

RANCANG BANGUN GAME PLATFORMER BINTANG KECIL MENGUNAKAN GODOT ENGINE

Ricky¹
Mesri Silalahi²

¹Mahasiswa Program Studi Sistem Informasi, Universitas Putera Batam

²Dosen Program Studi Sistem Informasi, Universitas Putera Batam

email: pb171510003@upbatam.ac.id

ABSTRACT

Entertainment industry comes in many shapes and form. One example of such things are digital video games. Despite the controversies and bad news around it, the industry shows no progress of slowing down. Many people show interest of developing video games, yet the opportunity seems abysmal, at least in this nation. This research paper intended to draw more people to the positive of video game, and possibly encourage them to try develop their own video games. Content of this paper include the basic process of developing a 2D platformer games using Godot Engine. Assets are homemade, meaning the drawing are self-made, with the exception of fonts.

Keywords: Godot Engine; Platformer; Video Game.

PENDAHULUAN

Akses pada perancangan dan pengembangan perangkat lunak sudah terbilang terjangkau, dengan adanya aplikasi seperti Visual Studio Code, Netbeans, dan aplikasi *Integrated Development Enviroment* lainnya. IDE tersebut umumnya digunakan untuk pengembangan perangkat lunak komersil. Contoh perangkat lunak komersil adalah sistem informasi, aplikasi *point of sale*, aplikasi pemesanan tiket, aplikasi pembayaran dalam jaringan, dan aplikasi umum lainnya. Aplikasi IDE tersebut sudah biasa diajarkan sebagai materi pada pembelajaran TIK di sekolah atau perkuliahan jurusan teknologi komputer.

Pembangunan *video game* juga termasuk perancangan dan pengembangan

perangkat lunak. Berbeda dengan pengembangan perangkat lunak yang biasa, pengembangan *video game* biasanya menggunakan *game engine* sebagai pengganti aplikasi IDE. *Game engine* yang sering ditemukan pada penelitian akademik yaitu Unity engine, Unreal Engine, RPG Maker, dan *game engine* lainnya. Salah satu *game engine* yang mulai dikenal kalangan pengembang *video game*, yaitu Godot Engine, akan menjadi *game engine* yang dibahas pada jurnal ini.

Selain kualitas, jenis/*genre* dari *video game* juga dapat memengaruhi ketertarikan pemain. Sejumlah jenis akan lebih populer dibandingkan jenis lainnya. Jenis/*genre platformer* akan menjadi jenis *video game* yang dirancang pada jurnal ini. Jenis/*genre* ini kalah bersaing dengan

genre First Person Shooter (Watson et al., 2015).

KAJIAN TEORI

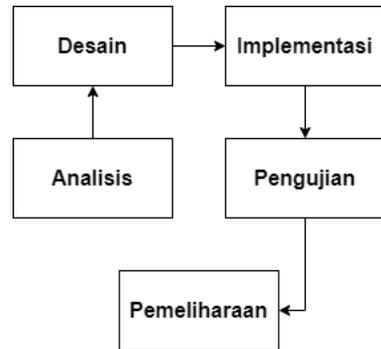
Sebagian besar teori mengenai sistem informasi dapat mendukung penelitian mengenai pengembangan video game.

2.1 Perancangan dan Pembangunan Perangkat Lunak

Untuk menjaga terarahnya proses pengembangan, diperlukan tahap perancangan. Fungsi lain penerapan tahap perancangan adalah membatasi ruang lingkup proyek (Hasyim et al., 2014). Tahap perancangan juga dapat digunakan untuk mengkomunikasikan fitur dan spesifikasi kepada rekan pengembang agar seluruh tim pengembang memiliki satu pemahaman tentang aplikasi yang sedang dibangun.

2.2 Metode *Waterfall*

Dinilai sebagai metode pengembangan perangkat lunak yang tertua, *Waterfall* menjadi salah satu metode yang paling populer di kalangan pengembangan perangkat lunak. Winston W. Royce disebut sebagai penemu metode pengembangan tersebut (Ind, Karambir, 2015).



