

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam tinjauan pustaka ini akan di uraikan tentang hasil penulisan lainnya yang pernah dilakukan dan mempunyai kaitan topik dengan penulisan yang penulis lakukan, di antaranya:

1. Penulisan yang dilakukan oleh Jery Ariska, M. Jazman (2016) dalam jurnalnya yang berjudul “RANCANG BANGUN SISTEM INFORMASI MANAJEMEN ASET SEKOLAH MENGGUNAKAN TEKNIK LABELLING QR CODE (STUDI KASUS: MAN 2 MODEL PEKANBARU)” dijelaskan bahwa Aset merupakan salah satu sistem informasi yang banyak digunakan oleh perusahaan termasuk juga sekolah. Implementasi Sistem Informasi Manajemen Aset dianggap mampu mentertibkan administrasi pengelolaan aset. Tertib dokumen aset berkaitan dengan upaya penyediaan dan pendataan data-data atau dokumen yang menyertai keberadaan aset, sedangkan tertib administrasi lebih dimaksudkan pada upaya membangun prosedur pengelolaan aset mulai saat pengadaan, penerimaan, perubahan data. Kemudian untuk memberi kemudahan bagi petugas dalam pencarian informasi barang, penulis melakukan teknik pelabelan aset menggunakan *QR Code*. Tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapat respon yang cepat pula (Ariska *et al.*, 2016) .
2. Penulisan yang dilakukan oleh Mohammad Adam Setioardi, Sukisno (2019) dalam jurnalnya yang berjudul “Perancangan Sistem Informasi Pengelolaan Barang Inventaris Berbasis Web Di SMAN 24 Kabupaten Tangerang”

dijelaskan bahwa, pengelolaan barang yang masih menggunakan spreadsheet masih memungkinkan adanya kesalahan, karena data yang ada menjadi terbagi mulai dari data barang yang akan masuk dan data barang yang masih ada, sehingga menyulitkan petugas dalam menginputkan atau pencarian data barang tertentu, kendala lain yang saat ini dihadapi oleh petugas pengelolaan barang inventaris ialah dalam mencatat nama peminjam barang yang masih dilakukan secara manual dan catatan biasa menyebabkan kehilangan data peminjam. Dengan ini penulis membuat sistem pengelolaan barang inventaris secara computer dan dirancang memakai PHP dan MySql. Sistem yang dianggap mampu membantu petugas administrasi dalam pengelolaan barang inventaris (Setioardi dan sukisno, 2019).

3. Penulisan yang dilakukan oleh Richard Anggaswara, Vera Suryani (2013) dalam judulnya “APLIKASI PENDUKUNG INVENTARISASI BARANG BERBASIS ANDROID (STUDI KASUS : IT TELKOM GEDUNG D RUANG D307 LOGISTIK BANDUNG)” dijelaskan bahwa, Sistem inventarisasi dibuat agar dapat membantu setiap perusahaan, sekolah atau instansi lainnya guna mengetahui setiap barang-barang yang mereka miliki karena kebanyakan dari mereka biasanya tidak mengetahui bagaimana keadaan setiap barang yang dimilikinya sehingga banyak yang sudah rusak, lecet atau malah tidak bisa digunakan kembali. Penggunaan sistem inventarisasi android terhadap IT Telkom Fakultas Informatika dilakukan dengan menempelkan kode disetiap barang (lemari, meja, rak, printer, komputer, perangkat keras sejenis dan barang-barang lainnya) dengan menggunakan QR kode sebagai profile terhadap barang karena code tersebut

menyimpan seluruh data dari tiap barang. QR kode dianggap lebih cepat dalam pembacaan (Richard Anggaswara dan Suryani, 2013).

