

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Tinjauan Pustaka adalah kompilasi dari ringkasan-ringkasan penelitian sebelumnya yang relevan dengan topik penelitian yang sedang dipersiapkan atau dilakukan. Dalam penelitian ini, peneliti menyusun Tinjauan Pustaka dengan memeriksa studi-studi sebelumnya yang relevan, bertujuan untuk memberikan dukungan serta konteks bagi penelitian yang sedang dilakukan. Beberapa penelitian terdahulu yang terkait dan relevan dengan topik penelitian penulis akan dipaparkan untuk memperkaya kerangka pemikiran dan menyusun dasar yang kokoh bagi penelitian yang sedang dilaksanakan.

**Tabel 2. 1** Tabel Literatur

<b>Literatur1</b>	Judul	Rancang Bangun Game Edukasi Pengenalan Nama Dan Suara Hewan Berbasis Multimedia (2020)
	Penulis	Sri Rahayu, Taupik Gunawan
	Metode	WBS
	Hasil	<b>Game edukasi pengenalan nama hewan dan suara hewan genre kuis</b>
<b>Literatur 2</b>	Judul	Pengembangan Media Pembelajaran Digital Berbasis Augmented Reality pada Topik Klasifikasi Hewan Berdasarkan Habitatnya (2020)
	Penulis	Yeni Nursanah, Daniah Adjani Putri
	Metode	SDLC
	Hasil	
<b>Literatur 3</b>	Judul	Game Edukasi Pengenalan Hewan Berdasarkan Habitatnya Untuk Siswa Sekolah Dasar (2021)
	Penulis	Yogiek Indra Kurniawan, Dhenok Prastyaningtyas Paramesvari, Widhiatmoko Herry Purnomo
	Metode	MDLC
	Hasil	Game ber genre Kuis
<b>Literatur 4</b>	Judul	Sistem Informasi Pengenalan Hewan Berbasis Multimedia Interaktif (2023)

	Penulis	Andi Soehartanto Putra, Firma Mukarromah, Usama Waliyyudin, Hersa Dwi Ikhsanti, Fatwa Meilidya, TB. Ahmad Fadhlan Shaquille
	Metode	R&D
	Hasil	<b>Game ber genre puzzle</b>
<b>Literatur 5</b>	Judul	Perancangan Aplikasi Pengenalan Hewan Berdasarkan Tempat Hidupnya Di Raudhatul Athfal (Ra) Amiruddin Mojokerto (2023)
	Penulis	Muhammad Reza, Muh Ariffudin Islam
	Metode	MDLC
	Hasil	<b>Game edukasi ber genre kuis</b>
<b>Literatur 6</b>	Judul	Aplikasi Game Edukasi Android Pengenalan Hewan Untuk Anak Usia Dini Menggunakan Construct 3 (2023)
	Penulis	Roni Saputra, Alif Hayatul Fikri
	Metode	MDLC
	Hasil	<b>Aplikasi Game Pengenalan Hewan Berdasaekan Darat, Air, dan Udara</b>

### 2.1.1 Tinjauan Literatur 1

Berdasarkan penelitian yang dilakukan oleh Sri Rahayu, Taupik Gunawan pada tahun 2020 yang berjudul “Rancang Bangun Game Edukasi Pengenalan Nama Dan Suara Hewan Berbasis Multimedia”. Penelitian tersebut mengangkat sebuah masalah yaitu kurang efektif dan ketertarikan siswa dalam belajar, maka dari itu penulis melakukan sebuah rancang bangun *game* edukasi menggunakan metode WBS (Nurhasanah & Putri, 2020).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi pengenalan tempat bersejarah di Indonesia dengan *genre quiz*.

Menurut penulis aplikasi *game* edukasi ini belum tersedia fitur skor tertinggi pada fitur *game*, Yang bisa menguji tingkat kebenaran dalam menjawab *game*.

### 2.1.2 Tinjauan Literatur 2

Berdasarkan penelitian yang dilakukan oleh Yeni Nursanah, Daniah Adjani Putri pada tahun 2020 yang berjudul “Pengembangan Media Pembelajaran Digital Berbasis Augmented Reality pada Topik Klasifikasi Hewan Berdasarkan Habitatnya”.

Penelitian tersebut mengangkat sebuah masalah pada Kurikulum Ilmu Pengetahuan Alam pada jenjang Sekolah Dasar pada kelas empat, maka dari itu penulis melakukan pengembangan media pembelajaran menggunakan metode SLDC (Rahayu & Gunawan, 2020).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi Hewan pada habitatnya dengan *genre AR*.

