

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penulisan ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penulisan, berikut ini merupakan tinjauan studi yang diambil yaitu, daftar literatur dapat dilihat pada tabel 2.1:

Tabel 2.1 Daftar Literatur

1.	Judul Penelitian	Aplikasi pengenalan hewan lindung menggunakan augmented reality dengan metode marker based tracking
	Penulis (Tahun)	Mir'atul Khusna Mufida, Muhammad Harun (2018)
	Masalah	Sekarang pengenalan hewan lindung Indonesia hanya melalui media teks, gambar ataupun videoyang terdapat dalam buku pembelajaran maupun internet. Dengan pengenalan yang terbatas menjadikan masih banyaknya masyarakat yang belum mengetahui tentang hewan lindung dan hal tersebut menyebabkan kurangnya kepedulian masyarakat untuk membantu melestarikan hewan lindung di Indonesia.
	Metode/Tools	Unity 3D
	Hasil Penelitian	Dengan Aplikasi Pengenalan Hewan Lindung menambah keinginan bagi masyarakat untuk lebih mengenal hewan lindung yang ada di Indonesia.
2.	Judul Penelitian	Aplikasi Pembelajaran Interaktif Pengenalan Satwa Sulawesi Utara Menggunakan Augmented Reality
	Penulis (Tahun)	Mestilia Meilin Mongilala, Virginia Tulenan, Brave A. Sugiarto (2019)
	Masalah	Satwa langka sering di bunuh untuk dijadikan hiasan, hewan peliharaan juga dimanfaatkan kulitnya menjadi suatu karya seni yang bernilai tinggi. Pengetahuan yang mamadai serta kesulitan dalam mendapatkan informasi menjadi salah satu faktor penyebabnya. Dengan demikian maka diperlukan sebuah teknologi yang mampu memberikan informasi secara mudah serta dapat menampilkan bentuk 3 dimensi satwalangka

	secara visual.
--	----------------

	Metode/Tools	MDLC (Multimedia Development Life Cycle)
	Hasil Penelitian	Menghasilkan aplikasi yang dapat memperkenalkan satwa langka dengan teknologi AR (Augmented Reality) yang berbasis android
3.	Judul Penelitian	Implementasi augmented reality untuk pengenalan hewan endemik indonesia berbasis android
	Penulis (Tahun)	Piter Budi Raharjo, Suryo Adi Wibowo, Mira Orisa (2020)
	Masalah	Informasi yang didapatkan kebanyakan bersumber pada buku manual dan e-book saja. Maka dibutuhkan metode yang lebih efektif.
	Metode/Tools	Unity 3D
	Hasil Penelitian	Dengan aplikasi pengenalan hewan endemik Indonesia selain dapat meningkatkan minat belajar siswa tentang hewan yang masih belum banyak diketahui, tetapi juga memberikan gambaran bentuk asli dari hewan tersebut layaknya seperti alat peraga.
4.	Judul Penelitian	Pembangunan perangkat lunak interaktif berbasis markerless augmented reality untuk pengenalan hewan pada taman kanak-kanak
	Penulis (Tahun)	Muhammad Qori' Untiarasani ¹ , Hanny Haryanto, Erna Zuni Astuti (2015)
	Masalah	Pengenalan hewan yang berlangsung masih menggunakan media kertas bergambar dengan penjelasan dari seorang pengajar dan sebagian juga sudah menggunakan alat peraga, namun alat peraga jumlahnya sangat terbatas, sehingga harus bergantian ketika menggunakannya. Peserta didik mengalami kesulitan terhadap materi yang disampaikan karena memerlukan imajinasi yang cukup terhadap ciri-ciri hewan yang disampaikan oleh pengajar.

