

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk menunjang penelitian ini, peneliti menggunakan beberapa literatur yang berkaitan dengan judul dan pokok pembahasan pada penelitian. Literatur yang telah ditinjau oleh peneliti dapat dilihat pada tabel 2.1 berikut:

Tabel 2.1 Tinjauan Pustaka

Nomor		Detail Jurnal
1	Judul	<i>Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android</i>
	Tahun Terbit	2020
	Penulis	Imam Ahmad, Rohmat Indra Borman, Jafar Fakhrurozi, Gavan Gorbi Caksana
	Metode Penelitian	<i>Extreme Programming</i>
	Hasil	Pada penelitian (I. Ahmad et al., 2020) dilakukan pengembangan aplikasi deteksi kemiripan judul skripsi dengan Pendekatan pengembangan <i>software XP</i> , Tahapan Pengembangan <i>XP</i> mampu membantu dalam menghasilkan <i>software</i> yang sesuai dengan fungsional <i>requirement</i> , Penelitian ini menghasilkan aplikasi berupa

		persentase kemiripan judul berdasarkan dengan judul skripsi sebelumnya yang diambil dari basis data skripsi yang ada, dengan pendekatan pengembangan sistem XP dan pencocokan kata menggunakan <i>Boyer-Moore</i> berbasis <i>Andorid</i> .
Nomor		Detail Jurnal
2	Judul	Pemanfaatan Teknologi <i>RFID</i> Untuk Pengelolaan Inventaris Sekolah Dengan Metode (R&D)
	Tahun Terbit	2019
	Penulis	S Fransisca, RN Putri
	Metode Penelitian	Metode R&D (<i>Research and development</i>)
	Hasil	Pada hasil penelitian (Fransisca & Putri, 2019) adalah pengelolaan data inventaris yang ada di sekolah menggunakan <i>RFID</i> dan <i>bar Code</i> membuat pencarian data laporan menjadi efektif, laporan yang telah dibuat mampu di simpan pada sistem sehingga dapat menghemat biaya pengeluaran dan juga dengan sistem yang telah di buat dapat mengurangi kesalahan pada saat melakukan pendataan inventaris sekolah
Nomor		Detail Jurnal

3	Judul	Rancang Bangun Sistem Pengelolaan Arsip Surat Berbasis <i>Web</i> Menggunakan Metode <i>Waterfall</i>
	Tahun Terbit	2019
	Penulis	Ade Suryadi, Yuli Siti Zulaikhah
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil	Hasil penelitian (Suryadi & Zulaikhah, 2019) adalah sistem informasi yang di bangun mengenai arsip surat berbasis <i>web</i> mampu meningkatkan kinerja karyawan sehingga pencarian arsip surat lebih akurat dan efisien. sistem dapat mendokumentasikan aktivitas surat masuk dan surat keluar sehingga dapat terpantau kegiatan karyawan.
Nomor		Detail Jurnal
	Judul	Perancangan Sistem Informasi Manajemen Kegiatan Mahasiswa Berbasis <i>Web</i>
	Tahun Terbit	2019
	Penulis	Ana Juita Oktasari, Denny Kurniadi
	Metode Penelitian	Model <i>waterfall</i>
	Hasil	pada hasil penelitian (Oktasari & Kurniadi, 2020) adalah sistem yang di buat mampu menyediakan

4		<p>pengelolaan penyimpanan data mengenai laporan kegiatan mahasiswa dengan baik, dari memasukkan kegiatan UKM (Unit Kegiatan Mahasiswa) hingga mengelompokkan keseluruhan kegiatan menjadi sebuah laporan.</p>
Nomor		Detail Jurnal
5	Judul	Sistem Informasi Manajemen Aset Sekolah Tinggi Teknologi Pagaralam Berbasis <i>Web</i>
	Tahun Terbit	2018
	Penulis	Yogi Isro' Mukti
	Metode Penelitian	<i>Web Engineering</i>
	Hasil	Pada hasil penelitian (Isro'Mukti, 2018) adalah sistem aplikasi <i>web</i> yang di buat memudahkan sekolah untuk melakukan pengawasan dan pelaporan aset data sekolah. aset sekolah yang baru atau sudah tidak terpakai dapat dengan mudah terpantau.
Nomor		Detail Jurnal
	Judul	Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis <i>Web</i> (Studi Kasus Koperasi Mitra Setia)