Menurut penulis masih terdapat kekurangan pada literatur ini yaitu penggunaan marker dengan menggunakan metode markerless sangat layak dicoba untuk meningkatkan pengalaman dan kegunaan dari aplikasi berbasis AR.

### **2.1.3 Tinjauan Literatur 3**

Berdasarkan penelitian yang dilakukan oleh Yogiek Indra Kurniawan, Dhenok Prastyaningtyas Paramesvari, Widhiatmoko Herry Purnomo pada tahun 2021 yang berjudul “Game Edukasi Pengenalan Hewan Berdasarkan Habitatnya Untuk Siswa Sekolah Dasar”. Penelitian tersebut mengangkat sebuah masalah yaitu dalam penyampaian materi pelajaran, khususnya pada pengenalan hewan berdasarkan habitatnya untuk siswa sekolah dasar, maka dari itu penulis membuat sebuah aplikasi *game* edukasi berbasis *augmented reality* dengan menggunakan metode ADDIE dalam pembelajaran khususnya topik seni ukiran kayu (Kurniawan et al., 2021).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi berbasis android dengan *genre quiz*.

### **2.1.4 Tinjauan Literatur 4**

Berdasarkan penelitian yang dilakukan oleh Andi Soehartanto Putra, Firma Mukarromah, Usama Waliyyudin, Hersa Dwi Ikhsanti, Fatwa Meilidya, TB. Ahmad

Fadhlan Shaquille pada tahun 2023 yang berjudul “Sistem Informasi Pengenalan Hewan Berbasis Multimedia Interaktif”. Penelitian tersebut mengangkat sebuah masalah pada TK Islam Bakti 113 yaitu guru yang mengalami kesulitan dalam mengajar dikarenakan banyaknya murid yang kurang memperhatikan proses pembelajaran, maka dari itu penulis melakukan pembuatan *game* edukasi dengan metode *waterfall* (Saputra & Fikri, 2023).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi pengenalan huruf alfabet dengan *genre arcade*.

### **2.1.5 Tinjauan Literatur 5**

Berdasarkan penelitian yang dilakukan oleh Muhammad Reza, Muh Ariffudin Islam pada tahun 2023 yang berjudul “Perancangan Aplikasi Pengenalan Hewan Berdasarkan Tempat Hidupnya Di Raudhatul Athfal (Ra) Amiruddin Mojokerto”. Penelitian tersebut mengangkat sebuah masalah dalam proses pembelajaran yang dianggap membosankan bagi anak-anak dikarenakan anak-anak lebih suka bermain daripada belajar, maka dari itu penulis melakukan pembuatan *game* edukasi dengan metode MDLC (muhammad reza, 2023).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi hewan pada habitanya dengan *genre quiz*.

Menurut penulis masih terdapat kekurangan pada literatur ini yaitu mengembangkan dan menambahkan sub tema seperti hewan peliharaan, hewan ternak, dan membuat media yang lebih menarik agi seperti menambahkan motion graphic, 3d, dan dubbing suara.

### 2.1.6 Tinjauan Literatur 6

Berdasarkan penelitian yang dilakukan oleh Roni Saputra, Alif Hayatul Fikri pada tahun 2023 yang berjudul “Aplikasi Game Edukasi Android Pengenalan Hewan Untuk Anak Usia Dini Menggunakan Construct 3”. Penelitian tersebut mengangkat sebuah masalah pada banyak anak yang masih belum mengenal hewan dan habitatnya, maka dari itu penulis melakukan pembuatan *game* edukasi dengan menggunakan metode R&D (Putra, Mukarromah, Waliyyudin, Ikhsanti, et al., 2023).

Hasil dari peneliti adalah sebuah aplikasi *game* edukasi hewan pada habitatnya dengan *genre quiz*.

Berdasarkan dari studi literatur yang ada penulis mengambil kesimpulan yaitu terdapat perbedaan pada beberapa dari penelitian yaitu identifikasi masalah dan tempat studi kasus yang akan dilakukan oleh penulis serta terdapat perbedaan dengan desain hasil aplikasi *game* edukasi yang dihasilkan dari penelitian yang akan dilakukan oleh penulis.