4. Penulisan yang dilakukan oleh Adelia Siti Rukoyah, Gunawan Abidillah, Asep Id Hadiana (2017) dalam judulnya “SISTEM INFORMASI INVENTARIS BARANG PADA RSUD SOREANG” dijelaskan bahwa, Penggunaan teknologi informasi yang diterapkan melalui sistem informasi inventaris memberikan kemudahan kepada pengguna dalam melaksanakan pekerjaannya. Kemudahan yang diperoleh yaitu memudahkan dalam mengelola data serta memudahkan pada saat melakukan evaluasi data berdasarkan kebutuhan. Berbagai fasilitas inventaris alat/barang pada rumah sakit ini yang perlu dilindungi banyak karakteristik yang sangat perlu dipahami oleh praktisi keamanan sistem informasi yaitu pada dasarnya fasilitas inventaris alat/barang rumah sakit yang ingin dijaga (Rukoyah, Abidillah dan Hadiana, 2017).
5. Penulisan yang dilakukan oleh Yuniati Rakhel, Arief Hidayat, Victor G. Utomo (2016) dalam judulnya “PERANCANGAN SISTEM INFORMASI INVENTARIS BERBASIS WEB MOBILE (Studi Kasus : STMIK ProVisi Semarang)” dijelaskan bahwa, Sistem inventaris adalah sebuah sistem yang membantu dalam menjalankan proses inventarisasi (pencatatan) dalam suatu organisasi mulai dari pencatatan barang masuk, penyimpanan sampai dengan barang keluar. Sistem inventaris di STMIK Provinsi masih menggunakan Microsoft Excel dalam pengolahan datanya. Hal ini menyebabkan bagian Rumah Tangga tidak dapat mengelola data inventaris dengan maksimal dikarenakan data barang yang cukup banyak.

Pencarian data barang atau tingkat kecepatan akses data (laporannya) menjadi sangat lambat karena harus mencari datanya satu per satu dan dibutuhkan satu hari berikutnya untuk hasil pelaporan data inventaris (Rakhel, Hidayat dan Utomo, 2016).

6. Penulisan yang dilakukan oleh I Gede Handika, Ayi Purbasari (2018) dalam judulnya “PEMANFAATAN FRAMEWORK LARAVEL DALAM PEMBANGUNAN APLIKASI E-TRAVEL BERBASIS WEBSITE” Framework laravel adalah sebuah kerangka kerja open source yang diciptakan oleh Taylor Otwell. Laravel merupakan framework bundle, migrasi dan artisan CLI (Command Line Interface) yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti Codeigniter, Yii, ASP.NET MVC, Ruby on Rails, Sinatra dan lain-lain. Laravel memiliki seperangkat sangat kaya fitur yang akan meningkatkan kecepatan pengembangan web (Gede Handika, Purbasari, 2018).

2.2. Landasan Teori

Landasan Teori berisi teori-teori relevan yang digunakan untuk menjelaskan tentang data-data yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan dan penyusunan penelitian.

2.2.1. Sistem Informasi

Sistem Informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran

(informasi), guna mencapai sasaran-sasaran perusahaan (Ariska *et al.*, 2016) (Muflihini, Dhika dan Handayani, 2020).

2.2.2. Inventarisasi

Inventarisasi adalah sebuah sistem yang membantu dalam menjalankan proses pencatatan data yang berhubungan dengan barang atau aset dalam suatu organisasi mulai dari pencatatan barang masuk, penyimpanan sampai dengan barang keluar pada sebuah instansi atau perusahaan (Rakhel, et al, 2016)(Rianto, et al, no date).

