

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang serupa dan relevan dengan penelitian bertujuan untuk membuktikan keaslian penelitian, untuk tinjauan pustaka dapat dilihat pada tabel 2.1 dibawah ini:

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Judul	Metode	Hasil
1.	(Adinda et al., 2021)	Implementasi Pelayanan Administrasi Kependudukan dan Catatan Sipil di Desa Bumiaji Kota Batu	<i>Kualitatif</i>	Hasil penelitian menunjukkan pelayanan publik berbasis elektronik di dinas kependudukan dan pencacatan sipil di desa bumi aji sudah berjalan dengan baik dapat dilihat dari beberapa aspek yaitu kompetensi petugas, biaya pelayanan, waktu penyelesaian, sarana prasarana, prosedur, dan produk pelayanan.
2.	(Maulana & Priatna, 2021)	Perancangan Sistem Pelayanan Administrasi Kependudukan dalam Menunjang Digitalisasi Desa	<i>Kualitatif</i>	Sistem yang dibangun memberikan pelayanan yang baik pada masyarakat dengan mencoba

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No	Peneliti	Judul	Metode	Hasil
				menyusun sistem pelayanan administrasi desa dengan harapan mempermudah masyarakat dalam mengakses pelayanan desa.
3.	(Padli et al., 2020)	Sistem Informasi Layanan Administrasi Kependudukan Pada Desa Maliaya Berbasis Web	<i>Kuantitatif</i>	Hasil penelitian ini adalah sebuah sistem informasi layanan penduduk pada desa maliaya berbasis website serta menggunakan bahasa pemrograman PHP dan MYSQL .
4.	(Syukron, 2019)	Perancangan Sistem Informasi Administrasi Kependudukan Desa Berbasis Website Pada Desa Winong	<i>Prototype</i>	Sistem informasi administrasi kependudukan yang berbasis website ini, dapat memudahkan pengelolaan data kependudukan. Sehingga dapat terwujud tertib administrasi kependudukan yang baik di desa Winong
5.	(Setyowati)	Implementasi Pelayanan	<i>SDLC</i>	Sistem ini dapat

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No	Peneliti	Judul	Metode	Hasil
	et al., (2019)	Administrasi Kependudukan Desa Kalimarah Wetan Purbalingga Berbasis Online		mengelola data warga sehingga setiap pengelolaan surat selalu terhubung pada database warga sehingga surat yang dikeluarkan sesuai dengan data warga di kelurahan dan format surat telah diatur oleh sistem sehingga terbantu dalam pengarsipan yang tertata dengan baik. Sistem

2.2. Definisi Pelayanan

Menurut Maulana & Priatna (2021) Pelayanan adalah kegiatan atau rangkaian kegiatan dalam rangka pemenuhan kebutuhan pelayanan sesuai dengan peraturan perundang-undangan bagi setiap warga negara dan penduduk atas barang, jasa, dan pelayanan administratif yang disediakan oleh penyelenggara pelayanan.

Menurut Adinda et al (2021) mendefinisikan bahwa pelayanan dapat diartikan sebagai aktivitas yang diberikan untuk membantu, menyiapkan, dan mengurus baik itu berupa barang atau jasa dari satu pihak ke pihak lain.

Berdasarkan pengertian pelayanan diatas dapat disimpulkan bahwa pelayanan adalah aktivitas atau kegiatan atau usaha yang dilakukan seseorang atau

sekelompok orang melalui hubungan interaksi antara satu pihak ke pihak lain yang menggunakan peralatan berupa organisasi atau lembaga perusahaan untuk kepuasan penerima.

2.3. Definisi Publik

Menurut Maulana & Priatna (2021) Istilah publik berasal dari bahasa Inggris public yang berarti umum, masyarakat, dan negara. Kata public sebenarnya sudah diterima menjadi bahasa Indonesia baku menjadi publik yang berarti umum, orang banyak, dan ramai.

Menurut Suandi (2019), mendefinisikan publik adalah sejumlah manusia yang memiliki kebersamaan berfikir, perasaan, harapan, sikap atau tindakan yang benar dan baik berdasarkan nilai-nilai norma yang merasa memiliki.

Berdasarkan pengertian publik di atas dapat disimpulkan bahwa publik adalah setiap kelompok orang yang memiliki minat atau nilai-nilai bersama dalam situasi tertentu, terutama kepentingan atau nilai-nilai mereka mungkin bertindak atas kesediaan.

2.4. Definisi Pelayanan Publik

Berdasarkan Undang Undang No. 25 Tahun 2009 tentang Pelayanan Publik yaitu pelayanan publik Pelayanan publik adalah segala bentuk kegiatan dalam rangka pengaturan, pembinaan, bimbingan, penyediaan fasilitas, jasa dan lainnya yang dilaksanakan oleh aparatur pemerintah sebagai upaya pemenuhan kebutuhan kepada masyarakat sesuai ketentuan perundang-undangan yang berlaku.

Menurut Suandi (2019) pelayanan publik adalah setiap kegiatan yang dilakukan oleh pemerintah terhadap sejumlah manusia yang memiliki setiap

kegiatan dan menawarkan kepuasan meskipun hasilnya tidak terikat pada suatu produk secara fisik.

Berdasarkan pengertian publik di atas dapat disimpulkan bahwa pelayanan publik adalah pemberian layanan atau melayani keperluan orang atau masyarakat yang mempunyai kepentingan pada organisasi itu sesuai dengan aturan pokok dan tata cara yang telah ditetapkan.

2.5. Administrasi Kependudukan

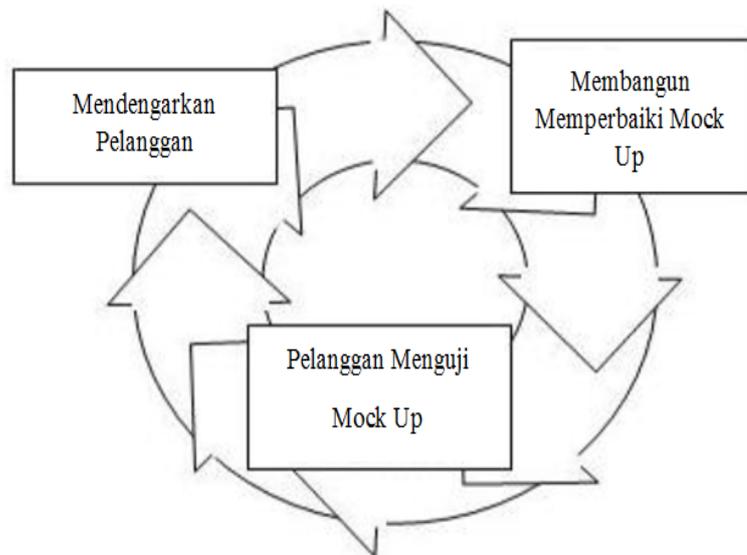
Menurut Setyowati et al (2019) administrasi kependudukan merupakan suatu pengelolaan data-data yang berkaitan dengan informasi penduduk meliputi data kependudukan, data kelahiran, data kematian, data perpindahan, dan data jumlah jiwa.

Menurut Lestari dkk (2021), administrasi kependudukan adalah rangkaian kegiatan penataan dan penerbitan dalam penerbitan dokumen dan data kependudukan melalui pendaftaran penduduk, pencatatan sipil, pengelolaan informasi administrasi kependudukan serta pendayagunaan hasilnya untuk pelayanan publik dan pembangunan sektor lain.

Berdasarkan pengertian administrasi kependudukan di atas dapat disimpulkan bahwa administrasi kependudukan merupakan suatu pengelolaan data yang berkaitan dengan informasi penduduk meliputi data kependudukan, data kelahiran, data kematian, data perpindahan, dan data jumlah jiwa. Kependudukan dan peristiwa penting memerlukan bukti yang sah untuk dilakukan pengadministrasian dan pencatatan sesuai dengan ketentuan undang-undang.

2.6. Metode Pengembangan Sistem

Model *prototype* dapat digunakan untuk menyambung ketidak pahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa & Shalahudin, 2018). Model *prototype* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model *Prototype*.

Sumber : (Rosa & Shalahudin, 2018)

Menurut (Rosa & Shalahudin, 2018) terdapat tahapan dalam proses *prototype* yaitu:

1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Membangun dan Memperbaiki *Mockup*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem yang disesuaikan dengan kebutuhan pengguna.

3. Menguji *Mockup*

Pada tahap ini, dilakukan pengujian *prototype* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan *prototype*, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih belum sesuai kembali ke tahap pertama.

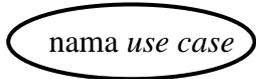
2.7. *Unified Modeling Language (UML)*

Alat pengembangan sistem yang digunakan pada penelitian ini yaitu *Unified Modeling Language (UML)*. *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahudin, 2018). Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *UML (Unified Modelling Language)*.

2.7.1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat (Rosa & Shalahudin, 2018).

Tabel 2.2 Simbol *Use Case Diagram*

No.	Simbol	Deskripsi
1	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar

Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)

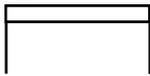
No.	Simbol	Deskripsi
		pesan antara unit atau aktor.
2	Aktor / <i>actor</i>  Actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i> <<include>> 	Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini menjalankan fungsionalnya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: (Rosa & Shalahudin, 2018)

2.7.2. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Hal yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Shalahudin, 2018).

Tabel 2.3 Simbol Activity Diagram

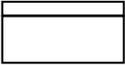
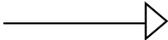
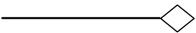
No.	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: (Rosa & Shalahudin, 2018)

2.7.3. Class Diagram

Class diagram adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem. *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan (Rosa & Shalahudin, 2018).

Tabel 2.4 Simbol Class Diagram

No.	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: (Rosa & Shalahudin, 2018)

2.8. Website

Menurut (Suprianto & Matsea, 2018) *website* adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.

Berdasarkan uraian, penulis menyimpulkan bahwa website adalah sebuah *software* yang berfungsi untuk menampilkan dokumen-dokumen pada suatu *web* yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet.

2.9. Freamwork CodeIgniter

CodeIgniter (CI) merupakan sebuah *framework* PHP (*Open Source*) atau gratis yang menerapkan metode Model View Controller (MVC) dalam pembuatan sebuah proyek web, dengan metode tersebut maka akan mempercepat pembangunan sebuah web. Selain itu CodeIgniter dilengkapi sistem keamanan yang baik dan cepat untuk dimuat di browser web. Berikut adalah pengertian *CodeIgniter* (CI) menurut para ahli: 15 Betha Sidik mengemukakan bahwa “*CodeIgniter* adalah sebuah framework php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal” (Destiningrum & Adrian, 2017).

CodeIgniter memiliki banyak fitur yang membantu para pengembang (*developer*) PHP untuk dapat membuat aplikasi web secara mudah dan cepat. *Framework CodeIgniter* memiliki desain yang lebih sederhana dan bersifat fleksibel (tidak kaku) dibandingkan dengan *framework* lainnya. *CodeIgniter*

mengizinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan.

CodeIgniter merupakan sebuah toolkit yang ditunjukkan untuk orang yang ingin membangun aplikasi web dalam bahasa pemrograman PHP. Beberapa keunggulan yang ditawarkan oleh *CodeIgniter* adalah sebagai berikut:

1. *CodeIgniter* adalah *framework* yang bersifat free dan open-source.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Selesai proses instalansi, *framework CodeIgniter* hanya berukuran kurang lebih 2MB (tanpa dokumentasi atau jika direktori user-guide dihapus). Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6MB.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.
4. *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu *file* tidak terlalu berisi banyak kode, hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara di kemudian hari.
5. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.

2.10. PHP

PHP adalah sekumpulan skrip atau bahasa program memiliki fungsi utama yaitu mampu mengumpulkan dan mengevaluasi hasil survey atau bentuk apapun keserver database dan pada tahap selanjutnya akan menciptakan efek beruntun, suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program dan mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML, juga merupakan bahasa pemrograman berbasis *server-side* yang dapat parsing *script php* menjadi *script*

web sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik (Rochman, Hakim dan Riswanto, 2018).

2.11. MySQL

Structured Query Language adalah suatu sistem basis data relation atau *Relational Database management System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan, sehingga sapat digunakan untuk aplikasi multi user (banyak pengguna). *MySQL* didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap program bebas menggunakan *MySQL* namun tidak bisa dijadikan produk turunan yang dijadikan *closed source* atau komersial (Rochman, Hakim dan Riswanto, 2018).

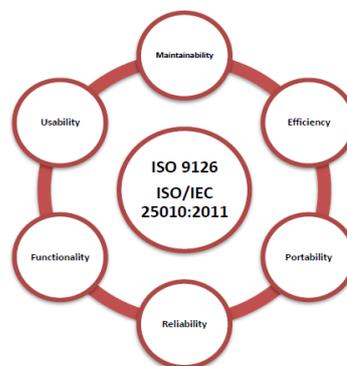
2.12. XAMPP

XAMPP adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan XAMPP ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Jadi fungsi dari XAMPP server itu sendiri merupakan server *website* kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakseskan web, untuk itu komputer kita harus menjadi server. Dapat disimpulkan *xampp* adalah aplikasi tools untuk menyediakan paket lunak yang berisi konfigurasi *Web Server, Apache, PHP, MySQL* untuk membantu dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web (Rochman, Hakim dan Riswanto, 2018).

2.13. Pengujian ISO 25010

Pada tahun 1991 *ISO 25010* diperkenalkan melalui definisi kualitas perangkat lunak (Rahayuda, 2017). *ISO 25010* adalah tolak ukur standar yang digunakan untuk mengukur analisis kualitas perangkat lunak yang digunakan oleh perusahaan, institusi atau organisasi. Ada enam karakteristik kualitas perangkat lunak utama *ISO 25010* yaitu:

- 1) *Functionality*, kemampuan untuk menutupi fungsi produk perangkat lunak yang memenuhi kebutuhan pengguna.
- 2) *Reliability*, kemampuan perangkat lunak untuk mempertahankan tingkat kinerja.
- 3) *Usability*, kemampuan yang berkaitan dengan penggunaan perangkat lunak.
- 4) *Efficiency*, kemampuan yang berkaitan dengan sumber daya fisik yang digunakan saat perangkat lunak dijalankan.
- 5) *Maintainability*, adalah kemampuan yang diperlukan untuk melakukan perubahan perangkat lunak.
- 6) *Portabilitas*, adalah kemampuan yang terkait dengan kemampuan untuk mengirim perangkat lunak ke lingkungan yang berbeda.



Gambar 2.6 Model Pengujian Perangkat Lunak ISO 25010
Sumber: (Rahayuda, 2017)

Masing-masing karakteristik kualitas perangkat lunak model ISO 25010 dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik Kualitas Perangkat Lunak Model *ISO 25010* dapat dilihat pada tabel 2.6:

Tabel 2.6 Karakteristik *ISO 25010*

Karakteristik	Sub Karakteristik	Deskripsi
<i>Functional Suitability</i>	<i>Functional Completeness</i>	Kemampuan perangkat lunak dalam menyediakan fungsi dapat mencakup semua tugas dan tujuan pengguna secara spesifik.
	<i>Functional Correctness</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan.
	<i>Functional Appropriateness</i>	Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik.
<i>Performance Efficiency</i>	<i>Resource Utilization</i>	Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria
	<i>Capacity</i>	Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria.
	<i>Time Behaviour</i>	Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria.
<i>Usability</i>	<i>Appropriateness Recognizability</i>	Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka.
	<i>Learnability</i>	Kemampuan perangkat lunak

Karakteristik	Sub Karakteristik	Deskripsi
		sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan.
	<i>Operability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu memenuhi kriteria dioperasikan dan dikontrol.
	<i>User Error Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.
	<i>User Interface Aesthetics</i>	Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna.
	<i>Accessibility</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks penggunaan
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal.
	<i>Availability</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen dapat beroperasi ketika diperlukan untuk digunakan.
	<i>Fault Tolerance</i>	Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen.
	<i>Recoverability</i>	Kemampuan perangkat lunak ketika produk atau sistem mengalami kejadian atau kegagalan dapat mengembalikan data dan

Karakteristik	Sub Karakteristik	Deskripsi
		menjalankan kondisi sistem yang diharapkan.
<i>Maintainability</i>	<i>Modularity</i>	Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain
	<i>Reusability</i>	Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.
	<i>Analyzability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.
	<i>Modifiability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.
	<i>Installability</i>	Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.
	<i>Replaceability</i>	Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada ketergantungan kepada

Karakteristik	Sub Karakteristik	Deskripsi
		perangkat lunak lain digunakan.

Sumber: (Rahayuda, 2017)

2.12. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sugiyono, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.7 di bawah ini.

Tabel 2.7 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Sumber : (Sugiyono, 2017)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat di bawah ini

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\% \quad \dots\dots\dots(1)$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.7 di bawah ini.

Tabel 2.7 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Sumber : (Sugiyono, 2017)