

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam melakukan penelitian, diperlukan mempelajari penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan. Penelitian yang terkait, digunakan sebagai referensi untuk membantu peneliti mengembangkan penelitian, agar terhindar dari penelitian yang sama. Berikut ini beberapa tinjauan pustaka yang mendukung penelitian yang disajikan pada tabel 2. 1.

Tabel 2. 1 Tinjauan Pustaka

1	Judul	Rancang Bangun <i>Game</i> Edukasi Tempat Bersejarah Di Indonesia
	Jurnal	Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)
	Volume dan Halaman	Vol. 1, No. 1, Juni 2020, <i>page-page</i> . 51-57
	Tahun	2020
	Penulis	Sanriomi Sintaro, Rahmat Ramdani, Slamet Samsugi
	Identifikasi Masalah	Banyak kalangan anak-anak yang terkesan tidak peduli terhadap peninggalan sejarah nenek moyang mereka, sehingga mengakibatkan kurangnya pemahaman tentang situs besejarah salah satunya arsitekur sejarah Indonesia. Oleh karena itu, penulis membangun sebuah media pembelajaran yang dapat meningkatkan minat dan pengetahuan anak tentang tempat-tempat
	Metode/Tools	1. Metode pengembangan sistem yang digunakan adalah <i>Game Development Life Cycle</i> , yang terdiri dari <i>Initiation, Pre-production, Production, Testing, Beta, Release</i> . 2. Metode perancangan pemodelan menggunakan metode <i>Finite Stage Machine</i> .
	Hasil Penelitian	Penelitian ini menghasilkan <i>Game</i> Edukasi Tempat Bersejarah Di Indonesia berbasis <i>Android</i> , sebagai media

		alternatif yang menarik dan menyenangkan agar dapat meningkatkan minat mengetahui tempat bersejarah di Indonesia.
--	--	---

Tabel 2. 1 Tinjauan Pustaka (Lanjutan)

2	Judul	Rancang Bangun Aplikasi <i>Game</i> “Petualangan Arjuna” Berbasis <i>Android</i> Dengan Pemodelan <i>Luther</i>
	Jurnal	Seminar Nasional Teknologi Informasi dan Komunikasi 2019 “Teknologi Humanis di Era <i>Society 5.0</i> ” e-ISSN: 2685-5615
	Volume dan Halaman	-
	Tahun	2019
	Penulis	Faisal Angga Setiadi
	Identifikasi Masalah	Penelitian ini mengangkat isu tentang bagaimana cara membangun <i>game</i> petualangan dengan menggunakan metode <i>luther</i> . Permainan yang akan dibuat menggambarkan seorang petualang yang bernama Arjuna yang sedang menjelajahi ke berbagai lokasi guna mencari pusaka sebagai syarat untuk membuka pintu kerajaan yang diambil paksa oleh Sang Butho Cakil
	Metode/Tools	1. Metode pengembangan sistem yang digunakan adalah <i>Multimedia Development Life Cycle</i> yang terdiri dari <i>Concept, Design, Material Collecting, Assembly, Testing, dan Distribution</i> . 2. Menggunakan metode <i>Luther</i> . 3. Menggunakan bahasa pemrograman C#.
Hasil Penelitian	Menghasilkan <i>Game</i> Petualangan Arjuna berbasis <i>Android</i> , sebagai sarana hiburan, melatih kecerdasan dan ketepatan terhadap suatu kondisi yang dapat digunakan untuk melatih kesiapan dan kreatifitas dalam menyelesaikan masalah. Dan dapat diambil kesimpulan bahwa, dalam merancang, membangun dan mengimplementasikan metode <i>luther</i> pada <i>Game</i> “Petualangan Arjuna” ini berhasil.	
3	Judul	Penerapan Model Pengembangan <i>Game</i> GDLC (<i>Game Development Life Cycle</i>) Dalam Membangun <i>Game Platform</i> Berbasis <i>Mobile</i>
	Jurnal	TEKNOKOM
	Volume dan Halaman	Vol. 2 No. 1 Maret 2019 ISSN: 2621-8070

	Tahun	2019
	Penulis	Rio Andriyat Krisdiawan, Darsanto

Tabel 2. 1 Tinjauan Pustaka (Lanjutan)

