sederhana, dan untuk membuat game tiga dimensi dibutuhkan tim pembuat yang lebih besar. Untuk itu pada penelitian ini dicoba untuk dibuat suatu prototype game dua dimensi yang menggunakan tampilan *isometric* untuk memberikan kesan adanya tiga dimensi tanpa perlu membuat sistem tiga dimensi sesungguhnya.

Game dua dimensi yang dibuat cukup sederhana, sekedar untuk menunjukkan tampilan *isometric*, yang dapat dijalankan pada komputer berbasis Windows. Pembuatan aplikasi dilakukan dengan menggunakan program Microsoft Visual C++ dan Microsoft DirectX 8.1 Software Development Kit serta *library* tambahan IsoHexDef dan WavLoader untuk membantu pembuatan aplikasi.

Prototype yang berhasil dibuat mempunyai dua level permainan, adanya beragam perintah yang dapat diberikan kepada unit, unit yang ada dapat menerima dan menjalankan perintah yang diberikan sesuai dengan kemampuan yang dimiliki, komputer dapat melakukan perhitungan untuk menggerakkan unit yang dimiliki, dan tampilan gambar yang digunakan dibuat dengan sudut pandang *isometric*.

Kata kunci: Kata kunci: 2D Game, Isometric, DirectX

ABSTRACT: *Two dimensional game development is easier than three dimensional one and the hardware requirement of two dimensional game is easier to meet, in addition the development of three dimensional game needs a bigger team. Based on those reasons, this research focuses on two dimensional isometric game to give three dimensional look without uses the real three dimensional system.*

The purpose of this application is making a simple two dimensional game with isometric view on Windows based computer. This application uses Microsoft Visual C++ and Microsoft DirectX 8.1 Software Development Kit and an additional library IsoHexDef and WavLoader.

The result shows that the application has two game levels and several units. Units are able to receive commands and act accordingly to extent its ability. The computer player can make calculations to move its own unit. The application uses isometric view.

Keywords: *2D Game, Isometric, DirectX*

1. PENDAHULUAN

Permainan komputer yang menggunakan tampilan tiga dimensi mempunyai tuntutan akan konfigurasi komputer yang lebih tinggi daripada permainan dengan tampilan dua dimensi. Meskipun demikian, tidak semua permainan yang menggunakan tampilan tiga dimensi menggunakan ke-

lahan dari tampilan yang dimiliki untuk *game play*. Selain itu pembuatan permainan tiga dimensi lebih sulit dibuat dari permainan dua dimensi. Biasanya pembuatan permainan tiga dimensi membutuhkan waktu dan tenaga yang lebih lama untuk membuat tampilannya daripada permainan dua dimensi. Karena itu pada penelitian ini permainan yang dibuat merupakan permainan dua dimensi yang menggunakan

sistem isometrik agar tampilan yang dihasilkan dapat lebih menarik dibandingkan jika menggunakan tampilan dua dimensi biasa.

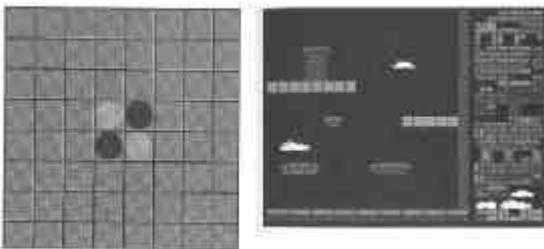
2. TEORI DASAR

2.1 DirectX

DirectX merupakan salah satu komponen Windows yang dibuat untuk membantu programmer dalam membuat program bagi Windows OS. DirectX merupakan sistem software yang menggabungkan video, audio, input, *networking*, *installation* dan lainnya. Dengan menggunakan DirectX seorang programmer dapat menggunakan kode yang sama untuk berbagai konfigurasi komputer.