Gambar 1. Model Pengembangan Waterfall
(Sumber: Peneliti, 2021)

2.3 Permainan Video/ *Video Game*

Video Game menggabungkan teks, video, suara, animasi, gambar, dan unsur multimedia lainnya dengan kemampuan interaktif yang dapat dijalankan pada perangkat elektronik (Adiwikarta & Dirgantara, 2017). Beriringan dengan perkembangan zaman, kemampuan perangkat elektronik menampilkan objek tiga dimensi juga mendorong munculnya *video game* tiga dimensi. Meskipun demikian, *video game* dua dimensi masih diminati banyak pemain. *Video game* itu sendiri dapat dikategorikan berdasarkan jenis/*genre*. Beberapa contoh *genre* yang sedang menguasai pasar *video game* yaitu *MOBA*, dan *Battle Royale*.

2.4 *Game Engine*

Mirip seperti Integrated Development Environment untuk pengembangan aplikasi umum, *game engine* dapat digunakan untuk pengembangan *video game*. Pengembang juga dapat menulis dan menggunakan *game engine* buatan sendiri. Layaknya objek tiga dimensi yang membutuhkan aplikasi pengolah objek tiga dimensi, *video game* juga memerlukan aplikasi khusus berupa

game engine (Fachroni et al., 2018). Contoh umum dari *game engine* adalah, Unreal Engine, Unity Engine, Cry Engine, dan *game engine* lainnya. Godot Engine dipilih karena penggunaannya tidak terikat biaya royalti atau lisensi yang berbayar.

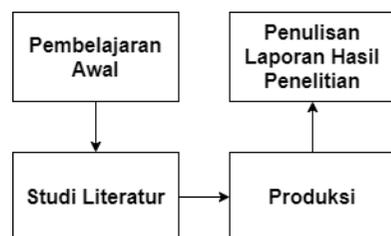
2.5 Aplikasi Pengolah Gambar : Krita

Pembuatan aset berupa gambar dan tampilan antarmuka memerlukan aplikasi pengolah gambar. Sebagian besar aspek yang diperhatikan ketika merancang sebuah *video game* adalah keberhasilan pemenuhan kebutuhan indera penglihatan. Terdapat indikator ketika pemain berhasil menyelesaikan sesuatu, misalnya ketika karakter mengoleksi objek, maka dapat diberikan animasi melompat ceria pada karakter sebagai indikator bahwa objek berhasil diambil. Aplikasi pengolah gambar yaitu Krita merupakan aplikasi gratis dan *open-source* sehingga calon pengembang *video game* dapat menggunakan aplikasi ini sebagai alternatif aplikasi berbayar lainnya.

METODE PENELITIAN

3.1 Desain Penelitian

Diperlukan desain penelitian untuk mendukung penyelesaian penelitian.



Gambar 2. Alur Proses Penelitian
(Sumber : Peneliti, 2021)

Pada tahapan Pembelajaran Awal, dikumpulkan informasi mengenai *game engine* yang populer di komunitas pengembang *video game*. Informasi mengenai *genre video game* yang sedang kalah bersaing juga diteliti. Pada tahapan studi literatur, dipelajari proses perancangan dan pengembangan *video game* dan perbedaannya dengan perancangan aplikasi umum. Pada tahapan produksi, dimulai proses perancangan dan pengembangan *video game*. *Genre* yang dipilih adalah *2D platformer*. Pada tahapan penulisan laporan hasil penelitian, dituliskan hasil perancangan dan proses pengembangan, dimulai dari penulisan konsep cerita, desain karakter, desain *level*, desain tampilan antar muka, sampai dengan penarikan simpulan dan saran.