	Identifikasi Masalah	Dalam penelitian ini, mengangkat isu tentang bagaimana cara menerapkan model pengembangan GDLC dalam membangun <i>game platform</i> .
	Metode/Tools	<ol style="list-style-type: none"> 1. Menggunakan metode <i>Game Development Lifw Cycle</i> sebagai metode pengembangan sistem. 2. <i>Unifield Modelling Language</i> sebagai metode perancangan sistem. 3. Menggunakan <i>Construct 2D</i> sebagai <i>game engine</i>.
	Hasil Penelitian	Menghasilkan sebuah <i>Game Platform</i> yang berjudul <i>PIKA Adventure</i> yang berbasis <i>Android</i> . <i>Game</i> ini merupakan <i>game</i> yang ringan dan memiliki unsur kesenangan bagi yang memainkannya. Dan dapat diambil kesimpulan bahwa, <i>Game Development Life Cycle</i> sangat cocok digunakan dalam mengembangkan pembuatan <i>game</i> , dan UML sangat memudahkan dalam perancangan sistem.
4	Judul	Perancangan <i>Game</i> OTW SARJANA Menggunakan Metode <i>Forward Chaining</i>
	Jurnal	SIMKOM
	Volume dan halaman	Vol. 6, No. 2, Juli 2021 ISSN: 2715-906X(Online) doi: 10.51717/simkom.v6i2.56
	Tahun	2021
	Penulis	Muhammad Bambang Firdaus, Damar Sanggara Habibie, Fadli Suandi, M Khairul Anam, Lathifah
	Identifikasi Masalah	Pada penelitian yang dilakukan peneliti, peneliti membuat sebuah <i>game</i> OTW SARJANA. <i>Game</i> ini menggambarkan seorang mahasiswa yang sedang kuliah untuk mencapai gelar sarjana. Lantaran waktu seorang mahasiswa yang sedang berkuliah akan menghadapi banyak rintangan yang ada di depannya, namun mahasiswa tadi wajib tetap fokus pada tujuan utamanya. <i>Game</i> yang dibuat oleh penulis merupakan <i>game</i> bergenre <i>side scrolling</i> , yang mengkondisikan karakter harus terus berpetualang untuk mencapai tujuan yang sesuai dengan yang ada di dalam <i>game</i> .

	Metode/Tools	<p>1. Metode yang digunakan adalah metode <i>Forward Chaining</i>. Metode ini digunakan pada saat pemain akan naik ke <i>level</i> selanjutnya, dan untuk menentukan nilai pada setiap <i>level</i>.</p> <p>2. <i>Adobe photoshop CS6</i></p>
--	---------------------	---

Tabel 2. 1 Tinjauan Pustaka (Lanjutan)

	Hasil Penelitian	Menghasilkan sebuah <i>Game</i> “OTW SARJANA” yang berbasis <i>Desktop</i> . Dan mendapatkan kesimpulan bahwa, metode <i>forward chaining</i> bisa diterapkan dan dipakai dalam <i>game</i> 2 dimensi bergenre <i>side scrolling</i> . Walaupun pada <i>gameplay</i> dan <i>testing</i> masih ditemukan masalah pada permainan yang dimainkan oleh pengguna. Tetapi dalam rata-rata waktu dan skor dapat disimpulkan bahwa pengguna bermain dengan kemampuan apa adanya.
5	Judul	Penerapan Metode <i>Finite State Machine</i> pada <i>Game</i> “Santri <i>on the Road</i> ”
	Jurnal	Walisongo Journal of Information Technology
	Volume dan Halaman	Vol. 3 No. 1 (2021) : 11-18 doi: http://dx.doi.org/10.21580/wjit.2021.3.1.7135
	Tahun	2021
	Penulis	Hafizzudin Sifaulloh, Juniardi Nur Fadila, Fresy Nugroho
	Identifikasi Masalah	Pada penelitian yang dilakukan peneliti, peneliti membuat sebuah <i>game endless running 2D</i> , yaitu <i>santri on the road</i> , yang menggambarkan para santri yang mencari ilmu yang berupa kitab, namun para santri menghadapi banyak kendala di dalam perjalanannya. Walaupun berat para santri terus mencari ilmu tak kenal lelah.
	Metode/Tools	<p>1. Menggunakan metode <i>Finite State Machine</i>.</p> <p>2. <i>Unity</i> sebagai <i>game engine</i>.</p>
	Hasil Penelitian	Menghasilkan sebuah <i>Game</i> “Santri <i>On the Road</i> ” yang berbasis <i>Android</i> , yang dapat dijadikan sebagai alternatif bagi anak-anak usia dini untuk belajar akan pentingnya dalam mencari ilmu dengan sungguh-sungguh. Dan dapat diambil kesimpulan bahwa implementasi <i>Finite State Machine</i> pada <i>game</i> ini berjalan dengan bagus.

