

BAB II

TINJAUAN PUSTAKA

2.1 Rancang Bangun

Rancang bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa komponen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi. Jadi, rancang bangun adalah proses mengubah hasil analisis ke dalam paket perangkat lunak dan kemudian membangun atau memperbaiki sistem [1]. Rancang bangun sangat berkaitan dengan perancangan sistem yang merupakan satu kesatuan untuk merancang dan membangun sebuah aplikasi [2].

2.2 Game

Game adalah salah satu jenis media hiburan yang banyak disukai orang untuk menghilangkan kejenuhan atau hanya untuk mengisi waktu luang [3]. *Game* adalah permainan yang diatur dengan aturan sehingga pemain dapat mengetahui siapa yang menang atau kalah. Permainan ini biasanya dimainkan dalam situasi yang tidak serius dan dengan tujuan menghilangkan stres dan menghabiskan waktu [4].

2.3 Jenis-Jenis Game

Jenis-jenis game antara lain adalah :

a. Education Game

Game ini bertujuan untuk memancing minat belajar sambil bermain dengan konteks pendidikan yang lebih mudah untuk dipahami.

b. Puzzle Game

Game ini memberikan tantangan kepada pemainnya dengan cara menjatukan sesuatu dari atas ke bawah. Pemain harus menyusunnya, susunan ini dilakukan secepat dan sebaik mungkin. Semakin lama akan semakin cepat dan semakin banyak objek yang jatuh. Contoh *puzzle game* adalah Tetris.

c. *Board Game*

Jenis *game* ini sama dengan *game board* tradisional, seperti monopoli dan *chess*. Versi elektronik hanya memindahkan versi tradisional ke layar komputer.

d. *Sport Game*

Selama *game* ini mengutamakan genre olahraga maka disebut genre *sport* tidak peduli *game* ini menggunakan gaya 2D atau 3D. contoh *sport game* adalah *Winning Eleven*, *FIFA* dan *NBA*.

2.4 *Game Edukasi*

Game edukasi adalah salah satu jenis *game* yang dapat membantu penggunanya belajar melalui media permainan yang mudah dipahami [3]. *Game* edukasi adalah jenis permainan yang dimaksudkan untuk mengajarkan keterampilan bermain kepada pemain, meningkatkan pemahaman mereka tentang topik tertentu, meningkatkan pemahaman mereka tentang peristiwa budaya atau historis, atau memperluas pengetahuan mereka tentang topik tertentu [5].

Game edukasi juga dikenal sebagai "permainan pendidikan", adalah permainan yang dapat digunakan untuk membantu siswa belajar. Permainan ini mengandung elemen yang berfungsi sebagai pendidik atau nilai-nilai pendidikan [6].

2.5 *Android*

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux yang memungkinkan pengembang membuat aplikasi mereka sendiri yang dapat digunakan pada berbagai piranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, perusahaan baru yang membuat piranti lunak ponsel. Kemudian dibentuk *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia* [7].

Android Nougat adalah varian *Android* yang rilis pada 2016 lalu, di mana *smartphone* *Nexus* dan *LG V20* jadi deretan *smartphone* yang mengusungnya di awal. Berbagai media teknologi menyebut bahwa kelebihan *Android Nougat*

sebagai salah satu yang revolusioner di *platform Android*, serta jadi OS yang mendapat banyak *review* positif.

2.6 C#

C# adalah Bahasa Pemrograman yang digabungkan dari berbagai bahasa pemrograman (C, C++, dan Java), sintaks yang digunakan adalah sintaks dari Java, serta sifat dari bahasa C# adalah bersifat *Case Sensitive*, besar kecil Huruf berpengaruh. C# (*C-Sharp*) adalah bahasa pemrograman yang dikembangkan oleh *Microsoft* yang berjalan pada *.NET Framework* (Gusrion, 2018). C # digunakan untuk mengembangkan aplikasi web, aplikasi desktop, aplikasi seluler, game, dan banyak lagi [8].

2.7 Firebase

Firebase adalah suatu layanan dari *Google* untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. *Firebase* adalah *platform* terpadu *Google* yang menyatukan berbagai fitur atau layanan canggih untuk aplikasi, termasuk backend seluler, analisis, dan fitur pertumbuhan. Berikut akan dijelaskan manfaat utama dari *Firebase* [9].

Sejarah *Firebase* singkatnya dimulai pada tahun 2011 oleh Andrew Lee dan James Tamplin. *Realtime Database* adalah produk *Firebase* pertama yang digunakan oleh *developer* untuk menyimpan dan menghubungkan data ke berbagai *user*. Dia kemudian berubah menjadi layanan pengembang aplikasi. Perusahaan tersebut dibeli oleh *Google* pada Oktober 2014.

