BAB I. PENDAHULUAN

1.1 Latar Belakang

Peduli terhadap lingkungan dapat diartikan sebagai sikap serta tindakan yang selalu mengupayakan pencegahan terhadap kerusakan alam di lingkungan sekitar, serta selalu mengupayakan untuk memperbaiki kerusakan yang telah terjadi pada alam. Karakter peduli lingkungan dapat dikatakan yaitu suatu sikap yang harus dimiliki oleh setiap orang untuk selalu mengupayakan perbaikan dan mengelola lingkungan hidup sekitar dengan baik dan benar, sehingga lingkungan dapat selalu dinikmati secara terus menerus tanpa merusaknya, serta menjaga dan melestarikan alam sekitar agar dapat bermanfaat untuk saat ini dan untuk masa yang akan datang (Purwanti, 2017).

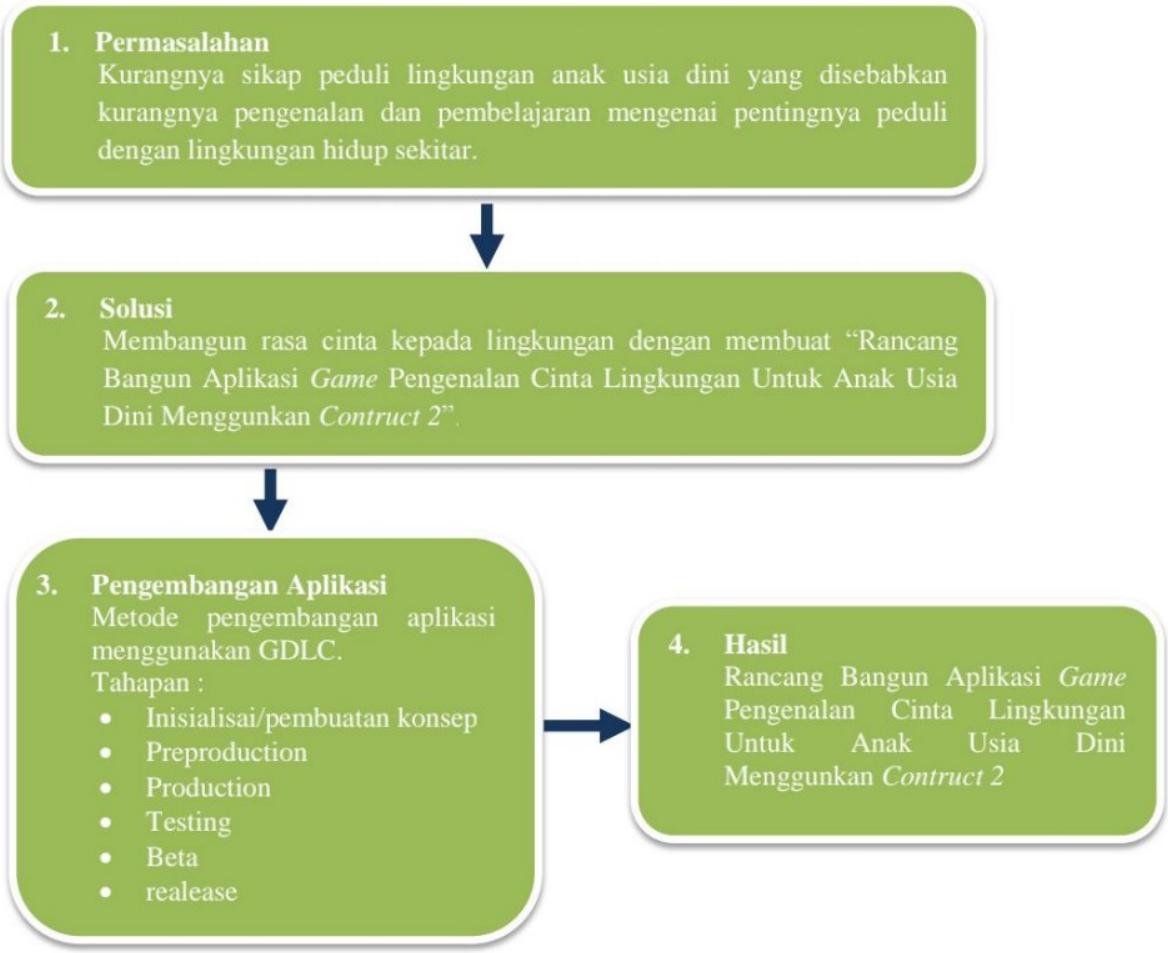
Seseorang yang tidak memiliki rasa terhadap pedulinya lingkungan, akan menimbulkan permasalahan terhadap kelestarian lahan hijau yang saat ini banyak disalah gunakan untuk pembangunan pemukiman warga, sehingga mudahnya terjadi bencana alam seperti banjir dan tanah longsor karena tidak adanya penyerapan air ketika hujan turun. Ketidak pedulian itu dapat dilihat dari sangat banyaknya lahan hijau seperti hutan, perkebunan, dan sawah yang saat ini beralih fungsi menjadi perkantoran, perumahan, sarana rekreasi, dan lainnya. Upaya untuk mengatasi permasalahan lingkungan tersebut salah satunya yaitu membentu karakter cinta lingkungan sejak usia dini. Dalam pembentukan karakter tersebut dilakukan melalui pembelajaran tentang lingkungan hidup sekitar. Sehingga dengan adanya