6	Tahun Terbit	2017
	Penulis	IGT Isa, GP Hartawan
	Metode Penelitian	Metode Berorientasi Objek
	Hasil	Pada hasil penelitian (Isa & Hartawan, 2017) adalah pengelolaan pelaporan berbasis aplikasi <i>web</i> mampu meningkatkan kinerja koperasi, dengan sistem yang terkomputerisasi dengan basis data, sehingga pencatatan data pekerjaan lebih akurat dan aman.

Kesimpulan yang dapat oleh peneliti yang di ambil dari tinjauan literatur tersebut dengan penelitian peneliti adalah perbedaan pada metode pengembangan sistem dan *platform* yang di gunakan. Pada penelitian yang akan diteliti, peneliti menggunakan metode pengembangan *Extreme Programming (XP)* sebagai metode pengembangan penelitiannya dan *website* sebagai *platform* untuk berdirinya sistem yang dibangun, karena selain bersifat fleksibel metode ini juga dapat menyederhanakan dan mempercepat dalam proses saat pengembangan sistem, dan untuk pengujian sistem penulis menggunakan pengujian ISO 25010.

2.2 Sistem Informasi Manajemen

Penerapan teknologi dibidang pemerintahan dapat memunculkan alur kerja yang baru, yaitu sistem informasi manajemen yang tersusun baik dengan teknologi saat ini. dengan mekanisme interaksi sistem antara pemerintah dengan masyarakat

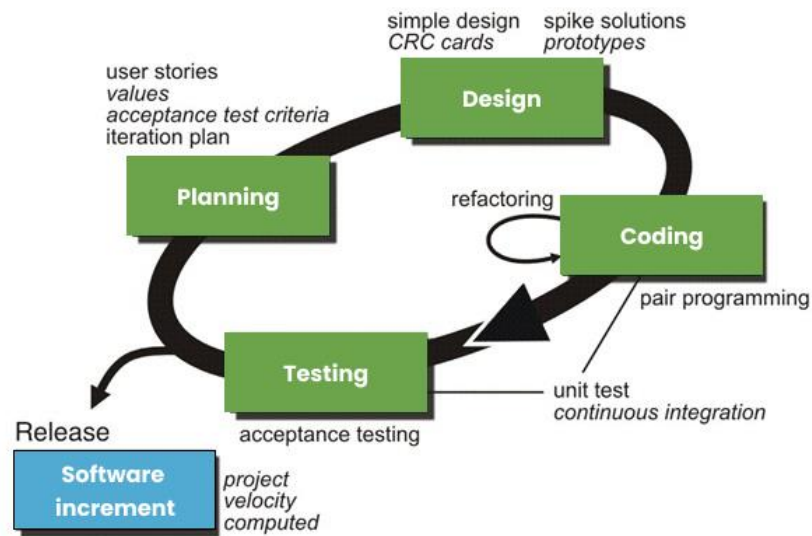
dan pemangku kepentingan lainnya, yang melibatkan penggunaan teknologi informasi khususnya internet, untuk meningkatkan kualitas layanan (Indrajit, 2016). Sistem informasi manajemen mampu meningkatkan kinerja produktivitas dan efisiensi di bidang pemerintahan, dan juga mampu mendorong inovasi lainnya yang akan di kembangkan di masa depan (Salju et al., 2016).

2.3 Metode Pengembangan Sistem

Menggunakan metode pengembangan sistem adalah bagian yang penulis gunakan sebagai alur kerja proses dalam pengembangannya sehingga penelitian dapat dikembangkan sesuai Tahap metode pengembangan sistem.