2.2. Rancang Bangun

Rancang bangun merupakan tahap dimana setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan fungsional, dan menggambarkan bagaimana sebuah sistem dibangun yang berupa perencanaan atau gambaran dari elemen yang terpisah kemudian digabungkan menjadi satu ke satuan yang utuh dan berfungsi, termasuk mengkonfigurasi komponen *software* dan *hardware* dari sebuah sistem (Aminah, 2018).

Menurut (Sunarto dan Krisdiawan, 2017) rancang bangun adalah suatu kegiatan yang menterjemahkan hasil analisis ke dalam bentuk *software* dan kemudian dibangun sistem baru atau memperbaiki sistem yang ada. Tujuan dari rancang bangun yaitu untuk memenuhi kebutuhan pengguna sistem dan berfokus pada desain sistem yang terperinci yang nantinya akan digunakan untuk membuat program komputer.

2.3. Game

Game merupakan sebuah sistem di mana *user* terlibat dalam konflik buatan. *User* berinteraksi dengan sistem dan konflik dalam sebuah permainan yang memiliki peraturan untuk membatasi perilaku *user* (Widodo dan Ahmad, 2018). Menurut (Sintaro, Ramdani dan Samsugi, 2020) *game* merupakan permainan yang menggunakan media elektronik, yang memiliki suara, gambar bergerak yang dibuat menarik dan memberikan kepuasan tersendiri kepada pengguna. Membuat *game* juga membutuhkan desain dari setiap bagian yang ada seperti *background*, *backsound*, tombol *button*. Semua bagian tersebut disebut dengan *resources game* (Jamal dan Siahaan, 2021).

2.4. Platform

Platform atau *platformer* merupakan salah satu genre *game* yang merupakan bagian dari genre aksi. Genre *platformer* biasanya penggunanya harus menggerakkan karakter yang ada di dalam *game* untuk terus bergerak menghadapi dan menghindari rintangan yang ada di depannya untuk mencapai suatu tujuan yang ada di dalam *game*. Pengguna mempunyai kemampuan untuk mengontrol karakter dalam *game* untuk menghadapi rintangan, atau membiarkan karakter yang dimainkan mati (Andana, Jeanny dan Jason, 2021).

2.5. 2 Dimensi (2D)

Dimensi dapat diartikan dengan memiliki banyak cara untuk menentukan posisi suatu objek yang berdasarkan terhadap acuan tertentu. Suatu objek bisa dikatakan berdimensi apabila bisa ditentukan dengan sebuah angka (Ratulangi, Saul dan Jacob, 2018). 2 Dimensi merupakan suatu objek yang memiliki 2 sisi yaitu bagian X dan Y. 2D dapat dibuat dengan menggunakan *bitmap* atau *vector*. Pada *game* 2D keseluruhan objek dan gerakan karakter terbatas yaitu hanya bergerak secara *vertical* dan *horizontal* (Khairani, Fadila dan Nugroho, 2021).

2.6. Unity

Unity merupakan sebuah *game engine* yang berbasis *cross-platform*. *Unity Technologies* dibangun tahun 2004 oleh David Helgason, Nicholas Francis dan Joachim Ante. Alasan dibangun *game engine* ini karena bentuk kepedulian mereka terhadap pengembang independen yang tidak dapat membeli *game engine*

dikarenakan harganya yang mahal. *Unity* dirilis pada tahun 2009 secara gratis, sehingga *Unity* mencapai popularitas tertinggi di dunia dengan jumlah lebih dari 1 juta *developer* pada tahun 2012 (Modami, Sengkey dan Tulenan, 2017).

