**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

1. **IDENTITAS PENGUSUL**

**NAMA : MOHAMMAD RIAN ARIF PRAKOSO**

**NRP : 5109100161**

**DOSEN WALI : ARYA YUDHI WIJAYA, S.Kom., M.Sc.**

1. **JUDUL TUGAS AKHIR**

**Perancangan dan Pembangunan Editor Level pada Pengembangan Permainan 3D Berteknologi XNA dalam Bentuk *Add-in* Visual Studio 2010.**

***Design and Construction of Level Editor for XNA 3D Game Development as Visual Studio 2010 Add-in.***

1. **URAIAN SINGKAT**

Industri pengembangan permainan mengalami perkembangan yang sangat pesat hingga sekarang dan memiliki pasar yang sangat luas, apalagi didukung pula oleh perkembangan berbagai perangkat yang dapat menjadi media permainan. Di bidang ini tentunya terjadi persaingan ketat antar pengembang satu sama lain. Mereka dituntut untuk meningkatkan produktivitas dalam mengembangkan permainan. *Game engine* memegang peran penting pada industri pengembangan permainan karena dapat menghasilkan permainan dengan kualitas yang baik dan cepat. Dengan *game engine*, melakukan pengembangan permainan menjadi sangat mudah karena memiliki fitur-fitur yang sangat bersahabat bagi pengguna, bahkan tanpa melakukan proses pemrograman sebuah permainan yang utuh dapat diselesaikan. Agar *game engine* bisa mendukung banyak *platform*, biasanya pembuat *game engine* mensyaratkan untuk melakukan pemutakhiran agar bisa mengembangkan permainan di *platform* yang berbeda, dan biasanya proses pemutakhiran ini membutuhkan biaya.

MonoGame, *library* yang memiliki struktur sama persis dengan XNA sudah mendukung pengembangan permainan di banyak platform. Karena merupakan *library* maka pengembangan permainan dengan *tools* ini membutuhkan proses pemrograman. Pengembangan permainan dengan pemrograman tentu saja berat dan memakan waktu sangat lama, apalagi permainan 3D.

Penulis menawarkan solusi berupa *tools* yang menjaga aset pengembangan permainan pada level kode sumber tetap ada. Solusi ini adalah sebuah *editor* level yang berupa *add-in* Visual Studio 2010, IDE yang digunakan untuk pengembangan permainan dengan *library* XNA. Dengan solusi ini, seluruh kode sumber akan tetap ada dan bisa dimanfaatkan ulang untuk melakukan pengembangan dengan *library* lain yang serupa semacam MonoGame.

1. **PENDAHULUAN**
   1. **LATAR BELAKANG**

Pengembangan permainan merupakan salah satu bidang serius dalam teknologi informasi. Sudah begitu banyak pengembang yang menggeluti bidang permainan. Para *developer* tersebut berlomba-lomba berkreasi mengembangkan permainan agar produk mereka diminati para penikmat permainan atau *gamer*. Sebuah permainan harus memiliki daya tarik tersendiri agar bisa diminati oleh *gamer*, bahkan membuat ketagihan atau adiktif. Agar menjadi permainan yang memiliki daya tarik, maka permainan tersebut haruslah menyenangkan, menantang, lucu, atau bahkan unik. *Genre* juga termasuk salah satu faktor daya tarik dari sebuah permainan.

Karakteristik dari sebuah permainan terdapat pada apa yang harus dilakukan oleh pemain. Hal ini biasa dikenal dengan istilah *gameplay*. Secara umum *gameplay* memiliki beberapa elemen yang antara lain adalah sebagai berikut.

* Pemain itu sendiri
* Peraturan
* Kondisi kemenangan

Permainan dibuat agar pemain merasa senang saat memainkannya. *Permainan* yang terlalu susah maupun terlalu mudah akan membuat pemain cepat merasa bosan. Oleh karena itu, perlu dilakukan perencanaan yang matang dalam mendesain permainan. Dimulai dari perancangan alur cerita, tingkat kesulitan dari tiap level, perancangan system, *gameplay*, dan sebagainya.

