

BAB II **LANDASAN TEORI**

2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan sumber referensi terhadap penelitian terdahulu seperti berikut:

1. Oleh Julius Ferbi, Willy Junaidi, Fransiska Prihatini Sihotang, Tahun 2020 dengan judul penelitian Sistem Informasi Manajemen Penyewaan Alat Berat pada PT Suriaman Sentosa. Masalah pada penelitian yaitu Kesulitan untuk mengetahui status alat berat, kekeliruan pada informasi jam operasional penyewaan alat berat, dan tidak ada informasi maintenance alat berat. Dalam menganalisis masalah menggunakan metode PIECES dengan hasil penelitian yaitu perancangan dan membangun sistem informasi berbasis *website* untuk menyeimbangkan data penyewaan, data alat berat, jam operasional alat berat, dan laporan yang berkaitan dengan alat berat.
2. Oleh Harmoko Tahun 2020 dengan judul Sistem Informasi Penyewaan Alat Drilling Berbasis Web Pada PT. Nana Yamano Teknik. Permasalahan pada penelitian yaitu terdapat kendala yaitu masih dikerjakan secara manual dan menyebabkan perusahaan terkena denda dari penyewa alat *drilling* karena *spare part* yang dibutuhkan alat *drilling* tidak tersedia. Dengan perancangan dan pendekatan menggunakan UML (*Unified Modelling Language*) dan *framework* menggunakan *Code Igniter* dengan hasil penelitian yaitu Sistem yang dihasilkan akan mempercepat pembelian barang dan pembayaran barang.

3. Oleh Minda Septiani, Nurul Afni, Resti Lia Andharsaputri Tahun 2019 dengan judul Perancangan Sistem Informasi Penyewaan Alat Berat. Penelitian tersebut memiliki permasalahan seperti proses penyimpanan data, proses transaksi sampai proses pembuatan laporan yang masih menggunakan sistem manual. Dalam perancangan sistem ini menggunakan metode *waterfall* dan menghasilkan sistem penyewaan alat berat yang akan mempermudah admin dalam membuat laporan dan meminimalkan resiko kehilangan data.
4. Oleh Haslinda, Hasrul Bakri, Harifuddin Tahun 2018 dengan judul Perancangan Sistem Informasi Penyewaan Alat Berat pada CV. Tri Daya Jaya Makassar. Permasalahan penelitian seperti informasi pengelolaan penyewaan alat berat menggunakan pengelolaan berbasis kertas. Dengan menggunakan metode pengembangan SDLC (*System Development Life Cycle*) hasil dari penelitian ini berupa sistem informasi penyewaan alat berat berbasis web yang mempermudah pengguna untuk menyewa alat berat yang diinginkan secara *online*.
5. Oleh M. Hadi Prayitno, Achmad Noe'man Tahun 2020 dengan judul Sistem Monitoring Pengiriman Alat Kerja Dengan Menggunakan Teknologi CouchDB. Permasalahan penelitian seperti proses pengantaran tidak diketahuinya keberadaan kendaraan pengantar dari sisi perusahaan dan menghasilkan penelitian berupa sistem monitoring ini perusahaan dapat mengetahui posisi alat kerja yang dikirim, serta memiliki sistem pelaporan dan verifikasi barang yang terintegrasi, antara petugas aplikasi administrasi dan aplikasi *tracking*. Sehingga mempermudah para pengambil keputusan untuk dapat meningkatkan pelayanan.

Berdasarkan kelima jurnal tersebut terdapat beberapa perbedaan pada penelitian yang dilakukan seperti berikut:

1. Metode yang digunakan berupa *extreme programming*
2. Fitur yang diolah pada lima jurnal hanya berupa sewa dan penalti memberikan fitur berupa fitur pelaporan jam kerja alat berat.
3. Laporan yang disajikan berupa laporan sewa dan jam kerja.
4. Dapat memberikan monitoring kepada perusahaan dan konsumen untuk melihat target jam kerja alat berat.