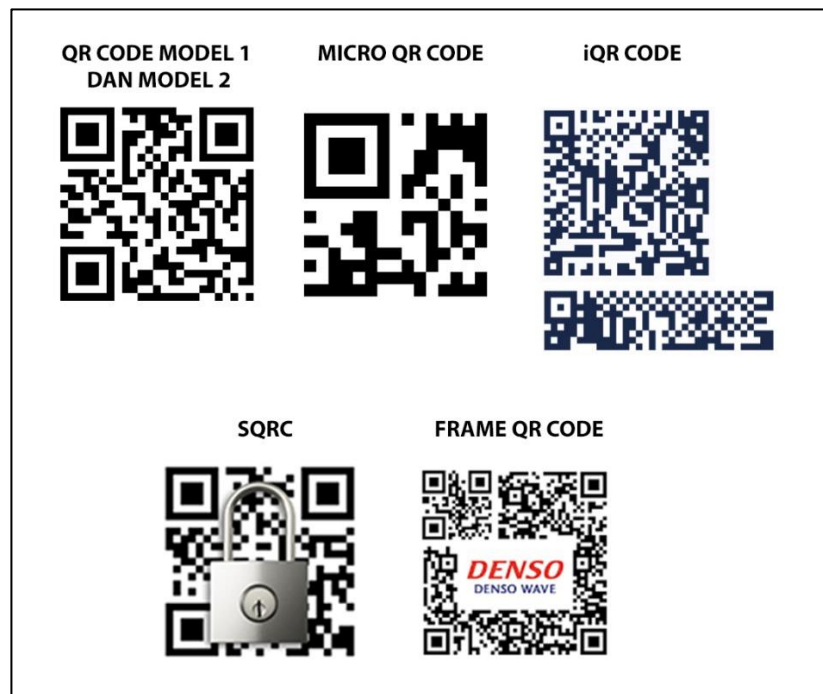
2.2.3. Framework Laravel

Framework laravel adalah sebuah kerangka kerja open source yang diciptakan oleh Taylor Otwell. *Laravel* merupakan *framework bundle*, migrasi dan artisan CLI (*Command Line Interface*) yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *Codeigniter*, *Yii*, *ASP.NET MVC*, *Ruby on Rails*, *Sinatra* dan lain-lain. *Laravel* memiliki seperangkat sangat kaya fitur yang akan meningkatkan kecepatan pengembangan web (Gede Handika, 2018).

2.2.4. QR Code

QR Code adalah bentuk evolusi kode batang atau istilah barcode yang merupakan satu dimensi menjadi dua dimensi yang bisa menyimpan kode lebih banyak seperti Numeric, Alphanumeric, Binary, Kanji dan Kana jika dibandingkan dengan kode batang (barcode) yang biasa dijumpai di setiap barang-barang yang ada di supermarket (Putu, Putra Yudha dan Mertasana, 2017)(Richard Anggaswara dan Suryani, 2013).

QR Code terdiri dari berbagai bentuk. Jenis-jenis *QR Code* terdiri dari *QR Code Model 1*, *QR Code Model 2*, *Micro QR Code*, *iQR Code*, *SQRC*, *Frame_QR*. *QR Code* dapat menyimpan data berjenis numerik sampai dengan jumlah 7.089 karakter. Selain itu, dapat menyimpan data berjenis alfanumerik sampai dengan 4.296 karakter. *QR Code* juga dapat menyimpan data kode binary sampai dengan 2.844 byte, dan data berbentuk huruf kanji sampai dengan 1.817 karakter. *QR Code* dapat menampung data secara horizontal dan vertikal. *QR Code* juga tahan terhadap kerusakan, karena dapat memperbaiki kesalahan sampai dengan 30%. Kelebihan lainnya, meskipun simbol *QR Code* dalam keadaan kotor ataupun rusak, akan tetapi data masih tetap dapat disimpan dan dibaca (Susanti *et al.*, 2018). Jenis – jenis *QR Code* disajikan seperti pada gambar 2.1.



Gambar 2. 1 Jenis – jenis *QR Code* (Susanti *et al.*, 2018)

Jadi Sistem Informasi Inventarisasi menggunakan *QR Code* adalah sebuah kerangka kerja yang membantu dalam pencatatan data yang berhubungan dengan

barang aset dalam suatu instansi atau perusahaan dengan memanfaatkan *QR Code* sebagai bentuk evolusi kode batang atau istilah barcode untuk proses pengelolaannya.