Seiring berkembangnya teknologi, banyak *tools* dibuat untuk membantu dalampembuatan permainan. *Tools* ini bisa berupa *library* maupun *game engine*. Secara mendasar perbedaan antara *library* dan *game engine* adalah *library* hanyalah kumpulan kelas dan fungsi-fungsi yang dibutuhkan dalam penulisan kode sumber permainan, sedangkan permainan *engine* biasanya berupa aplikasi yang memiliki fitur-fitur tertentu sehingga memudahkan proses pembuatan permainan, bahkan tanpa perlu menulis kode. Biasanya setiap *game engine* berbasis pada *library* tertentu. *Tools* ini dibuat disesuaikan dengan kebutuhan para pengembang permainan, termasuk diantaranya *genre*, teknologi yang digunakan, dan target *platform* dimana permainan tersebut akan di-*deploy*.

Salah satu teknologi permainan yang populer di kalangan pengembang permainan adalah XNA. XNA adalah *library* permainan yang dikembangkan oleh Microsoft. *Library* ini digemari oleh para pengembang permainan, khususnya programmer, karena kemudahan penggunaannya dan memiliki performa yang baik.

Selain XNA terdapat pula *library* permainan yang serupa, yaitu MonoGame. MonoGame dikembangkan oleh komunitas dan diimplementasikan secara *open source*. MonoGame dapat digunakan untuk membangun permainan dengan target platform yang bervariasi, antara lain iOS, Android, Windows (baik OpenGL maupun DirectX), Mac OS X, Linux, Windows 8 Store, Windows Phone 8, PlayStation Mobile, dan OUYA console. Fungsi-fungsi maupun kelas-kelas yang terdapat pada MonoGame sama persis dengan yang ada pada XNA 4.0. Hal ini disengaja agar para pengembang permainan XNA dapat dengan mudah mem-*porting­* permainannya ke platform di luar Microsoft.

Proyek XNA dan MonoGame sama-sama bisa dikerjakan dengan Microsoft Visual Studio 2010. Kebanyakan *developer* produk dengan teknologi Microsoft pasti familiar dengan *Integrated Development Environment* (IDE) ini, karena merupakan salah satu IDE yang hebat yang sangat memanjakan *developer*-nya dengan fitur-fiturnya.

Berdasarkan latar belakang di atas, penulis menawarkan solusi berupa *editor* level untuk permainan 3D yang menjadi bagian dari Visual Studio 2010 dalam bentuk *add-in*. *Add-in* ini dilengkapi dengan *tools* untuk menyunting level dalam permainan, seperti peletakan model 3D menjadi objek dalam level, pengaturan *terrain*, dan lain-lain. Setiap aksi yang dilakukan dengan menggunakan *add-in* ini akan men-*generate­* kode sumber dalam bahasa C# dan XNA yang akan menjadi *source* untuk dikompilasi menjadi permainan. Dengan begini, *developer* tetap bebas dan fleksibel untuk menyunting kode sumber hasil *generate*. Dan tentunya, kode sumber ini bisa di-*porting* ke teknologi MonoGame.

* 1. **RUMUSAN MASALAH**

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut.

* + - 1. Bagaimana membangun *add-in* pada Visual Studio 2010 dengan teknologi XNA.
      2. Bagaimana mengintegrasikan *tools* pada Visual Studio 2010 dengan *add-in*.
      3. Bagaimana melakukan peletakan dari model 3D menjadi objek yang ditampilkan oleh *editor* secara tepat berdasarkan sudut pandang *editor*.
      4. Bagaimana men-*generate* kode sumber dari aksi yang dilakukan pada *editor*.
      5. Bagaimana memvisualisasikan tingkat kesulitan level dilihat dari perilaku objek *non-playable character* (NPC) seperti jalur patroli dan jangkauan penglihatan.
      6. Bagaimana melakukan pengaturan warna, tekstur, dan tinggi-rendah *terrain*.
      7. Bagaimana memberikan efek fisika pada objek-objek dalam level.
  1. **BATASAN MASALAH**

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut.

1. *Add-in* dibangun dengan *library* XNA dan JigLib untuk efek fisika.
2. Model 3D yang dapat dijadikan aset berformat \*.fbx.
3. Kode sumber yang di-*generate* menggunakan bahasa pemrograman C#.
4. *Add-in* dibangun untuk dipasang pada Visual Studio 2010.
5. *Add-in* membutuhkan kartu grafis agar dapat berjalan.
   1. **TUJUAN DAN MANFAAT TUGAS AKHIR**

Tujuan dari Tugas Akhir ini adalah sebagai berikut.