pembelajaran tersebut diharapkan agar anak-anak dapat memiliki karakter yang mencerminkan peduli terhadap alam dan lingkungan sekitarnya (Ismail, 2021).

Upaya yang harus dilakukan untuk mencegah terjadinya masalah lingkungan di masa depan yaitu dengan cara meningkatkan kualitas lingkungan serta menumbuhkan sikap dan karakter peduli lingkungan terhadap anak-anak usia dini. Seseorang yang memiliki sikap peduli lingkungan dalam kehidupan masyarakat biasanya melakukan tindakan seperti tidak merusak lingkungan, melestarikan, mencegah, dan memperbaiki lingkungan alam (Adriansyah et al., 2019). Sejalan dengan pendapat tersebut (Tamara, 2016) menjelaskan bahwa seseorang yang selalu bersikap peduli terhadap lingkungan biasanya dalam kehidupan sehari-hari atau bermasyarakat diartikan sebagai tindakan seseorang terhadap lingkungannya dengan menciptakan lingkungan yang bersih dan asri. Menurut (Sabardila et al., 2020) cara untuk menumbuhkan rasa peduli terhadap lingkungan salah satunya yaitu dengan cara melakukan penanaman atau penghijauan di sekitar lingkungan. Tindakan tersebut dalam arti luas yaitu segala sesuatu yang memulihkan, memelihara, dan meningkatkan kondisi lahan sehingga dapat berproduksi dan berfungsi secara optimal, baik untuk pengatur tata air atau pelindung lingkungan.

Anak usia dini merupakan masa individu yang unik, pada masa ini anak disebut dengan masa *Golden Age* karena sedang dalam proses pertumbuhan serta perkembangan. Anak usia dini berada pada rentang umur 0-8 tahun dan sedang menjalani proses perkembangan yang sangat pesat dan fundamental bagi kehidupan selanjutnya(Wiwik Pratiwi, 2017).

dengan berdasarkan pertanyaan penelitian (*research question*), dan menjelaskan suatu himpunan dari beberapa konsep serta hubungan di antara konsep-konsep tersebut Polancik (dalam Mardah, 2021). Masalah yang muncul dari kurangnya sikap peduli lingkungan anak usia dini yang disebabkan kurangnya pengenalan dan pembelajaran mengenai pentingnya peduli dengan lingkungan hidup sekitar. Solusi yang diberikan yaitu membangun rasa cinta kepada lingkungan dengan membuat *"Rancang Bangun Aplikasi Game Pengenalan Cinta Lingkungan Untuk Anak Usia Dini Menggunakan Construct 2".* Pembuatan aplikasi ini dilakukan dengan metode pengembngan sistem *Game Development Life Cycle (GDLC)* yang terdiri dari 6 fase pengembangan yaitu inisialisasi/pembuatan konsep, *preproduction*, *production*, *testing*, *beta* dan *realease*. Berdasarkan dari latar belakang dapat disusun suatu kerangka pemikiran yang disajikan dalam gambar berikut.