2.3.1 Metode *Extreme Programming*

Menurut (Suryantara et al., 2017) *Extreme programming* adalah sebuah metode Pengembangan perangkat lunak untuk meningkatkan perkembangan teknologi perangkat lunak dan kemampuan untuk menanggapi perubahan kebutuhan pelanggan. Dalam pengembangan sistem menggunakan metode ini ada empat langkah yang harus di lalui, yaitu terdiri dari perencanaan, desain, *coding* dan yang terakhir testing. Berikut adalah gambar kerangka dari cara kerja metode *Extreme Programming* (XP) dapat dilihat pada gambar 2.1.



Gambar 2.1 Tahapan Metode *Extreme Programming*

1. Perencanaan (*Planning*)

Tahapan perancangan (*Planning*) merupakan tahapan dasar pada metode xp yang dilakukan untuk menetapkan sistem yang akan dibuat. Tahapan ini sangat penting dilakukan karena untuk membuat sistem yang sesuai dengan kebutuhan dan perancangan yang sesuai oleh pengguna. Sistem yang direncanakan akan menentukan bagaimana pengembangan selanjutnya yang sesuai dengan kesepakatan yang telah disepakati, adapun berbagai cara agar mendapatkan perencanaan yang diinginkan dengan cara :

- a) Pengguna memberikan penjelasan sistem yang telah ada atau sedang berjalan dan memberikan gambaran bagaimana sistem yang baru.
- b) Kemudian dari hasil penjelasan pengguna yang telah di paparkan peneliti mendapatkan inti nilai pada sistem yang sudah berjalan dan sistem yang akan dibangun, sehingga peneliti mampu memutuskan sistem yang akan dibangun.

- c) Kemudian dari hasil kesepakatan nilai - nilai yang di dapat, peneliti menentukan *acceptance test criteria* yaitu sebuah kriteria keputusan yang akan menjadi acuan terhadap sistem yang akan dibangun.
- d) Setelah itu peneliti dapat menyimpulkan akan melakukan tahapan-tahapan dalam *relase* dan perbaikan pada tahap *iteration plan* untuk menetapkan pengujian terhadap sistem yang dibangun.

2. Desain (*Design*)

Tahapan selanjutnya setelah perancangan, maka peneliti akan melanjutkannya ke tahap desain. Pada tahapan pengembangan desain ini peneliti dimulai dengan merancang pemodelan sistem yang ingin dibuat, dan kemudian memberikan rancangan alur sistem dan permodelan berbasis data untuk menggambarkan sistem yang akan dibangun.

- a) Design yang dibuat akan menggunakan desain sederhana yang diawali dengan pembuatan UML dan perancangan desain lainnya seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.
- b) Tahapan desain metode XP menggunakan *CRC Card*, yaitu untuk mengetahui dan mengatur alur kerja OOC (*Object Oriented Class*) sesuai dengan sistem yang akan dibangun. adapun alternatif atau cara lainnya jika pada perancangan desain tidak sesuai dengan kebutuhan pengguna maka dapat dilakukan dengan.
- c) *Spike Solution*, yaitu pada saat penerapan desain yang akan dibuat menjadi sangat sulit. Maka metode XP akan menggunakan *Spike Solution* yang dimakan sistem yang akan dibuat menyesuaikan langsung dengan kebutuhan inti dari adanya sistem tersebut.

d) *Prototype* yaitu suatu bagian perancangan berupa *user interface* dalam bentuk *wireframing* yang bertujuan untuk mempermudah pengguna dalam melihat desain sistem.

3. Pengkodean (*Coding*)

Tahapan Pengkodean, peneliti akan membuat sistem sesuai dengan kebutuhan pengguna sesuai dengan kesepakatan Perencanaan (*Planning*) dan desain yang telah dibuat. Pada proses pengkodean berikut ini adalah hal yang dapat dilakukan :

- a) *Pair Programming* yaitu tahapan pengembangan sistem sesuai dengan bahasa pemrograman dan media penyimpanan yang sudah disepakati.
- b) *Refactory* merupakan bagian tahapan yang dilakukan jika terjadi hal yang tidak terduga pada pengkodean sistem yang dibangun sehingga dapat dilakukannya perbaikan sehingga sistem yang dibuat dapat sesuai kebutuhan.