1. Membuat *tools* untuk mempermudah pengembangan permainan dengan teknologi XNA dengan dengan prosedur yang sama dengan pengembangan permainan XNA *from scratch*.
2. Programmer tidak perlu beradaptasi dengan aplikasi selain Visual Studio 2010 untuk melakukan kustomisasi pada kode sumber.
3. Mudah melakukan *porting* kode sumber agar bisa dikompilasi ke target *platform* selain Microsoft dengan *library* yang mirip dengan XNA seperti MonoGame.
4. **TINJAUAN PUSTAKA**
   1. **Desain Permainan**

Membangun permainan merupakan sebuah tantangan tersendiri karena membutuhkan imajinasi dan kreatifitas untuk membuatnya menjadi hidup dan menghibur orang lain. Kesenangan adalah kunci untuk memperoleh ide dalam membuat permainan. Akan tetapi, kesenangan bagi setiap orang pasti berbeda-beda. Oleh karena itu, ada beberapa faktor yang perlu dipertimbangkan seperti opini orang-orang terhadap permainan yang bisa didapat dari situs-situs web tentang tinjauan permainan, genre yang akan dibuat, dan juga pendapat dari diri sendiri.

Setelah mendapat ide permainan apa yang akan dibuat, yang perlu dilakukan selanjutnya adalah mendesain sistem. Berikut ini adalah beberapa hal yang perlu diperhatikan untuk mendesain sistem.

1. *Gameplay*
2. Tantangan

Tantangan bisa berupa dalam salah satu atau kombinasi dari beberapa bentuk yang ada di bawah ini.

* 1. Tantangan waktu.
  2. Tantangan ketangkasan.
  3. Tantangan kesabaran.
  4. Tantangan ingatan/wawasan.
  5. Tantangan logika.
  6. Tantangan pengendalian sumber daya.

1. Perulangan dan alur

Apabila ide dan sistem sudah ditentukan, maka perlu dipikirkan aspek-aspek dari permainan. Menurut acuan dari [[1](#Fei05)] dan [[2](#Rog10)] aspek-aspek permainan tersebut antara lain adalah sebagai berikut.

* Karakter pemain.
* Lingkungan permainan, seperti *terrain* atau lahan, bangunan, pencahayaan, dan atmosfer
* Orang atau makhluk lain yang tidak dimainkan oleh pemain baik sebagai pihak pemain, netral, maupun musuh.

Pada akhir dari permainan, perlu adanya sejumlah penghargaan yang diberikan kepada pemain. Pemain harus mendapatkan kesan yang baik di penutupan [[1](#Fei05)]**.**

* 1. **.NET Framework 4.0**

.NET adalah bagian integral dari banyak aplikasi yang berjalan pada Windows dan menyediakan fungsionalitas umum terhadap aplikasi tersebut. .NET Framework adalah *framework* perangkat lunak yang jalan di atas sistem operasi Windows dan *platform* pengembangan aplikasi yang menyediakan layanan-layanan untuk pembangunan, *deploy*, dan menjalankan aplikasi desktop, web, aplikasi telepon, dan web service. .NET Framework terdiri dari *common language runtime* (CLR), yang menyediakan manajemen memori dan berbagai layanan sistem, dan *class library* luas yang menyediakan kode teruji, dapat digunakan ulang untuk semua area umum dalam pengembangan aplikasi [[3](#Mic131)].

.NET Framework memperkenalkan *common type system*, atau CTS. Spesifikasi CTS mendefinisikan semua kemungkinan tipe data dan konstruksi pemrograman yang didukung oleh CLR dan bagaimana mereka dapat atau tidak dapat berinteraksi satu sama lain sesuai dengan spesifikasi *common language infrastructure* (CLI). Dengan begini, .NET Framework mendukung pertukaran tipe dan instansiasi objek antara *library* dan aplikasi yang ditulis dengan bahasa-bahasa pemrograman yang didukung, seperti C#, VB.NET, dan F# [[4](#Wik131)].

