

BAB 2

LANDASAN TEORI

2.1 Visual Novel

Visual Novel adalah jenis cerita fiksi interaktif yang umumnya menggunakan grafik statis yang dilengkapi dengan kotak percakapan untuk menyampaikan narasi dan ucapan setiap karakter. Pada dasarnya, Visual Novel mirip dengan "*choose-your-own-adventure game*", tetapi lebih menyerupai buku digital karena formatnya yang padat teks. Penggunaan gambar dan suara yang vital menjadikan Visual Novel sebagai media interaktif yang biasanya berkulat pada cerita fiksi, meskipun dapat juga digunakan sebagai buku interaktif digital [3].

Visual novel termasuk salah satu jenis animasi cerita di mana memfokuskan pembaca pada visualisasi gambar beserta teks dan suara dari sinopsis cerita yang dipilih oleh pembaca. Dalam sejarahnya penggunaan visual novel ini bukanlah sebuah hal yang baru. Pada 1988, Onami merilis sebuah game rancangan hideo Kojima berjudul "snatcher" yang memberi dampak yang cukup besar dalam perkembangan visual novel. Game ini menghadirkan beberapa animasi bergerak, musik, dan juga art dengan cyberpunk yang detail.

Visual novel dapat digunakan sebagai konsep pembelajaran karena dapat membantu siswa menikmati proses belajar dan membantu meningkatkan kualitas pembelajaran. Formatnya yang berbasis *storytelling* memungkinkan penyampaian cerita dan dialog yang menarik, sehingga memotivasi pengguna untuk mendalami kisahnya. Pada intinya, visual novel *game* membawa pengguna ke dalam ceritanya dan mengajak mereka untuk terlibat dalam interaksi dengan alur cerita [4].

2.2 Cerita Rakyat

Cerita rakyat dapat dipahami sebagai sebuah kisah atau cerita yang berasal dari masyarakat zaman dahulu dan berkembang secara luas dari mulut ke mulut hingga pada akhirnya dikenal secara luas. Cerita rakyat merupakan cerminan budaya suatu masyarakat yang terwujud melalui bahasa. Cerita-cerita ini sarat dengan berbagai aspek budaya, seperti kepercayaan, adat istiadat, aturan, kegiatan ekonomi, struktur keluarga, dan nilai-nilai sosial yang dianut masyarakat. Cerita rakyat sendiri merupakan warisan budaya yang perlu dilestarikan hingga generasi selanjutnya. Dahulu kala, cerita rakyat diwariskan secara lisan dari generasi ke generasi, menjadi bagian penting dalam kehidupan masyarakat [3]. Namun, karena cerita rakyat disampaikan secara lisan, maka sangat sulit untuk mengetahui siapa

pengarangnya. Selain itu, cerita rakyat juga termasuk jenis cerita fiktif sehingga tidak dapat dibuktikan kebenarannya.

2.3 Blender

Blender merupakan sebuah perangkat lunak grafik 3D yang digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif, dan permainan video. Umumnya Blender dikenal luas oleh masyarakat sebagai paket pembuatan 3D gratis dengan sumber terbuka. Blender sangat cocok untuk individu atau studio kecil yang ingin mendapatkan keuntungan dari pipeline terpadu dan proses pengembangan yang responsif.

Blender bersifat *open-source*, yang terkenal dengan keserbagunaannya. Kemampuannya mencakup berbagai aspek kreasi 3D, seperti pemodelan, animasi, penyuntingan, dan rendering, termasuk rendering real-time dan non-real-time. Salah satu keunggulan utama Blender adalah sifatnya yang cross-platform. Blender dapat berjalan dengan lancar di berbagai sistem operasi, termasuk Linux, Windows, maupun Macintosh, Blender menggunakan OpenGL untuk antarmuka pengguna, memastikan pengalaman yang konsisten dan responsif di semua platform yang didukung [5].



Gambar 2.1 Logo Blender

2.4 Aplikasi

Aplikasi adalah jenis perangkat lunak komputer yang dirancang untuk menyelesaikan tugas atau fungsi tertentu yang **secara langsung** bermanfaat bagi pengguna. Berbeda dengan perangkat lunak sistem yang fokus pada integrasi kemampuan komputer, aplikasi **berfokus pada penerapan kemampuan tersebut** untuk menyelesaikan tugas yang **menguntungkan pengguna**. Contohnya adalah pengolah kata, lembar kerja, dan pemutar media[6].