Menurut (Harahap, Sucipto dan Jupriyadi, 2020) *unity* adalah sebuah *software* yang digunakan dalam membangun bermacam-macam aplikasi. Sebagian besar pengguna *unity* adalah untuk membangun aplikasi *game*. Selain itu, *unity* juga bisa digunakan untuk membuat berbagai macam aplikasi presentasi, *website*, dan juga *augmented reality*.



Gambar 2. 1 Logo *Unity*

2.7. *Android*

Android merupakan sistem operasi yang digunakan untuk *smartphone* atau tablet yang tersedia banyak fitur didalamnya untuk memudahkan kehidupan manusia yang sampai sekarang terus berkembang semakin canggih (Galih Pradana dan Nita, 2019). Menurut (Damayanti, Akbar and Sulistiani, 2020) *android* adalah sistem operasi berbasis *Linux* yang dikembangkan oleh *Open Handset Alliance* terdiri dari *software*, *hardware* dan *provider* seperti *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, dan *T-Mobile* yang mempunyai keunggulan terhadap kelengkapan aplikasi yang tersedia dan kemudahan menambahkan aplikasi sesuai keinginan. *Android* adalah generasi baru dari *platform mobile*, yang memberikan kebebasan

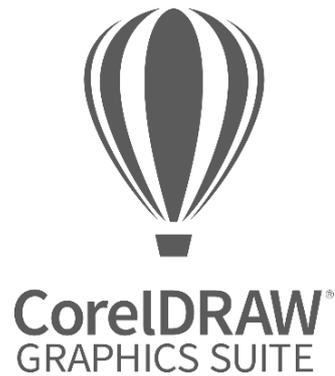
kepada *developer* untuk melakukan pengembangan sistem sesuai keinginannya (Hasanudin, Salim and Robby, 2017).



Gambar 2. 2 Logo Android

2.8. CorelDRAW

CorelDRAW merupakan program komputer yang melakukan *editing* pada garis *vector*. *CorelDRAW* memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan dalam bidang publikasi atau percetakan dan pekerjaan yang membutuhkan proses visualisasi (Rohmawati, 2019). *CorelDRAW* merupakan suatu *software* yang berbasis desain serta digunakan untuk merancang berbagai macam gambar grafis seperti *logo*, *poster* dan *banner*. *Software* ini juga bisa menjadi ladang penghasilan dan sangat populer di kalangan desainer (Galih Pradana dan Nita, 2019).



Gambar 2. 3 Logo *CorelDRAW*

2.9. Unified Modelling Language (UML)

UML merupakan salah satu standar bahasa yang membuat analisa dan *design*, mendefinisikan *requirement*, dan merancang arsitektur pemrograman berorientasi objek (Josi, 2017). UML menjelaskan model yang tepat, tidak ambigu dan lengkap. UML lebih menekankan langkah penting dalam keputusan Analisa, *design*, dan implementasi sistem perangkat lunak (Nugroho dan Rohimi, 2020).

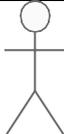
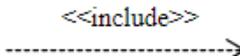
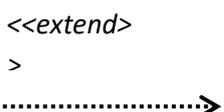
Dalam pemodelan sistem yang akan dibangun, peneliti menggunakan beberapa model diagram UML yaitu :

2.9.1. Use Case Diagram

Use case diagram merupakan model untuk menggambarkan sistem yang akan dibuat. *Use case* menggambarkan interaksi antara aktor dan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di pada sistem yang akan dibangun dan siapa yang berhak menggunakan fungsi tersebut (Simatupang dan Sianturi, 2019).

Berikut ini adalah simbol *use case diagram*, yang disajikan pada tabel 2. 2.

Tabel 2. 2 Simbol *Use Case Diagram* (Simatupang dan Sianturi, 2019)

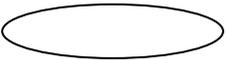
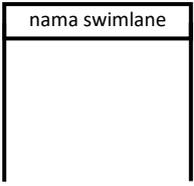
No.	Simbol	Nama	Keterangan
1		<i>Actor</i>	Pengguna, sistem lain, atau alat yang berhubungan dengan <i>Use Case</i>
2		<i>Use Case</i>	Fungsi yang disediakan oleh sistem sebagai tempat pertukaran pesan antar unit atau aktor
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> pada sistem
4		<i>Include</i>	Merupakan suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya.
5		<i>Extend</i>	Merupakan suatu <i>Use Case</i> tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.
6		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya pada sistem

2.9.2. Activity Diagram

Activity diagram merupakan aktivitas atau *workflow* suatu sistem yang ada pada *software*. *Activity diagram* menggambarkan aktivitas yang dilakukan oleh sistem, bukan apa yang dilakukan oleh *actor* (Simatupang dan Sianturi, 2019).