4. Pengujian (*Testing*)

Tahapan Pengujian yang akan dilakukan oleh pengguna yang akan dilakukan dengan berbagai pengujian yang sesuai dengan *acceptance test* yang telah di tentukan dan disetujui, pengujian sistem yang dilakukan akan berfokus pada keseluruhan fitur dan kegunaan sistem. Kemudian setelah sesuai dengan kebutuhan dan kesepakatan maka sistem dapat di *release*.

2.4 Unified Modeling Language (UML)


Menurut (T. Ahmad et al., 2019), *Unified Modeling Language (UML)* merupakan sebuah penggambaran atau suatu hal yang divisualisasikan sehingga dapat mudah dipahami dari perilaku sistem yang telah dibuat. UML adalah sebuah bahasa yang digunakan untuk menggambarkan model dari sistem. maka dari itu UML hanya digunakan untuk membuat model. penggunaan UML tidak terbatas dengan metode tertentu, namun UML lebih diutamakan dalam metode berorientasi *object*.

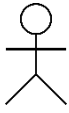



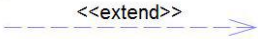
Terdapat tiga diagram *Unified Modeling Language (UML)* dengan fungsi yang berbeda :

2.4.1 Use Case Diagram

Menurut (Chonoles, 2017), *Use case* merupakan sebuah perilaku atau *behavior* dalam *Unified Modeling Language (UML)* dengan menggambarkan sebuah kasus yang terdapat fungsional dari pengembangan sistem yang dibuat. *use case* digunakan untuk menggambarkan alur kerja atau fungsi apa saja yang terdapat pada sistem yang dapat digunakan oleh pengguna dan sistem itu sendiri. berikut ini adalah simbol yang digunakan dalam penggambaran *use case* diagram dapat dilihat pada tabel 2.2 berikut :

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Keterangan
1		<i>Use case</i> merupakan fungsionalitas yang disediakan oleh sistem sebagai unit untuk bertukar pesan antar unit atau aktor. Biasanya, <i>use case</i> diberi nama

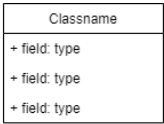

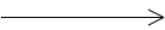



		menggunakan kata kerja di awal frase nama <i>use case</i> .
2		Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang dihasilkan, sehingga meskipun simbol agen adalah gambar seseorang, agen belum tentu orang. Biasanya aktor diberi nama menggunakan nama di awal kalimat nama aktor.
3		Asosiasi merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi dalam diagram <i>use case</i> atau <i>use case</i> yang berinteraksi dengan aktor. Asosiasi adalah simbol yang digunakan untuk menghubungkan <i>link</i> antar elemen.
4		Generalisasi merupakan hubungan umum-khusus antara dua <i>use case</i> penggunaan di mana fungsi adalah fungsi yang lebih umum.
5		<code><<Include>></code> simbol ini merupakan hubungan <i>use case</i> tambahan ke <i>use case</i> di mana <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> tersebut untuk menjalankan fungsinya atau sebagai kondisi untuk <i>use case</i> tersebut.
6		<code><<Extend>></code> simbol ini merupakan hubungan <i>use case</i> tambahan dengan <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri meskipun tidak ada <i>use case</i> tambahan.


2.4.2 Class Diagram

Menurut (Chonoles, 2017), *Class Diagram* adalah sebuah penggambaran kelas-kelas yang mendefinisikan struktur sistem yang diperlukan. di dalam kelas terdapat bagian- bagian yang terdiri dari atribut, *method*, atau operasi. berikut ini adalah penjelasan dari atribut dan *method* :

1. Atribut adalah sebuah variabel yang dimiliki oleh sebuah *class*.
2. *Method* merupakan sebuah fungsi yang dapat di panggil dari sebuah kelas.