2.5 3D

Objek 3D memiliki bentuk, ruang, dan volume yang posisinya ditentukan koordinat X, Y, dan Z. Mereka tersusun dari titik (vertex), garis (edge), dan permukaan (face) yang saling terhubung. Titik adalah lokasi dasar dengan koordinat, garis terbentuk dari dua titik, dan permukaan terbentuk dari tiga titik dan garis yang membentuk bidang tertutup. Gabungan kesemuanya ini disebut mesh, representasi digital objek 3D yang utuh[7].

2.6 C#

C Sharp, sering disingkat **C#**, adalah bahasa pemrograman canggih yang dibuat oleh Microsoft. Bahasa ini berorientasi objek, artinya ia memungkinkan pemrogram untuk membuat program dengan cara yang lebih terstruktur dan mudah dipahami. **C#** didesain berdasarkan bahasa **C++** yang populer, namun dengan beberapa penyederhanaan dan tambahan fitur yang terinspirasi dari bahasa lain seperti Java, Delphi, dan Visual Basic.

C# merupakan bahasa pemrograman handal yang tidak hanya piawai dalam **pengembangan game**, tetapi juga terintegrasi dengan mulus dengan **Unity**. Kemampuan ini menjadikan **C#** pilihan tepat untuk membangun **game 2D dan 3D** yang memukau[8].

2.7 Unity

Unity adalah salah satu **mesin game** paling mutakhir saat ini, memungkinkan para pengembang untuk menciptakan pengalaman bermain game yang imersif dan memukau. Didirikan pada tahun 2004 oleh David Helgason, Nicholas Francis, dan Joachim Ante, **Unity** telah berkembang menjadi platform yang digunakan oleh para pengembang game di seluruh dunia[8]. **Unity** tidak hanya dapat digunakan untuk pengembangan game, tetapi juga bisa digunakan untuk membuat media pembelajaran. **Unity** dapat membuat game berbasis 2D maupun 3D[9][10].



Gambar 2.2 Logo Unity

Adapun fitur-fitur yang dimiliki oleh Unity 3D antara lain sebagai berikut [16]:

1. *Integrated development environment* (IDE) atau lingkungan pengembangan terpadu.
2. Penyebaran hasil aplikasi pada banyak platform.
3. Engine grafis menggunakan Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iOS), dan proprietary API (Wii).
4. Game Scripting melalui Mono. Scripting yang dibangun pada Mono, implementasi open source dari NET Framework. Selain itu pemrograman dapat menggunakan UnityScript (bahasa kustom dengan sintaks JavaScript-inspired), bahasa C# atau Boo (yang memiliki sintaks Python-inspired).

2.8 Use Case Diagram

Diagram Use Case merupakan alat visual yang menggambarkan interaksi antara **aktor** (pengguna atau sistem eksternal) dan **sistem** yang sedang dirancang. Diagram ini membantu memvisualisasikan fungsionalitas sistem dan bagaimana pengguna akan berinteraksi dengannya, Penelitian yang dilakukan oleh Olsson dan Lindstrom [11] menunjukkan bahwa penggunaan diagram Use Case dalam tahap awal pengembangan perangkat lunak dapat meningkatkan pemahaman pemangku kepentingan tentang persyaratan sistem dan mengurangi risiko miskomunikasi. Berikut merupakan beberapa simbol use case diagram.

1. Actor

Aktor adalah elemen penting dalam diagram kasus penggunaan yang membantu memahami bagaimana sistem digunakan oleh entitas eksternal. Meskipun bukan bagian dari sistem itu sendiri, aktor memainkan peran penting dalam mendefinisikan fungsionalitas dan batasan sistem



2. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Pada gambar berikut merupakan Simbol *Use Case* pada *Use Case*

Diagram

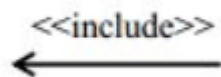


3. Association

Association dalam diagram kasus penggunaan adalah alat penting untuk memvisualisasikan hubungan antar elemen dan memahami bagaimana sistem bekerja. Association membantu mendefinisikan interaksi antara aktor dan kasus penggunaan, serta hubungan antar kasus penggunaan itu sendiri.