Berikut ini adalah simbol *Activity diagram*, yang disajikan pada tabel 2. 3:

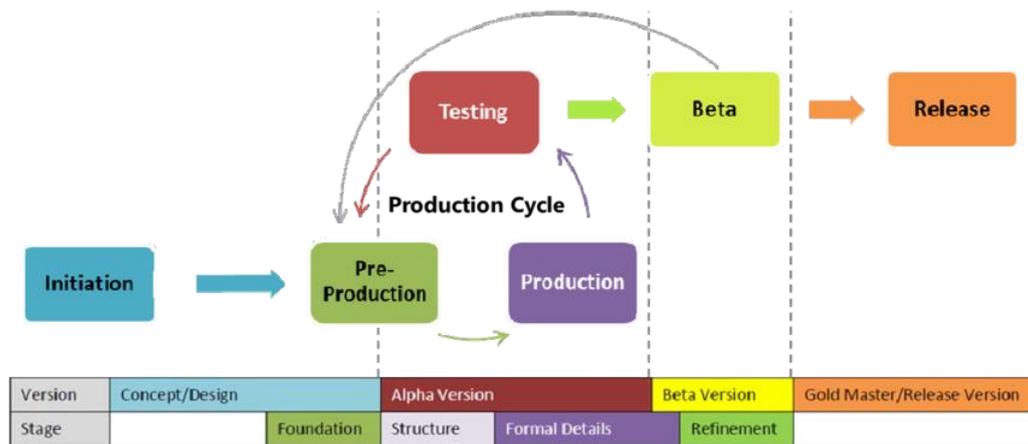
Tabel 2. 3 Simbol *Activity Diagram* (Simatupang dan Sianturi, 2019)

No.	Simbol	Nama	Keterangan
1		Status awal	Status awal yang dilakukan sistem.
2		Status akhir	Status akhir yang dilakukan sistem
3		Penggabungan	Lebih dari satu aktivitas digabungkan jadi satu.
4		Percabangan	Terdapat pilihan aktivitas lebih dari satu.
5		Aktivitas	Aktivitas yang dilakukan sistem.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.10. Metode Pengembangan Sistem

Metode yang dipakai pada pembuatan *game* ini adalah *Game Development Life Cycle* (GDLC). GDLC adalah salah satu metode pengembangan *game* yang

terdiri dari enam tahap, yang dimulai dari fase Inisiasi, fase Pra-produksi, fase Produksi, fase Pengujian, Beta, dan Rilis (Krisdiawan, 2018).



Gambar 2. 4 Metode Game Development Life Cycle (GDLC) (Andriyat Krisdiawan dan Darsanto, 2019)

1. *Initiation*

Initiation merupakan tahap awal yang dimulai dari membuat konsep *game* yang akan dibangun, menentukan jenis *game* seperti apa yang akan dibangun, dan menentukan target pengguna dari *game* yang akan dibangun. Hasil dari tahap *initiation* adalah konsep dan deskripsi dari *game* yang akan dibangun (Krisdiawan, 2018).

2. *Pre-production*

Pre-production merupakan fase yang penting dalam siklus *production*. Dalam *pre-production* juga merupakan tahap menentukan *genre game*, *design*, *game play*, tantangan, dan pembuatan *Game Design Document* (GDD) (Krisdiawan, 2018).

3. *Production*

Production merupakan tahap yang sangat penting, karena pada tahap ini dilakukan pembuatan *asset*, dan melakukan *coding*. Dalam tahap *production* juga melakukan kegiatan membuat fitur, membuat *level game*, serta melakukan penyesuaian dan penyempurnaan permainan (Krisdiawan, 2018).

4. *Testing*

Testing dilakukan setelah tahap *production*. *Testing* dilakukan untuk menguji apakah ada kesalahan fungsional atau tidak. Hasil dari melakukan *testing* adalah laporan *bug*, adanya perubahan, dan keputusan *developer* (Krisdiawan, 2018).