Tabel 2.3 Simbol-Simbol *Class Diagram*




No	Simbol	Keterangan
1		Kelas yang terdapat pada struktur.
2		<i>Interface</i> (Antar muka) merupakan sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Asosiasi berarah, hubungan kelas dengan makna dari satu kelas digunakan oleh kelas lain, asosiasi sering dikaitkan dengan simbol.
4		Generalisasi, hubungan antar kelas dalam arti generalisasi-spesialisasi (umum-khusus).
5		<i>Dependency</i> (Ketergantungan) merupakan hubungan antar kelas dalam arti ketergantungan antar kelas.
6		Agregasi merupakan hubungan antar kelas dengan makna semua bagian.



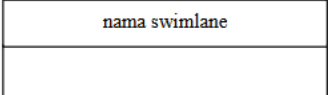
7		Asosiasi merupakan Hubungan antar kelas dalam pengertian umum, asosiasi pada umumnya juga disertai dengan <i>multiplicity</i> .
---	---	---

2.4.3 Activity Diagram

Menurut (Chonoles, 2017), *Activity Diagram* atau disebut dengan diagram aktivitas merupakan sebuah *workflow* atau alur kerja atau dapat disebut juga alur kerja dari sistem yang sedang berjalan dari perangkat lunak. *Activity diagram* berguna untuk menggambarkan alur sistem yang dibuat dari dalam, yang berguna untuk tidak menggambarkan dari aktivitas pengguna melainkan hanya dari sistem yang ada di perangkat lunak. *Activity diagram* terdapat beberapa simbol yang digunakan berikut adalah simbol tersebut dapat dilihat pada tabel 2.4 :

Tabel 2.4 Simbol-Simbol *Activity Diagram*

No	Simbol	Keterangan
1		Status awal (mulai) merupakan sebuah diagram aktivitas yang memiliki status awal.
2		Aktivitas merupakan suatu kegiatan yang dilakukan sistem, biasanya dimulai dengan kata kerja.
3		<i>Decision</i> merupakan asosiasi percabangan yang dimaksud untuk jika ada pilihan aktivitas lebih dari satu.

4		<i>Join</i> merupakan asosiasi penggabungan lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir (Selesai) merupakan sebuah diagram aktivitas memiliki sebuah status akhir.
6		<i>Swimlane</i> merupakan memisahkan organisasi bisnis yang bertanggung jawab atas aktivitas yang terjadi.

2.5 CodeIgniter

CodeIgniter adalah sebuah *framework* gratis yang dapat digunakan untuk mengembangkan aplikasi apa pun, *CodeIgniter* lebih sering digunakan untuk pengembangan aplikasi *web* dengan pengembangan jenis *Model View Controller* (MVC) sehingga pengembangan *web* dapat lebih fleksibel (Alvarez & Alvarez, 2017). *CodeIgniter* sendiri merupakan salah satu *framework* yang memiliki dasar bahasa pemrogramannya adalah PHP dan bersifat *Open Source*.

2.6 Rest API dan Restful API

Rest API merupakan sebuah desain arsitektur yang berada pada *API* itu sendiri, sedangkan *RESTful API* merupakan *REST* yang berkerja sebagai *client* dari sistem yang dikembangkan agar dapat mengakses data dari server menggunakan *Global ID* atau *Universal Resource Identifiers* (URIs) (Oktasari & Kurniadi, 2020). Format yang digunakan pada *REST API* ini dalam berkomunikasi adalah

menggunakan format *text*, *XML* atau *JSON*. Berikut ini adalah metode *HTTP* yang digunakan untuk melakukan komunikasi antara *client* dan *server* :

1. *GET*, merupakan metode yang digunakan untuk membaca data dari *REST server*.
2. *POST*, merupakan metode yang digunakan untuk mengubah atau menambah data baru ke *REST Server*.
3. *DELETE*, merupakan metode yang digunakan untuk menghapus data pada *REST server*.
4. *OPTIONS*, merupakan metode yang digunakan untuk mendapatkan operasi yang dapat digunakan mengolah data dari *REST server*.

