

PEMBUATAN GAME 3D PLATFORMER SIDESCROLLING “LEGEND”

Riyan Jaya Adi Pratama¹⁾ Jeanny Pragantha²⁾ Rendi Kristyadi³⁾

¹⁾ Mahasiswa Program Studi Teknik Informatika
Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta 11440 Indonesia
email: riyanjayaadipratama@gmail.com

²⁾ Dosen Program Studi Teknik Informatika
Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta 11440 Indonesia
email: jeannyp@fti.untar.ac.id

³⁾ Dosen Program Studi Teknik Informatika
Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta 11440 Indonesia
email : rendi88@gmail.com

ABSTRAK

Game “Legend” adalah game dengan genre Platformer berjenis Sidescrolling 3D. Game ini dirancang dengan menggunakan game engine Unity dengan C# sebagai bahasa pemrogramannya. Game ini memiliki tujuan memberikan permainan dan pengalaman bermain yang dapat menghibur para pemainnya. Pengujian dilakukan dengan metode blackbox testing, alpha testing oleh dosen pembimbing, dan beta testing dengan melalui survei pada 30 responden. Hasil pengujian menunjukkan bahwa Sebagian besar pemain merasa tertarik untuk bermain lagi karena beberapa alasan yaitu konsep game menarik, menantang, menghibur, unik dan gameplay yang menarik. Genre yang termasuk populer dikalangan responden, dan sebagian besar sudah tidak asing terhadap genre tersebut. Pada game ini namun masih memiliki kekurangan yaitu pada variasi musuh dan sistem platform yang hanya sebatas pada sistem operasi Windows maupun macOS.

Kata Kunci

C#, Game 3D, Legend, macOS, Platformer Game, Sidescrolling, Unity, Windows

1. Pendahuluan

Dewasa ini, seiring dengan berjalan dan berkembangnya zaman memainkan sebuah permainan atau game dapat dimainkan oleh siapapun kapanpun dan dimanapun. Dengan kemajuan teknologi masyarakat tidak perlu repot memikirkan tempat dan waktu bermain baik untuk sekedar *refreshing* ataupun *having fun*. Game pun dewasa ini diciptakan untuk berbagai keperluan dan tujuan yang berbeda tidak hanya sekedar sarana *entertainment* tetapi game dapat pula dijadikan sebagai media edukasi, promosi hingga ajang kompetisi seperti yang sering dilihat pada beberapa media online.

Game dapat dimainkan pada beberapa macam perangkat seperti Android, IOS, dan Personal Computer. Seiring berjalannya waktu banyak aplikasi yang mempermudah dalam pembuatan game, seperti Unity3D, HTML5 Game Engine, Game Maker, RPG Maker dan lain - lain. Aplikasi ini sering digunakan oleh orang yang ingin membuat game sendiri[1].

Berbagai macam genre dan tipe game telah dibuat baik oleh perusahaan besar maupun perseorangan, salah satunya yang paling populer adalah *platformer*. Game “Legend” ini sendiri terinspirasi oleh game “Medical Attack” buatan Franky Sutanto (NPM : 535112001) Mahasiswa Teknik Informatika, Fakultas Teknologi Informasi[2]. Dalam game *platformer* biasanya memiliki atau disisipkan cerita yang unik tergantung masing masing pembuatnya baik itu menyelamatkan seseorang, berpetualang hingga akhir, atau melawan boss penguasa diakhir setiap cerita atau *stage*[3].

2. Dasar Teori

Dalam melakukan perancangan suatu game, diperlukan dasar-dasar teori yang digunakan dalam pembuatan game agar dapat dihasilkan game yang berkualitas dan sesuai yang diinginkan. Beberapa dasar teori yang mendukung pembuatan dari game ini antara lain metode perancangan, genre game, Game engine Unity, Autodesk Maya 2018.

2.1 Metode Perancangan

Dalam melakukan perancangan sebuah game, diperlukan metode-metode tertentu yang digunakan dalam tahap perancangan game sebelum memasuki tahap pembuatan[4]. Rancangan game disebut sebagai dokumen desain game (*Game design document*)[5].

Adapun bagian-bagian dari dokumen desain *game* adalah:

1. High Concept

Secara umum *game* yang akan dibuat seperti apa gambaran umumnya semua dibuat pada saat pembuatan konsep permainan. Tujuan pembuatan bagian ini agar partisipan baru dalam proyek *game* dapat mengerti tentang apa yang akan dibuat. Pembuatan *High concept* bertujuan untuk memudahkan pengembang menjelaskan *game* yang dibuat. Di dalam *High Concept* pada suatu *game* berisi *storyboard* dari karakter yang akan dimainkan. *Storyboard* dapat berisi judul, *genre*, *Feature*, *Target Market* atau *Audience* dan Kontrol *game*[6].