2.2.5. Pengertian PHP

PHP adalah bahasa pemrograman tingkat tinggi yang saat ini populer dikalangan programmer. Dalam pembuatan aplikasi PHP bisa menjadi pilihan bahasa yang mudah untuk dikuasai dengan keunggulannya diantaranya cepat dengan menyisipkannya di dalam kode HTML, mudah digunakan, dan bisa dijalankan hampir di semua sistem operasi yang ada saat ini (Gede Handika, 2018).

2.2.6. MySQL

MySQL adalah aplikasi yang digunakan sebagai penyimpanan data dari sebuah sistem atau aplikasi yang cepat dalam pengelolaan databasenya. MySQL juga dikenal sebagai perangkat lunak yang mudah digunakan dan cocok untuk aplikasi yang memerlukan media penyimpanan data yang responsive (Witirani *et al.*, 2017).

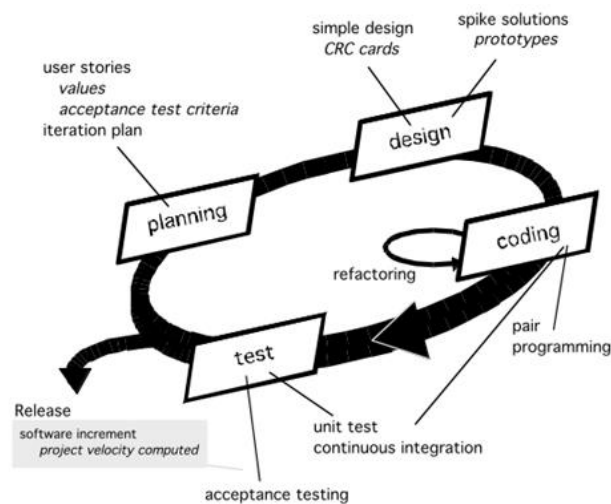
2.2.7. Laragon

Laragon adalah sebuah universal development environment untuk PHP, Python, dan terisolasi serta mudah dipakai. Laragon memiliki kelebihan diantaranya yaitu bisa diakses menggunakan localhost, bisa dipindahkan tanpa merusak sistem, tidak mempengaruhi komputer lokal, dan cocok digunakan untuk pengembangan web (Irianto, Anshori dan Mas'udi, 2020).

2.3. Metode Pengembangan Sistem

Terdapat beberapa metode dalam pengembangan sistem, salah satunya yaitu metode pengembangan *Extreme Programming*(XP). *Extreme Programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan – perubahan requirement yang sangat cepat (Putu, Putra Yudha dan Mertasana, 2017).

Model pengembangan tradisional mengacu soal perencanaan, analisa, dan perancangan sistem, dengan waktu yang lama untuk masing- masing tahap karena luasnya cakupan, maka XP menawarkan cara yang berbeda. XP menawarkan tahapan - tahapan tersebut dalam waktu yang singkat dan berulang. Berdasarkan uraian diatas maka dapat diambil kesimpulan bahwa model XP dimana semua proses yang dilakukan secara berurutan sesuai dengan urutan yang ada. Berikut adalah gambaran model *Extreme Programming* (XP), disajikan seperti pada gambar 2.2.



Gambar 2. 2 Model *Extreme Programming* (Supriyatna, 2018)

Adapun tahapan pembangunan aplikasi menggunakan metode *Extreme Programming* adalah sebagai berikut:

1. *Planning* (Perencanaan)

Tahapan ini dimulai dengan mendengarkan kumpulan kebutuhan aktifitas suatu sistem yang memungkinkan pengguna memahami proses bisnis untuk sistem dan mendapatkan gambaran yang jelas mengenai fitur utama, fungsionalitas dan keluaran yang diinginkan. Dalam pembangunan aplikasi pada tahapan ini dimulai dari mengidentifikasi permasalahan yang timbul pada sistem yang sedang berjalan, kemudian dilakukan analisa kebutuhan pengguna terhadap sistem yang akan dibangun.

2. *Design* (Perancangan)

Pada tahapan perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Selain itu dibuatkan juga pemodelan basis data untuk menggambarkan hubungan antar data. Pemodelan sistem yang digunakan yaitu *Unified Modelling Language* (UML) yang terdiri dari beberapa diagram antara lain Use- Case Diagram, Activity Diagram, Component Diagram dan Deployment Diagram.