* 1. **XNA**

XNA adalah seperangkat *tools* yang bekerja secara *managed runtime environment* yang disediakan oleh Microsoft untuk memfasilitasi pengembangan dan manajemen permainan. XNA berbasis .NET Framework, yang berjalan pada Windows, Windows Phone, dan XBox. Dalam banyak hal, XNA dikenal sebagai analogi .NET terhadap sistem pengembangan permainan yang lebih dikenal, DirectX, tetapi ditujukan kepada pengembang yang terutama tertarik untuk membuat permainan ringan yang berjalan pada banyak platform Microsoft. XNA merupakan *platform* basic untuk XNA Live Indie Games.

Perangkat *tools* XNA diumumkan pada 24 Maret 2004, di Konferensi Pengembangan permainan di San Jose, California. XNA Build *Community Technology Preview* yang pertama dirilis pada 14 Maret 2006. XNA Game Studio 2.0 dirilis pada Desember 2007, kemudian dilanjutkan XNA Game Studio 3.0 yang dirilis pada Oktober 2008. XNA Game Studio 4.0 dirilis pada 16 September 2010 yang disertai dengan Windows Phone Development Tools.

XNA Framework mengenkapsulasi detail teknologi kelas bawah dalam penulisan kode permainan, menjamin *framework* ini sendiri menjaga perbedaan *platform* dimana permainan di-*porting* antara satu *platform* dengan lainnya, dan karenanya bisa membuat *developer* untuk lebih fokus pada konten dan *gameplay*. XNA Framework mengintegrasikan beberapa *tools*, seperti Cross-platform Audio Creation Tool (XACT), untuk membantu pembuatan konten. XNA Framework menyediakan dukungan untuk pembuatan permainan 2D dan 3D [[5](#Wik13)].

* 1. **Visual Studio 2010**

Visual Studio 2010 adalah sebuah *integrated development environment* dari Microsoft. Visual Studio adalah kumpulan lengkap *tools* untuk membangun aplikasi web ASP.NET, web service XML, aplikasi desktop, dan aplikasi *mobile* [[6](#Mic132)].

Visual Studio mengandung *editor* kode dengan dukungan IntelliSense sebagai sebaik-baiknya refaktorisasi kode. *Debugger* terintegrasi bekerja pada *source-level* dan *machine-level*. Terdapat juga *tools* lain yang disertakan seperti desainer form, desainer web, desainer kelas, dan desainer skema basis data. Visual Studio juga menerima *plug-in* atau *add-in* yang bisa menambah fungsionalitas pada hampir tiap level­, termasuk menambah dukungan untuk sistem *source-control* (seperti Subversion) dan menambah *toolsets* baru seperti editor dan desainer visual untuk bahasa domain yang lebih spesifik atau *toolsets* untuk aspek pengembangan perangkat lunak lainnya (seperti klien Team Foundation Server: Team Explorer) [[7](#Wik132)].

1. **METODOLOGI**
   1. **Deskripsi Singkat**

*Add-in* yang akan dikembangkan adalah *tools* untuk melakukan *editing* level pada pengembangan permainan 3D pada Microsoft Visual Studio 2010. *Add-in* ini akan tampil sebagai jendela di dalam Visual Studio sebagai semacam *form designer* yang menampilkan visualisasi dari kode sumber pada proyek XNA dengan sudut pandang yang bisa diatur. Di samping itu juga terdapat *toolbox* atau menu-menu untuk membantu *developer* dalam mendesain levelpermainan. Dengan begitu, *developer* akan semakin mudah untuk mengembangkan permainan 3D dengan adanya fitur-fitur seperti peletakan aset model 3D berformat \*.fbx menjadi objek dalam level, melakukan pengaturan *terrain*, memberikan efek fisika ke objek yang ditentukan, dan memvisualisasi tingkat kesulitan permainan dengan adanya mode *view* untuk menampilkan perilaku NPC seperti jalur patroli dan jangkauan penglihatan.

Diagram aliran perancangan permainan XNA setelah *add-in* ini dipasang pada Visual Studio 2010 adalah seperti pada gambar.