Gambar 1.1 Kerangka Pemikiran

1.4. Kontribusi

Adanya Rancang Bangun Aplikasi *Game* Pengenalan Cinta Lingkungan Untuk Anak Usia Dini Menggunakan *Construct 2,* sebagai berikut :

1. Diharapkan tidak ada lagi kerusakan alam pada lingkungan sekitar,
2. Membangun rasa cinta terhadap anak usia dini untuk peduli terhadap lingkungan sekitar
3. Menimbulkan rasa peduli untuk menjaga alam dari kerusakan.

BAB II. TINJAUAN PUSTAKA

2.1 Pentingnya Menjaga Lingkungan Hidup

Peduli terhadap lingkungan dapat diartikan sebagai sikap serta tindakan yang selalu mengupayakan pencegahan terhadap kerusakan alam di lingkungan sekitar, serta selalu mengupayakan untuk memperbaiki kerusakan yang telah terjadi pada alam. Karakter peduli lingkungan dapat dikatakan yaitu suatu sikap yang harus dimiliki oleh setiap orang untuk selalu mengupayakan perbaikan dan mengelola lingkungan hidup sekitar dengan baik dan benar, sehingga lingkungan dapat selalu dinikmati secara terus menerus tanpa merusaknya, serta menjaga dan melestarikan alam sekitar agar dapat bermanfaat untuk saat ini dan untuk masa yang akan datang (Purwanti, 2017).

Kesadaran akan pentingnya menjaga lingkungan masih sangat rendah. Banyak anak-anak, remaja bahkan orang dewasa yang masih membuang sampah sembarangan. Kesadaran akan pentingnya menjaga lingkungan tersebut harus ditanamkan dari sejak dini. Upaya yang harus dilakukan dengan adanya kebijakan pendidikan karakter di sekolah, sehingga kesadaran tersebut dapat terwujudkan. Berkaitan dengan perilaku yang cenderung tidak peduli lingkungan maka merubah perilaku tersebut menjadi prioritas utama dalam mengatasi krisis lingkungan. Kepedulian terhadap lingkungan dapat dianggap sebagai suatu bentuk perhatian terhadap fakta-fakta dan perilaku dari diri sendiri dengan konsekuensi tertentu untuk menjaga lingkungan di sekitar kita (Efendi, 2020).

2.2 *Game* dan Teknologi Pendukung

Game merupakan permainan yang terkomputerisasi yang dibuat menggunakan teknik dan metode animasi. *Game* sendiri merupakan media hiburan yang banyak dipilih anak-anak untuk menghilangkan kebosanan atau hanya untuk sekedar mengisi waktu luang saja. Selain itu juga *game* dapat menjadi sebuah media belajar untuk meningkatkan perkembangan otak dalam daya motorik, afeksi, kognitif, spiritual, dan keseimbangan sehingga dapat mencerdaskan kemampuan otak anak-anak (Amami Pramuditya et al., 2017). Dalam proses pembuatan *game* memerlukan *software* sebagai teknologi pendukung salah satunya yaitu *Game Builder Construct 2.*

Game Builder Construct 2 dibuat untuk merancang *game* berbasis 2D. Dengan menggunakan *software Construct 2* pengembang permainan dapat pengembangkan dan permainan dapat diterbitkan ke beberapa *platform* seperti *HTML 5 website, Google Chrome Webstore, Facebook, Phonegap (Android), Windows Phone, Windows 8.* Pada *construct 2* terdapat 70 *visual effect* yang menggunakan *engine WebGL.* Selain itu juga *software* ini dilengkapi dengan 20 *built-in plugin* dan *behavior* (perilaku objek) sehingga kita dapat membuat *sprite, objek teks*, mengkoneksikan dengan *facebook*, menambah musik, memanipulasi penyimpanan data *game* dan lain sebagainya (Bastian et al., 2019).

