# BAB II

**LANDASAN TEORI**

1. Game Edukasi

*Game* adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin. Perkembangan video game yang semakin hari semakin pesat menimbulkan sebuah ide di bidang perindustrian untuk menjadikan vidieo game sebagai media edukasi yang efektif dan interaktif. Game edukasi yang semakin popular di kalangan pendidikan mengakibatkan para pendidik berpikir bahwa di zaman serba digital ini dalam memberikan pendidikan kepada peserta didik dapat menggunakan komponen rancangan game dan menerapkannya pada kurikulum pembelajaran dengan menggunakan industry berbasis game edukasi. Menurut Amanda & Rianto (2018) dalam penelitiannya dijelaskan bahwa game edukasi merupakan sebuah media permainan yang dirancang sedemikian rupa secara khusus untuk memberikan pembelajaran tertentu kepada peserta didik, mengembangkan konsep dan pemahaman peserta didik dan membimbing mereka dalam melatih kemampuan, serta memberikan motivasi kepada peserta didik untuk memainkan game edukasi. Tujuan pembuatan game edukasi adalah sebagai alat untuk membantu pendidikan, seperti halnya beljaar mengenal warna, belajar mengenal huruf dan angka, belajar berhitung, danbahkan sampai belajar bahasa asing (Sri, 2021).

1. **Android**

Android adalah sebuah sistem operasi yang dirancang oleh *Google* yang berbasiskan kernelLinux dan berbagai perangkat lunak *open source* lainnya serta biasa digunakan untuk perangkat dengan layar sentuh seperti smartphone dan tablet.

Walau Android adalah milik *Google, Google* tidak memungut biaya bagi siapapun yang ingin menggunakan Android karena Android adalah perangkat lunak open source. Hal ini sangat bertolak belakang dengan sistem operasi Windows 10 *Mobile* milik Microsoft, dimana perusahaan harus membayar royalti jika ingin menggunakan sistem operasi tersebut. Sedangkan iOS milik *Apple* hanya bisa digunakan di perangkat *iPhone* dan *iPad* saja, dan *Apple* tidak merilisnya untuk perangkat lain.

1. **Unity**

Unity 3D adalah sebuah game engine yang berbasis cross-platform. Unity dapat digunakan untuk membuat sebuah game yang bisa digunakan pada perangkat komputer, ponsel pintar android, iPhone, PS3, dan bahkan X-BOX. Unity adalah sebuah sebuah tool yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. Unity bisa untuk games PC dan games Online. Untuk games Online diperlukan sebuah plugin, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser. Unity tidak dirancang untuk proses desain atau modelling, dikarenakan unity bukan tool untuk mendesain. Jika ingin mendesain, pergunakan 3D editor lain seperti 3dsmax atau Blender. Banyak hal yang bisa dilakukan dengan unity, ada fitur audio reverb zone, particle effect, dan Sky Box untuk menambahkan langit. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. Flexible and EasyMoving, rotating, dan scaling objects hanya perlu sebaris kode. Begitu juga dengan Duplicating, removing, dan changing properties. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker. Berbasis .NET. Artinya penjalanan program dilakukan dengan Open Source .NET platform, Mono. (Bagus & Mahendra, 2016)

1. Flowchart

Flowchart atau bagan alir adalah bagan (*chart*) yang menunjukan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari Langkah-langkah dan urut-urutan prosedur dari suatu program. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Farid, 2021.).

2.4.1. Simbol *Flowchart*

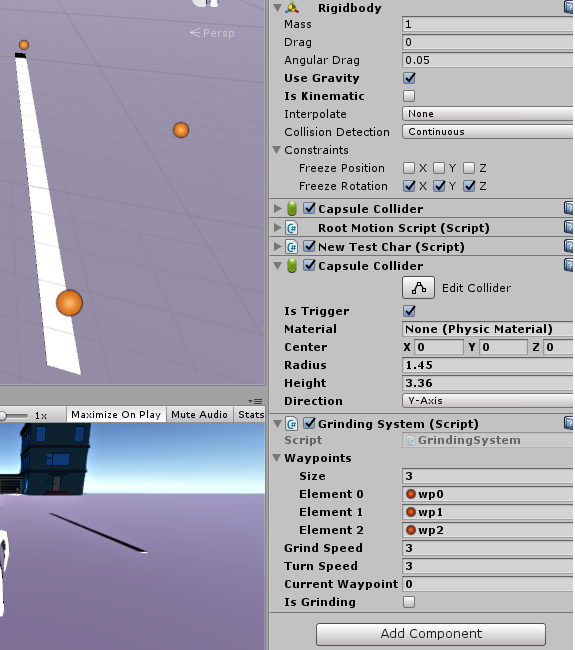
**Tabel 2.4.1 Simbol yang digunakan merancang suatu *Flowchart* sebagai berikut :**

|  |  |
| --- | --- |
| Komponen | Keterangan |
|  | *Terminal*, sebagai awal (*start*) dan akhir (*end*) suatu bagan alir. |
|  | *Proccess*, sebagai himpunan penugasan yang akan dilaksanakan secara berurutan. |
|  | *Input*/*Output*, sebagai memasukkan data atau menampilkan data |
|  | *Decission*, sebagai cabang untuk memutuskan arah percabangan sesuai kondisi yang dipenuhi |
|  | *Predefined* *Process*, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain |
|  | *Connector*, sebagai penghubung apabila bagan alir terputus, contoh pergantian halaman. |
|  | *Flowline*, sebagai penunjuk arah proses |

1. Blender 3D

Blender 3d Blender adalah rangkaian kreasi 3D yang gratis dan open source. Blender mendukung konsep 3D secara keseluruhan—modeling, rigging, animasi, simulasi, rendering, compositing, dan motion tracking, bahkan video editing dan pembuatan game. Pengguna yang telah mahir menggunakan API milik Blender yang ditujukan pada scripting python untuk menyesuaikan aplikasi ini dan menulis tools yang telah dikhususkan; biasanya semua ini dimuat pada fitur Blender versi selanjutnya. Blender sangat cocok bagi studio kecil dan perorangan yang mendapatkan untung dari konsep pemersatuannya dan proses pengembangan yang responsif. (Meilin Mongilala dkk., t.t.)