## 2.2 Hewan

Hewan, yang dalam bahasa Latin disebut *Animalia*, merupakan salah satu dari kerajaan makhluk hidup dalam klasifikasi biologi di bawah *Domain Eukarya*. Menurut yang dikutip dari jurnal (Putra, Mukarromah, Waliyyudin, Dwi Ikhsanti, et al., 2023) Dalam bahasa Inggris, istilah "hewan" diterjemahkan sebagai "animal," yang berasal dari bahasa Latin "animalis" yang berarti memiliki napas. Dalam penggunaan sehari-hari yang tidak resmi, kata tersebut biasanya merujuk pada makhluk hidup selain manusia. Hewan merujuk kepada binatang atau satwa yang menghabiskan sebagian atau seluruh siklus kehidupannya di darat, air, dan/atau udara, baik itu dilindungi, dipelihara, atau berada dalam habitat alaminya.

Hewan darat, juga dikenal sebagai hewan terestrial, merujuk pada makhluk hidup yang sebagian besar atau seluruhnya berada di daratan atau lingkungan terestrial. Contoh hewan darat mencakup kucing, sapi, lipan, dan laba-laba. Hal ini berbeda dengan hewan akuatik atau hewan yang berhabitat di air, yang mayoritas atau seluruhnya hidup di dalam air, seperti ikan, lobster, dan gurita, serta amfibi, yang hidup dengan memanfaatkan habitat baik di air maupun di darat, seperti katak atau salamander. Sebagian besar serangga termasuk dalam kategori hewan darat, seperti semut, kupu-kupu, belalang, dan kecoak, namun ada juga serangga semi-akuatik seperti capung yang fase larvanya hidup di dalam air.

**Tabel 2. 2** Contoh Hewan Berdasarkan Habitat

No	Contoh Hewan	Habitat
1	Kupu-Kupu, burung, lebah, cacing dan rayap, darat seperti kucing, gajah dan singa.	Darat
2	Ikan, udang, cumi-cumi dan gurita.	Air
3	Katak, buaya dan kura-kura	Darat & Air

### 2.3 Game Edukasi

Game Edukasi merupakan jenis permainan yang diciptakan dengan tujuan untuk meningkatkan pemahaman atau pembelajaran dengan cara yang menyenangkan, sehingga proses belajar tidak terasa membosankan. Sebuah penelitian menghasilkan sebuah aplikasi game edukasi dengan genre puzzle yang fokus pada pembelajaran huruf-huruf alfabet. Komponen-komponen yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

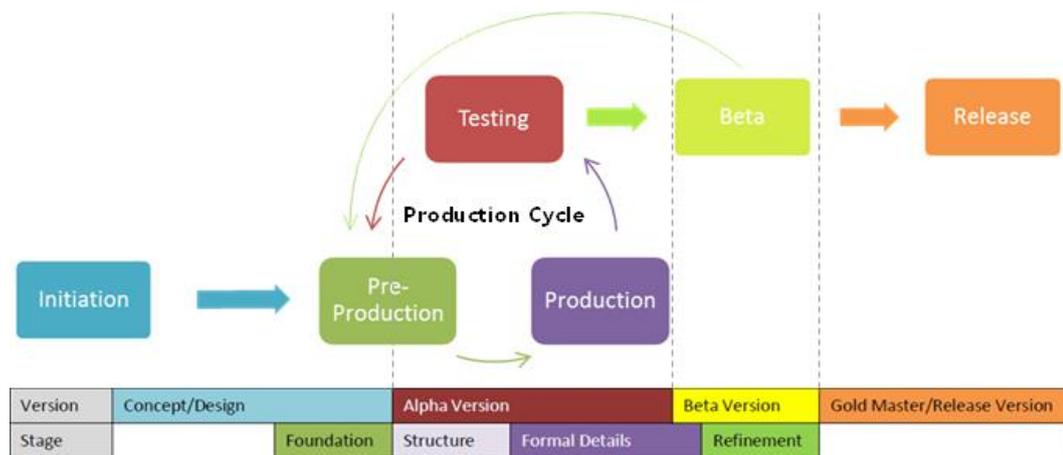
- a. *Game Rule* : adalah aturan perintah, cara menjalankan, fungsi objek dan karakter permainan di dunia *game*.
- b. *Character* : pengguna sebagai karakter utama maupun karakter lainnya yang mempunyai ciri dan sifat tertentu.