2.7 CRC (Class Responsibility Collaborator) Card

Class Responsibility Collaborator (CRC) Cards merupakan teknik untuk mengajar desain berorientasi objek dan digunakan lebih luas dalam pemodelan perangkat lunak. Mereka menyusun *subset* persyaratan pengguna dalam bentuk yang berguna untuk pengembang berorientasi objek (Putra, 2020).

2.8 Bootstrap

Bootstrap merupakan sebuah *framework* yang mendasari kepada responsivitas sebuah *web* yang bekerja pada *HTML*, *CSS* dan *JS*. *Bootstrap* digunakan agar *web* yang dikembangkan akan menyesuaikan tampilan *layout* dari perangkat yang kita gunakan melalui *device* yang berbeda seperti *smarthphone* dan PC atau Laptop. Menurut (Horowitz, 2019) *Bootstrap* merupakan sebuah metode yang digunakan untuk mengestimasi sebuah *estimator* atau statistik uji dengan

mengambil *sampel* data dari pengguna yang memiliki perangkat berbeda namun dengan pengalaman yang sama.

2.9 *User Story*

User Story merupakan urutan beberapa cerita yang didapatkan dari hasil diskusi antara *programmer* dan *user* mengenai hal-hal apa saja yang harus bisa dilakukan sistem sesuai dengan keinginan *user*. *User Story* akan dijadikan acuan pembuatan fitur-fitur yang harus ada pada sistem (Wautelet et al., 2014).

2.10 *Value*

Definisi *Value* adalah keinginan atau keyakinan sentral tentang keadaan akhir atau perilaku yang diinginkan yang melampaui situasi tertentu, memandu pilihan dan penilaian kita atas keputusan dan dengan demikian perilaku kita, menjadi bagian *integral* dari cara kita berada dan bertindak, dan dengan demikian membentuk karakter kita (Petra, 2011). Itu berarti pusat keinginan atau keyakinan tentang keputusan akhir yang memandu pilihan, mengevaluasi keputusan kita dan menjadi bagian dari cara kita bertindak dan sesuatu yang membentuk karakter kita.

2.11 *Acceptance Test Criteria*

Acceptance test criteria merupakan sebuah langkah membuat daftar kriteria penerimaan untuk cerita pengguna, berguna untuk menentukan batasan pengguna (Endramawan & Sifa, 2017).

2.12 *Iteration Plan*

Perencanaan iterasi adalah peristiwa di mana semua anggota tim memutuskan berapa banyak pemulihan tim yang dapat mereka lakukan selama

iterasi berikutnya. Tim merangkum pekerjaan menjadi serangkaian tujuan berulang yang berkomitmen (Nurkholis & Saputra, 2021).

2.13 *Prototype Desain*

Desain adalah penyusunan rencana atau kesepakatan untuk pembangunan (pembangunan) pada suatu sistem (Mulyadi, 2013). *Prototyping* adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal.

Prototyping memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat yang akan dibuat (Purnomo, 2017).

Penelitian yang dilakukan (Aryani et al., 2019), *Prototyping* adalah suatu metode yang digunakan untuk pengembangan suatu sistem ataupun perangkat lunak. Dengan metode *prototyping* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi. Agar proses pembuatan *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan pada tahap awal, yaitu pengembang dan pengguna harus satu pemahaman bahwa *prototype* dibangun untuk mendefinisikan kebutuhan awal (Purnomo, 2017).