2.3 *Software* Pendukung *Asset*

Dalam pembuatan *Asset desain* dapat menggunakan *software CorelDraw.* Berdasarkan situs *website* (Corel, 2022), *File CorelDraw* (CDR) terutama adalah gambar grafik vektor. Vektor mendefinisikan gambar sebagai daftar grafik primitif

(persegi panjang, garis, teks, busur, dan elips). Vektor dipetakan poin demi poin ke halaman, jadi jika Anda mengurangi atau memperbesar ukuran grafik vektor, gambar aslinya tidak akan terdistorsi. Grafik vektor dibuat dan diedit dalam aplikasi desain grafis, seperti *CorelDraw*, tetapi Anda juga dapat mengedit grafik vektor dalam aplikasi pengeditan gambar seperti *Corel Photo-Paint*. Anda dapat menggunakan gambar vektor dari berbagai format dalam program desktop *publishing*. Sedangkan dalam pembuatan *Asset audio* memerlukan *software FL Studio.*

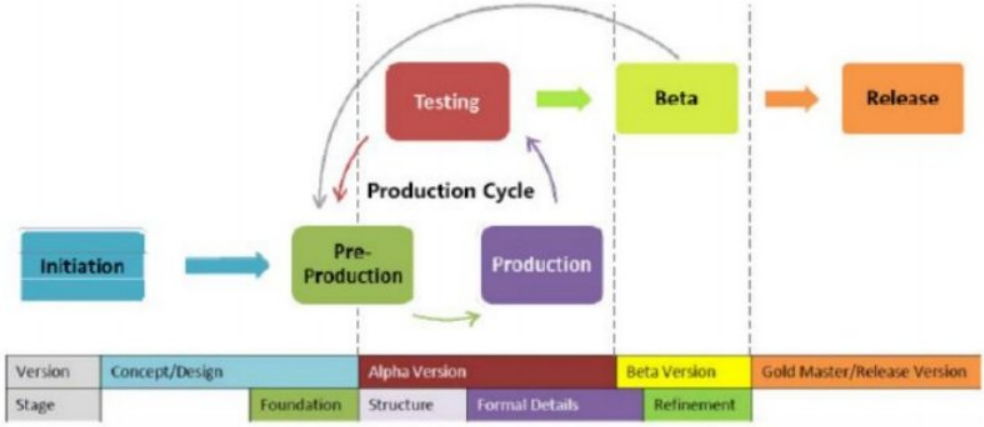
Berdasarkan situs *website* (Image-Line, 2022), *FL Studio* adalah lingkungan produksi musik berfitur lengkap yang mampu merekam, mengurutkan, dan mencampur *audio multi-track* untuk membuat trek musik berkualitas profesional. Dengan *VST hosting, mixer fleksibel*, dukungan MIDI canggih, dan *ReWire*, tidak ada gaya musik yang berada di luar jangkauan. Lagu atau loop dapat diekspor 23 sebagai .wav, .mp3, .ogg, .flac, atau .mid. Sejalan dengan itu (Eriga, 2017) menjelaskan *FL Studio* merupakan aplikasi *software multimedi*a pengolah *audio* yang sangat populer dan paling banyak digunakan para *sound engineer* dan *sound designer* untuk membuat dan mengolah berbagai bentuk efek *audio*.