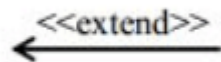
4. Include

Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya.



5. Extend

Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri.



2.9 Activity Diagram

Aktiviti diagram merupakan alat bantu pemodelan visual yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem, proses bisnis, ataupun menu dalam perangkat lunak. Diagram ini memfokuskan pada **aktivitas yang dilakukan oleh sistem** itu sendiri, bukan pada tindakan yang dilakukan oleh aktor atau sistem [12]. Berikut merupakan contoh simbol yang terdapat pada activity diagram.

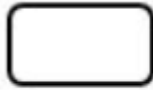
1. Status Awal

Digunakan sebagai penanda dimulainya suatu aktivitas yang akan dilakukan.



2. Aktivitas

Digunakan untuk menggambarkan suatu proses yang terjadi pada sistem.



3. Decision / Percabangan

Digunakan untuk Menggambarkan percabangan dimana pilihan aktivitas lebih dari satu.



4. Join

Digunakan sebagai penghubung antara simbol.



5. Status Akhir

Digunakan sebagai penanda bahwa suatu aktivitas sudah berakhir



6. Swimlane

Digunakan untuk pemisah antara aktivitas yang terjadi pada aktor dan aplikasi.



2.10 Sequence Diagram

Sequence Diagram merupakan peta interaksi antar objek dalam sebuah sistem. Diagram ini menyajikan gambaran urutan pesan yang dipertukarkan antar objek, dimana alur percakapan yang tersusun rapi. Tak hanya objek internal sistem, diagram ini juga menyajikan interaksi dengan pengguna, display, dan elemen eksternal lainnya[13].

2.11 Class Diagram

Class Diagram adalah alat untuk memvisualisasikan dan memahami struktur statis sistem perangkat lunak. Diagram ini membantu pemodel sistem, pengembang perangkat lunak, dan pemangku kepentingan untuk memahami bagaimana sistem bekerja dan mengidentifikasi potensi masalah desain. *Class diagram* memperlihatkan hubungan antar kelas baik private atau publik dan penjelasan detail tiap – tiap kelas dalam model desain dari suatu sistem. Proses analisis dalam *class diagram* memperlihatkan aturan – aturan dan tanggung jawab terhadap entitas yang dibuat [19].

1.12 2D

Animasi 2D merupakan animasi yang memiliki bentuk dua dimensi, maksudnya adalah Animator menciptakan karakter dan latar belakang dalam bentuk dua dimensi lalu menggerakkannya secara berurutan untuk menciptakan ilusi gerakan. Ciri khas animasi 2D adalah tampilan karakter yang sederhana dan gerakan yang terbatas pada sumbu X dan Y. Proses pembuatannya yang melibatkan penggambaran manual satu per satu gambar menjadikan animasi 2D sebagai teknik animasi tradisional yang unik[14] .

1.13 3D

Animasi 2D dan 3D memiliki pendekatan yang sangat berbeda dalam proses pembuatannya. Animasi 2D umumnya dimulai dengan sketsa tangan, lalu diwarnai dan dianimasikan secara frame-by-frame. Setiap gerakan kecil harus digambar secara manual, sehingga prosesnya cenderung lebih memakan waktu. Sebaliknya, animasi 3D lebih banyak melibatkan penggunaan perangkat lunak komputer untuk membangun model tiga dimensi dari objek dan karakter. Setelah model selesai, animator dapat memindahkan dan memanipulasi model tersebut secara digital untuk menciptakan gerakan. Proses ini memungkinkan animator untuk membuat animasi yang lebih kompleks dan realistis dalam waktu yang lebih singkat[15].

Animasi yang paling diminati saat ini adalah yang memiliki kualitas grafis yang lebih tinggi, terutama animasi 3D. Perkembangan teknologi telah memungkinkan pembuatan animasi dengan detail grafis yang sangat baik, seperti yang ditemukan dalam animasi 3D [20]. Animasi 3D dalam komputer menggambarkan gambar dengan kedalaman dengan dimensi X, Y, dan Z, meskipun pada dasarnya tetap merupakan representasi 2D di layar kaca. Berbeda dengan animasi 2D yang datar, animasi 3D memiliki volume dan bentuk yang lebih nyata dan dapat dilihat dari berbagai sudut pandang [21]