	Metode/Tools	UML
	Hasil Penelitian	Aplikasi ini mampu memberikan detail gambar 3D berikut dengan keterangan sehingga membuat anak-anak lebih memahami tentang pembelajaran hewan
5.	Judul Penelitian	Rancang Bangun Aplikasi Pengenalan Satwa Langka di Indonesia Menggunakan Augmented Reality
	Penulis (Tahun)	Riana Indriani, Bayu Sugiarto, Agus Purwanto (2016)

	Masalah	Indonesia terkenal akan keanekaragaman satwa yang tersebar di pulau-pulau di Indonesia, namun sebab kurangnya media untuk memperkenalkan satwa langka di Indonesia menjadikan satwa langka yang ada tidak dikenal sehingga rentan akan perburuan sehingga terancam punah
	Metode/Tools	Blender, MDLC (Multimedia Development Life Cycle)
	Hasil Penelitian	Aplikasi Augmented Reality ini dapat digunakan sebagai media pengenalan satwa langka di Indonesia, diharapkan aplikasi ini selanjutnya bisa dikembangkan untuk berbagai platform juga dapat memuat informasi tentang seluruh satwa langka yang ada di Indonesia.

2.2 Augmented Reality

Augmented Reality (AR) adalah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia virtual yang dibuat oleh komputer sehingga batas antara keduanya menjadi sangat tipis. Dengan teknologi AR, lingkungan nyata di sekitar akan dapat berinteraksi dalam bentuk digital (virtual) (Raharjo, dkk., 2020). Dengan Augmented Reality dapat membuat suatu objek mati seakan-akan dihidupkan dengan bantuan kamera yang dapat diakses di komputer atau smartphone. Dengan sebuah marker kita dapat melihat benda dua dimensi atau tiga dimensi dalam sebuah layar sebagai titik acuan fokus kamera (Karundeng, dkk., 2018).

Spesifikasi perangkat minimal untuk menampilkan *Augmented Reality* dengan baik adalah RAM 1 GB, Camera 5 MP, Android 4.4, dan Prosesor Quad core 1 GHz (Zainudin, dkk., 2016).

2.3 Hewan Endemik

Hewan endemik merupakan Spesies hewan alami yang hidup dan mendiami suatu wilayah atau daerah tertentu, sehingga menjadi ciri khas dari suatu wilayah yang tidak ditemukan di lokasi lainnya (Aristides, dkk., 2016).

2.4 Blender

Blender merupakan serangkaian kreasi 3 dimensi yang gratis serta open source. Blender mendukung konsep 3D secara keseluruhan seperti modeling, rigging, animasi, simulasi, rendering, compositing, dan motion tracking, bahkan video editing dan pembuatan game (Mongilala, dkk, 2019).

2.5 Unity 3D

Unity 3D adalah perangkat lunak yang dapat membangun suatu permainan 3 Dimensi (3D). Unity mendukung pengembangan aplikasi android, yang mana diperlukan suatu pengaturan lingkungan pengembang android pada perangkat sebelum dapat menjalankan aplikasi yang di buat oleh software unity 3D (Rachmanto dan Noval, 2018).

2.6 Aplikasi

Aplikasi adalah suatu unit perangkat lunak yang dibuat dengan tujuan membantu melayani kebutuhan seseorang dalam melakukan aktivitas agar menjadi lebih baik dan lebih mudah jika menggunakan suatu aplikasi (Syafrizal, dkk., 2018). Menurut Abdurahman dan Asep dalam (Hendrati, 2018) aplikasi merupakan perangkat yang telah siap untuk digunakan oleh seorang user. Secara umum aplikasi merupakan alat terapan yang difungsikan secara khusus sesuai kemampuan yang dimilikinya.

2.7 Android

Android merupakan salah satu sistem operasi berbasis Linux dan open source. Sistem Operasi Android mempunyai banyak pengguna dikalangan masyarakat dikarenakan sifat Android yang open source (Syafrizal, dkk., 2018). Aplikasi yang akan dibuat adalah berbasis android dengan tujuan agar lebih

mudah dalam penggunaan aplikasi ini jika menggunakan smartphone dengan sistem operasi android.

^{2.8} **Vuforia**

Vuforia merupakan library *Augmented Reality Software Development Kit* (SDK) yang digunakan sebagai pendukung adanya augmented reality pada android. Vuforia menganalisa gambar dengan menggunakan pendeteksi marker dan menghasilkan informasi 3D dari marker yang sudah dideteksi. Kemampuan vuforia yang dapat merubah kertas dengan latar kosong menjadi tampilan dengan grafis 3D yang menarik, karena adanya teknologi augmented reality. Dengan vuforia, developer dapat memaksimalkan kemampuan teknologi augmented reality untuk menciptakan konten seperti, aplikasi, game, iklan dan presentasi (Raharjo, dkk., 2020).