- c. *Object* : adalah sesuatu yang penting dan digunakan sebagai pemain agar dapat memecahkan suatu permasalahan, pemain diharapkan harus mempunyai keahlian dan pengetahuan untuk dapat memainkannya.
- d. *Text, Graphic* dan *Sound* : merupakan kombinasi dari berbagai sumber, baik dari media teks, grafik dan suara, walaupun tidak harus semuanya di dalam sebuah *game*.
- e. *User Interface* : merupakan fitur-fitur yang mengkomunikasikan *user* dengan *game*.
- f. *Score Model* : adalah *instrument* yang dapat digunakan untuk menghitung jumlah atau mendata dan menampilkan hasil dari pemain Ketika bermain sebuah *game*. Elemen ini menjadi suatu alat yang penting dikarenakan menjadi suatu acuan untuk si pemain dalam menyelesaikan tantangan.

## 2.4 Metode GDLC

*Game Development Life Cycle* (GDLC) adalah metodologi yang sangat penting dalam industri game, digunakan untuk mengelola dan mengarahkan setiap aspek dari proses pengembangan game, mulai dari ide awal hingga produk akhir. Dengan menerapkan GDLC, pengembang dapat memastikan bahwa setiap tahap dalam proses pengembangan dilakukan secara sistematis dan terstruktur. Ini mencakup berbagai fase mulai dari perencanaan awal, desain, dan pengembangan, hingga pengujian dan peluncuran. GDLC membantu tim pengembang dalam mengidentifikasi dan mengatasi potensi masalah lebih awal, serta memastikan bahwa proyek tetap sesuai dengan anggaran dan jadwal yang ditetapkan. Dengan pendekatan yang terorganisir ini, GDLC berfungsi sebagai panduan yang efektif untuk mencapai hasil akhir yang berkualitas dan memenuhi ekspektasi pengguna.

Melalui penerapan GDLC, setiap elemen dalam game dapat dikembangkan dengan lebih terarah dan efisien, menghasilkan produk akhir yang lebih baik. (Mufida et al., 2021). Metode *Game Development Life Cycle* (GDLC) memiliki beberapa tahapan yaitu initiation, pre-production, production, testing, release. Fase/tahapan dari GDLC dapat dilihat pada gambar 2.1 berikut ini:



**Gambar 2. 1** Fase GDLC

Sumber: (Mufida et al., 2021)

- Tahap *Initiation*, Pada tahap ini, dilakukan identifikasi masalah utama dan peneliti melakukan tinjauan literatur terhadap beberapa jurnal penelitian yang relevan.
- Tahap *Pre-Production*, Di tahap ini, peneliti mengidentifikasi alur permainan yang akan dikembangkan serta menentukan jenis atau genre game yang akan dihasilkan.
- Tahap *Production*, Pada tahap ini, peneliti merancang aplikasi dan mengimplementasikan game edukasi yang telah direncanakan.
- Tahap *Testing*, Pada tahap ini, dilakukan pengujian untuk mengetahui apakah ada kesalahan saat aplikasi dijalankan.

- e. Tahap *Release*, Tahap akhir ini merupakan tahap di mana aplikasi tidak mengalami kesalahan saat dijalankan dan dilakukan peluncuran aplikasi.

## 2.5 *Unified Modeling Language (UML)*



**Gambar 2. 2** *Logo Unified Modeling Language*

UML, atau *Unified Modeling Language*, adalah bahasa pemodelan standar yang terdiri dari serangkaian diagram terintegrasi, yang dikembangkan untuk membantu pengembangan sistem dan perangkat lunak. UML digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak. Selain itu, UML juga bermanfaat dalam pemodelan bisnis dan sistem non-perangkat lunak lainnya. Dengan menyediakan berbagai jenis diagram seperti diagram use case, diagram kelas, dan diagram urutan, UML memfasilitasi komunikasi yang efektif antara pengembang dan pemangku kepentingan, serta membantu memastikan bahwa semua aspek sistem terdefinisi dengan baik dan konsisten (Charles et al., 2022). Pada penelitian ini penulis menggunakan diagram UML diantaranya sebagai berikut:

1. *Use Case Diagram*, memodelkan proses bisnis.
2. *Activity Diagram*, memodelkan perilaku *use case* dan *objects* di dalam *system*.

