

Bab 1 Pendahuluan

1.1 Latar Belakang

Platformer merupakan genre game dimana karakter pemain akan melewati *platform* yang disediakan untuk mencapai *objective* dari game tersebut. Beberapa game yang memiliki genre *platformer* adalah *Super Mario Bros*, *Contra*, *Rayman*, *Megaman*, dan *Prince of Persia*. Judul tersebut merupakan beberapa dari sekian banyak judul game dengan genre *platformer* untuk target game *console*, dan masih banyak pula judul-judul *platformer* untuk *PC game*, *web-based game*, bahkan *mobile game*. Dari setiap judul game tersebut, terdapat kesamaan pada tiap komponen yang membangun game tersebut, antara lain: *platforms*, *obstacles*, *movement aids*, *collectible items*, dan *Triggers*[7].

Pada game studio, proses produksi game biasanya ditangani oleh 3 peran: *Game Designer*, *Artist*, dan *Engineer*[9]. *Game designer* bertugas untuk membuat *Game Design Document (GDD)* yang berisi *story line*, *level design*, karakter yang ada pada game, *objective*, hingga *experience* yang akan didapat oleh pemain. *Artist* bertugas pada pembuatan *game assets* termasuk audio dan visual pada game. Sedangkan *engineer* bertanggung jawab untuk merealisasikan rancangan game dengan menggabungkan ide pada GDD dan game asset hingga menjadi sebuah game yang interaktif.

Dengan adanya kesamaan karakteristik *gameplay* pada game *platformer* dan keterbatasan sumber daya yang dimiliki oleh suatu perusahaan, tim membutuhkan sebuah pendekatan khusus agar proses produksi, terutama pada game *platformer*, dapat berjalan sesuai target dengan memanfaatkan sumber daya yang ada pada proses produksi sebelumnya. Selain itu diperlukan pula kondisi dimana tiap anggota tim dapat fokus pada bidang keahlian masing-masing[5]. *Game designer* dapat melakukan *balancing* dan desain *level* – *Artist* dapat mengatur animasi atau mengubah *asset* – *Engineer* dapat fokus pada fitur game, sehingga proses produksi game dapat dikerjakan secara *concurrent*.

Penggunaan *game engine* merupakan salah satu solusi untuk memudahkan dalam pengembangan game pada. *Game engine* mampu menangani permasalahan untuk kondisi yang spesifik. Namun game engine yang ada memiliki keterbatasan[9]. *Game engine* hanya dapat digunakan oleh *engineer* karena untuk melakukan perubahan harus melalui proses *scripting*.

Sama halnya dengan aplikasi, game akan berkembang seiring bertambahnya kebutuhan game tersebut. Perubahan dapat berupa penambahan atau pengurangan konten yang nantinya akan melalui proses *testing*, *tweaking*, dan *balancing*. Pemanfaatan framework dan game engine membutuhkan proses *compiling* sebelum game dapat dijalankan. Hasil dari pengujian berupa feedback akan dievaluasi

sehingga terjadi perubahan pada program game. Game akan dicompile ulang untuk kemudian diuji kembali hingga didapat kondisi game yang siap dipasarkan. Selain itu dengan adanya perubahan pada game, penambahan dan pengurangan game object dan fitur game dilakukan dengan menambahkan class baru. Semakin banyak game object unik ditambahkan, semakin banyak pula class yang akan dibuat dan semakin sulit untuk dapat mengatur game object tersebut. Kondisi tersebut akan kita temukan pula pada pengembangan game platformer. Pengujian akan terus dilakukan hingga didapat kondisi level sesuai ketika pembuatan game scene, merancang level, menambah dan mengurangi object, dan mengubah fitur dari game platformer.

Solusi lain adalah dengan menggunakan *Data-driven programming* pada game framework atau game engine. *Data-driven programming* memisahkan antara *data* dan *logic* yang dibutuhkan oleh game[14]. *Data-driven programming* dapat meminimalkan proses kompilasi. Game dapat berubah dengan hanya memanipulasi data tanpa harus bersentuhan langsung dengan *source code*. Dengan demikian, tanpa harus memiliki kemampuan dasar pemrograman, setiap peran dalam proses produksi dapat bekerja secara *concurrent* untuk membuat game platformer yang diinginkan.