2.9 Marker

Menurut Rumanjar (dalam Rachmanto dan Noval, 2018) Marker merupakan suatu penanda khusus yang mempunyai pola yang sudah ditentukan sehingga pada saat kamera di arahkan ke marker maka marker akan terdeteksi dan objek 3 dimensi akan tampil. Augmented reality terus menerus melakukan perkembangan, salah satunya perkembangan yang dilakukan adalah pada bagian marker. *Marker* yang pertama kali adalah based tracking. *Marker based tracking* arker ini biasanya merupakan suatu ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang yang berwarna putih. Melalui posisi yang dihadapkan pada sebuah kamera komputer atau *smartphone*, maka komputer atau *smartphone* akan melakukan proses menciptakan dunia virtual dua dimensi atau tiga dimensi. *Marker Based Tracking* ini sudah lama dikembangkan sejak tahun 1980-an dan pada awal tahun 1990-an mulai dikembangkan untuk penggunaan *augmented reality* (Mufida & Harun, 2018). Kemudian ada pula *Merkeless Augmented Reality*, yaitu metode dimana pengguna tidak perlu lagi menggunakan sebuah marker untuk menampilkan elemen-elemen digital (Meilin Mongilala, dkk., 2019).

2.10 Multimedia

Multimedia merupakan kombinasi teks, seni suara, gambar, animasi dan video yang disampaikan dengan computer atau dimanipulasi secara digital Ada 3 jenis multimedia (Meilin Mongilala, dkk., 2019), yaitu:

2.10.1 *Multimedia Interaktif*

Disebut multimedia interaktif ketika pengguna di izinkan dapat mengontrol apa dan kapan elemen-elemen multimedia akan dikirimkan.

2.10.2 *Multimedia Hyperaktif*

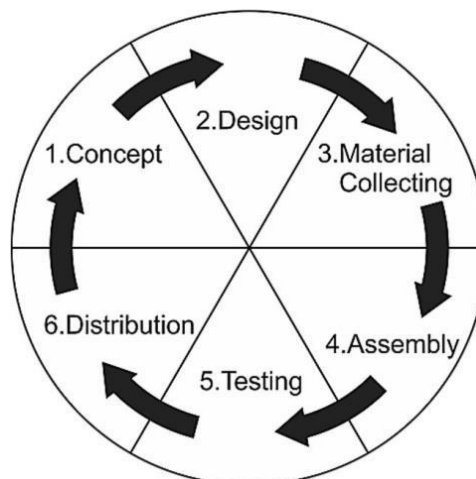
Mampu menyediakan suatu struktur dari elemen-elementerkait dengan pengguna yang dapat mengarahkannya.

2.10.3 *Multimedia Linear*

Pengguna hanya menjadi penonton dan menikmati produk multimedia yang disajikan dari awal hinggaakhir. Multimedia dapat meningkatkan antarmuka komputer secara visual sehingga menghasilkan keuntungan yang memuaskan pengguna serta dapat menarik perhatian dan ketertarikan orang untuk mempelajarinya. Multimedia dapat memperkuat ingatanterhadap suatu informasi dan jika dirancang dengan menggunakan kreatifitas yang tepat dapat dimanfaatkan sebagai media hiburan.

2.11 Metode Pengembangan Sistem

Penelitian ini menggunakan metode *Multimedia Development Life Cycle (MDLC)* menurut Luther dalam (Sutopo, 2003). Luther mendefinisikan langkah-langkah pengembangan perangkat lunak multimedia dengan 6 tahap, seperti gambar berikut.



Gambar 2.1 Metode pengembangan *Multimedia Development Life Cycle (MDLC) Model Luther* (Sutopo, 2003).

1. *Concept*

Tujuan pengembangan perangkat lunak multimedia di definisikan pada tahapan ini, melingkupi identifikasi pengguna aplikasi, jenis aplikasi (presentasi, interaktif, dst.), tujuan aplikasi (pemberitahuan, hiburan, pengajaran, dst.), dan hal-hal umum. Aturan dasar untuk tahapan design juga sudah diatur dan ditentukan pada tahapan ini, seperti gaya, ukuran aplikasi, platform yang akan digunakan, dan lainnya.