### 2.5.1 Use Case Diagram

*Use Case* adalah model pemodelan untuk perilaku sistem informasi yang akan dibangun. Mereka menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem itu sendiri. Fungsinya adalah untuk mengidentifikasi dan menggambarkan fungsi-fungsi yang dapat diakses oleh aktor-aktor tersebut. Dengan menggunakan *Use Case*, pengembang dan analis sistem dapat memahami secara detail bagaimana setiap aktor berinteraksi dengan sistem, serta fungsi apa saja yang tersedia dan bagaimana mereka saling terkait. Ini membantu memastikan bahwa sistem yang dikembangkan akan memenuhi kebutuhan pengguna secara efektif dan efisien. Simbol *use case* dapat dilihat sebagai berikut:

**Tabel 2. 3** Tabel Simbol *Use Case*

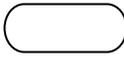
No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.
2	<i>Extends</i>		Relase <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
3	<i>Include</i>		Relasi <i>use case</i> tambahan kesebuah <i>use case</i> yang ditambahkan melalui <i>use case</i> ini untuk menjalankan fungsinya untuk sebagai syarat dijalankan <i>use case</i> ini.
4	<i>Use case</i>		Menjelaskan yang dilakukan aktor dari sistem untuk mencapai tujuan tertentu.
5	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.
6	<i>System</i>		Menspesifikasikan data paket yang menampilkan system secara terbatas

Sumber : (Dias & Muhallim, 2022)

### 2.5.2 Activity Diagram

*Activity diagram* atau diagram aktivitas adalah alat visual yang digunakan untuk menggambarkan workflow atau aliran kerja dalam sebuah sistem, proses bisnis, atau menu yang ada dalam perangkat lunak. Penting untuk dicatat bahwa diagram aktivitas menggambarkan aktivitas yang dilakukan oleh sistem itu sendiri, bukan aktivitas yang dilakukan oleh aktor atau pengguna. Hal ini memungkinkan para pengembang untuk memahami secara jelas bagaimana berbagai aktivitas dalam sistem berinteraksi dan saling terkait, sehingga mempermudah analisis dan pengembangan untuk mencapai efisiensi dan tujuan yang diinginkan dalam desain perangkat lunak tersebut. Simbol *activity diagram* dapat dilihat pada tabel berikut:

**Tabel 2. 4** Tabel Simbol *Activity Diagram*

No	Nama	Simbol	Keterangan
1	<i>Activity</i>		Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2	<i>Action</i>		State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3	<i>Initial Node</i>		Bagaimana objek dibentuk atau diawali.
4	<i>Activity Final Node</i>		Bagaimana objek dibentuk dan dihancurkan
5	<i>Fork Node</i>		Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : (Dias & Muhallim, 2022)

### 2.6 Flowchart

Flowchart adalah gambaran grafis dari langkah-langkah dan urutan prosedur suatu program, yang membantu analisis dalam membagi masalah menjadi bagian-

bagian yang lebih kecil dan mendukung analisis terhadap berbagai alternatif operasional. (Ridlo, 2017). Flowchart adalah diagram grafis yang menggambarkan aliran proses dengan menggunakan bentuk-bentuk geometris seperti lingkaran, persegi, wajik, oval, dan sebagainya untuk mengilustrasikan langkah-langkah dan urutannya, dengan menghubungkan setiap simbol tersebut menggunakan panah, Menurut Tague (2005), tujuan digunakannya *flowchart* antara lain:

- a. Untuk mngembangkan pemahaman tentang bagaimana proses dilakukan.
- b. Untuk mempelajari perbaikan proses.
- c. Untuk berkomunikasi dengan orang lain bagaimana proses dilakukan.
- d. Untuk keperluan komunikasi yang lebih baik di antara orang-orang yang terlibat dalam proses yang sama.
- e. Untuk mendokumentasikan proses.
- f. Untuk merencanakan sebuah kegiatan.

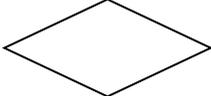
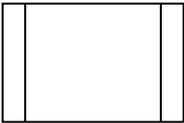
Flowchart dapat dikelompokkan menjadi beberapa jenis berdasarkan fungsi, proses, dan tingkat kepentingan pengguna. Ada lima jenis flowchart, antara lain:

- a. Flowchart Sistem, yang menggambarkan alur kerja dan prosedur keseluruhan dari suatu sistem.
- b. Flowchart Paperwork, yang melacak alur data yang ditulis melalui sistem, termasuk proses pengisian formulir dan penyimpanan laporan.
- c. Flowchart Skematik, mirip dengan flowchart sistem, yang menggambarkan sistem atau prosedur secara visual.
- d. Flowchart Program, yang merupakan versi rinci dari flowchart sistem, menjelaskan langkah-langkah program atau prosedur secara lebih terperinci.

- e. Flowchart Proses, teknik penggambaran rekayasa industri yang memecah dan menganalisis langkah-langkah dalam prosedur atau sistem tertentu.