2.8 Unity

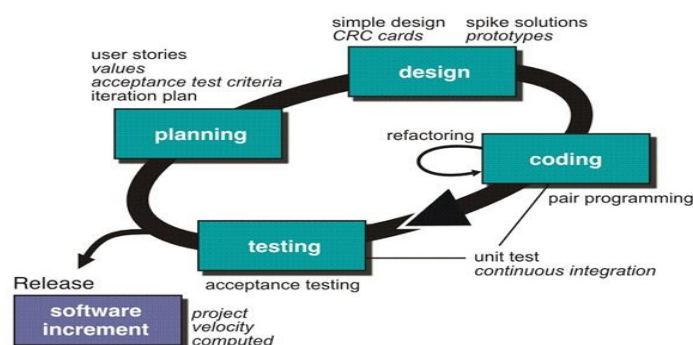
Game engine Unity 3D merupakan sebuah *software* (perangkat lunak) yang dirancang untuk dapat menciptakan atau mengembangkan video *Game*. Fungsi utama yang disediakan oleh *game engine* biasanya mencakup *renderer engine* (mesin render) yang berguna untuk me-render 2D atau 3D grafis, *physics engine* untuk membuat objek 3D berlaku layaknya sebagai benda nyata (terpengaruh gravitasi, bertabrakan), *sound* (suara), *script*, animasi, kecerdasan buatan (AI),

jaringan, *streaming*, manajemen memori, *threading*, dan grafik animasi. Ada banyak *Game engine* yang dirancang untuk membuat Game untuk berbagai platform seperti konsol video Game dan sistem desktop seperti *Microsoft Windows*, *Linux*, dan *Mac OS* (Pudyanto, 2014). Kelebihan dari *Unity 3D*, *Unity* dapat dioperasikan pada *platform Windows* dan *Mac OS* dan dapat menghasilkan *Game* untuk *Windows*, *Mac*, *Linux*, *Wii*, *iPad*, *iPhone*, *Google Android* dan juga *browser* [10].

Unity menyediakan *script editor* yang terpasang bersamaan di dalamnya, yaitu *MonoDevelop*. *Script editor* tersebut menyediakan lingkungan pengembangan terintegrasi yang digunakan oleh peneliti untuk melakukan proses pengkodean program komputer. *MonoDevelop* memiliki fitur dukungan penyelesaian kode dalam bahasa C#, template kode, dan pemeriksaan error yang terintegrasi dengan *Unity* [11].

2.9 Extreme Programming

Suatu metode yang paling umum digunakan untuk pengembangan perangkat lunak cepat adalah metode *extreme programming*. Metode pengembangan perangkat lunak yang dikenal sebagai *Extreme Programming (XP)* bertujuan untuk meningkatkan kualitas perangkat lunak agar sesuai dengan perubahan dan kebutuhan pelanggan [12]. Terutama untuk pengembangan aplikasi dan permainan untuk ponsel dengan jumlah programmer yang terbatas, *extreme programming* adalah metode pengembangan cepat yang sering digunakan. Metode ini menuntut banyak perubahan dalam proses pengembangannya.



Gambar 2. 1 Metode *Extreme Programming*

Metode *extreme programming* mempunyai 4 tahapan yaitu:

a. *Planning*

Dalam tahapan ini, beberapa kegiatan perencanaan dilakukan, termasuk menemukan masalah, menilai kebutuhan sistem dan pengguna, dan menetapkan jadwal pelaksanaan pengembangan sistem.

b. *Design*

Proses ini merupakan bagian dari proses perancangan sistem, di mana pemodelan sistem dan pemodelan arsitektur sistem dilakukan dengan menggunakan diagram *Unified Modelling Language* (UML).

c. *Coding*

Ini adalah implementasi dari perancangan sistem dengan *user interface* yang dibuat menggunakan bahasa pemrograman. Dalam penelitian ini, bahasa pemrograman yang digunakan adalah *c#*.

d. *Testing*

Tahapan yang terakhir ialah testing, di tahap ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi. Pengujian ini dilakukan untuk mengetahui apakah aplikasi dapat berjalan sesuai harapan untuk dicapai.

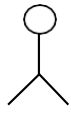

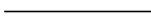

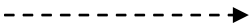
2.10 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan perangkat lunak yang telah distandarisi sebagai media penulisan cetak biru (blueprints) perangkat lunak (Pressman). UML dapat digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi beberapa komponen sistem yang ada dalam perangkat lunak [13].