5. *Beta*

Beta merupakan pengujian dari pihak ketiga. Pengujian *beta* sama dengan pengujian sebelumnya. Ada dua jenis metode pengujian *beta* yaitu tertutup dan terbuka. Pada tahap *beta* tertutup hanya individu tertentu yang dipilih menjadi *tester*, sedangkan pada *beta* terbuka siapa saja bisa mengajukan diri untuk menjadi *tester* (Krisdiawan, 2018).

6. *Release*

Release merupakan tahap akhir, dimana aplikasi yang telah selesai dibangun siap untuk diluncurkan ke publik (Krisdiawan, 2018).

2.11. ***Game Design Document (GDD)***

Game design document adalah kumpulan bahan yang digunakan oleh desainer untuk memberikan informasi tentang *game* yang dirancang, sambil mengubah ide-ide yang abstrak menjadi tertulis (Handayani, M dan Al Fatta, 2017).

Menurut (Kurniawan, Rachman dan Aldya, 2018) *game design document* adalah kumpulan informasi *game* yang tersusun dan mengubah konsep yang abstrak menjadi rencana tertulis, yang berisi Spesifikasi, Gambaran, Mekanisme Permainan, Level Permainan, Kondisi Menang-Kalah, Rancangan Tingkat Kesulitan, Alur Permainan, dan Seni.

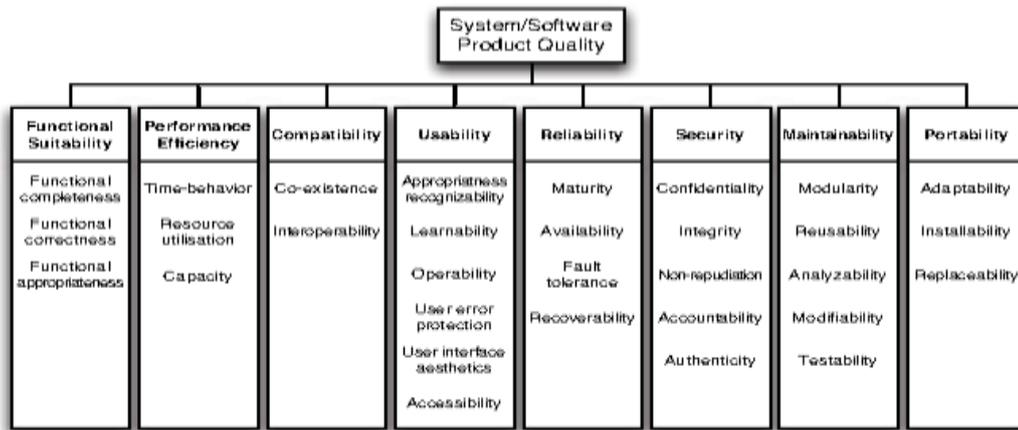
2.12. Storyboard

Storyboard adalah teknik yang mempelajari tentang perencanaan urutan suatu *scene* atau *scene* berupa gambar visual untuk memudahkan perancang dalam membuat alur cerita (Galih Pradana dan Nita, 2019). Menurut (Yamani, Adiwihardja dan Palasara, 2019) *storyboard* adalah sketsa gambar yang disusun menurut skenario. Dengan *storyboard*, kita dapat lebih mudah menyampaikan ide cerita kepada orang lain, karena kita dapat memandu imajinasi seseorang untuk mengikuti gambar yang disajikan.

2.13. ISO 25010

ISO 25010 merupakan salah satu tolak ukur kualitas yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. *ISO/IEC 25010* menggantikan standar *ISO/IEC 9126* (*ISO/IEC 25010*, 2011). Menurut (Gunawan dan Triantoro2, 2017) *ISO 9126* dianggap tidak relevan dengan teknologi saat ini, karena perubahan teknologi yang sangat pesat maka dibutuhkan standar pengujian perangkat lunak yang sesuai dengan kebutuhan.

ISO 25010 mempunyai delapan karakteristik, yang dapat dilihat pada gambar 2. 5.