2.14 *Fungsional dan Usability*

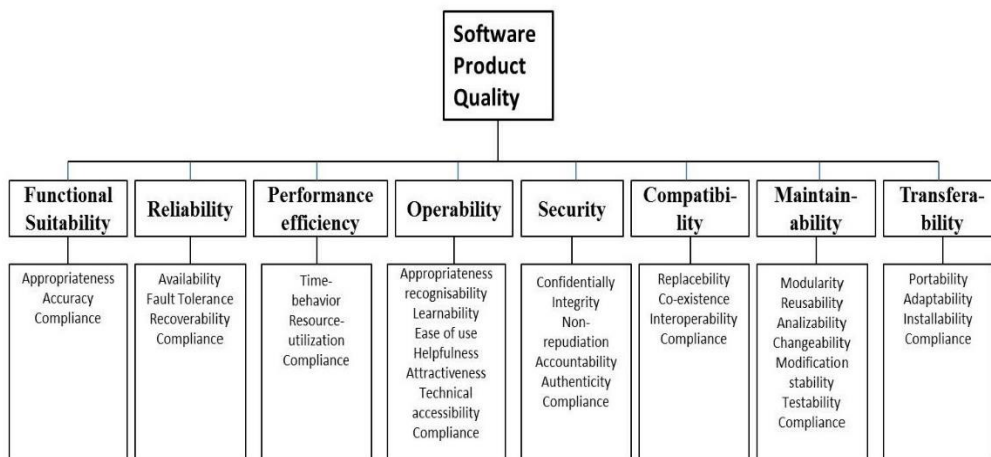
Karakteristik *functional suitability* merupakan sejauh mana suatu produk atau sistem yang memenuhi kebutuhan ketika digunakan pada kondisi tertentu (Aryani et al., 2019). *Functional suitability* mempunyai tiga sub karakteristik yaitu *functional completeness*, *functional correctness*, dan *functional appropriateness*.

Usability merupakan karakteristik sejauh mana sebuah produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan efektif, efisien, dan kepuasan dalam konteks pengguna. Kemudian *reliability* merupakan karakteristik sejauh mana sistem, produk, atau komponen melakukan fungsi tertentu di bawah kondisi tertentu dalam waktu yang telah ditetapkan. Karakteristik *performance efficiency* merupakan tingkat kinerja relatif terhadap sumber daya yang digunakan dalam kondisi yang ditetapkan (Dwi Mulyawan et al., 2021).

2.15 Pengujian ISO 25010

ISO 25010 merupakan sebuah pengujian yang terdiri dari delapan karakteristik yang dibagi menjadi bagian - bagian yang berhubungan dengan sistem perangkat lunak yang statis dan sifat dinamis sistem komputer (Haoues et al., 2017).

Berikut adalah gambar delapan karakteristik dari ISO 25010 pada gambar 2.2:



Gambar 2.2 Model ISO 25010

Berdasarkan dari delapan karakteristik di atas, berikut ini adalah penjelasan dari setiap karakteristik tersebut :

1. *Functional Suitability*, merupakan sistem yang memberikan sebuah fungsional untuk dapat menunjang kebutuhan sistem pada saat hal tertentu terjadi.
2. *Reliability*, merupakan tingkatan sebuah sistem yang dapat menjaga kinerja pada tingkatan tertentu pada saat digunakan.
3. *Performance Efficiency*, merupakan tingkatan sistem yang menyediakan performa yang optimal dalam kinerjanya dengan *resource* yang digunakan.
4. *Usability* adalah tingkatan sebuah sistem yang dapat mudah dimengerti dan menarik oleh pengguna.
5. *Security*, merupakan sebuah tingkatan sistem yang digunakan dapat melindungi data pengguna, akses sistem, modifikasi ataupun kerusakan yang berbahaya terhadap sistem.
6. *Compatibility*, merupakan kemampuan sistem dalam menjalankan atau mengirimkan informasi.
7. *Maintainability*, merupakan tingkatan sistem yang dapat diubah sewaktu-waktu jika diperlukan yang melibatkan tentang perbaikan, pengembangan untuk menyesuaikan kebutuhan dimasa mendatang sesuai spesifikasi fungsinya.
8. *Portability*, merupakan tingkatan sistem yang dapat dengan mudah di pindahkan ke tempat penyimpanan atau ke ruangan yang berbeda.