3. *Coding* (Pengkodean).

Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program yang menghasilkan prototipe dari perangkat lunak.

4. *Testing* (Pengujian)

Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus

pada fitur dan fungsionalitas dari keseluruhan sistem kemudian ditinjau oleh pengguna sistem. Metode yang digunakan dalam melakukan pengujian terhadap aplikasi adalah Black-Box Testing dengan melakukan pengujian terhadap masukan dan keluaran yang dihasilkan sistem.

5. *Software Increment* (Peningkatan Perangkat Lunak).

Tahapan ini merupakan tahap pengembangan sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambahkan layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem.

2.4. Bahasa Permodelan Terpadu (*Unified Modeling Language*)






Menurut (Roger S. Pressman, 2012) (Sonata, 2019), Unified Modeling Language (UML) adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak.

2.4.1. *Use Case Diagram*

Menurut (Roger S. Pressman, 2012) (Sonata, 2019) use case diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram use case. Para aktor terhubung oleh garis ke use case yang mereka kerjakan.

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang ada pada diagram use case disajikan pada tabel 2.1 :

Tabel 2. 1 Simbol *Use Case Diagram*



Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i></p>
<p>Ekstensi/<i>extend</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan</p>
<p><<<i>extend</i>>></p> 	<p>dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>


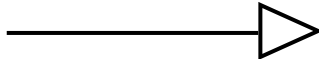

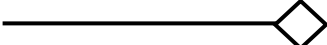
Simbol	Deskripsi
Menggunakan/Include/uses <<include>>→	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya

2.4.2. Class Diagram

Menurut (Roger S. Pressman, 2012) (Sonata, 2019), Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas, bagian bawah merupakan *operation* dari kelas tersebut. (Roger S. Pressman, 2012) (Sonata, 2019) simbol-simbol yang ada pada diagram disajikan pada tabel 2.2.

Tabel 2. 2 Class Diagram



Simbol	Deskripsi
Kelas <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>nama_kelas</p> <p>+atribut</p> <p>+operasi()</p> </div>	Kelas pada struktur sistem
Antarmuka/Interface <div style="display: flex; align-items: center; justify-content: center; margin: 10px auto;"> <div style="margin-right: 10px;">nama_interface</div>  </div>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/association <div style="text-align: center; margin: 10px auto;">  </div>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

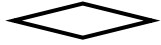

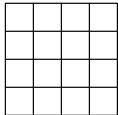


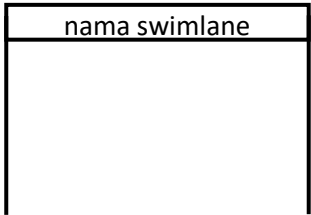
Simbol	Deskripsi
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.4.3. Activity Diagram

Menurut (Roger S. Pressman, 2012) (Sonata, 2019), Sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan. (Roger S. Pressman, 2012) (Sonata, 2019) Simbol-simbol yang ada pada *activity diagram* disajikan pada tabel 2.3 :