2.2 Sistem Informasi

Menurut Krismiaji (2015), Sistem informasi adalah Cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Menurut Nas (2018), Sistem informasi adalah Sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

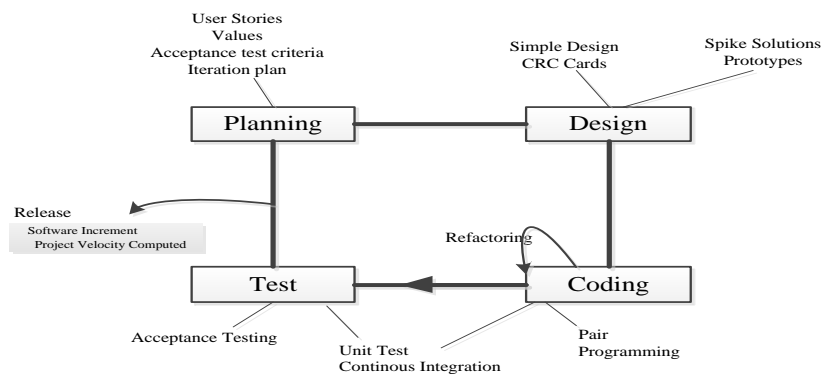
2.3 Alat Berat

Menurut Haslinda *et al.*, (2019) Alat berat adalah peralatan mesin berukuran besar yang didesain untuk melaksanakan fungsi konstruksi seperti pengerjaan tanah, konstruksi jalan, konstruksi bangunan, perkebunan, dan pertambangan.

Menurut Ferbi *et al.*, (2020) Alat berat adalah mesin berukuran besar yang didesain untuk melaksanakan fungsi konstruksi seperti pengerjaan tanah (*earthworking*) dan memindahkan bahan bangunan. Alat berat umumnya terdiri atas lima komponen, yaitu implemen, alat traksi, struktur, sumber tenaga dan transmisinya (*power train*), serta sistem kendali.

2.4 Metode *Extreme Programming*

Menurut Suryantara (2017) *extreme programming* berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming*.



Gambar 2.1 *Extreme Programming*
(Suryantara, 2017)

2.4.1 Sejarah Singkat *Extreme Programming*

Extreme Programming merupakan salah satu metodologi rakayasa perangkat lunak yang banyak digunakan untuk pengembangan aplikasi oleh para developer, diperkenalkan oleh Kent Beck yang ditunjukan untuk menangani sebuah proyek penggajian yang dikenal sebagai C3 (Chrysler Comprehensive Compensatun). Proyek tersebut dimulai pada maret 1996 yan terancam gagal karena rumitnya sistem mengalami kegagalan pada proses testing, kemudian pihak Chrysler menyewa Kent Beck sebagai kunsultan dibidang *software engineerung* yang kemudian disebut sebagai pencetus XP.

2.4.2 Kerangka Kerja *Extreme Programming*

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning, design, coding* dan *testing*.

1. *Planning*/Perencanaan

Tahap ini dimulain dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

a. *User Stories*

Merupakan cerita pengguna terkait kebutuhan sistem seperti input proses dan output dan harapan dari keinginan pengguna tersebut.

b. *Values*

Suatu nilai yang diambil dari stories berupa nilai kebutuhan yang lebih spesifik

c. *Acceptance Test Criteria*

Penentuan tahapan pengujian yang akan dilakukan pada saat sistem selesai dibangun.

d. *Iteration Plan*

Proses menentukan berapa lama sistem yang akan dikembangkan selesai dilakuka berdasarkan waktu dan pertemuan

2. *Design/Perancangan*

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas.

a. *Simple Design*

Desain terlebih dahulu dibuat sesederhana mungkin (tidak menambahkan suatu fungsi) agar lebih cepat terselesaikan (Keep It Simple). Fungsi baru ditambahkan saat benar benar dibutuhkan

b. *CRC Cards*

Class, Responsibilities, and Collaboration (CRC) cards digunakan tim pengembang mengumpulkan ide (brainstorming) dalam membuat desain perangkat lunak. CRC card biasanya digunakan saat pertama kali menentukan class apa yang diperlukan

c. *Spike Solutions*

Tujuan dari membuat Spike Solution adalah untuk memecahkan masalah teknis atau masalah dalam mendesain. Saat menemui masalah teknis, dapat menempatkan dua developer untuk menyelesaikan masalah itu untuk mengurangi resiko kegagalan

d. *Prototype*

Menampilkan hasil perancangan sistem sebelum proses pengkodean

3. *Coding/Pengkodean*

Hal utama dalam pengembangan menggunakan XP yaitu *pair programming*

a. *Refactoring*

Mengijinkan programer melakukan perbaikan-perbaikan pada kode program walaupun hal tersebut tidak direncanakan dalam desain.