Simbol-simbol *flowchart* yang biasanya dipakai adalah simbol-simbol *flowchart* standar yang dikeluarkan oleh ANSI dan ISO. Simbol-simbol *flowchart* dapat dilihat pada tabel berikut ini:

**Tabel 2. 5** Tabel Simbol Flowchart

Simbol	Nama	Arti
	Input/Output	Merepresentasikan input data atau output data yang diproses atau informasi.
	Proses	Mempresentasikan operasi
	Penghubung	Keluar ke atau masuk dari bagian lain <i>flowchart</i> khususnya halaman yang sama
	Anak Panah	Merepresentasikan alur kerja
	Keputusan	Keputusan dalam program
	<i>Terminal Points</i>	Awal/Akhir <i>flowchart</i>
	<i>Predefined Process</i>	Rincian operasi berada ditempat lain
	<i>Punched Card</i>	Input/Output yang menggunakan kartu berlubang
	Dokumen	Input/Output dalam format yang dicetak

Sumber : (Ridlo, 2017)

## 2.7 CorelDraw X7



**Gambar 2.3** Logo CorelDraw X7

*CorelDRAW* merupakan sebuah aplikasi desain grafis digital berbasis vektor yang fokus pada penciptaan karya 2D. Dengan kekuatan aplikasi ini, pengguna dapat dengan mudah menciptakan berbagai macam desain seperti logo, flex, brosur, kartu undangan, dan dokumen desain lainnya. Keunggulan utama *CorelDRAW* terletak pada kemudahannya dalam membuat desain-desain tersebut dengan beragam fitur yang intuitif dan efisien.

Selain kemudahan penggunaannya, *CorelDRAW* juga menawarkan beberapa keunggulan yang membuatnya unggul dibandingkan dengan aplikasi desain lainnya. Salah satunya adalah kemampuannya dalam mengolah vektor, yang memberikan hasil desain yang tajam dan skalabilitas yang tinggi tanpa kehilangan kualitas. Selain itu, *CorelDRAW* juga dilengkapi dengan berbagai fitur canggih seperti alat perspektif, pembuatan efek khusus, dan dukungan untuk berbagai format file yang membuatnya menjadi pilihan yang sangat fleksibel untuk kebutuhan desain profesional.

## 2.8 Unity



**Gambar 2. 4** Logo *Unity*

*Unity Game Engine* adalah perangkat lunak yang memungkinkan pembuatan video *game* berbasis dua atau tiga dimensi, dan dapat diakses secara gratis. Selain untuk menciptakan *game*, *Unity* juga mendukung pembuatan konten interaktif lainnya seperti visual arsitektur, animasi 3D *real-time*, *augmented reality*, dan *virtual reality*. Pengguna dapat mengunduh *Unity Game Engine* secara gratis dengan mengunjungi situs resmi *Unity* dan menginstal *Unity Hub*, yang kemudian memungkinkan pengguna untuk memasang *Unity Long Term Support* sesuai dengan kebutuhan versi yang diinginkan. Prosedur ini mempermudah akses dan penggunaan *Unity Game Engine* bagi para pengembang dan kreator konten interaktif.

*Unity Game Engine* tidak hanya menyediakan alat untuk pembuatan video *game*, tetapi juga menjadi *platform* yang kuat untuk menghasilkan beragam konten interaktif, mulai dari simulasi arsitektur hingga pengalaman *virtual reality*. Melalui *Unity Hub*, pengguna dapat mengelola instalasi dan versi perangkat lunak secara efisien sesuai dengan proyek yang sedang dikerjakan. Dengan tersedianya *Unity Long Term Support*, pengguna dapat memastikan bahwa mereka menggunakan versi perangkat lunak yang stabil dan didukung secara optimal oleh tim

pengembang *Unity*, sehingga memberikan pengalaman pengembangan yang lebih lancar dan andal.

## 2.9 Skala Likert

Skala Likert adalah metode yang digunakan untuk mengukur atau menilai pendapat dari responden terhadap suatu pernyataan. Angket yang menggunakan skala Likert ini umumnya terdiri dari serangkaian pernyataan yang disertai dengan beberapa pilihan jawaban. Jawaban yang diberikan oleh responden kemudian akan dihitung menggunakan skor tertentu. Dengan demikian, skala Likert memungkinkan peneliti untuk mengukur tingkat setuju atau tidak setuju responden terhadap pernyataan yang diberikan.