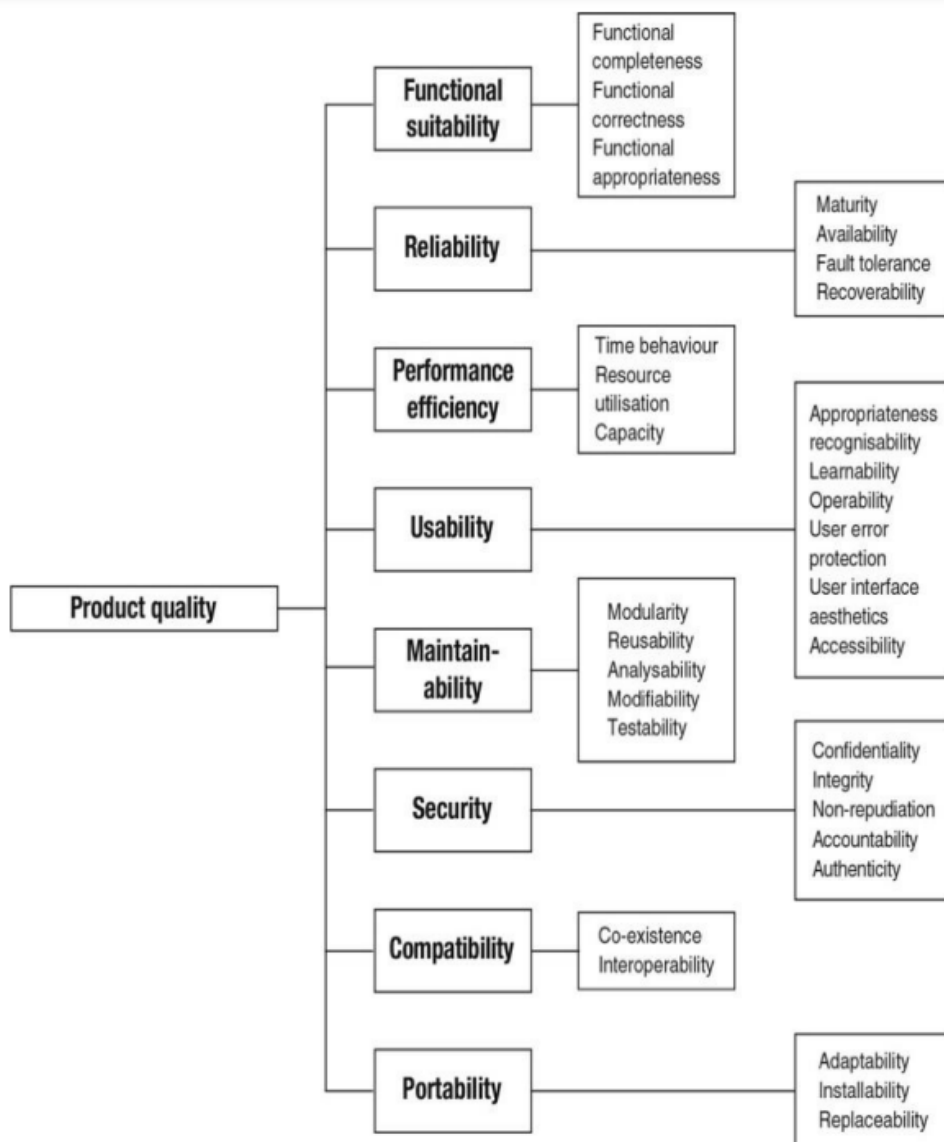
Tabel 2. 3 Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

Simbol	Deskripsi
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.4.4. ISO 25010

ISO 25010 adalah pengujian standard yang digunakan untuk menguji sebuah sistem atau perangkat lunak yang telah dibuat. ISO 25010 mengevaluasi perangkat lunak yang dilakukan dengan detail berdasarkan kualitas produk yang terdiri dari delapan karakter uji yang dimiliki oleh ISO 25010 yakni *functionality suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability*, *security*, *maintainability*, dan *portability* (Murdiani dan Umar, 2020). Karakteristik uji dapat dilihat pada gambar dibawah ini.



Gambar 2. 3 ISO 25010

Berdasarkan gambar diatas terdapat delapan karateristik yang dijelaskan sebagai berikut:

1. *Functional suitability* adalah pengujian yang dilakukan dengan melihat kebutuhan dari sistem yang dibangun terpenuhi atau tidak. Adapun ciri yang dimiliki diantaranya:
 - a. Kelengkapan fungsi, melihat fungsi yang berjalan mencakup tujuan dan tugas dari pengguna sistem.