b. *Pair Programming*

Proses penulisan kode program sesuai dengan kebutuhan pengguna

4. *Testing/Pengujian*

Tahap ini fokus pda pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

a. *Acceptance Test*

Pengujian dilakukan berdasarkan permintaan dari customer. Apakah program sesuai dengan apa yang diminta oleh customer.

b. *Unit Test*

Pengujian dilakukan pada setiap unit atau fungsi dalam sistem.

2.4.3 Nilai Inti *Extreme Programming*

Nilai inti pada pengembangan sistem yaitu :

1. *Communication* (Komunikasi)

Konmunikasi antar tim yang digunakan untuk saling berbagi pengetahuan dalam pengembangan.

2. *Simplicity* (Kesederhanaan)

Melakukan semua proses dengan sederhana dan mencoba mencari solusi yang paling sederhana.

3. *Feedback* (Masukan)

Masukan untuk mengetahui kemajuan dari proses dan kualitas perangkat lunak yang dibuat.

4. *Courage* (Kesalahan)

Kesuksesan pengembangan aplikasi harus memiliki keberanian, keyakinan dan integritas dalam pekerjaan.

5. *Respect* (Menghormati)

Menerapkan siklus pendek dan integrasi *continue*.

2.4.4 Aspek Dasar *Extreme Programming*

Aspek dasar pada penerapan metode *extreme programming* yaitu :

1. *The Planning Game*

Proses pendek dan cepat, mengutamakan aspek teknik, memisahkan unsur bisnis dengan unsur teknis dan pertemuan intensif antara klien dengan developer. Pada XP proses ini menggunakan terminologi “game” karena Beck menyarankan untuk menggunakan teknik *score card* dalam menentukan *requirements*.

2. *Small Releases*

Menyelesaikan bagian –bagian aplikasi dan melakukan persentasi kepada *client*, setelah mendapatkan persetujuan maka dilakukan penerapan keaplikasi.

3. *Metaphor*

Menggambarkan visi yang luas terhadap tujuan dari pengembangan perangkat lunak. Dengan Tujuan diharapkan komunikasi antara klien dengan developer akan berlangsung lebih baik.

4. *Simple Design*

Menghindari desain yang rumit dalam sebuah pengembangan perangkat lunak. Dengan desain yang simpel apabila terjadi perubahan dapat meminimalkan kesalahan.

5. *Refactoring*

Melakukan perubahan pada kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program tersebut tanpa mengubah cara program tersebut bekerja.

6. *Testing*

Membuat test terhadap aplikasi berdasarkan model test yang telah ditentukan.

7. *Pair Programming*

Dua orang programmer saling bekerjasama di komputer yang sama untuk menyelesaikan sebuah unit.

8. *Colletive Ownership*

Saling berbagi pengetahuan agar tidak saling ketergantungan pada programmer tertentu ataupun berbagai hambatan akibat perbedaan gaya menulis program dapat diperkecil.

9. *Coding Standard*

Dengan adanya coding standards yang telah disepakati terlebih dahulu maka pemahaman terhadap program akan menjadi mudah untuk semua programmer dalam tim.

10. *Continuous Integration*

Melakukan build sesering mungkin berbagai kesalahan pada program dapat dideteksi dan diperbaiki secepat mungkin.

11. *40-hours Week*

Beck berpendapat bekerja 8 jam sehari dan 5 hari seminggu adalah maksimal untuk tiap programmer.

12. *On-Site Customer*

XP menganjurkan bahwa ada anggota dari klien yang terlibat pada proses pengembangan perangkat lunak. Apabila ada kesalahan dalam pengembangan diharapkan klien dapat segera memberikan masukan untuk koreksinya.