2. *Design*

Tujuan tahapan ini adalah menentukan secara detail arsitektur, gaya, dan semua material yang akan digunakan pada perangkat lunak multimedia yang akan dikembangkan. Perangkat lunak *authoring* akan mulai berguna pada tahapan ini karena *system authoring* akan mengambil alih tugas pendokumentasian *design* dan menyimpan informasi dalam suatu bentuk yang dapat dengan mudah berpindah secara langsung ke tahapan *content collection* dan *assembly* tanpa entri manual. Fitur *authoring* yang membantu dalam tahapan ini adalah: *outlining*, *storyboarding* dan *scripting*.

3. *Material Collecting*

Dari daftar konten yang sudah dibuat pada tahap *design*, pengembang harus mengumpulkan konten materialnya. Pekerjaan ini sering dikerjakan secara parallel

dengan tahapan *assembly*, terutama jika *system authoring* mengizinkan konten *dummy* untuk digunakan sampai konten yang sebenarnya ada. Konten material didapat dari sumber luar atau membuat sendiri sesuai kebutuhan. Sistem *authoring* akan membantu dengan berbagai cara.

4. *Assembly*

Seluruh material dan yang dibutuhkan digabungkan di tahap ini. Tergantung pada apa yang sudah dikerjakan pada tahap sebelumnya, ada kemungkinan tidak ada pekerjaan pada tahap ini atau justru harus dikerjakan keseluruhan. Ada *system authoring* yang secara otomatis membangun struktur program dari *percancangan* yang sudah diberikan dan pengembang hanya memasukkan konten material ke tempat-tempat yang sudah disediakan.

5. *Testing*

Ketika aplikasi sudah dibangun dan konten material sudah masuk ke dalamnya, aplikasi harus di tes untuk meyakinkan bahwa semuanya berjalan sesuai dengan keinginan. Hal ini sudah dilakukan juga ketika sedang pada tahap *assembly* dan sangat penting ketika *system authoring* mengizinkan pengembang untuk melihat dan memeriksa pekerjaan yang sedang berlangsung.

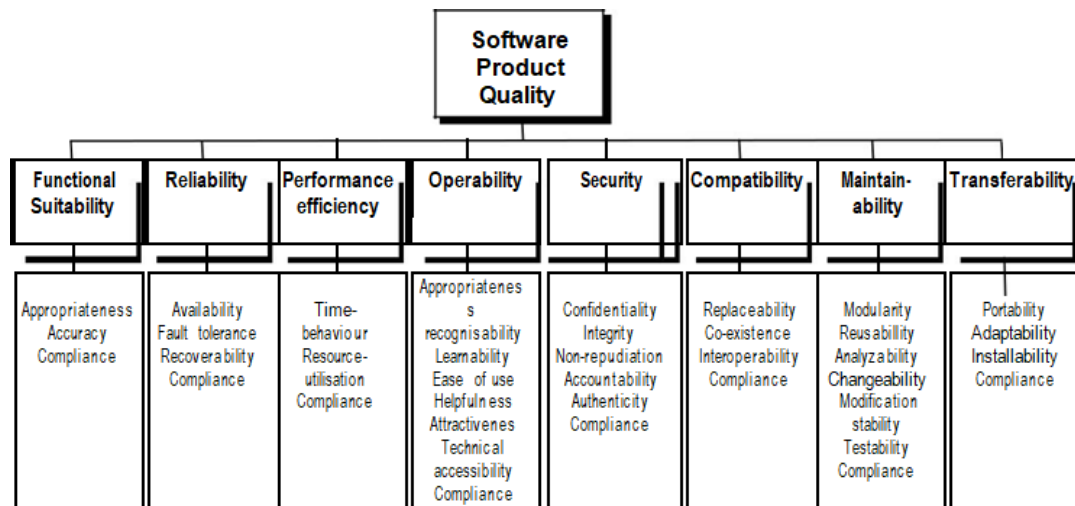
6. *Distribution*

Cara pendistribusian aplikasi yang sudah sepenuhnya siap digunakan harus disiapkan dan disesuaikan dengan lingkungan yang sebenarnya. Pengembang sebaiknya mempunyai beberapa *alternative* untuk pendistribusian ini, misalnya dengan *cdrom/dvdrom*, *flashdisk*, atau internet. Hal ini juga terkait dengan jumlah dan besarnya *file* yang ada maupun penataan file yang diperlukan oleh aplikasi.