DirectX memberikan kontrol yang besar bagi programmer yang menggunakannya hampir setara dengan kontrol terhadap *hardware* secara langsung. Hal ini dimungkinkan karena adanya teknologi COM (*Component Object Model*) serta kumpulan *driver* dan *library* yang ditulis oleh Microsoft dan pembuat *hardware*. Selain itu adanya kumpulan aturan *function*, *variable*, dan struktur data yang ditetapkan oleh Microsoft untuk digunakan oleh pembuat *hardware* bagi *driver* mereka.

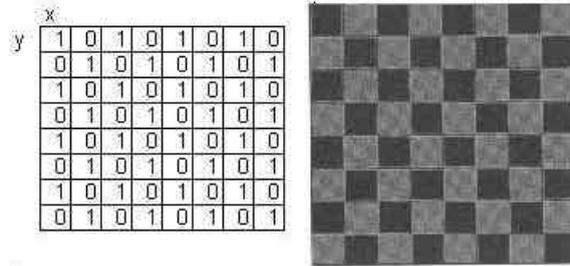
DIRECTX memiliki enam buah komponen yang dapat digunakan, yaitu : DirectX Graphics, DirectX Audio, DirectXInput, DirectXPlay, DirectXShow dan DirectXSetup.



Gambar 1. Permainan Menggunakan Sistem Tile

Hal ini mudah terlihat pada permainan yang menggunakan papan seperti catur, dimana papan yang ada terbagi dalam bujursangkar yang menandai area permainan. Tetapi permainan yang menggunakan *tile* tidak harus selalu tampak layar yang dimiliki tampak terbagi-bagi secara

rapi dengan jelas. Hal itu tergantung dari tipe permainan yang hendak dibuat.



Gambar 2. Tampilan Layar Menggunakan Dua Jenis Kotak

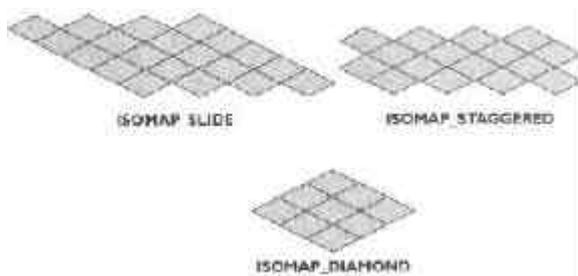
Gambar 2 menunjukkan bagaimana gambar dari sebuah layar permainan yang hanya terdiri dari dua buah tipe kotak yang berbeda disimpan. Gambar yang digunakan dalam permainan ini biasanya sudah disiapkan ke dalam bentuk *tile* untuk memudahkan pemrograman yang dilakukan yang disebut juga *tileset*.

Permainan yang tidak menggunakan sudut pandang *isometric* biasanya lebih mudah dibuat karena layar yang ada sudah berbentuk persegi. Koordinat dimulai dari kiri atas. Pada permainan tipe ini biasanya tidak ada persepsi kedalaman. Sehingga untuk menambahkan ilustrasi kedalaman digunakan *overlapping*, *scrolling*, dan *parallax scrolling*, yaitu sesuatu yang ada di belakang atau depan obyek yang dijadikan fokus dan dijalankan secara paralel dengan obyek tersebut. Obyek yang ada di depan dijalankan lebih cepat daripada background obyek yang ada di belakangnya.

Sudut pandang *isometric* biasanya dipakai untuk menyediakan representasi gambar tiga dimensi dengan memberikan ilustrasi kedalaman atau menyediakan sudut pandang yang lain pada permainan dengan menggunakan gambar bitmap. *Isometric* merupakan istilah yang biasanya digunakan untuk mereferensikan *orthographic view* dari sebuah bentuk, dimana semua sudut pandang 45° dari tiga sisi sebuah bentuk terlihat sama. Karena itu pada *isometric* tidak terdapat titik lenyap, gambar yang ada di depan tidak lebih besar daripada gambar yang ada dibelakang. Permainan yang mempunyai sudut pandang *isometric* biasanya disebut sebagai permainan *isometric*. Ada

beberapa cara untuk menggambar dua dimensi *tile* agar mempunyai *isometric* perspektif. Cara yang pertama dengan menggambar semua image mempunyai sudut pandang *isometric*. Cara yang kedua dengan membuat *tile* mempunyai bentuk diamond dan mensejajarkannya sehingga gambar yang terbentuk akan tertumpuk dengan rapi.