2.4.5 Tujuan *Extreme Programming*

Tujuan metode *extreme programming* untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif dan mengurangi biaya selama ada perubahan dalam pengembangan perangkat lunak menggunakan siklus pengembangan perangkat lunak singkat.

2.4.6 Daur Hidup Metodologi *Extreme Programming*

Metode XP dapat diterapkan bila:

1. Adanya perubahan yang sangat cepat
2. Memiliki resiko yang tinggi pada pembuatan aplikasi

3. Dalam tim pengembangan aplikasi dengan sedikit programmer
4. Mampu mengotomatisasikan uji sistem
5. Keterlibatan peran serta klien secara langsung
6. Harus ada komunikasi yang baik

2.4.7 Keuntungan dan Kerugian *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu:

1. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi
2. Saling menghargai antar developer dan meningkatkan komunikasi
3. Dapat menjadi pembelajaran bagi orang lain
4. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
5. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat
6. Developer tidak berkerja secara berlebihan
7. Dengan XP dapat membuat keputusan yang bersifat teknis


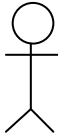

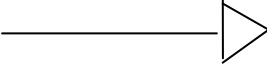
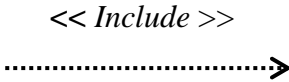
2.5 *Unified Modelling Language (UML)*

Menurut Rosa and Shalahuddin (2019) UML (*unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

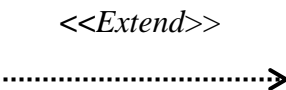
2.5.1 Use Case Diagram

Menurut Rosa and Shalahuddin (2019) *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Usecase</i> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)


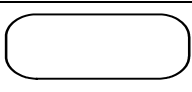
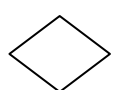

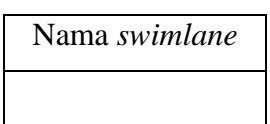

No	Simbol	Deskripsi
6.		Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

Sumber : (Rosa and Shalahuddin, 2019)

2.5.2 Activity Diagram

Menurut Rosa and Shalahuddin (2019) *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.2 berikut ini :

Tabel 2.2 Simbol *Activity Diagram*

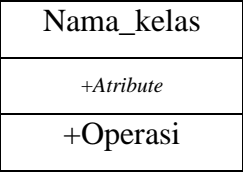


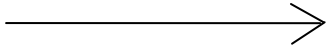
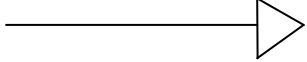
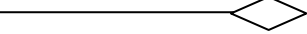
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber : (Rosa and Shalahuddin, 2019)

2.5.3 Class Diagram

Menurut Rosa and Shalahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/<i>Interface</i></p>  <p>Nama_<i>Interface</i></p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa and Shalahuddin, 2019)

2.6 Website

Menurut Urbietta *et al.* (2019), *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya.

Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.6.1 Dreamweaver

Menurut Agustian (2017) Macromedia Dreamweaver adalah program untuk membuat dan mengedit dokumen HTML secara visual dan mengelola halaman sebuah situs. Dreamweaver menyediakan banyak perangkat yang berkaitan dengan pengkodean dan fitur seperti HTML, CSS, JavaScript, PHP, ASP, ColdFusion, dan XML.

2.6.2 Mysql

Menurut Sabar *et al.*, (2019), MySQL adalah basis data yang bersifat *open source* sehingga banyak di gunakan untuk media. Walaupun gratis, MySQL tetap berkualitas dan sudah cukup memberikan performance yang memadai. Penggunaan PHP MyAdmin lebih mudah digunakan karena menggunakan *interface* yang lebih mudah dipahami”.