2.12 Metode Pengujian Sistem

Metode pengujian dalam penelitian ini menggunakan Standar *ISO /IEC 25010:2011*, dimana pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. *ISO/IEC 25010* memperkenalkan tipe kualitas (*quality in use*) dimana mengikuti elemen yang telah diketahui **Invalid source specified..** *ISO/IEC 25010* merupakan standar

internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan perkembangan dari ISO 9126. Model kualitas *ISO 25010* mempunyai delapan ukuran kualitas yang ditetapkan oleh *ISO/IEC 25010* yang dapat dilihat pada Gambar 2.7.



Gambar 2.2 Karakteristik ISO 25010 **Invalid source specified.**

2.12.1 *Functional Suitability*

Functional Suitability merupakan tingkat dimana produk perangkat lunak menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika perangkat lunak digunakan dalam kondisi tertentu. Subkarakteristik *Functional Suitability* meliputi *appropriateness*, *accuracy*, dan *compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Functional Suitability* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Penjelasan Sub karakteristik *Functional Suitability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Functional Suitability</i>	<i>Appropriateness</i>	Sejauh mana produk perangkat lunak menyediakan fungsi yang tepat untuk tugas-tugas tertentu dan tujuan pengguna?
	<i>Accuracy</i>	Sejauh mana produk perangkat lunak memberikan hasil yang tepat atau spesifik dengan tingkat presisi yang diperlukan?

	<i>Compliance</i>	Tingkat dimana perangkat lunak mematuhi standar, konvensi, atau peraturan dalam undang-undang dan peraturan serupa yang berkaitan dengan kesesuaian fungsional
--	-------------------	--

2.12.2 Reliability

Reliability didefinisikan sejauh mana perangkat lunak dapat mempertahankan tingkat kinerja dalam kondisi tertentu. Subkarakteristik *Reliability* meliputi *availability*, *fault tolerance*, *recoverability*, *reliability compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Reliability* yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Penjelasan Subkarakteristik *Reliability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Reliability</i>	<i>Availability</i>	Sejauh mana komponen perangkat lunak beroperasi dan tersedia saat diperlukan untuk digunakan.
	<i>Fault tolerance</i>	Sejauh mana produk perangkat lunak dapat mempertahankan tingkat kinerja tertentu dalam kasus kesalahan perangkat lunak atau pelanggaran antarmuka yang ditentukan.
	<i>Recoverability</i>	Tingkat dimana produk perangkat lunak dapat menetapkan kembali tingkat kinerja yang ditentukan dan memulihkan data yang secara langsung dalam kasus kegagalan.
	<i>Reliability compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keandalan.

2.12.3 Performance efficiency

Performance efficiency merupakan sejauh mana perangkat lunak memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Subkarakteristik *Performance efficiency* meliputi *time behaviour*, *resource utilization*, *performance efficiency compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Performance efficiency* yang dapat dilihat pada Tabel 2.4.

Tabel 2.4 Penjelasan Subkarakteristik *Performance efficiency*

Karakteristik	Subkarakteristik	Penjelasan
<i>Performance efficiency</i>	<i>Time behaviour</i>	Sejauh mana perangkat lunak memberikan respons yang tepat, waktu pemrosesan dan laju keluaran ketika menjalankan fungsinya dalam kondisi yang ditentukan.
	<i>Resource utilization</i>	Sejauh mana perangkat lunak menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak menjalankan fungsinya dalam kondisi yang ditentukan.
	<i>Performance Efficiency Compliance</i>	Sejauh mana perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan efisiensi kinerja.