1.2 Rumusan Masalah

Dari penjelasan pada bagian latar belakang didapat rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan *data-driven programming* pada *game engine* untuk menangani pengembangan game platformer?
2. Berapa nilai *usability* yang didapat oleh *data-driven game engine* dalam menangani pengembangan game platformer?

1.3 Tujuan

Adapun tujuan dari tugas akhir ini adalah:

1. Mengimplementasikan *data-driven programming* kedalam *game engine* untuk kasus game *2d platformer*.
2. Mengetahui nilai *usability* data-driven game engine dalam menangani pengembangan game platformer.

1.4 Batasan Masalah

Adapun batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Studi kasus yang digunakan adalah *2d platformer game*.
2. Game hanya dapat dimainkan oleh satu pemain.

3. Implementasi *data-driven programming* menggunakan *XML* sebagai *database*.
4. Target platform dari game yang dibuat adalah *PC/Desktop*.

1.5 Metodologi Penelitian

1. Studi literature

Tahap awal dari penelitian dengan melakukan studi literatur dilakukan dengan mencari bahan mengenai game platformer 2d yang cocok, arsitektur Game Object Component (GOC) yang sesuai dengan game platformer 2d, dan juga mengenai desain Data driven programming yang cocok untuk menganalisis game.

2. Analisis kebutuhan

Dari studi literature akan didapat beberapa kebutuhan sistem, seperti fungsionalitas yang dibutuhkan oleh objek-objek game sehingga dapat didesain komponen. Selain itu mendefinisikan data-data yang berkaitan dengan game objek dan akan disimpan kedalam database. Diperlukan juga desain dari game yang sesuai untuk jenis game platformer 2d seperti desain level, desain entitas game dan behavior masing-masing entitas tersebut, serta desain game flow (urutan game scene yang akan digunakan).

3. Perancangan system

Dari analisa kebutuhan, tahap selanjutnya adalah merancang arsitektur dari game engine beserta penerapan data driven programming dan menentukan template xml untuk memudahkan dalam penyimpanan data.

4. Implementasi

Tahap implementasi dilakukan dengan cara menggabungkan semua metode menjadi satu sistem sehingga dapat mendukung desain game yang telah dirancang.

5. Simulasi, pengujian, dan pengumpulan data

Data driven game engine akan diuji berdasarkan skenario pengujian untuk mengetahui sejauh mana *data-driven game engine* dapat memenuhi kebutuhan dalam pengembangan game *2d-platformer*. Selain itu, pengujian *data driven game engine* dilakukan dengan mengambil data dari responden. *Game engine* akan digunakan oleh responden untuk membuat game *platformer* yang terdiri dari satu *level* dalam waktu 60 menit. Selama proses pembuatan game akan didapat data komponen game platformer yang dapat dipenuhi oleh responden. Selanjutnya kuisioner akan diajukan kepada responden untuk dapat dianalisis lebih lanjut.

6. Analisis data

Dari komponen game platformer dan data kuisisioner, selanjutnya akan dilakukan analisis untuk mengetahui seberapa besar pengaruh data driven programming pada game engine untuk menangani kasus game platformer.

1.6 Sistematika Penulisan

Sistematika penulisan pada tugas akhir ini adalah sebagai berikut:

Bab 1 Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

Bab 2 Landasan Teori

Bab ini berisi dasar teori yang berhubungan dengan tugas akhir ini yang secara umum terdiri atas *Game engine*, *Platformer game*, *component based software engineering*, dan *data driven programming*.

Bab 3 Desain dan Implementasi

Bab ini berisi desain studi kasus untuk game *2d-platformer*, desain dari *data driven game engine* untuk menangani game platformer, desain pengujian dari data driven game engine.

Bab 4 Pengujian dan Analisis

Bab ini berisi mengenai pengujian *data driven game engine* pada studi kasus game *2d-platformer* yang terbagi menjadi dua pengujian: perbandingan antara pengembangan game *2d-platformer* pada *non data-driven programming* dan *data-driven programming*.

Bab 5 Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran dari hasil pengujian pada bab sebelumnya berdasarkan pada rumusan masalah dan tujuan yang ada pada tugas akhir ini.