- b. Kesesuaian fungsi, melihat fungsi yang berjalan sesuai dengan pengguna.
 - c. Ketepatan fungsi, melihat hasil yang didapatkan benar sesuai dengan kebutuhan pengguna sistem.
2. Reliability adalah pengujian yang dilakukan dengan tujuan mengetahui apakah sistem yang dibangun dapat berfungsi pada kondisi dan rentang waktu tertentu. Adapun ciri yang dimiliki diantaranya:
- a. Ketersediaan, melihat apakah sistem bisa digunakan pada saat akan diakses.
 - b. Kematangan, melihat apakah sistem mampu memenuhi kebutuhan dalam keadaan baik.
 - c. Dapat dipulihkan, melihat apakah sistem dapat memulihkan data ketika terjadi gangguan.
 - d. Toleransi kesalahan, melihat apakah sistem bisa beroperasi meski terdapat kesalahan pada perangkat lunak maupun perangkat keras pada sistem.
3. Performance Efficiency adalah pengujian untuk melihat kinerja penggunaan sumber daya dalam waktu tertentu. Adapun ciri yang dimiliki diantaranya:
- a. Kapasitas, melihat batasan yang dimiliki sistem.
 - b. Perilaku waktu, melihat respon dari sistem saat melakukan pengoperasian suatu fungsi di dalam sistem.
 - c. Pemanfaatan sumber daya, melihat sumber daya yang digunakan apakah memenuhi kebutuhan sistem.

4. *Usability* adalah pengujian yang dilakukan dengan tujuan melihat apakah sistem yang dibuat sesuai dengan kebutuhan penggunanya. Adapun ciri yang dimiliki diantaranya:
 - a. *Learnability*, melihat apakah pengguna dapat menjalankan perangkat lunak yang dibuat dengan belajar menggunakannya secara efektif.
 - b. Ketepatan pengenalan, melihat apakah sistem telah memenuhi keinginan para pengguna.
 - c. Operabilitas, melihat apakah sistem yang dibangun mudah dioperasikan oleh para pengguna.
 - d. Aksesibilitas, melihat apakah sistem dapat digunakan oleh semua kelompok atau pengguna.
 - e. Estetika, melihat apakah tampilan dari sistem telah dibuat sesuai dengan keinginan pengguna.
 - f. *Protect*, melihat apakah sistem dapat melindungi pengguna jika terjadi kesalahan yang disebabkan oleh pengguna itu sendiri.

5. *Maintainability* adalah pengujian yang dilakukan dengan melihat sistem apakah bisa diubah. Adapun ciri yang dimiliki diantaranya:
 - a. Modularitas, melihat sistem terdiri dari komponen yang terpisah jika diubah apakah berdampak kecil pada komponen lainnya di dalam sistem.
 - b. *Modifiability*, melihat apakah sistem dapat dimodifikasi secara efektif.
 - c. *Testability*, melihat tingkat efisiensi dengan membentuk kriteria pengujian suatu perangkat lunak.

6. *Security* adalah pengujian yang dilakukan apakah sistem dapat melindungi data dan informasi dari penggunanya. Adapun ciri yang dimiliki diantaranya:
 - a. Kerahasiaan, memastikan sistem dapat menjaga akses yang diberikan dalam mengubah data.
 - b. Keaslian, memastikan pengguna yang memiliki akses asli yang dapat di klaim secara legal.
7. *Compatibility* adalah pengujian yang dilakukan terhadap kemampuan sistem yang dibangun pada saat berbagi informasi atau dijalankan secara bersamaan dengan sistem yang sama. Adapun ciri yang dimiliki diantaranya:
 - a. Koeksistensi, melihat apakah sistem bisa berbagi informasi tanpa merusak komponen lain yang ada pada sistem.
 - b. Interoperabilitas, melihat sistem menggunakan informasi yang sama.
8. *Portability* adalah pengujian yang dilakukan untuk melihat ketahanan sistem jika digunakan di perangkat atau media yang berbeda. Adapun ciri yang dimiliki diantaranya:
 - a. Adaptabilitas, melihat apakah sistem mampu beradaptasi dengan lingkungan berbeda.
 - b. Penginstalan, melihat apakah sistem dapat diinstal di perangkat yang berbeda.