2.12.4 Operability

Operability merupakan sejauh mana produk perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik bagi pengguna bila digunakan dalam kondisi tertentu. Subkarakteristik *Operability* meliputi *appropriateness recognisability*, *learnability*, *ease of use*, *helpfulness*, *attractiveness*, *technical accessibility*, dan *compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Operability* yang dapat dilihat pada Tabel 2.5.

Tabel 2.5 Penjelasan Subkarakteristik *Operability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Operability</i>	<i>Appropriateness Recognisability</i>	Sejauh mana perangkat lunak memungkinkan pengguna untuk mengenali apakah perangkat lunak sesuai dengan kebutuhan pengguna?
	<i>Learnability</i>	Sejauh mana perangkat lunak memungkinkan pengguna untuk mempelajari aplikasinya?
	<i>Ease of use</i>	Sejauh mana perangkat lunak memudahkan pengguna untuk mengoperasikan dan mengendalikannya?
	<i>Helpfulness</i>	Sejauh mana perangkat lunak dapat membantu pengguna?
	<i>Attractiveness</i>	Apakah antarmuka terlihat baik?
	<i>Technical Accessibility</i>	Tingkat pengoperasian perangkat lunak untuk pengguna dengan cacat tertentu?
	<i>Compliance</i>	Apakah perangkat lunak sudah mematuhi standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan pengoperasian?

2.12.5 Security

Security merupakan perlindungan item sistem dari akses yang tidak disengaja atau berbahaya, penggunaan, modifikasi, perusakan dan pengungkapan. Subkarakteristik *Security* meliputi *confidentiality*, *integrity*, *non-repudiation*, *accountability*, *authenticity*, *security compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *security* yang dapat dilihat pada Tabel 2.6.

Tabel 2.6 Penjelasan Subkarakteristik *Security*

Karakteristik	Subkarakteristik	Penjelasan
<i>Security</i>	<i>Confidentiality</i>	Sejauh mana perangkat lunak memberikan perlindungan dari pengungkapan data atau informasi yang tidak sah, baik disengaja atau disengaja.
	<i>Integrity</i>	Sejauh mana ketepatan dan kelengkapan aset dijaga.
	<i>Non-repudiation</i>	Sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak kemudian.
	<i>Accountability</i>	Sejauh mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
	<i>Authenticity</i>	Sejauh mana identitas suatu subjek sumber daya dapat dibuktikan sebagai yang diklaim.
	<i>Security compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keamanan.

2.12.6 Compatibility

Compatibility merupakan kemampuan dua atau lebih komponen perangkat lunak untuk bertukar informasi dan untuk melakukan fungsi yang diperlukan saat berbagi perangkat keras atau perangkat lunak yang sama. Subkarakteristik *compatibility* meliputi *replaceability*, *co-existence*, *interoperability*, *compatibility compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *compatibility* yang dapat dilihat pada Tabel 2.7.

Tabel 2.7 Penjelasan Subkarakteristik *Compatibility*

Karakteristik	Subkarakteristik	Penjelasan
<i>Compatibility</i>	<i>Replaceability</i>	Sejauh mana perangkat lunak dapat digunakan ditempat perangkat lunak lain yang ditentukan untuk tujuan yang sama dilingkungan yang sama.
	<i>Co-existence</i>	Sejauh mana perangkat lunak dapat bekerja sama dengan perangkat lunak independen lainnya dalam lingkungan umum berbagi sumber daya umum tanpa ada dampak yang merugikan.
	<i>Interoperability</i>	Sejauh mana perangkat lunak dapat dioperasikan secara kooperatif dengan satu atau lebih perangkat lunak lainnya.
	<i>Compatibility compliance</i>	Sejauh mana perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan kompatibilitas.

2.12.7 Maintainability

Maintainability merupakan sejauh mana perangkat lunak dapat dimodifikasi. Modifikasi dapat mencakup koreksi, peningkatan atau adaptasi perangkat lunak terhadap perubahan lingkungan, dan persyaratan serta spesifikasi fungsional. Subkarakteristik *Maintainability* meliputi *modularity*, *reusability*, *analyzability*, *changeability*, *modification stability*, *testability*, *maintainability compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *maintainability* yang dapat dilihat pada Tabel 2.8.

