BAB II
LANDASAN TEORI

1.1 Tinjauan Pustaka

Terdapat penelitian yang telah dilakukan dalam penerapan sistem dalam konteks ini menggunakan sistem bengkel. Sehingga, dalam penelitian ini sangat diperlukannya tinjauan pustaka sebagai alat dalam penerapan sistem pada algoritma ini, agar dapat mengindentifikasi kesenjangan, menghindarkan terjadinya pengulangan, mengetahui algoritma yang sudah digunakan sebelumnya,. Daftar literatur dapat dilihat di table 2.1

Table 2.1 Tabel Literatur

No.	Nama peneliti	Tahun	Judul
1.	(Ahmad Supriatna, Anita Ratnasari S.Kom, M.Kom)	2019	Analisa dan Perancangan Sistem Infromasi Servis Mobil dan Penyediaan Mekanik pada Sony Otomotif
2.	(Iyus Supriadi)	2019	Sistem Informasi Layanan Bengkel Online Motor di Kota Jaya Pura Berbasis Android.
3.	(A.Ferico Octaviansyah Pasaribu1 , Dedi Darwis , Agus Irawan, Ade Surahman)	2019	Sistem Informasi Geogreafis Untuk Pencarian Lokasi Bengkel di Kota Bandar Lampung.

4.	(Reza Oksri Nengsi, Rahmat Hidayat, Yulindon)	2019	Kajian Aplikasi Pelayanan Bengkel Berbasis Android

1.2 Aplikasi *Mobile*

Aplikasi *mobile* atau sering juga disingkat dengan istilah *Mobile App*s adalah aplikasi dari sebuah perangkat lunak yang dalam pengoperasiannya dapat berjalan diperangkat *mobile* (Smartphone, Tablet, iPod, dll), dan memiliki sistem operasi yang mendukung perangkat lunak secara standalone. Platform pendistribusibusian aplikasi *mobile* yang tersedia, biasanya dikelola oleh owner dari *mobile* operating system, seperti *store* (*App*le *App*), *store (Google Play*), *Store* (Windows Phone) dan world (BlackBerry *App*) (Siegler, 2018).

Aplikasi *mobile* dapat berasal dari aplikasi yang sebelumnya telah terpasang di dalam perangkat *mobile* maupun juga yang dapat diunduh melalui tempat pendistribusiannya. Secara umum, aplikasi *mobile* memungkinkan penggunanya terhubung ke layanan internet yang biasanya hanya diakses melaului PC atau Notebook. Dengan demikian, aplikasi *mobile* dapat membantu pengguna untuk lebih mudah mengakses layanan internet menggunakan perangkat *mobile* mereka (Wang & Yang, 2018).

Dibandingkan dengan *mobile* phone terdahulu, smartphone dan tablet PC menawarkan berbagai fungsi yang jauh lebih luas. Aplikasi *mobile* semakin banyak digunakan untuk pengelolaan berbagai tugas dalam kehidupan sehari-hari. Saat ini, lebih dari 900.000 aplikasi telah tersedia di *App*le *App Store* (Sistem Operasi: iOS, Pengembang: *App*le) dan kira-kira 700.000 lebih aplikasi telah disediakan juga di

Google Play *Store* (Sistem Operasi: Android, pengembang: *Google*) (Turban, 2018). Melalui aplikasi *mobile* , pengguna juga dapat mengakses sejumlah informasi informasi penting menggunakan smartphone yang terkoneksi dengan layanan internet. Keunggulan utama dari aplikasi *mobile* yaitu memberikan kemudahan pengguna dalam mendapatkan informasi secara portable tanpa menggunakan PC atau netbook dan pemanfaaatannya dalam memperoleh informasi secara up to date terpenuhi tanpa terhalang waktu dan tempat keberadaan pengguna perangkat *mobile* serta areanya yang dapat terjangkau jaringan komunikasi internet (Turban, 2018). Selain itu, Akses pada sebuah website dapat dilakukan melalui aplikasi *mobile* menggunakan perangkat *mobile* pengguna. Ukuran layar dan resolusi yang secara otomatis menyesuaikan dengan ukuran halam web versi *mobile* mengurangi pemakaian bandwith atau tidak memerlukan bandwith yang terlalu besar (Jadhav & dkk, 2019).