Dalam penerapannya, skala Likert menawarkan dua jenis pertanyaan utama yang dapat digunakan dalam menyusun kuesioner, yaitu pertanyaan positif dan negatif. Pertanyaan positif mengarah pada aspek-aspek yang dianggap menguntungkan atau diinginkan, sementara pertanyaan negatif menyoroti aspek-aspek yang dianggap merugikan atau tidak diinginkan. Dengan adanya kedua jenis pertanyaan ini, skala Likert memungkinkan peneliti untuk mendapatkan gambaran yang lebih komprehensif tentang pendapat dan sikap responden terhadap suatu topik atau pernyataan yang diajukan (Retnawati 2015), yang akan dijelaskan sebagai berikut:

- a. Pertanyaan positif
  1. Skor 5 untuk Sangat (Setuju/Baik)
  2. Skor 4 untuk (Setuju/Baik)
  3. Skor 3 untuk (Netral/Cukup)
  4. Skor 2 untuk Tidak (Setuju/Baik)

5. Skor 1 untuk Sangat Tidak (Setuju/Baik)

b. Pertanyaan negative

1. Skor 5 untuk Sangat Tidak (Setuju/Baik)

2. Skor 4 untuk Tidak (Setuju/Baik)

3. Skor 3 untuk (Netral/Cukup)

4. Skor 2 untuk (Setuju/Baik)

5. Skor 1 untuk Sangat (Setuju/Baik)

Total skor diperoleh dari penilaian dalam setiap kategori, kemudian dihitung menggunakan rumus tertentu untuk menghasilkan persentase. Persentase tersebut memiliki kategori yang menunjukkan hasil dari persentase untuk suatu pertanyaan tertentu. Berikut adalah kategori hasil persentase:

- a. Jika persentase berada pada rentang 0% - 19,999%, maka penilaiannya adalah "Sangat Tidak (Setuju/Baik)".
- b. Jika persentase berada pada rentang 20% - 39,999%, maka penilaiannya adalah "Tidak (Setuju/Baik)".
- c. Jika persentase berada pada rentang 40% - 59,999%, maka penilaiannya adalah "(Cukup/Netral)".
- d. Jika persentase berada pada rentang 60% - 79,999%, maka penilaiannya adalah "(Setuju/Baik)".
- e. Jika persentase berada pada rentang 80% - 100%, maka penilaiannya adalah "Sangat (Setuju/Baik)".

## 2.10 Metode Pengujian



**Gambar 2. 5** Gambar ISO 25010

Sumber: (ISO 25010, n.d.)

Berdasarkan pada (ISO 25010, n.d.) Model kualitas ISO/IEC 25010 bertindak sebagai kerangka evaluasi untuk menilai kualitas produk, memutuskan aspek-aspek kualitas mana yang akan menjadi fokus saat mengevaluasi properti perangkat lunak. Kualitas sebuah sistem diukur berdasarkan sejauh mana sistem tersebut memenuhi kebutuhan yang terungkap dan tersirat dari berbagai pemangku kepentingan. Kriteria-kriteria tersebut, seperti fungsionalitas, kinerja, keamanan, pemeliharaan, dan lainnya, sesuai dengan representasi dalam model kualitas yang membagi kualitas produk menjadi karakteristik dan sub-karakteristik yang relevan. ISO/IEC 25010 menetapkan delapan karakteristik kualitas utama yang membentuk dasar evaluasi, yang disajikan dalam format visual untuk memandu pemahaman dan implementasi lebih lanjut.

Pemilihan karakteristik dan sub-karakteristik kualitas yang relevan dalam model ISO/IEC 25010 sangat penting dalam memastikan evaluasi kualitas produk yang komprehensif dan akurat. Karakteristik kualitas tersebut mencakup aspek-aspek seperti fungsionalitas, kinerja, kompatibilitas, keamanan, dan lainnya yang secara langsung mempengaruhi pengalaman pengguna dan keberhasilan implementasi sistem. Dengan menggunakan model kualitas ini, para pemangku

kepentingan dapat mengidentifikasi prioritas dalam pengembangan perangkat lunak, memastikan bahwa produk memenuhi standar yang diinginkan, serta menyediakan panduan yang jelas untuk evaluasi kualitas yang terperinci dan efektif. Model kualitas produk yang didefinisikan dalam ISO/IEC 25010 terdiri dari delapan karakteristik kualitas yang ditunjukkan pada gambar berikut:

1. *Functional Suitability*, Karakteristik ini mewakili sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Functional completeness* - Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
  - b. *Functional correctness* - Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang diperlukan.
  - c. *Functional appropriateness* - Sejauh mana fungsi-fungsi tersebut memfasilitasi pencapaian tugas dan tujuan tertentu.
2. *Performance efficiency*, Karakteristik ini mewakili kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Time behaviour* - Sejauh mana respons dan waktu pemrosesan serta tingkat keluaran suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.
  - b. *Resource utilization* - Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.