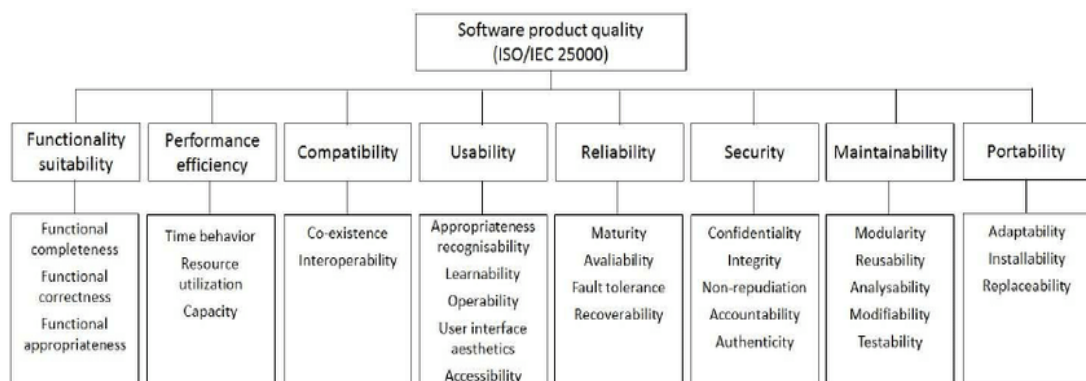
MySQL adalah nama sebuah *database server* yang menangani akses database yang selalu dalam bentuk pernyataan SQL (*Structured Query Language*) yaitu suatu bahasa yang digunakan untuk mengakses *database* relasional.

2.7 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

2.7.1 ISO 25010

Menurut *International Organisation for Standardisation* (2011), ISO/IEC 25010 adalah Model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* sebagai bentuk konsep pengujian terhadap kelayakan sistem yang telah dibangun.



Gambar 2.2 ISO/IEC 25010

Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

- Kualitas dalam model penggunaan,
- Model kualitas produk, dan

c. Data model kualitas

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Model ini berlaku untuk sistem komputer dan produk perangkat lunak. Karakteristik yang didefinisikan oleh kedua model tersebut relevan untuk semua produk perangkat lunak dan sistem komputer. Karakteristik dan subkarakteristik memberikan terminologi yang konsisten untuk menentukan, mengukur dan mengevaluasi kualitas sistem dan perangkat lunak. Mereka juga menyediakan seperangkat karakteristik kualitas yang sesuai dengan persyaratan kualitas yang dapat dibandingkan untuk kelengkapan.

2.7.2 Tahapan Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui proses pengujian yang akan dilakukan, berikut adalah tahapan pengujian ISO/IEC 25010:

1. *Functional Suitability*

Sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan di bawah kondisi tertentu. Sub pengujian tersebut sebagai berikut:

a. *Functional Completeness* (Kelengkapan Fungsional)

Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan

b. *Functional Correctness* (Kebenaran Fungsional)

Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.

Alat pengujian yang digunakan yaitu kuisisioner yang memiliki 2 jawaban pertanyaan seperti sukses atau gagal.

2. *Usability*

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Sub pengujian tersebut sebagai berikut :

a. *Appropriateness Recognizability*

Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.

b. *Learnability*

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran yang ditentukan untuk menggunakan produk atau sistem dengan efektifitas, efisiensi, bebas dari resiko dan kepuasan dengan tujuan tertentu konteks penggunaan.

Alat pengujian yang digunakan yaitu kuisisioner yang memiliki 5 jawaban pertanyaan seperti Sangat Setuju, Setuju, Ragu-ragu, Tidak Setuju dan Sangat Tidak Setuju.

3. *Portability*

Tingkat efektivitas dan efisiensi dimana sistem, produk atau komponen dapat ditransfer dari satu perangkat keras, perangkat lunak atau lingkungan operasional atau penggunaan lainnya ke yang lain. Berikut sub pengujian tersebut:

a. *Adaptability*

Sejauh mana suatu produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan yang berbeda atau berkembang.

b. *Coexistence*

Sejauh mana sistem dapat bekerja pada perangkat lunak yang ada.

Alat yang digunakan untuk pengujian tersebut berupa *cross browsing compability testing* seperti *chrome*, *firefox* dan *opera*.

4. *Performance efeciency.*

kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang dinyatakan, berikut sub pengujian tersebut:

a. *Time Behaviour*

Sejauh mana respon dan waktu pemrosesan dan tingkat *throughput* dari suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan

b. *Resource Utilization*

Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan

Alat pengujian yang digunakan yaitu dengan media web test secara online sehingga mampu menampilkan detail performa sistem, adapun link web test yaitu <https://www.webpagetest.org/>.