Gambar 2. 5 *ISO 25010* (Gunawan dan Triantoro2, 2017)

Dari delapan karakteristik tersebut, peneliti akan menggunakan tiga karakter pada *ISO 25010*, yaitu :

1. *Functional Suitability*

Functional Suitability merupakan kemampuan perangkat lunak untuk menyediakan fungsi yang sesuai dengan kebutuhan pengguna di dalam keadaan tertentu (Gunawan dan Triantoro2, 2017). Penjelasan masing-masing sub-karakteristik dapat dilihat pada tabel 2. 4:

Tabel 2. 4 Penjelasan Sub-karakteristik *Functional Suitability* (Gunawan dan Triantoro2, 2017)

Karakteristik	Sub-karakteristik	Penjelasan
<i>Functional Suitability</i>	<i>Functional Completeness</i>	Apakah system bisa mencakup semua tugas dan tujuan pengguna secara spesifik?.
	<i>Functional Correctness</i>	Apakah sistem bisa menyediakan hasil yang sesuai dengan kebutuhan?

	<i>Functional appropriateness</i>	Apakah sistem bisa memfasilitasi tujuan pengguna pada kondisi tertentu?
--	-----------------------------------	---

2. Usability

Usability merupakan sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan efektif, efisiensi, dan kepuasan dalam penggunaan (Gunawan dan Triantoro2, 2017). Penjelasan masing-masing sub-karakteristik dapat dilihat pada tabel 2. 5:

Tabel 2. 5 Penjelasan Sub-karakteristik *Usability* (Gunawan dan Triantoro2, 2017)

Karakteristik	Sub-karakteristik	Penjelasan
<i>Usability</i>	<i>Appropriateness Recognizibility</i>	Apakah sistem sudah sesuai dengan kebutuhan pengguna?
	<i>Learnability</i>	Apakah pengguna bisa menggunakan sistem dengan mudah, efisien, efektif, dan merasa puas saat menggunakan sistem pada saat tertentu?
	<i>Operability</i>	Apakah sistem mudah dipahami dan digunakan oleh pengguna?
	<i>User Error Protection</i>	Apakah sistem bisa melindungi pengguna dari kesalahan yang dilakukan oleh pengguna?
	<i>User Interface Aesthetics</i>	Apakah tampilan antarmuka menyenangkan dan memuaskan bagi pengguna pada saat menggunakan sistem?
	<i>Accessibility</i>	Apakah sistem bisa digunakan oleh semua kalangan?

3. Portability

Portability merupakan keefektifan sistem ketika *transfer* dari satu *device* ke *device* lainnya (Gunawan dan Triantoro2, 2017). Penjelasan masing-masing sub-karakteristik dapat dilihat pada tabel 2. 6:

Tabel 2. 6 Penjelasan Sub-karakteristik *Portability* (Gunawan dan Triantoro2, 2017)

Karakteristik	Sub-karakteristik	Penjelasan
<i>Portability</i>	<i>Adaptability</i>	Apakah sistem dapat dipindahkan ke lingkungan lain?
	<i>Installability</i>	Apakah sistem dapat di <i>install</i> pada kondisi tertentu?
	<i>Replaceability</i>	Apakah sistem dapat menggantikan sistem lain untuk tujuan yang sama dan lingkungan yang sama?

2.14. Skala Likert

Skala Likert adalah skala pengukuran yang digunakan untuk mengukur sikap, pendapat, persepsi seseorang terhadap suatu peristiwa sosial (Bahrun, Alifah dan Mulyono, 2017). Menggunakan skala likert, variabel yang diukur diubah menjadi indikator variabel. Selain itu, indikator variabel ini akan digunakan sebagai titik awal untuk pengembangan proyek dalam bentuk pernyataan dan pertanyaan.

Skala likert memiliki ciri-ciri, ada dua jenis pertanyaan yaitu pertanyaan positif dan negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, 1. Sedangkan, untuk pertanyaan negatuf diberi skor 1, 2, 3, 4, 5 (Pranatawijaya *et al.*, 2019).

2.15. Skala Guttmen

Skala Guttmen adalah salah satu skala pengukuran yang digunakan dalam penelitian yang hanya menyediakan dua pilihan jawaban, seperti ya-tidak, sukses-gagal. Untuk jawaban positif maka diberi nilai 1 dan jawaban negative maka diberi nilai 0 (Pranatawijaya *et al.*, 2019).