- c. *Capacity* - Sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.
3. *Compatibility*, Sejauh mana suatu produk, sistem, atau komponen dapat bertukar informasi dengan produk, sistem, atau komponen lain, dan/atau menjalankan fungsi yang diperlukan sambil berbagi lingkungan perangkat keras atau perangkat lunak yang sama. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Co-existence* - Sejauh mana suatu produk dapat menjalankan fungsi-fungsi yang diperlukan secara efisien sambil berbagi lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak merugikan pada produk lain.
  - b. *Interoperability* - Sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.
4. *Usability*, Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Appropriateness recognizability* – Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.
  - b. *Learnability* - Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran tertentu untuk menggunakan produk atau sistem dengan efektif, efisien, bebas dari risiko dan kepuasan dalam konteks penggunaan tertentu.

- c. *Operability* - Sejauh mana suatu produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.
  - d. *User error protection* - Sejauh mana suatu sistem melindungi pengguna dari kesalahan.
  - e. *User interface aesthetics* - Sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
  - f. *Accessibility* - Sejauh mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu.
5. *Reliability*, Sejauh mana suatu sistem, produk, atau komponen menjalankan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
- a. *Maturity* - Sejauh mana suatu sistem, produk atau komponen memenuhi kebutuhan keandalan dalam operasi normal. *Ketersediaan* - Sejauh mana suatu sistem, produk atau komponen dapat beroperasi dan dapat diakses ketika diperlukan untuk digunakan.
  - b. *Fault tolerance* - Sejauh mana sistem, produk, atau komponen beroperasi sebagaimana mestinya meskipun terdapat kesalahan perangkat keras atau perangkat lunak.
  - c. *Recoverability* - Sejauh mana, jika terjadi gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terkena dampak langsung dan memulihkan kondisi sistem yang diinginkan.

6. *Security*, Sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga orang atau produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasinya. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Confidentiality* - Sejauh mana suatu produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
  - b. *Integrity* - Sejauh mana sistem, produk, atau komponen mencegah akses tidak sah, atau modifikasi, program atau data komputer.
  - c. *Non-repudiation* - Sejauh mana suatu tindakan atau peristiwa dapat dibuktikan telah terjadi sehingga peristiwa atau tindakan tersebut tidak dapat disangkal lagi di kemudian hari.
  - d. *Accountability* - Sejauh mana tindakan suatu entitas dapat ditelusuri secara unik ke entitas tersebut.
  - e. *Authenticity* - Sejauh mana identitas subjek atau sumber daya dapat dibuktikan sesuai dengan yang diklaim.
7. *Maintanability*, Karakteristik ini mewakili tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat dimodifikasi untuk memperbaikinya, memperbaikinya atau menyesuaikannya dengan perubahan lingkungan, dan persyaratan. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Modularity* - Sejauh mana suatu sistem atau program komputer terdiri dari komponen-komponen terpisah sedemikian rupa sehingga perubahan pada satu komponen mempunyai dampak minimal pada komponen lainnya.

- b. *Reusability* - Sejauh mana suatu aset dapat digunakan di lebih dari satu sistem, atau dalam membangun aset lain.
  - c. *Analysability* - Tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari perubahan yang direncanakan pada satu atau lebih bagian-bagiannya, atau untuk mendiagnosis kekurangan atau penyebab kegagalan produk, atau untuk mengidentifikasi bagian-bagian yang perlu dianalisa. dimodifikasi.
  - d. *Modifiability* - Sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan cacat atau menurunkan kualitas produk yang ada.
  - e. *Testability* - Tingkat efektivitas dan efisiensi kriteria pengujian yang dapat ditetapkan untuk suatu sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah dipenuhi.
8. *Portability*, Tingkat efektivitas dan efisiensi sistem, produk, atau komponen yang dapat ditransfer dari satu perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya ke lingkungan lain. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
- a. *Adaptability* - Sejauh mana suatu produk atau sistem dapat diadaptasi secara efektif dan efisien untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya yang berbeda atau terus berkembang.
  - b. *Installability* - Tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat berhasil diinstal dan/atau dihapus instalasinya dalam lingkungan tertentu.

- c. *Replaceability* - Sejauh mana suatu produk dapat menggantikan produk perangkat lunak lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.