2.4 GDLC

Menurut (Krisdiawan, 2018) GDLC atau *Game Development Life Cycle* merupakan sebuah teknik atau metode pengembangan *game* dengan menggunakan pendekatan interaktif yang memiliki 6 tahapan dalam pengembangan *game* yaitu tahapan inisiasi , pra produksi, produksi, pengujian alpha, pengujian beta, dan rilis. Tahapan tersebut terbagi menjadi 3 kelompok bagian proses utama, sebagai berikut :

1. Proses inisiasi merupakan proses pembuatan konsep awal dan rancangan sebuah *game*.
2. Proses produksi yang memiliki beberapa proses mulai dari proses pra-produksi, produksi, dan pengujian alpha dan beta
3. Proses yang terakhir yaitu Rilis..

Proses GDLC dapat digambarkan seperti gambar berikut :



Gambar 2.1 Proses GDLC

Berdasarkan gambar tersebut maka proses awal dari pembuatan *game* menggunakan metode GDLC yaitu Tahapan Inisiasi yang merupakan proses awal dari pembuatan game yaitu menentukan rancangan dari sebuah *game*, tahapan ini di mulai dari menentukan *game* yang akan dibuat, topik yang akan diangkat, target pengguna dari game tersebut. Hasil akhir dari tahapan ini berupa rancangan *game* dan deskripsi dari *game* itu yang akan dibuat. Selanjutnya tahapan kedua yaitu Preproduction, tahapan ini merupakan tahapan penting dalam pembuatan *game*. Mulai dari pembuatan desain game yang berfokus pada mendefinisikan *genre*

permainan, *gameplay*, *game* mekanik, alur dari cerita *game*, karakter, tantangan, faktor kesenangan, aspek teknis, dan dokumentasi elemennya dalam *Game Design Document* (GDD). Hasil dari proses ini adalah desain pada *game* yang didokumentasikan pada DDG.

Setelah berhasil merancang GDD selanjutnya masuk pada tahapan *Production,* proses ini merupakan proses inti dimana pada tahapan ini dilakukan penciptaan *asset*, pembuatan kode, dan integrasi kedua elemen. Hasil akhir dari tahapan ini yaitu *game* yang siap diujikan. Selanjutnya masuk pada tahapan pengujian *Alpha* atau *Alpha Testing,* proses ini merupakan tahapan dimana *developer* akan melakukan pengujian *internal* dengan beberapa pengguna, hal ini dilakukan untuk menguji seluruh aspek fungsional pada *game* yang selesai dikembangkan telah berhasil atau masih terdapat *bug* pada *game*. Hasil akhir dari tahapan ini yaitu laporan *bug* hasil pengujian.

Setelah berhasil melewati proses *Alpha testing,* selanjutnya masuk pada tahapan *Beta Testing* atau pengujian *Beta* dimana akan dilakukan pengujian *eksternal* atau pengujian langsung ke pengguna untuk mengetahui *bug* pada *game*. Hasil akhir pada tahapan ini adalah laporan hasil pengujian. Selanjutnya setelah *game* dirasa sudah memenuhi standar maka akan masuk pada tahapan terakhir yaitu *Release* dimana pada tahapan ini *game* yang telah berhasil melalui tahapan pengujian akan di rilis ke pengguna.

2.8 Firebase Realtime Database

Firebase Realtime Database merupakan sebuah jenis *database* yang penyimpanannya pada *cloud, firebase* dapat digunakan untuk berbagai platform seperti *Android, iOS* dan *Web*. Struktur data pada *database firebase* disimpan dengan struktur *Java Script Object Notation* (JSON). Data akan disinkronisasi secara otomatis kepada sistem atau aplikasi yang terhubung dengannya. Sehingga aplikasi yang menggunakan *database* ini akan secara otomatis mengupdate data terbaru yang ada pada *server database firebase* tersebut (Sudiartha et al., 2018).