2.10.1 Use Case Diagram

Use case diagram adalah gambar yang menjelaskan bagaimana pengguna akan memakai suatu sistem atau program komputer, dengan menggunakan simbol-simbol tertentu untuk memperjelas alurnya.





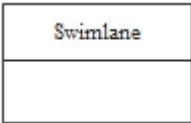

Tabel 2. 1 Simbol *Use Case Diagram*

No	Simbol	Keterangan Fungsi
1.	<i>Actor</i> 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	<i>Use Case</i> 	<i>Use Case</i> adalah deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya atau sebaliknya dari bawah ke atas.
5.	<i>Defendency</i> 	<i>Defendency</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (<i>independen</i>).

2.10.2 Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *use case* yang memiliki alur aktivitas.

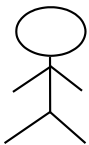

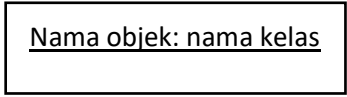
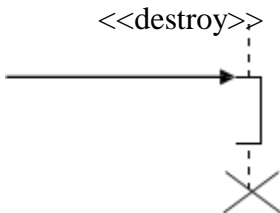
Tabel 2. 2 Simbol *Activity Diagram*

No	Simbol	Keterangan Fungsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		<i>Swimlane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.10.3 Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya.

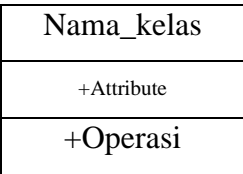


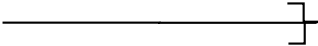
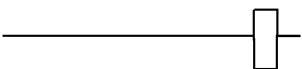
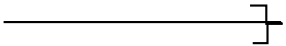
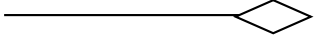
Tabel 2. 3 Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  Atau <u>Nama aktor</u> Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor
2.	Garis hidup <i>/lifeline</i> 	Menyatakan kehidupan suatu objek
3.	Objek 	Menyatakan objek yang berinteraksi peran
4	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i>

2.10.4 Class Diagram

Class diagram mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2. 4 Simbol *Class Diagram*

No	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan simbol
4.	<p>Asosiasi Berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan simbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / <i>dependency</i></p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.11 Penelitian Terkait

Penelitian adalah sebuah perjalanan intelektual yang membawa kita pada pemahaman mendalam terhadap suatu fenomena atau masalah tertentu. Dalam konteks ini, penelitian ini digagas untuk mengeksplorasi dan menganalisis aspek-aspek tertentu yang berkaitan dengan judul penelitian. Penelitian terkait dengan judul penelitian yang dilakukan saat ini terlihat pada tabel di bawah ini.

Tabel 2. 5 Penelitian Terkait

No	Peneliti	Judul	Hasil Penelitian	Metode
1.	[3]	Rancang Bangun Game Edukasi Perhitungan Dasar Matematika Sekolah Dasar Kelas 3, 4 dan 5 Menggunakan Construct 2	<i>Game</i> edukasi perhitungan dasar matematika dapat memberikan pengalaman baru dalam dunia pembelajaran dan diharapkan mampu mencapai standar kompetensi siswa sekolah dasar dan meningkatkan minat anak sekolah dasar untuk belajar matematika khususnya belajar berhitung.	<i>Game Development The Life Cycle (GDLC)</i>
2.	[14]	Aplikasi Belajar Matematika Sekolah Dasar Kelas V Berbasis <i>Mobile Android</i>	Aplikasi media pembelajaran berbasis android pada mata pelajaran matematika kelas V tingkat sekolah dasar dapat	<i>Extreme Programming (XP)</i>

Tabel 2. 6 Penelitian Terkait (Lanjutan)

No	Peneliti	Judul	Hasil Penelitian	Metode
			digunakan sebagai media belajar mandiri dan membantu meningkatkan minat belajar siswa.	
3.	[15]	RANCANG BANGUN GAME EDUKASI TEMPAT BERSEJARAH DI INDONESIA	Game ini berpotensi meningkatkan pengetahuan mengenai tempat bersejarah di Indonesia khususnya tempat bersejarah yang ada di Pulau Sumatera dan Pulau Jawa seperti yang ada di dalam game.	<i>Game Development The Life Cycle (GDLC)</i>
4.	[16]	E-Report Engineering System for Senior High School in Lampung	Aplikasi berbasis web seluler dibangun secara online oleh menghasilkan fitur seperti pemrosesan data laporan berupa laporan penilaian berdasarkan materi dan kemajuan siswa secara keseluruhan dan laporan yang dapat diakses secara online oleh orang tua.	<i>Rational Unified Procces (RUP)</i>