2. Gameplay / Game Mechanics

Tujuan *Gameplay* sendiri adalah untuk menjelaskan apa yang akan dan seharusnya pemain lakukan didalam *game* tersebut.

a. Control Design

Dalam memainkan *game* diperlukan adanya kontrol yang digunakan untuk menggerakkan karakter pemain. Maka pengembang harus menentukan jenis kontrol yang digunakan untuk memainkan *game* tersebut.

b. Character Design

Dalam permainan *game* dibutuhkan karakter utama yang dikendalikan dalam sebuah *game*. Karakter desain terdiri dari karakter utama dan karakter tambahan di dalam *game*.

c. Object Design

Bagian ini menggambarkan berbagai karakter, barang, dan obyek yang dapat ditemui di dalam *game*. Setiap komponen dijelaskan interaksinya dengan pemain, dunia permainan, dan lawan.

d. Level Design

Pada bagian ini, desainer *game* memecah semua kejadian yang dirasakan oleh pemain dan bagaimana mereka berubah dan berkembang pada satuan waktu tertentu. Bagian ini dapat menjadi sebuah buku panduan untuk tim artistik dan desainer *game* tentang bagaimana suasana yang harus dibangun untuk *game*. Rancangan level design melibatkan tahapan dan misi yang harus diselesaikan oleh pemain.

e. Sound Design

Dalam permainan *game* sound design digunakan untuk memasukkan suara di dalam *game*. Rancangan sound design seperti sedang di dalam kota, di tengah hutan, di dalam goa dan lain-lain.

3. Storyline

Bagian ini menggambarkan secara umum cerita yang disediakan di dalam *game*. Bagian ini tidak perlu mencakup seluruh interaksi yang dialami pemain atau berbagai *subquest* yang tersedia di dalam *game*.

4. Rancangan Menu

Pada bagian ini dijelaskan secara detil berbagai opsi permainan yang disediakan dan tidak terlalu berpengaruh pada jalannya permainan.

5. Target Audience

Bagian ini menjelaskan untuk siapa *game* tersebut dibuat dan bagaimana ini akan menarik mereka.

6. Hardware Platform

Spesifikasi terendah yang diperlukan agar pemain dapat memainkan permainan tersebut tanpa terkendali oleh Perangkat keras.

7. Pembuatan Game

Seluruh konsep yang telah terbentuk dicoba dan di implemetasikan menjadi sebuah permainan. Pengumpulan atau pembuatan Aset yang dilakukan beserta *scripting* menjadi peranan paling penting dalam tahap ini.

8. Testing

Setelah permainan selesai dibuat maka perlu dilakukan beberapa tahap uji coba untuk melihat apakah hasil sesuai dengan konsep yang di ajukan di awal, dan untuk melihat apakah masih ada berbagai *bug* yang terdapat dalam *game* tersebut. Berikut beberapa tahap uji yang akan dilakukan:

a. Alpha Testing

Pengujian untuk mengetahui kelayakan suatu produk, dilakukan untuk mengidentifikasi semua permasalahan dan berbagai *bug* sebelum merilis kepada public. Tujuannya untuk meningkatkan kualitas permainan dan mempersiapkan untuk pengujian selanjutnya yaitu *Beta Testing*.

b. Beta Testing

Pengujian yang dapat dilakukan setelah melakukan pengujian Alpha, yaitu pengujian terakhir yang dilakukan sebelum *game* tersebut dirilis. *Feedback* dari konsumen merupakan keunggulan dari *beta testing*. Tujuannya adalah untuk meningkatkan kualitas permainan dengan mempertimbangkan umpan balik dari pemain dan memastikan kesiapan *game* tersebut kedepannya.

3. Genre Game

Game memiliki banyak jenis dan sering juga diklasifikasikan menggunakan *genre* dengan cara mengelompokkan *game* berdasarkan karakteristik yang mirip. Genre menjelaskan kearah mana kategori *game* tersebut.

4. Game Engine Unity

Unity merupakan sebuah *game engine* yang dibuat oleh Unity Technologies dan digunakan untuk membuat video *games* untuk PC, konsol, perangkat mobile, dan website.