Menurut (Ilham Firman Maulana, 2020) *Firebase Realtime Database* juga merupakan sebuah *Platfrom Database* yang banyak dipakai pada sistem atau aplikasi *realtime.* Saat suatu aplikasi yang terhubung pada *firebase* mengalami perubahan data maka aplikasi tersebut akan mengupdate data pada aplikasi secara otomatis baik itu perangkat *mobile* maupun *website*. *Database* ini dapat digabungkan dengan berbagai macam *framework* lain seperti *node, java, javascript* dan lainnya. Adapun fitur yang yang disediakan oleh *firebase* adalah sebagai berikut:

1. *Analytics* atau Analisis merupakan suatu fitur yang dapat dipakai untuk mengamati perilaku pengguna dalam penggunaan aplikasi dan tampilan dalam satu *dashboard.*
2. *Develop* atau Mengembangkan merupakan suatu fitur yang berupa perpesanan cloud, otentikasi, basis data waktu nyata, penyimpanan, hosting, testlab, dan pelaporan kerusakan.
3. *Grow* atau semakin meningkat merupakan suatu fitur yang dapat dipakai

untuk mempublikasikan sebuah produk aplikasi.

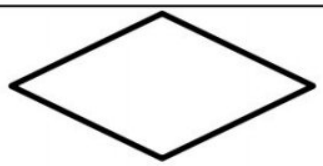
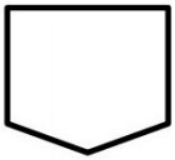
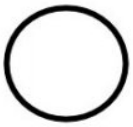
2.9 *Flowchart*

Flowchart adalah merupakan suatu jenis penulisan dari algoritma dalam menyelesaikan suatu permasalahan yang berbentuk bagan alir dengan menggunakan simbol-simbol. *Flowchart* dapat memudahkan dalam melakukan pengecekan penyelsaian permasalahan yang ada. *Flowchart* juga dapat digunakan sebagai fasilitas untuk komunikasi antara *programmer* dan sistem analis dalam melaksanakan proyek. (Santoso & Nurmalina, 2017). Adapun simbol-simbol yang ada pada *flowchart* sebagai berikut :

Tabel 2. 1 Simbol-simbol *Flowchart*

Simbol	Fungsi
	Simbol Terminal merupakan simbol yang berfungsi untuk menunjukan proses awal dan akhir dari sebuah permasalahan.
	Simbol Input/Output merupakan simbol yang berfungsi untuk menunjukan input dan output sebuah program.
	Simbol Proses merupakan simbol yang berfungsi untuk menunjukan proses pengolahan data.

Tabel 2.2 Lanjutan Simbol-simbol *Flowchart*

Simbol	Fungsi
	Simbol Keputusan merupakan simbol yang berfungsi untuk menyatakan suatu pilihan dalam suatu kondisi tertentu.
	Simbol persiapan merupakan simbol yang berfungsi untuk memberikan nilai awal pada suatu variabel atau pencacah.
	Simbol proses terdefinisi merupakan simbol yang berfungsi untuk proses yang detailnya dijelaskan terpisah.
	Simbol Penghubung ke halaman lain merupakan simbol yang berfungsi untuk menghubungkan diagram alir pada halaman yang berbeda
	Simbol Penghubung ke halaman yang sama merupakan simbol yang berfungsi untuk menghubungkan diagram alir pada halaman yang sama
	Simbol Arah alir, simbol yang berfungsi untuk menunjukan arah alir proses.

Sumber : (Santoso & Nurmalina, 2017)

2.10 UML

UML *(Unified Modelling Language*) merupakan standar bahasa yang digunakan untuk merancang suatu program yang berfungsi untuk menggambarkan atau memvisualisasikan desain dari sebuah sistem (Avrilia, 2017). Dalam penelitian ini penulis menggunakan UML jenis *Use Case Diagram* dan *Activity Diagram.*

Menurut (Sukamto, 2018) *Use Case Diagram* adalah bentuk pemodelan perilaku (*behavior*) dari sebuah sistem yang akan dibuat. *Use case* berfungsi untuk memaparkan interaksi yang dapat dilakukan oleh satu atau lebih aktor terhadap sistem, serta untuk mengetahui fungsi apa yang terdapat pada sistem dan siapa saja yang dapat menggunakan fungsi tersebut. Adapun Simbol-simbol pada *use case* sebagai berikut:

Tabel 2. 3 Simbol-Simbol *Use Case*

Simbol	Nama	Keterangan
	*Use case*	Berfungsi untuk menggambarkan interaksi antara sistem dan aktor.
	Aktor/*actor*	Berfungsi untuk mewakili orang, proses, atau sistem lain yang berinteraksi dengan use case.