3.2 Metode Pengembangan *Video Game* : *Waterfall*

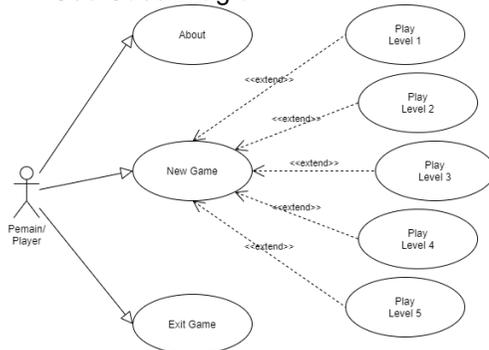
Meskipun terdapat proses pengembangan *video game* seperti *GDLC* (*Game Development Life Cycle*), tidak menutup kemungkinan untuk menggunakan metode pengembangan perangkat lunak biasa untuk pengembangan *video game*. Metode *waterfall* dinilai sebagai metode yang sesuai untuk pemula dalam pengembangan perangkat lunak. Untuk menutup kelemahan dari metode *waterfall*, modifikasi terhadap metode tersebut akan dilakukan. Pada tahapan analisis tidak akan melibatkan calon pengguna, karena tidak diperlukan pemenuhan kebutuhan calon pengguna. Akan tetapi, tidak menutup kemungkinan untuk melibatkan pihak yang dapat membantu pengujian *video game* sebagai pengganti calon pemain. Pihak tersebut dikenal sebagai *beta tester*, yang bertugas untuk menguji

purwarupa *video game* berulang kali dan bertugas menemukan kesalahan pada *video game* sehingga dapat dilaporkan kepada pihak pengembang (Hernández Bécares et al., 2017).

HASIL DAN PEMBAHASAN

Perancangan menggunakan *Unified Modelling Language* untuk memudahkan komunikasi ide dan konsep kepada pembaca. Empat diagram untuk merepresentasikan seluruh diagram adalah *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*. *Video game* diberi judul atau nama "Bintang Kecil".

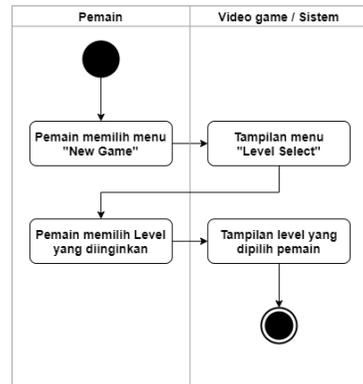
4.1 Use Case Diagram



Gambar 3. Diagram Use Case (Sumber : Peneliti, 2021)

Pada diagram *use case*, dijelaskan tombol yang dapat diakses pada menu utama. Terdapat pilihan untuk *New Game*, *About*, dan *Exit Game*.

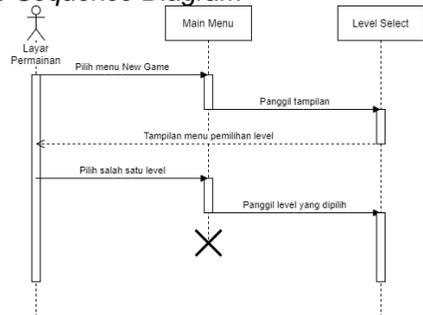
4.2 Activity Diagram



Gambar 4. Salah satu diagram Activity (Sumber : Peneliti, 2021)

Activity Diagram ditulis sebanyak jumlah *use case*, dengan pengecualian untuk *use case play level 1-5*, karena aktivitas yang dilakukan sama. Perbedaan hanya terletak pada desain *level* dan lokasi bintang.

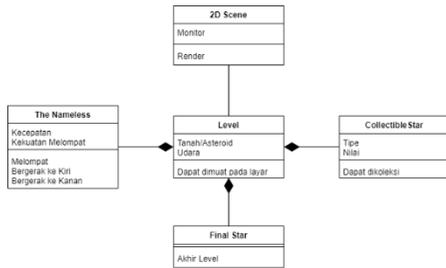
4.3 Sequence Diagram



Gambar 5. Salah satu diagram sequence (Sumber : Peneliti, 2021)

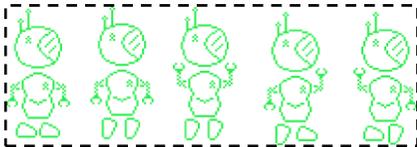
Sequence diagram dibuat sebanyak jumlah *use case*, kecuali untuk *use case play level 1-5*, dengan alasan yang sama dengan sebelumnya.

4.4 Class Diagram



Gambar 6. Class Diagram
(Sumber : Peneliti, 2021)

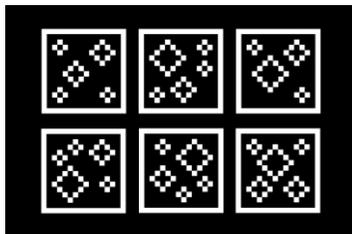
4.5 Desain Karakter



Gambar 7. Karakter Utama
(Sumber : Peneliti, 2021)

Karakter dirancang dan digambar dengan mempertimbangkan tema, yaitu tata surya yang berada pada ambang kematian. Karakter diberi pakaian astronot dan memiliki kemampuan berjalan ke kiri, kanan, dan melompat tinggi.