Program Unity dapat digunakan untuk membuat *game* dengan mudah dan cepat, dan yang lebih

menguntungkan lagi bahwa *game* yang dibuat dapat dimainkan di berbagai platform. Hal yang paling menarik dari Unity adalah kemampuannya dalam mendukung berbagai perangkat teknologi.

5. Autodesk Maya 2018

Autodesk Maya adalah perangkat lunak aplikasi 3D grafis yang dapat dijalankan pada windows, macOS, dan Linux, di buat oleh Alias Systems Corporation dan sekarang dilanjutkan oleh Autodesk, Inc. Perangkat lunak ini berguna untuk merancang 3D interaktif baik itu untuk pembuatan aset video game, 3D film animasi, dan *visual effect*.

6. Finite State Machines

Finite State Machines adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State (Keadaan), Event (kejadian) dan action (aksi)[5].

7. Alur Aplikasi

Game “Legend” merupakan *game* dengan *genre platformer*, berbasis *3D side scrolling*. Konsep permainan *game* ini menekankan pada karakter pemain yang berusaha melewati berbagai rintangan dan melawan musuh pada setiap *stage*. *Game* ini dirancang dengan menggunakan Unity dan Bahasa pemrograman C#.

Untuk memainkan *game* ini pemain dapat memilih *Menu play* dan pemain akan diarahkan pada pemilihan *stage*. Pemain tidak dapat memilih *stage* selanjutnya sebelum dapat menyelesaikan *stage sebelumnya*. Dimasing-masing *stage* pemain mendapatkan 2 kesempatan berupa *lifepoints* dan *healthpoints* yang merupakan indicator pada darah pada pemain.

Game yang dirancang ini terdiri dari 5 *stage* dengan tingkat kesulitan dan boss yang berbeda-beda. Pemain dapat menemukan berbagai varian senjata didalam *game*. Jika pemain menemukan varian senjata yang sama maka peluru akan terisi full kembali dan jika pemain menghabiskan peluru tersebut maka senjata tersebut akan hilang .

8. Hasil Pengujian

Setelah selesai melalui tahap pembuatan, maka dilakukan tahap pengujian pada *game LEGEND*. Pengujian dilakukan agar dapat mengetahui *game* berjalan dengan baik dan kekurangannya.

3. Pengujian

3.1. Metode Pengujian

Pengujian *game* “Legend” dilakukan dengan menggunakan 3 metode pengujian, antara lain:

1. Blackbox testing

Pengujian *blackbox testing* dilakukan dengan menjalankan setiap modul yang terdapat dalam *game* “Legend” untuk mengetahui apakah terdapat *error* pada modul yang dijalankan.

2. Alpha Testing

Pengujian *alpha testing* dilakukan oleh orang yang mengerti pemograman. Pengujian *alpha testing game* “Legend” dilakukan oleh dosen pembimbing skripsi.

3. Beta Testing

Pengujian *beta testing* dilakukan oleh setiap orang yang ingin memainkan *game* “Legend” ini. Setelah selesai memainkan *game* ini, pemain akan mengisi kuisioner yang diberikan.

3.1.1. Blackbox Testing

Blackbox Testing dilakukan dengan menguji setiap fungsi dari *game* yang telah dibuat. Pengujian dilakukan dengan cara memeriksa setiap modul yang terdapat dalam *game* ini. Berikut modul-modul yang diujikan adalah sebagai berikut:

1. Pengujian Modul Main Menu

Pada Modul *Main menu* terdapat tombol *Campaign*, pemilihan *stage*, *Tutorial*, *Leaderboards*, *credits* dan *quit game*. Hasil pengujian yang dilakukan dalam modul ini sudah sesuai dengan rancangan, ketika *player* memulai permainan ataupun melanjutkan permainan, dan sesuai dengan rancangan yaitu pemain harus menyelesaikan *stage* sebelumnya sebelum dapat melanjutkan *stage* selanjutnya. Tampilan modul *main menu* dapat dilihat pada **Gambar 1** berikut ini.



Gambar 1 Modul Main Menu

[4.2. Pengujian Modul Gameplay](#)

Pada saat pengujian *modul gameplay*. Terdapat modul-modul yang terdiri dari beberapa bagian yang diujikan yaitu:

a. Pengujian Karakter Utama

Pada saat permainan dilakukan, pemain akan menggunakan karakter utama sebagai Krispy untuk menyelesaikan *game* tersebut. Tampilan permainan karakter utama dapat dilihat pada **Gambar 2** berikut ini.



Gambar 2 Pengujian karakter utama

b. Pengujian Karakter Interaksi Dengan Objek

Pada saat permainan dilakukan, karakter utama akan berinteraksi dengan setiap objek yang terdapat di dalam *game* tersebut. Tampilan permainan karakter berinteraksi dengan setiap objek dapat dilihat pada **Gambar 3** berikut ini.



Gambar 3 Pengujian interaksi karakter dengan Objek

c. Pengujian *Enemy*

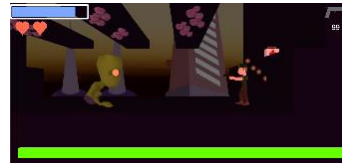
Pada saat permainan dilakukan, setiap *enemy* akan bergerak dan mempunyai serangan yang berbeda-beda yang terdapat dalam *game* tersebut. Berikut contoh tampilan permainan *enemy* dapat dilihat pada **Gambar 4** berikut ini.



Gambar 4 Pengujian enemy

d. Pengujian *Boss*

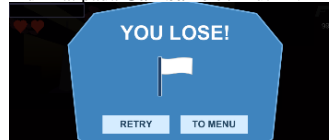
Pada saat permainan dilakukan, *Boss Dorkbutt* mempunyai pola pergerakan dan serangan yang unik didalam *game* Berikut contoh tampilan karakter *Boss Alien Dorkbutt* dapat dilihat pada **Gambar 5** dan berikut ini.



Gambar 5 Pengujian Boss

e. Pengujian *Player Mati*

Pada saat permainan dilakukan, *player* mempunyai 2 *life*, apabila *player* kehilangan nyawanya tetapi sudah mendapatkan area check point maka pemain akan *respawn* pada area checkpoint dan kehilangan kesempatan / nyawanya. Tampilan permainan *Player* mati dapat dilihat pada **Gambar 6** berikut ini.



Gambar 6 Pengujian Player Mati

f. Pengujian *Player Menang*

Permainan akan selesai ketika *player* berhasil menyelesaikan *stage* dan otomatis akan ada *feedback* tampilan *win game*. Tampilan *player* menang dapat dilihat pada **Gambar 7** berikut ini.



Gambar 7 Pengujian player Menang

2. Pengujian Modul *Pause Game*

Pada modul *pause game* terdapat di dalam modul *gameplay* yang berguna untuk pemain dapat menghentikan permainan secara langsung ketika menekan tombol *esc*. Tampilan permainan modul *pause game* dapat dilihat pada **Gambar 8** berikut ini.



Gambar 8 Pengujian modul pause

3. Pengujian Modul *Exit*
Pada modul *exit game* terdapat di dalam modul *main menu* yang berguna untuk user dapat keluar dari *game* ketika menekan tombol *quit game*.
4. Pengujian Modul About
Pada modul *about* akan menampilkan rancangan tentang penulis dan dari menu pengujian tersebut

3.1.2 Alpha Testing

Pengujian *Alpha testing* dilakukan oleh pihak yang mempunyai peran sebagai perwakilan dari calon pengguna yang akan memainkan *game* ini. Pengujian *Alpha testing* pada *game* "Legend" akan dilakukan oleh dosen pembimbing sebagai pihak yang mengerti konsep dan tujuan dari pembuatan *game* ini. pengujian dilakukan di Universitas Tarumanagara, Fakultas Teknologi Informasi.

3.1.3 Beta Testing

Pengujian *Beta testing* dilakukan kepada setiap orang setelah selesai melakukan *Alpha testing*. Pengujian ini dilakukan pada tanggal 18 hingga dan 28 Juni dengan memberikan kuisioner dan percobaan *game* "Legend" melalui alat komunikasi *email* dan *messenger* kepada berbagai responden yang berusia 10 tahun ke atas. Kuisioner berisi 13 pertanyaan yang berisi mengenai pengetahuan dari *genre game*, penilaian cerita dan *gameplay* dan pendapat dari *game* "Legend". Berikut adalah ringkasan dari hasil *beta testing*:

1. 25 responden (78.1%) menjawab mengetahui tentang *game* berjenis *platformer* dan 7 responden (21.9%) tidak mengetahui tentang *game* berjenis tersebut.
2. 25 responden (78.1%) menjawab tampilan utama *game* "Legend" mudah dimengerti, 6 responden (18.8%) menjawab Biasa saja dan 1 responden (3.1%) menjawab lainnya.
3. 27 responden (84.4%) tampilan pemilihan stage mudah dimengerti, 4 responden (12.5%) menjawab Biasa saja dan 1 responden (3.1%) menjawab lainnya.
4. 31 responden (96.9%) menjawab tingkat kesulitan stage sudah sesuai. 1 responden (3.1%) menjawab lainnya.
5. 28 responden (87.5%) tampilan permainan legend sangat mudah dimengerti, 3 responden (9.4%) menjawab biasa saja dan 1 responden (3.1%) menjawab lainnya.
6. 21 responden (65.6%) menjawab tingkat kesulitan dari *game* "Legend" merupakan sangat mudah, 6 responden (18.8%) menjawab biasa saja dan 5 responden menjawab cukup sulit.

7. 22 responden (68.8%) menjawab penggunaan control saat memainkan *game* tersebut sangat mudah dan 10 responden (31.3%) menjawab cukup mudah.
8. 31 responden (96.9%) menjawab akan memainkannya kembali dan 1 responden (3.1%) menjawab mungkin.

Bukti dokumentasi pengujian beta testing dapat dilihat pada **Gambar 9**, **Gambar 10** berikut ini



Gambar 9 Pengujian beta testing Syahrul



Gambar 10 Pengujian beta testing Santi

4. Kesimpulan dan Saran

Setelah selesai melakukan pengujian pada *game* "Legend", dari data dan komentar yang muncul dapat ditarik kesimpulan sebagai berikut:

1. *Game* "Legend" sudah cukup memberikan tantangan kepada pemain dalam menyelesaikan setiap *stage* yang ada.
2. "Legend" merupakan *game* bergenre *platformer* dengan *gameplay* yang tidak begitu asing sehingga dapat dengan mudah dimengerti oleh sebagian besar pemain.
3. *Game* "Legend" sudah cukup memberikan fitur yang bagus dan unik.
4. *Game* "Legend" dapat memberikan hiburan yang cukup sehingga pemain ingin memainkan *game* ini lagi.
5. *Keyboard* sebagai media *input* dirasa kurang cocok dan tergolong susah untuk dimainkan
6. *Graphic User Interface* atau GUI pada *game* "Legend" cukup mudah dimengerti dan dipahami

Selain kesimpulan yang diambil dari data dan komentar yang muncul saat pengujian, terdapat saran-saran yang dapat mengembangkan *game* ini. Berikut ini adalah saran-saran yang disampaikan:

1. Cerita dalam *game* "Legend" dibuat menjadi lebih panjang.

2. Penambahan *stage* supaya dapat membuat *game* "*Legend*" menjadi lebih menarik.
3. Ditambahkan amunisi pada senjata tambahan tetapi damage dikurangi agar tantangan pemain dapat ditingkatkan.
4. Ditambahkan *enemy* yang bervariasi supaya dapat membuat *game* menjadi lebih menarik.
5. Di *Build* pada perangkat mobile sehingga mobilitas pada saat bermain lebih mudah, baik ke Android maupun kedalam iOS.

REFERENSI

- [1] Bagus Aji Santoso, [Lima Game Engine Pilihan Untuk Membuat Game Android](https://www.codepolitan.com/lima-game-engine-pilihan-untuk-membuat-game-android), <https://www.codepolitan.com/lima-game-engine-pilihan-untuk-membuat-game-android>, 22 Februari 2017
- [2] Franky, Sutanto. "Pembuatan Game Platformer "Medical Attack" Menggunakan Unity." 2016. (Skripsi tidak di publikasikan)
- [3] Wikipedia, [Platform Game](https://en.wikipedia.org/wiki/Platform_game), https://en.wikipedia.org/wiki/Platform_game diakses 04 February 2018
- [4] Bob Bates, *Game Design* Second Edition. (Boston : Thomson Course Technology PTR, 2004), h. 203.
- [5] Richard Rouse, *Game Design: Theory & Practice* Second Edition. (Plano : Wordware Publishing Inc, 2005), h. 361.
- [6] Christer Ericson, *Real Time Collision Detection*, (Boca Raton: CRC Press, 2004), h. 13.

Riyan Jaya Adi Pratama, mahasiswa tingkat akhir Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.

Jeanny Pragantha, memperoleh gelar Ir. dari Institut Teknologi Bandung pada tahun 1986. Kemudian memperoleh gelar M.Eng. dari Asian Institute of Technology, Bangkok pada tahun 1989. Saat ini aktif sebagai dosen Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.

Rendi Kristyadi, memperoleh gelar S.Kom dari Universitas Bina Nusantara pada 2010, melanjutkan S2 di Universitas Bina Nusantara dan memperoleh gelar Magister Manajemen Sistem Informasi pada tahun 2012. Beliau aktif dan berkecimpung dalam industri *Game Development* di Indonesia.