Gambar 1. Diagram Alir Pembuatan Game XNA dengan *Add-in*

* 1. **Arsitektur**

Visual Studio 2010

Extensibility

Add-In

XNA

Gambar 2. Arsitektur

*Add-in* mengimplementasi *Interface* IDTExtensibility2 dari *namespace* Extensibility dari Visual Studio 2010 yang memperbolehkan *add-in* untuk mengakses *event* *add-in*, misal salah satunya adalah *event* OnConnection yang akan berjalan saat Visual Studio memuat *add-in*. Selain itu, *add-in* ini tentunya juga akan mengacu pada *library* XNA agar dapat memvisualisasikan model 3D pada jendela *add-in*.

* 1. **Diagram *Use Case* dan Penjelasan Fitur**

****

Gambar 3. Diagram *Use Case*

*Add-in* ini dapat dimanfaatkan baik oleh desainer maupun programmer. Fitur-fitur yang ada pada *add-in* ini adalah sebagai berikut.

1. Penyunting lahan

Dengan fitur ini terdapat kontrol untuk mengatur ketinggian titik atau area pada lahan level.

1. Meletakkan model 3D menjadi objek pada level.

Aset model 3D yang berformat \*.fbx dapat diletakkan ke level dengan cara *drag and drop* ke tampilan level menjadi objek dengan posisi di level sesuai sudut pandang tampilan.

1. Menambahkan komponen-komponen ke objek.

Setiap objek yang sudah diletakkan di level bisa memiliki satu atau lebih komponen. Macam dari komponen-komponen tersebut antara lain:

* 1. Efek fisika

Komponen ini digunakan agar objek target dapat terpengaruh efek fisika di lingkungan level seperti gravitasi dan tabrakan.

* 1. Rute patroli

Komponen ini digunakan untuk dapat mengatur titik-titik yang akan dilewati objek saat mode patroli. Pada komponen ini dapat diatur kecepatan bergerak objek antar titik. Titik-titik dan garis yang akan dilalui objek tersebut akan divisualisasi saat objek dipilih.

* 1. Jangkauan penglihatan dan aksi

Komponen ini digunakan untuk mengatur besarnya jarak jangkauan penglihatan dan jangkauan aksi objek. Ketika objek dipilih, komponen ini divisualisasikan dengan bentuk area lingkaran yang mengitari objek.

* 1. **Uji Coba dan Evaluasi**

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan data atau skenario yang telah dipersiapkan sebelumnya. Uji coba dan evaluasi perangkat dilakukan untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan. Uji coba dilakukan terhadap fungsional perangkat lunak. Setiap tahap pengembangan suatu fungsional langsung diadakan pengujian apakah fungsi tersebut sudah berjalan dengan baik. Apabila fungsi sudah berjalan baik, bisa dilakukan tahap pengembangan fungsional selanjutnya. Begitu seterusnya hingga semua fungsional perangkat sudah selesai. Dan terakhir dilakukan pengujian secara keseluruhan, yaitu pengujian dari fungsional pertama yang dimiliki perangkat hingga yang terakhir.

1. **JADWAL KEGIATAN**

Tabel 1. Jadwal Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Bulan (Tahun 2013) | | | | | | | | | | | | | | | | |
| Februari | | | | Maret | | | | April | | | | | Mei | | | |
| Analisa kebutuhan dan studi literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**

x

|  |  |
| --- | --- |
| [1] | John Feil and Marc Scattergood, *Beginning Game Level Design*. Boston: Thompson Course Technology, 2005. |
| [2] | Scott Rogers, *Level Up! The Guide to Great Video Game Design*. United Kingdom: John Wiley & Sons, Ltd., 2010. |
| [3] | Microsoft MSDN. (2013, Februari).NET Framework 4. [Online]. <http://msdn.microsoft.com/en-us/library/w0x726c2%28v=vs.100%29.aspx> |
| [4] | Wikipedia. (2013, Februari).NET Framework. [Online]. <http://en.wikipedia.org/wiki/.NET_Framework> |
| [5] | Wikipedia. (2013, Februari) Microsoft XNA. [Online]. <http://en.wikipedia.org/wiki/Microsoft_XNA> |
| [6] | Microsoft MSDN. (2013, Februari) Introducing Visual Studio. [Online]. <http://msdn.microsoft.com/en-us/library/fx6bk1f4%28v=vs.100%29.aspx> |
| [7] | Wikipedia. (2013, Februari) Microsoft Visual Studio. [Online]. <http://en.wikipedia.org/wiki/Microsoft_Visual_Studio> |

x

x