4.6 Desain Blok Tempat Berdiri



Gambar 7. Batu Asteroid
(Sumber : Peneliti, 2021)

Desain blok batu asteroid digunakan sebagai tempat agar karakter utama dapat

berdiri dan melompat. Desain mengikuti tema cerita, yaitu gaya gravitasi pada tata surya yang sudah tidak stabil sehingga batu asteroid berserakan pada luar angkasa.

4.7 Desain Bintang



Gambar 8. Bintang yang dapat dikoleksi
(Sumber : Peneliti, 2021)

Berdasarkan cerita dari *video game*, tugas karakter utama adalah mengoleksi bintang. Semakin banyak jumlah bintang yang dikumpulkan, maka semakin tinggi nilai yang didapatkan.

4.8 Desain Menu Utama

Pada menu utama akan disediakan tombol *New Game*, *About*, dan *Exit Game*. Fungsi tombol *New Game* adalah untuk memulai permainan. Fungsi pada tombol *About* adalah untuk menampilkan layar informasi mengenai aplikasi dan pihak yang terlibat dengan pengembangan *video game*.



Gambar 9. Desain Menu Utama
(Sumber : Peneliti, 2021)

SIMPULAN

Setelah melakukan pengembangan *video game* Bintang Kecil, dapat ditarik simpulan sebagai berikut:

1. Godot Engine dapat dijadikan sebagai *game engine* alternatif dari Unreal Engine atau Unity Engine. Meskipun kualitas *video game* bukan tergantung dari *game engine* yang digunakan, terdapat keuntungan khusus jika menggunakan Godot Engine, yaitu jika pengembang berencana menjual *video game*, maka tidak perlu mengeluarkan biaya untuk royalti kepada pemilik Godot Engine.
2. Aplikasi pendukung seperti aplikasi pengolah gambar dapat menggunakan yang gratis dan *open-source*. Pengembang pemula memiliki pilihan selain membeli aplikasi pengolah gambar komersil atau berbayar.

DAFTAR PUSTAKA

- Adiwikarta, R., & Dirgantara, H. B. (2017). Pengembangan Permainan Video Endless Running Berbasis Android Menggunakan Framework Game Development Life Cycle. *Kalbiscientia Jurnal Sains Dan Teknologi*.
- Fachroni, M. Y., Wibowo, H., & Syaifuddin, S. (2018). Perancangan Game Physical Puzzle Rolling Kingdom. *Journal of Animation and Games Studies*.
<https://doi.org/10.24821/jags.v4i1.1>

873

- Hasyim, N., Hidayah, N. A., & Latisuro, S. W. (2014). Rancang Bangun Sistem Informasi Koperasi Berbasis Web Pada Koperasi Warga Baru MTS N 17 Jakarta. *Jurnal Sistem Informasi*.
- Hernández Bécares, J., Costero Valero, L., & Gómez Martín, P. P. (2017). An approach to automated videogame beta testing. *Entertainment Computing*.
<https://doi.org/10.1016/j.entcom.2016.08.002>
- ind, Karambir, S. T. (2015). A Simulation Model for the Spiral Software Development Life Cycle. *International Journal of Innovative Research in Computer and Communication Engineering*.
<https://doi.org/10.15680/ijrcce.2015.0305013>
- Watson, B., Wang, Y., & Zheng, J. (2015). Popularity and player satisfaction of video games: A case study of Xbox 360. *Conference Proceedings - 2014 IEEE Games, Media, Entertainment Conference, IEEE GEM 2014*.
<https://doi.org/10.1109/GEM.2014.7118435>



Penulis pertama, Ricky, merupakan mahasiswa Prodi Sistem Informasi Universitas Putera Batam.



Penulis kedua, Mesri Silalahi, S.Kom., M.SI., merupakan Dosen Prodi Sistem Informasi Universitas Putera Batam.