Tabel 2.8 Penjelasan Subkarakteristik *Maintainability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Maintainability</i>	<i>Reusability</i>	Sejauh mana aset dapat digunakan lebih dari satu sistem perangkat lunak, atau dalam membangun aset lainnya.
	<i>Analyzability</i>	Tingkat dimana perangkat lunak dapat didiagnosis untuk kekurangan atau penyebab kegagalan dalam perangkat lunak.
	<i>Changeability</i>	Sejauh mana perangkat lunak memungkinkan modifikasi tertentu untuk diimplementasikan.
	<i>Modification stability</i>	Sejauh mana perangkat lunak dapat menghindari efek tak terduga dari modifikasi perangkat lunak.
	<i>Testability</i>	Sejauh mana perangkat lunak yang dimodifikasi untuk divalidasi.
	<i>Maintainability compliance</i>	Sejauh mana perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan pemeliharaan.

2.12.8 *Transferability*

Merupakan sejauh mana perangkat lunak dapat ditransfer dari satu lingkungan ke lingkungan lain. Subkarakteristik *transferability* meliputi *portability*, *adaptability*, *installability*, *transferability compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *transferability* yang dapat dilihat pada Tabel 2.9.

Tabel 2.9 Penjelasan Subkarakteristik *Transferability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Transferability</i>	<i>Portability</i>	Kemudahan sistem atau komponen yang dapat ditransfer dari satu perangkat keras atau perangkat lunak ke perangkat lain
	<i>Adaptability</i>	Apakah perangkat lunak dapat disesuaikan dengan lingkungan tertentu yang berbeda?
	<i>Installability</i>	Apakah perangkat lunak dapat diinstal dan dihapus pada lingkungan tertentu?
	<i>Transferability compliance</i>	Apakah perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan portabilitas.

Pada penelitian ini pengujian berfokus pada karakteritik *Functional Suitability*, *Performance efficiency* dan *Operability*.

3.13 Skala *Likert*

Skala *likert* merupakan skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono, 2013) skala likert dapat digunakan untuk mengukur sikap seseorang dengan menyatakan setuju atau tidak setuju terhadap subjek, obyek atau kejadian tertentu.

Dengan menggunakan skala likert, variabel dijabarkan menurut urutan variabel, sub variabel, indikator, dan deskriptor. Deskriptor kemudian dijadikan titik tolak untuk membuat butir instrumen berupa pernyataan atau pertanyaan yang perlu dijawab oleh responden.

Item-item dalam skala likert menyediakan respon dengan kategori yang berjenjang, dan biasanya memiliki jenjang lima, yaitu: sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju. Setiap kategori tersebut diberi nilai atau skor. Pernyataan pada skala likert terdiri dari pernyataan positif dan pernyataan

negatif. Contoh lima jenjang dalam skala likert dapat dilihat pada tabel 2.10 berikut.

Tabel 2.10 Jenjang dalam skala *likert* (Sugiyono, 2013).

Pernyataan positif	Nilai	Pernyataan Negatif	Nilai
Sangat setuju	5	Sangat setuju	1
Setuju	4	Setuju	2
Ragu-ragu (Netral)	3	Ragu-ragu (Netral)	3
Tidak Setuju	2	Tidak Setuju	4
Sangat tidak setuju	1	Sangat tidak setuju	5

Rumus perhitungan skala Likert adalah sebagai berikut:

$$\% \text{ Skor Aktual} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100$$

Keterangan:

1. Skor aktual adalah hasil jawaban seluruh responden atas kuisioner yang telah diajukan.
2. Skor ideal adalah nilai tertinggi atau semua responden diasumsikan memilih jawaban dengan skor tertinggi.

Kemudian hasil perhitungan yang didapatkan dari angket, selanjutnya dibandingkan dengan rentang kriteria interpretasi skor untuk menyatakan hasil yang didapatkan dengan rentang di tabel 2.11 berikut.

Tabel 2.11 Rentang kriteria interpretasi (Sugiyono, 2013).

No	Rentang Kriteria	Kriteria
1	0% - 20%	Sangat Tidak Baik
2	21% - 40%	Tidak Baik
3	41% - 60%	Kurang Baik
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik