

BAB II LANDASANTEORI

2.1 Tinjauan Pustaka

Untuk mendukung penelitian ini, digunakan beberapa literatur yang berkaitan dengan judul dan pokok bahasan pada penelitian. Adapun literatur yang digunakan dapat dilihat pada Tabel 2.1 berikut

Tabel 2. 1 Tinjauan Pustaka

No	Penulis, Tahun	Judul	Hasil
1	(Haryadi & Yanto, 2018)	Sistem Informasi Berbasis Web Pada SMP Negeri 1 Sungai Raya	Pada penelitian ini dibangun sistem informasi pendaftaran siswa baru berbasis web pada sekolah SMK Multimedia Mandiri Jakarta untuk dapat menjangkau siswa dari luar daerah. Sistem dibangun dengan alat pengembangan sistem <i>Extreme Programming</i> , pengembangan perangkat lunak digambarkan dengan model UML dan pengujian nya menggunakan <i>Blackbox</i> .
2	(Sholatul et al., 2019)	Sistem Informasi Sekolah SMA (Swasta) Berbasis Web di SMA Fajar Dunia	Pada penelitian ini dibangun sistem informasi sekolah berbasis web yang memudahkan guru untuk melihat nilai dan mata pelajaran siswa sehingga guru tidak perlu melihat mata pelajaran di kertas tapi melainkan guru bisa melihat di tampilan web. Sistem dibangun dengan alat pengembangan sistem <i>waterfall</i> , pengembangan perangkat lunak digambarkan dengan model UML dan menggunakan metode pengujian <i>black box testing</i> .

Tabel 2. 2 Tinjauan Pustaka (lanjutan)

No	Penulis, Tahun	Judul	Hasil
3	(Rizqullah et al., 2020)	Sistem Inforasi Pelayanan dan Pelaporan Siswa SMAN 5 Mataram	Pada penelitian ini dibangun sistem inforasi pelayanan dan pelaporan siswa SMAN 5 Mataram yang dapat membantu sekolah dalam menyediakan layanan dan laporan dari siwa untuk meningkatkan kualitas sekolah. Sistem dibangun dengan alat pengembangan sistem <i>waterfall</i> , pengembangan perangkat lunak digambarkan dengan model UML, <i>framework</i> yang gunakan <i>Laravel</i> .
4	(Nabila & Wati, 2022)	Implementasi <i>Framework Codeigniter</i> Pada Sistem Informasi Pendataan Prestasi Akademik dan Non-akademik Siswa SMA Negeri 4 Cibinong Berbasis Web	Pada penelitian ini dibangun sistem informasi pendataan prestasi akademik dan non-akademik siswa SMA Negeri 4 Cibinong berbasis web agar pengguna dapat langsung memasukkan data prestasi dan memudahkan admin dalam pengelolaan dan pelaporan data prestasi. Sistem dibangun dengan alat pengembangan sistem <i>waterfall</i> , pengembangan perangkat lunak digambarkan dengan model UML, <i>framework</i> yang gunakan <i>Codeigniter</i> .
5	(Budiarti & Risyanto, 2020)	Impelementasi Metode <i>Extreme Programming</i> Untuk Merancang Sistem Informasi Pendaftaran Siswa Baru Berbasis Web Pada Sekolah SMK Multimedia Mandiri Jakarta	Pada penelitian ini dibangun suatu sistem yang dapat menyampaikan informasi SMP Negeri 1 Sungai Raya untuk membantu orang tua dalam menentukan sekolah yang sesuai dengan putra-putri mereka. Sistem dibangun dengan alat pengembangan sistem <i>research and development</i> , pengembangan perangkat lunak digambarkan dengan model UML .

Yang pertama oleh Dody Arief Haryadi dan Hendri Yanto dari Universitas Mataram dengan judul perancangan sistem informasi berbasis web pada SMP Negeri 1 Sungai Raya. Penelitian yang dilakukan oleh penulis karena belum ada sistem informasi pada sekolah SMP Negeri 1 Sungai Raya. Pada penelitian yang dilakukan, dijelaskan bahwa Penelitian ini dilakukan dengan menggunakan pendekatan “Penelitian Pengembangan” (*Research and Development*) adalah strategi pengembangan sistem yang menekankan kecepatan dalam pengembangan melalui keterlibatan pengguna dalam pembangunan secara cepat, interaktif, dan inkremental dari suatu serangkaian *prototype* dari suatu sistem yang dapat berkembang menjadi suatu sistem akhir atau versi tertentu.

Kemudian penelitian yang dilakukan oleh Nanda Sholatul Akbar, Dwi Wardiyanto, Marsya Geby Diamanti, dan Lydia Vintari, dari Mercu Buana dengan judul sistem informasi sekolah SMA (Swasta) berbasis web di SMA Fajar Dunia. Penelitian yang dilakukan oleh penulis dilatar belakangi sistem informasi penilai dan membuat *invoice* dilakukan secara manual, dalam keseharian masih ditemukan masalah-masalah yang terjadi terutama pada lamanya proses pencatatan pada penelitian yang dilakukan, dijelaskan bahwa pembuatan sistem dilakukan dengan menggunakan metode *waterfall*. *Model waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Hal ini disebut *waterfall* karena proses mengalir "secara sistematis dari satu tahap ke tahap lainnya dalam mode ke bawah" (*Pressman, 2012*). Metode ini mempunyai struktur yang jelas dan terarah dalam setiap tahapan perancangan dan implementasinya

Penelitian yang dilakukan oleh Muhammad Naufal Rizqullah, I Gede Pasek Suta Wijaya dan Novian Maududi dari Universitas Mataram dengan judul sistem informasi pelayanan dan pelaporan siswa SMAN 5 Mataram. Penelitian yang dibahas oleh penulis untuk memudahkan sekolah dalam mendapatkan laporan dari siswa, perlu untuk dibuat sistem informasi yang dapat digunakan sebagai media bagi siswa untuk menyampaikan laporan atau keluhan. Pada penelitian yang dilakukan digunakan metode *waterfall*. Metode *waterfall* digunakan sebagai dasar pengembangan sistem yang akan dibangun. Metode *waterfall* memungkinkan pengerjaan sistem yang dilakukan secara linear atau secara berurutan. Setiap langkah-langkah pengerjaan harus dikerjakan berurutan tanpa bisa dilewati. Apabila langkah pertama belum dikerjakan, maka sangat tidak memungkinkan untuk mengerjakan langkah selanjutnya. Sehingga untuk menyelesaikan pengembangan sistem, sebuah keharusan bahwa penyelesaian setiap langkah pekerjaan harus diselesaikan dengan tuntas.

Oleh Suci Nabila dan Theresiawati dari Universitas Pembangunan Nasional Veteran Jakarta dengan judul implementasi *framework codeigniter* pada sistem informasi pendataan prestasi akademik dan non-akademik siswa SMA Negeri 4 Cibinong berbasis web. Penelitian ini dilakukan karena proses pendataan prestasi baik secara akademik maupun non-akademik siswa di SMA Negeri 4 Cibinong masih mengalami kendala, yaitu pendataan yang dilakukan belum berjalan secara optimal karena masih bersifat konvensional. Hal ini tentunya memiliki beberapa keterbatasan, mengakibatkan proses pendataan yang perlu melalui beberapa tahapan dan membutuhkan waktu yang lama, tidak tersedianya penyajian data

prestasi akademik dan non-akademik siswa, dan penyimpanan data yang tidak terorganisir.

Oleh Yusnia Budiarti dan Risyanto dari STMIK Mandiri Jakarta dengan judul implementasi metode *extreme programming* untuk merancang sistem informasi pendaftaran siswa baru berbasis web pada sekolah SMK Multimedia Mandiri Jakarta. Masalah pokok penelitian sekolah masih menggunakan sistem pendaftaran siswa baru secara manual, dimana calon siswa harus datang langsung terjadi masalah seperti tidak terbacanya tulisan tangan calon siswa baru yang menyebabkan kesalahan data siswa serta hilangnya dokumen-dokumen penting pendaftaran siswa di sekolah karna tidak didokumentasikan dengan baik. Pada penelitian yang dilakukan menggunakan metode *extreme programming*. *Extreme programming* adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel.

Adapun perbedaan sistem yang akan dibangun pada SDN 6 Gedong Air dengan penelitian terdahulu antara lain, pertama perbedaannya terletak pada alat pengembangan sistem pada penelitian kali ini alat pengembangan sistem yang akan digunakan adalah *extreme programming*, yang kedua sistem yang akan dibangun pada SDN 6 Gedong Air berfokus pada informasi pelayanan seperti informasi *profil* sekolah, agenda sekolah, fasilitas sekolah, prestasi sekolah dan ada penambahan fitur seperti pendaftaran peserta didik baru, yang terakhir perbedaan dari segi tampilan, dari segi tampilan pada sistem yang akan dibangun akan dibuat design yang enak dilihat dan tidak membuat bingung pengguna.

2.2 Sistem

Menurut Abdul Kadir (2018), Sistem adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai

2.3 Informasi

Informasi adalah sekumpulan data yang bersumber dari fakta-fakta dan diolah sedemikian rupa sehingga menjadi lebih bermanfaat bagi yang menggunakannya. Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi yang membutuhkan dan bermanfaat bagi pengambilan keputusan saat ini atau yang akan mendatang (Hasan & Muhammad, 2020).

2.4 Sistem Informasi

Menurut Abdul Kadir (2018), Sistem Informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu diproses (data menjadi informasi), dan dimaksudkan untuk mencapai tujuan tertentu.

2.5 Pelayanan

Menurut Kotler (Laksana, 2018:85), pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh satu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun.

2.6 Sekolah

Menurut Sunarto dalam buku yang ditulis oleh Abdullah (2011) juga, pada saat ini kata sekolah telah berubah artinya menjadi bangunan atau lembaga untuk belajar dan mengajar serta tempat memberi dan menerima pelajaran.

2.7 *Extreme Programming*

Extreme Programming (XP) adalah metodologi dalam pengembangan rekayasa perangkat lunak dan juga merupakan satu dari beberapa *agile software*

development methodologies yang berfokus pada *coding* sebagai aktivitas utama di semua tahap *software development lifecycle*. (Gumelar, Astuti & Sunarni, 2017). *Extreme programming* (XP) yang dikenal dengan metode *technical how to* atau bagaimana suatu tim mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik. (Carolina & Rusman, 2019).

Menurut Carolina & Rusman, 2019 *Extreme programming* memiliki 4 tahap proses yaitu :

1. Planning

Yaitu tahap perencanaan dimulai dengan membuat *user stories* yang menggambarkan output, fitur, dan fungsi-fungsi dari software yang akan dibuat. *User stories* tersebut kemudian diberikan bobot seperti prioritas dan dikelompokkan untuk selanjutnya dilakukan proses *delivery* secara *incremental*.

2. Design

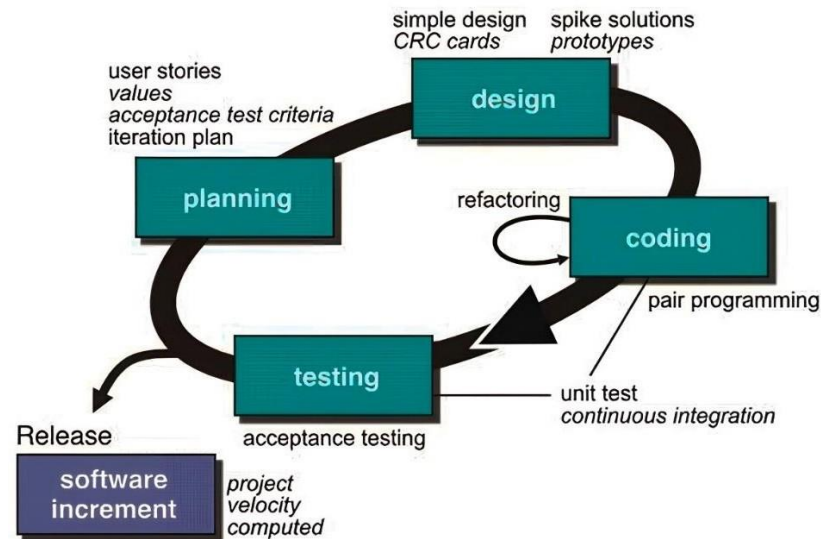
Tahap *design* pada *extreme programming* mengikuti prinsip *keep in simple* (KIS). Dimana untuk penggunaan *design* yang sulit, *extreme programming* menggunakan *spike solution*. *Extreme programming* juga mendukung adanya *refactoring*.

3. Coding

Proses ini diawali dengan membangun serangkaian unit test. Lalu pengembang akan berfokus untuk mengimplementasikannya. Dalam XP diperkenalkan *pair programming*.

4. Testing

Pada tahap ini dilakukan pengujian kode pada unit test. Dalam XP, diperkenalkan XP *acceptance test* atau *customer test*. Test ini dilakukan oleh *customer* yang berfokus kepada fitur dan fungsi sistem secara keseluruhan.



Gambar 2. 1 Tahapan proses *Extreme Programming*
(Sumber : Carolina & Rusman, 2019)

2.8 Framework

Framework adalah kumpulan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu (Basuki, 2017).

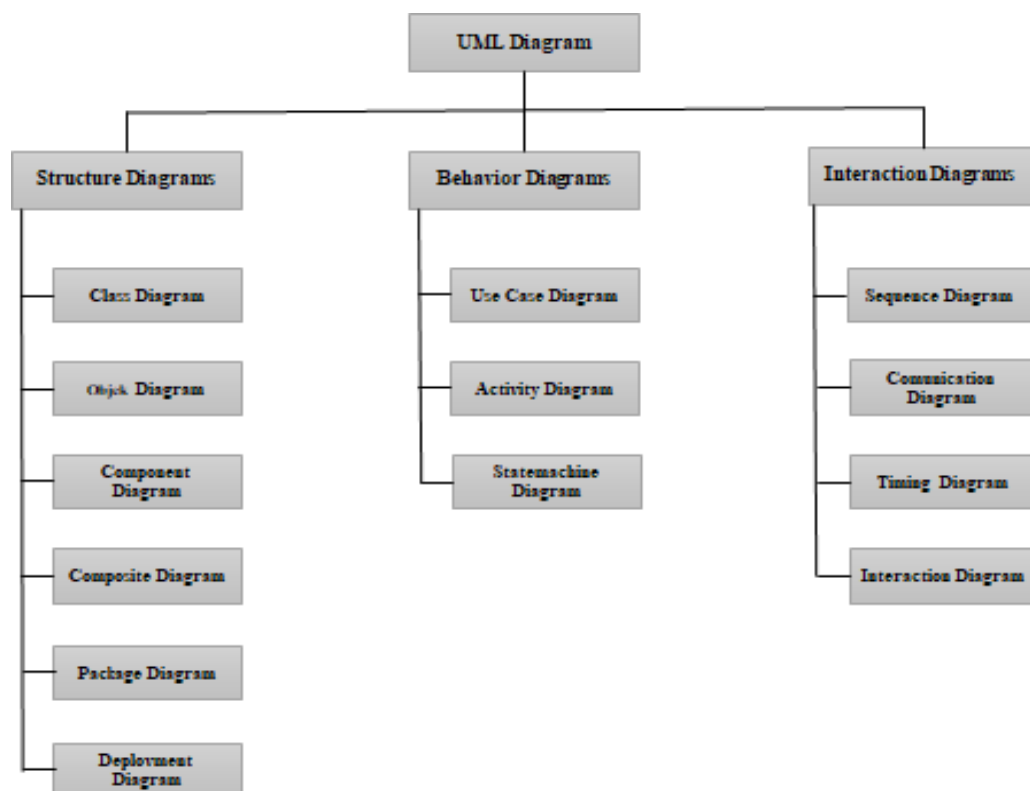
2.9 Codeigniter

Codeigniter adalah *framework* aplikasi web yang open source untuk bahasa pemrograman PHP. Dari sisi pemrograman, *Codeigniter* kompatibel dengan PHP4 dan PHP5, sehingga akan berjalan dengan baik pada web host yang banyak dipakai saat ini. *Codeigniter* menggunakan pola desain model *View-Controller*

(MVC), *Codeigniter* adalah kerangka kerja pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur. *Codeigniter* memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti helpers and libraries untuk mengimplementasi tugas yang biasa dilakukan. (Arrhioui et al., 2017)

2.10 Unified Modeling Language (UML)

UML adalah bahasa pemodelan yang digunakan dalam pengembangan sebuah perangkat lunak. *UML* digunakan dengan memodelkan atau menganalisis serta merancang kerangka didalam pemograman, digunakan sebagai bentuk visualisasi dari perangkat lunak yang akan dibangun. Dalam *UML* dibagi menjadi tiga kategori dan terdapat tiga belas jenis diagram didalamnya (Julianto and Setiawan, 2019). Berikut pembagian kategori dapat dilihat pada gambar dibawah ini.

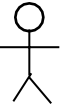
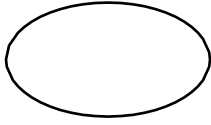
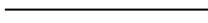
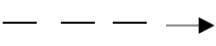



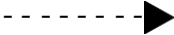
Gambar 2. 2 UML (Unified Modeling Language)
(Sumber : (Munawar, 2018))

2.10.1 Use Case Diagram

Use case adalah sebuah diagram yang menggambarkan atau pemodelan dari perangkat lunak yang akan dibangun. *Use Case* berguna untuk mendeskripsikan kegiatan yang dilakukan antara aktor satu dengan yang lain yang berhubungan didalam sistem, *Use Case* juga digunakan sebagai pemetaan terhadap akses yang diberikan (Julianto and Setiawan, 2019).

Tabel 2. 3 Use Case Diagram

No	Simbol	Keterangan
1.	Aktor/Actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibangun
2.	Use Case 	Fungsionalitas yang disediakan sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
3.	Association 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Extend/Ekstensi 	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.	Generalization/Generalisasi 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.




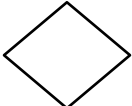

6.	Menggunakan/ <i>Include/Uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat.
----	---	---

Sumber : (Munawar, 2018)

2.10.2 Activity Diagram

Activity diagram adalah yang menggambarkan alur kegiatan atau aktifitas yang terjadi di dalam sistem yang akan dibangun. Berikut simbol yang terdapat pada *activity diagram* (Julianto and Setiawan, 2019)

Tabel 2. 4 Activity Diagram

No	Simbol	Keterangan
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
3.	Aktifitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
4.	Percabangan/ <i>Decision</i> 	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
5.	Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi 1.

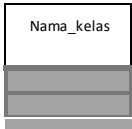
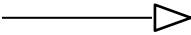
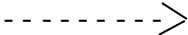
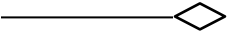
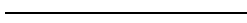
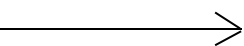
Sumber : (Munawar, 2018)

2.10.3 Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dimulai dari kelas-kelas yang ada dan kelas tersebut memiliki atribut dan operasi

didalamnya. Berikut dapat dilihat dari simbol dari *class diagram* (Julianto and Setiawan, 2019).

Tabel 2. 5 Class Diagram

No	Simbol	Keterangan
1.	<p>Kelas</p> 	Kelas pada struktur sistem
2.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-generelisasi-spesialisasi (umum khusus).
3.	<p>Kebergantungan/<i>Dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
4.	<p>Agregasi/<i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).
5.	<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
6.	<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai <i>multiplicity</i> .

Sumber : (Munawar, 2018)

2.11 Black Box

Black Box Testing (Pengujian kotak hitam) yaitu bertujuan untuk menunjukkan fungsi PL tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemukhtahirannya (Nurajizah & Aziz, 2019).

Sedangkan pengertian lain tentang *Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak (Jaya, 2018). Menurut Pressman dalam (Khasanah, Kesuma, & Wijianto, 2018) “*black box testing* merupakan pengujian yang memungkinkan *software engineer* mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program”.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori menurut (Suryawan, Prasetya, Budiawan, Nip, & Nipt, 2019) sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang.
2. Kesalahan *Interface*.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan Kinerja.
5. Inisialisasi dan kesalahan terminasi

Kelebihan yang dimiliki oleh *Black Box Testing* :

1. Dapat memilih subset test secara efektif dan efisien
2. Dapat menemukan cacat
3. Memaksimalkan *testing investmen*.

2.12 ISO 25010

ISO/IEC 25010 adalah standar model kualitas dari sebuah sistem dan perangkat lunak yang menggantikan *ISO/IEC 9126* tentang *software engineering*.

Dikutip dari (P. Miguel, Mauricio and Rodríguez, 2014) Menurut IEE *Standard Glossary of Software Engineering Terminology* kualitas produk software didefinisikan sebagai

1. Sejahtera mana suatu sistem, komponen atau proses memenuhi persyaratan yang ditentukan.
2. Sejahtera mana suatu sistem, komponen atau proses memenuhi kebutuhan atau harapan pengguna.

Karakteristik ISO/IEC 25010 ini dibagi menjadi beberapa karakteristik antara lain yaitu :

2.12.1 *Functional Suitability*

Karakteristik sejahtera mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi dan keadaan tertentu. *Functional Suitability* memiliki sub karakteristik yaitu :

1. *Functional completeness*, sejahtera mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
2. *Functional Correctness*, sejahtera mana produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.
3. *Functional Appropriateness*, sejahtera mana fungsi memfasilitasi pencapaian tugas dan tujuan tertentu.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.12.2 *Reliability*

Sejahtera mana suatu sistem, produk atau komponen melakukan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Karakteristik ini terdiri dari sub karakteristik berikut :

1. *Maturity*, sejahtera mana sistem, produk, atau komponen beroperasi dan dapat diakses saat diperlukan untuk digunakan.

2. *Availability*, sejauh mana sistem, produk atau komponen beroperasi dan dapat diakses saat diperlukan untuk digunakan.
3. *Fault Tolerance*, sejauh mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun ada kesalahan perangkat keras atau perangkat lunak.
4. *Recoverability*, sejauh mana dalam hal gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan membangun kembali keadaan sistem yang diinginkan.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.12.3 Performance Efficiency

Karakteristik ini mewakili kinerja relative terhadap jumlah sumber daya yang akan digunakan dalam kondisi yang dinyatakan. Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Time Behaviour*, sejauh mana respon dan waktu pemrosesan serta tingkat input suatu produk atau sistem saat menjalankan fungsinya untuk memenuhi persyaratan.
2. *Resource Utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem ketika menjalankan fungsinya.
3. *Capability*, sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.12.4 Usability

Sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam penggunaannya. Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Appropriateness recognizability*, sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai untuk kebutuhan mereka.
2. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan pembelajaran menggunakan produk atau sistem dengan efektivitas, efisiensi, kebebasan dari risiko yang didapat dan tingkat kepuasan dalam ranah penggunaan tertentu.
3. *Operability*, sejauh mana produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.
4. *User Error Protection*, sejauh mana sistem melindungi pengguna dari membuat kesalahan.
5. *User Interface Aesthetics*, sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
6. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh orang-orang dengan jangkauan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam ranah penggunaan tertentu.

2.12.5 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 otorisasi mereka. Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Confidentiality*, sejauh mana produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
2. *Integrity*, sejauh mana sistem, produk atau komponen mencegah akses tidak sah masuk ke dalam atau memodifikasi program atau *database*.
3. *Non-repudation*, sejauh mana tindakan dapat dibuktikan telah terjadi sehingga peristiwa atau Tindakan tidak dapat dihindari.
4. *Accountability*, sejauh mana Tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
5. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat dibuktikan sebagai yang diklaim.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.12.6 Compatibility

Sejauh mana suatu produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen lain dan menjalankan fungsi yang diperlukan sambal berbagai lingkungan perangkat keras atau perangkat lunak yang sama. Berikut karakteristik ini terdiri dari berikut :

1. *Co-existence*, sejauh mana suatu produk dapat melakukan fungsi yang diperlukan secara efisien sambal berbagai lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak yang merugikan pada produk lain.

2. *Interoperability*, sejauh mana dua atau lebih sistem produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.14.7 Maintainability

Karakteristik ini mewakili tingkat efektivitas dan efisiensi dimana produk atau sistem dapat dimodifikasi untuk memperbaikinya atau menyesuaikannya dengan perubahan lingkungan dan persyaratan. Karakteristik ini terdiri dari sub karakteristik yaitu :

1. *Modularity*, sejauh mana sistem atau program komputer terdiri dari komponen-komponen sedemikian rupa sehingga perubahan pada satu komponen memiliki dampak minimal pada komponen lainnya.
2. *Reusability*, sejauh mana asset dapat digunakan di lebih satu sistem.
3. *Analysability*, tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari perubahan yang dimaksudkan untuk satu atau lebih bagiannya atau untuk mendeteksi suatu produk untuk kekurangan atau penyebab kegagalan untuk mengidentifikasi bagian untuk dimodifikasi.
4. *Modifiability*, sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan kekurangan atau menurunkan kualitas produk yang ada.
5. *Testability*, tingkat efektivitas dan efisiensi dengan kriteria pengujian yang dapat ditetapkan untuk sistem produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.14.8 *Portability*

Tingkat efektifitas dan efisiensi dimana sistem produk atau komponen dapat ditransfer dari satu perangkat keras, lunak atau operasional dan penggunaan lainnya. Karakteristik ini terdiri dari sub yaitu :

1. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat.
2. *Installability*, tingkat efisiensi dimana produk atau sistem dapat berhasil dapat berhasil dipasang dan dilepas di lingkungan tersebut.
3. *Replaceability*, sejauh mana suatu produk dapat menggantikan sistem yang sudah ada.

(P. Miguel, Mauricio and Rodríguez, 2014).

2.13 *Skala Likert*

Menurut (Sugiyono 2018), skala *likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok tentang kejadian sosial. Dengan *skala likert*, *variabel* yang akan diukur dijabarkan ke dalam *variabel indikator*. *Variabel indikator* kemudian digunakan sebagai kriteria untuk mengelompokkan item alat yang dapat berupa data atau pertanyaan. Berikut adalah kategori penilaian *skala likert* pada penilaian *usability* yaitu "Sangat Setuju "SS" bernilai 5 (lima), Setuju "ST" bernilai 4 (empat), Ragu-Ragu "R" bernilai 3 (tiga), Tidak Setuju "TS" bernilai 2 (dua) dan Sangat Tidak Setuju "STS" bernilai 1 (satu). Skala Likert dapat dilihat pada Tabel 2.17

Table 2.1 Skala Likert

Jawaban	Skor
Sangat Setuju (SS)	5

Setuju (S)	4
Ragu-Ragu	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Sumber : (Sanusi, 2018)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini $persentase = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$. *Persentase* kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.8

Table 2. 2 Skala Konversi Nilai

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

Sumber : (Sanusi, 2018)