Sistem *tile* dengan *Isometric* biasanya digunakan pada permainan RPG (*Role Playing Game*) atau permainan strategi yang membutuhkan sistem ini agar permainan yang dibuat terlihat mempunyai unsur kedalaman, serta memudahkan dalam menggambar bangunan yang tinggi, juga memudahkan untuk membuat suatu *character* terlihat jauh.



Gambar 3. Tiga Macam Bentuk Peta Isometric

Bentuk dari layar yang ditampilkan oleh *isometric* sendiri juga terdiri dari beberapa macam bentuk yang umum dipakai *slide map*, *staggered map*, dan *diamond map*. Lihat Gambar 3. Tetapi yang paling sering dipakai *staggered map* digunakan untuk permainan *strategy* dan *diamond map* untuk permainan *RTS (Real Time Strategy)* dan *RPG (Role Playing Game)*. Pada penelitian ini tipe peta yang dipilih termasuk jenis *diamond map* karena dapat memberikan hasil tampilan yang bagus dan paling sering digunakan.

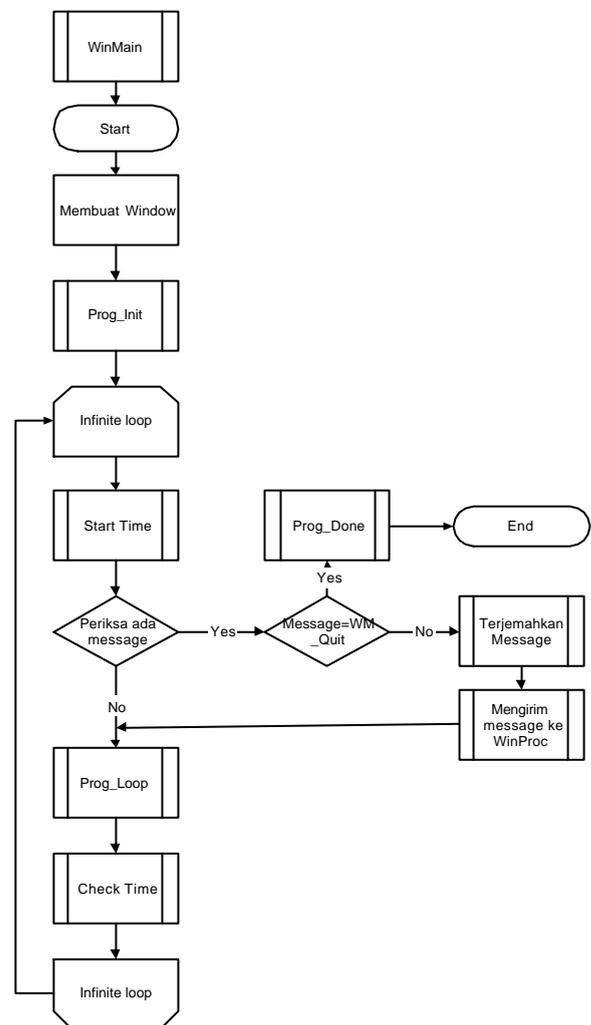
3. PERANCANGAN PROTOTYPE

3.1 Garis besar sistem WinMain

WinMain dalam sistem ini terdiri dari beberapa bagian yang utama yaitu membuat windows, inialisasi awal, pemeriksaan waktu, pemeriksaan *message*, program

utama, dan pembersihan program. Pemeriksaan waktu dilakukan diawal dan diakhir perulangan. Tujuannya membatasi kecepatan tampilan pada tampilan program aplikasi. Perulangan yang dilakukan pada WinMain yang dibuat merupakan perulangan yang tidak berakhir. Perulangan baru dihentikan bila diantara *message* yang ada di *message queue* terdapat *message WM_Quit*. Selain *message WM_Quit*, *message* lainnya yang diterima akan diterjemahkan dan diteruskan ke WindowProc.

Secara garis besar sistem kerja dari aplikasi ditunjukkan oleh Gambar 4.



Gambar 4. WinMain

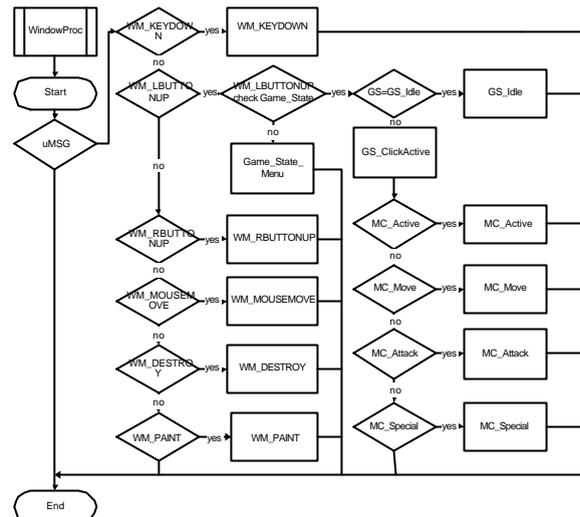
3.2 Perencanaan WindowProc

WindowProc dari sistem ini digunakan untuk memproses *message* yang dikirim ke Window. WindowProc yang digunakan da-

lam program aplikasi ini hanya memproses beberapa jenis *message* saja dalam menjalankan program aplikasi. *Message* lainnya yang tidak diproses oleh WindowProc akan diteruskan ke *default event handler* dari Windows. WindowProc yang digunakan untuk pemrosesan beberapa *message* dengan kegunaannya sebagai berikut:

- WM_KEYDOWN digunakan untuk memeriksa apakah ada penekanan tombol *esc*. Jika ada maka kondisi dari aplikasi akan berhenti atau keluar.
- WM_LBUTTONDOWN digunakan untuk memeriksa penekanan tombol kiri mouse yang secara umum digunakan untuk memberikan perintah.
- WM_RBUTTONDOWN digunakan untuk memeriksa penekanan tombol kanan mouse. Digunakan untuk mengembalikan kondisi aplikasi ke idle yang berarti tidak ada unit yang aktif dan tampilan menu perintah yang telah muncul akan dihilangkan.
- WM_MOUSEMOVE digunakan untuk memeriksa pergeseran mouse. yang digunakan untuk mengatur perpindahan layar (*scrolling*) dan kecepatannya berpindah. Selain itu WM_MOUSEMOVE juga digunakan untuk mengatur perubahan bentuk kursor.
- WM_DESTROY digunakan untuk menghentikan aplikasi yang sedang berjalan serta keluar dari program.

Didalam pemrosesan *message* WM_LBUTTONDOWN terdapat pengaturan perubahan beberapa kondisi yang terhubung dengan kondisi aplikasi dalam prosedur Prog_Loop yang digunakan untuk proses aplikasi. Kondisi yang terdapat di dalam WM_LBUTTONDOWN yang dijalankan disaat kondisi aplikasi sedang berjalan meliputi : GS_Idle, GS_ClickActive, MC_Active, MC_Move, MC_Attack dan MC_Special. Secara garis besar sistem kerja WindowProc ditunjukkan oleh Gambar 5.



Gambar 5. WindowProc

3.4 Perencanaan Sistem Perulangan Aplikasi

Setelah inisialisasi, Prog_Loop akan dijalankan dalam perulangan secara terus - menerus. Setiap kali Prog_Loop dijalankan, diperiksa kondisi aplikasi yang sedang terjadi. Kemudian dipilih tindakan berdasarkan kondisi itu. Ada tiga kondisi yang digunakan, kondisi aplikasi secara keseluruhan, kondisi aplikasi sewaktu sedang berjalan, dan kondisi untuk unit komputer. Kondisi aplikasi yang sedang berjalan ditunjukkan dengan GS dan akan dijalankan pada waktu kondisi aplikasi secara keseluruhan sedang menunjukkan Game_State_Running. Demikian juga pada kondisi untuk AI yang ditunjukkan dengan 'AI'. Kondisi ini dijalankan setelah kondisi aplikasi menunjukkan GS_AI, yang mana digunakan untuk menampung kondisi dari unit kompute. Saat kondisi ini berjalan maka kondisi untuk menjalankan unit komputer diaktifkan dan *user* tidak dapat memberikan perintah ke unit. GS_AI terbagi menjadi lima kondisi yaitu AI_Idle, AI_Think, AI_FindPath, AI_Move, dan AI_DoneMove.

Secara garis besar perencanaan sistem aplikasi ditunjukkan oleh Gambar 6 dibawah ini:



Gambar 9. Unit komputer Mengejar Unit Terdekat

Apabila unit *user* dan unit komputer saling berhadapan maka keduanya dapat saling melakukan penyerangan terhadap satu sama lain. Kemampuan untuk menyerang dari unit komputer serupa dengan kemampuan unit *user*, artinya batasan dan perhitungan yang dipakai untuk melakukan perhitungan bagi unit *user* juga dipakai bagi unit komputer. Selain itu unit dari komputer memiliki kelemahan yaitu tidak mempunyai kemampuan dan gerakan *special*.

Setiap kali ada unit yang melakukan penyerangan terhadap unit lainnya, maka di atas kepala unit yang menjadi target akan ditampilkan hasil dari penyerangan tersebut. Ada dua hasil yang mungkin yaitu serangan gagal mengenai target atau berhasil mengenai target. Apabila serangan berhasil maka hasil penyerangan akan menampilkan nilai kerusakan yang ditimbulkan. Jika serangan gagal maka hasil penyerangan akan menampilkan tulisan *MISS*.

Apabila semua unit dari unit komputer sudah dikalahkan maka penekanan tombol *end* akan membawa permainan beralih ke dunia baru. Lihat Gambar 10. Pada dunia baru peta yang digunakan akan berubah selain itu tenaga dan kemampuan *special* dari unit akan dipulihkan kembali ke nilai maksimal. Serta tingkat kesukaran akan bertambah ditandai dengan semakin banyaknya musuh dan adanya musuh yang memiliki pengalaman yang lebih tinggi dari yang biasa.



Gambar 10. Dunia baru

Dunia baru merupakan lanjutan dari tingkatan sebelumnya. Unit yang dikendalikan *user* akan tetap memiliki pengalaman yang diperoleh hasil dari mengalahkan musuh sebelumnya.

Pada aplikasi ini setelah mengalahkan tingkatan kedua dari permainan di dunia berikutnya maka permainan akan berakhir sebab tingkat kedua pada aplikasi ini merupakan tingkat terakhir.

5. KESIMPULAN

- Aplikasi yang dibuat sudah berhasil memiliki sudut pandang *isometric*.
- Permainan yang dibuat berhasil memiliki level permainan yang bertingkat, dengan menggunakan peta baru setelah berhasil mengalahkan level sebelumnya.
- Permainan yang dibuat masih mempunyai kekurangan dalam hal *game play*, yaitu keterbatasan pada ragam unit, tingkatan dunia, dan objek benda yang digunakan.
- Aplikasi yang dibuat belum memiliki untuk interaksi user dengan aplikasi, seperti *customize unit*, *save*, dan *load*.

DAFTAR PUSTAKA

- Andre LaMothe, *Trick of the Windows Game Programming Gurus, 2nd edition*, Sams Publishing, 2002.

2. Chapman, Davis. *Teach Yourself Visual C++ 6 in 21 Days*. Indianapolis: Sams Publishing, 1998.
3. Ernest Pazera, *Isometric Game Programming with DirectX 7.0*, Premier Press, 2001.
4. Game Development, <http://www.game-dev.net>
5. MSDN Developer Center, <http://msdn.microsoft.com/>
6. Sullivan, G.A. *Special Edition Using Visual Studio 6.0*. Indianapolis : Que Corporation, 1997