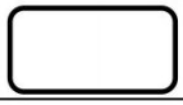
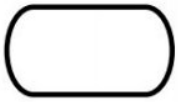
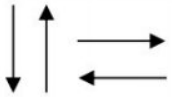
Tabel 2. 4 Lanjutan Simbol-simbol *Use Case*

Simbol	Nama	Deskripsi
	Assosiasi /*Association*	Berfungsi untuk komunikasi antara aktor dan use case.
	Generalisasi/ generalization	Berfungsi untuk menggambarkan spesialisasi dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<<***include***>>	*Include*	Berfungsi untuk menunjukan *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan.
 <<***extend***>>	*Extend*	Berfungsi untuk menunjukan use case tambahan ke sebuah use case di mana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

(Sukamto, 2018),

Menurut (Syarif & Nugraha, 2020), *activity diagram* merupakan sebuah diagram yang berfungsi untuk menggambarkan suatu aliran kerja atau aktivitas dalam sebuah sistem. Adapun simbol-simbol pada *activity diagram* dapat dilihat sebagai berikut:

Tabel 2. 5 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	*Activity*	Berfungsi untuk menunjukan interaksi antara antar muka atau tampilan..
	*Action*	Berfungsi untuk menggambarkan eksekusi dari suatu aksi.
	*Intial node*	Berfungsi untuk menunjukan awal dari aktivitas.
	*Activity final node*	Berfungsi untuk menunjukan akhir dari aktivitas.
	*Decision*	Berfungsi untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
	*Line Connector*	Berfungsi untuk menghubungkan antara simbol dengan simbol lainnya.

Sumber : (Syarif & Nugraha, 2020)

2.11 Metode Pengujian *System Usability Scale* (SUS)

Pengujian pada *game* ini menggunakan metode *System Usability Scale* (SUS) yang diperkenalkan oleh John Brooke. *System usability scale* (SUS) ini memiliki 10 pertanyaan dimana setiap pertanyaannya diberikan pilihan skor dengan masing-masing jawaban bernilia 1-5, sangat tidak setuju (STS) dengan nilai 1, tidak setuju (ST) dengan nilai 2, ragu-ragu (RR) dengan nilai 3, setuju (S) mempunyai nilai 4, dan sangat setuju (SS) dengan nilai 5, pilihan tersebut sangat berguna untuk mendapatkan jawaban berdasarkan pada berapa banyak mereka setuju pada pertanyaan tersebut terhadap aplikasi atau fitur yang diujikan. Dengan pertanyaan yang telah

distandarisasi sehingga dapat memberikan nilai rata-rata *Usability* dengan skala 0-100, dari hasil perhitungan metode SUS akan dikonversi kedalam nilai yang dapat dijadikan pertimbangan apakah aplikasi yang telat dibuat layak atau tidak layak untuk diterapkan (Ramadhan, 2019). Pertanyaan *System usability scale* (SUS). Disajikan dalam Tabel berikut :

Tabel 2.6 Pertanyaan *System usability scale* (SUS)

Pertanyaan
Saya berfikir akan menggunakan sistem ini lagi
Saya merasa sistem ini rumit
Saya merasa sistem ini mudah digunakan
Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
Saya merasa ada banyak hal yang tidak konsisten (tidak sersi dalam sistem ini)
Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
Saya merasa sistem ini membingungkan
Saya merasa tidak ada hambatan dalam menggunakan sistem ini
Saya perlu membiasakan diri terlebih dahulu sebelum menggunkan sistem ini

Sumber : (Susilo, E. 2019).

2.11.1 Aturan Perhitungan *System Usability Scale* (SUS)

Dalam perhitungan SUS terdapat beberapa aturan saat menghitung skor pada kuesioner seperti berikut :

1) Setiap pertanyaan yang bernomor ganjil, stiap skor yang didapatkan akan dikurangi 1.
2) Setiap pertanyaan yang bernomor genap,skor akhir 5 dikurangi dari skor pertanyaan dari pengguna.
3) Hasil skor SUS pada penjumlahan kemudian dikali 2,5

Skor pada SUS di masing-masing responden dicari rata-ratanya denagan cara menjumlahkan semua skor lalu dibagi dengan jumlah responden (Susilo, E. 2019). Seperti rumus berikut:

$$\bar{x} = \frac{\sum x}{n}$$

Penjabaran : $\bar{x}$ = nilai rata-rata

$\sum x$ = jumlah nilai SUS

n = jumlah responden

2.11.2 Cara Perhitungan *System Usability Scale* (SUS)

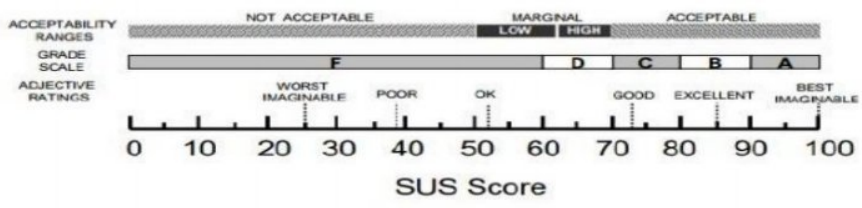
Perhitungan *System Usability Scale* (SUS) dengan menuliskan hasil dari responden di aplikasi excel atau aplikasi lainnya. Mulai dari Q1 sampai Q10 yang merupakan nomer pertanyaan dengan angka sebagai jawaban dari responden (Susilo, E. 2019). Disajikan dalam Tabel berikut :

Tabel 2.7 Perhitungan *System Usability Scale* (SUS)

NO	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai (jml x 2,5)
1	Responden 1	5	2	4	2	5	2	2	2	5	4	33	82,5
2	Responden 2	5	2	4	2	5	2	2	2	5	2	31	77,5
3	Responden 3	4	4	4	4	4	4	4	4	4	4	40	100
..	Responden …	..	..	..	..	..	..	..	..	..	..	…	…
	Jumlah Rata-Rata												….

Sumber : (Susilo, E. 2019).

Kesimpulan dari cara hitung *System Usability Scale* (SUS) adalah setelah dihitung dan mendapatkan skor rata-rata SUS dari semua responden kemudian skor tersebut disesuaikan dengan penilaian SUS masuk kategori mana hasil pengujian dengan skor rata-rata yang telah di dapat. Dari banyaknya skor rata-rata SUS penelitian nilai 68 akan dianggap di atas rata-rata sedangkan nilai di bawah 68 dianggap dibawah rata-rata. Kesimpulan perhitungan SUS juga bisa ditentukan melalui penilaian seperti disajikan dalam gambar berikut.



Gambar 2.2 Skor SUS

Usability biasanya dapat digunakan untuk mengukur tingkat kegunaan suatu sistem atau peralatan. Menurut *International Organization for Standardization* – ISO 9241-11:1998 disebutkan bahwa *usability* adalah tingkat kegunaan dari suatu produk yang dapat digunakan oleh pengguna untuk mencapai tujuan yang ditentukan secara efektif, efisien, dan memberikan kepuasan. *Usability* merupakan sebuah konsep pembuatan sistem yang mudah dipelajari dan digunakan. *Usability* adalah proses yang sangat penting dalam desain interaksi yang meliputi: perilaku, efisiensi, efektifitas, fleksibilitas, keamanan, utilitas, kemudahan dipelajari, dan kemudahan diingat. Salah satu metode yang dapat digunakan untuk melakukan evaluasi tingkat kegunaan website adalah *System Usability Scale* (Soejono et al., 2018).