1. Rigidbody



**Gambar 2.6. Tampilan UI *Rigidbody***

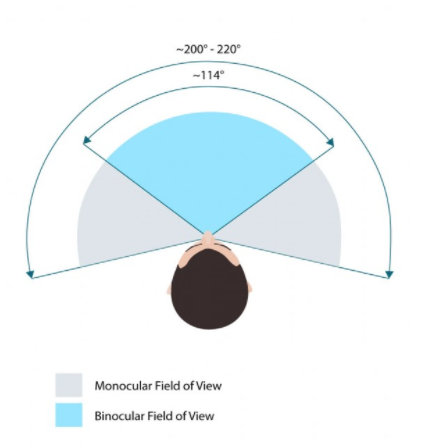
(Sumber :<https://docs.unity3d.com/550/Documentation/Manual/classRigidbody.html>)

*Rigidbody* memungkinkan *GameObjects* untuk bertindak di bawah kendali fisika. *Rigidbody* dapat menerima kekuatan dan torsi untuk membuat objek pengguna bergerak secara realistis. *GameObject* apa pun harus berisi benda padat yang akan dipengaruhi oleh gravitasi, dan bergerak di bawah kekuatan tambahan melalui skrip, atau berinteraksi dengan objek lain.

Jika collider digunakan sebagai pemicu, misalkan objek sudah mencapai area tertentu dan perlu memunculkan cutscene, maka dapat menggunakan fungsi “Is Trigger.” Jika yang diinginkan adalah objek yang bereaksi terhadap tabrakan fisik dengan objek lain, maka perlu menambahkan komponen Rigidbody. Untuk tabrakan bukan pemicu digunakan OnCollisionEnter, OnCollisionExit, dan OnCollision Stay.

OnCollisionEnter dipanggil saat collider objek menyentuh collider objek lain yang mempunyai Rigidbody. Saat sedang bersentuhan (bertabrakan), OnCollisionStay dipanggil. Jika sentuhan diakhiri, OnCollisionExit dipanggil. Cara yang sama berlaku jika objek sebagai pemicu (diatur sebagai Trigger), dan yang digunakan adalah OnTriggerEnter, OnTriggerStay, dan OnTriggerExit (Widowati, 2020).

1. FOV ( Field of View )



**Gambar 2.7. Gambaran dari Field of View**

( Sumber : <https://whatis.techtarget.com/definition/field-of-view-FOV>)

Field of view (FOV) adalah area terbuka yang dapat dilihat seseorang melalui matanya atau melalui perangkat optik. Dalam hal perangkat optik dan sensor, FOV menggambarkan sudut di mana perangkat dapat mengambil radiasi elektromagnetik.

FOV memungkinkan cakupan area daripada titik fokus tunggal. Dalam realitas virtual (VR), FOV besar sangat penting untuk mendapatkan pengalaman yang mendalam dan hidup. FOV yang lebih luas juga menyediakan jangkauan sensor yang lebih baik atau aksesibilitas untuk banyak perangkat optik lainnya.

1. Waterfall

Metode yang digunakan adalah *waterfall,* Menurut (Simarmata, 2010) “Metode pengembangan model waterfall dengan mempunyai langkah-langkah antara lain mendefinisikan masalah, analisis kebutuhan, merancang prototipe, implementasi, integrasi/pengujian dan rilis/pemeliharaan”. Sedangkan Menurut Rosa dan Shalahuddin (Nurhadi, 2018) Model SDLC air terjun (waterfall) sering juga disebut model sekuensial linier (Sequential Linear) atau alur hidup klasik (Classic Life Cycle)”.

Menurut Rosa dan Shalahuddin (Nurhadi, 2018) berikut adalah penjelasan mengenai analisis, desain, pengkodean, pengujian dan tahap pedukung (support).

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mengekspresikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasikan kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

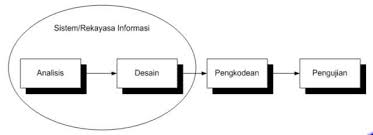
Desain perangkat lunak adalah proses langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan kerepresentasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan Kode Program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tehnik ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada peranagkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan diinginkan.



**Gambar 2.8. Alur pengujian**

1. Storyboard

*Storyboard* dalam bentuknya yang paling dasar adalah presentasi garis besar ide menggunakan gambar dan (biasanya) kata-kata. *Storyboard* adalah kelanjutan proses perancangan skenario menggunakan gambar namun *storyboard* lebih menyajikan tujuan yang unik dan spesifik dari gerakan *scripting* untuk animasi dan film (Jacobs, 2015).

*Storyboard* adalah sarana komunikasi yang penting dan efektif digunakan dalam pembuatan *film*. Pembuat *film* menggunakan pensil / pena dan bingkai kertas untuk skenario *storyboard*. Baru-baru ini seniman *storyboard* telah menggunakan grafis berbasis pena tablet untuk memvisualisasikan dan berkomunikasi secara efektif mengenai ide yang telah dibuat (Jhala dkk., 2008).