1.3 *Google* Maps

Google Maps adalah layanan pemetaan web yang dikembangkan oleh *google*. Layanan ini memberikan citra satelit, peta jalan, panorama, kondisi lalu lintas, dan perencanaan rute untuk berpergian dengan berjalan kaki, kendaraan, sepeda (versi peta), atau angkutan umum, *google Maps* dimulai sebagai program desktop CC+, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada *Where 2 Technologies*. (Wikipedia, 2017)

1.4 Pengertian *Website*

Menurut (Bekti, 2018), menyimpulkan bahwa website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing di hubungkan dengan jaringan-jaringan halaman. Menurut (Bekti, 2018), secara garis besar website dapat berfungsi sebagai:

1. Media Promosi

   Sebagai media promosi yang dapat di bedakan menjadi media promosi utama, misalnya website yang berfungsi sebagai mesin pencarian atau online shop, atau sebagai penunjang promosi utama, namun website dapat berisi informasi yang lebih lengkap daripada media promosi yang tidak menggunakan digital atau online seperti koran atau majalah.

2. Media Pemasaran

   Pada toko online atau sistem afiliasi, website merupakan media pemasaran yang sangat baik dibandingkan dengan toko konvensional sebagaimana di dunia nyata, untuk membangun online shop diperlukan modal yang relatif lebih kecil, dan dapat beroperasi selama 24 jam walaupun pemilik website tersebut sedang istirahat atau sedang tidak ditempat, serta dapat diakses darimana saja.

3. Media Informasi

   Website portal, radio atau tv online menyediakan informasi yang bersifat keseluruhan karena dapat diakses darimana saja selama dapat terhubung ke

internet, sehingga dapat menjangkau lebih luas daripada media informasi konvensional seperti koran, majalah lain yang bersifat lokal.

4. Media Pendidikan

   Adanya komunitas yang membangun aplikasi website khusus yang berisi informasi atau media artikel yang syarat dengan informasi ilmiah misalnya wikipedia.

5. Media Komunikasi

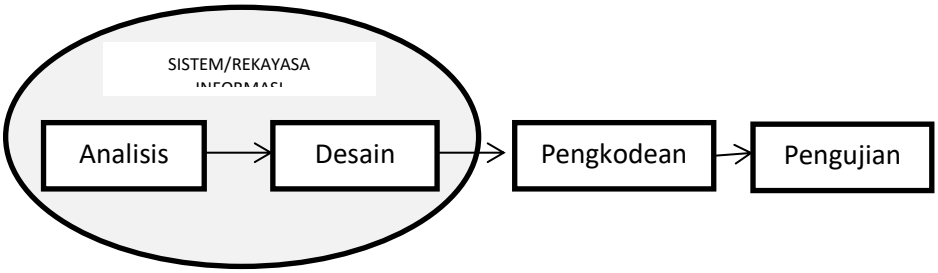
   Sekarang banyak terdapat website yang di bangun secara khusus untuk berkomunikasi seperti forum chat yang dapat memberikan fasilitas bagi para anggotanya untuk saling berkomunikasi atau berbagi informasi untuk membantu pemecahan masalah tertentu.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa pengertian website adalah halaman-halaman yang digunakan untuk menampilkan informasi teks atau gambar sebagai media promosi atau pemasaran.

1.5 Model Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alir hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian, dan tahap pendukung (*support*) (Rosa & Shalahuddin., 2019).

Berikut adalah gambar model air terjun ditunjukkan pada gambar 2.1.



Gambar 1.1 Sistem Model *Waterfall*

1. Analisis

   Proses pengumpulan kebutuhan di lakukan secara baik untuk mendskripsikan kebutuhan perangkat lunak agar dapat di pahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan secara baik.

2. Desain

   Desain merupakan proses multi desain yang fokus pada desain pembuatan program perangkat lunak yang termaksud dalam struktur data, arsitektur perangkat lunak, representasi perangkat lunak, dan prosedur pengkodingan. Tahap ini memanupulasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke memanupulasi desain agar pengguna dapat mengimplementasikan program pada tahap selanjutnya. Desain perangkat lunak yang di hasilkan pada tahap ini juga perlu untuk didokumentasikan.

3. Pembuatan Kode Program

   Desain harus di implementasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain atau kebutuhan yang telah di buat pada tahap desain.

4. Pengujian

   Pengujian yang berfokus pada perangkat lunak atau software secara dari segi lojik dan fungsionalnya dan memastikan bahwa seluruh bagian sudah di uji. Hal ini di laksanakan untuk meminimalisir kesalahan *(error)* dan memastikan keluaran hasilnya sesuai dengan yang di inginkan.

5. Pendukung *(support)* atau pemeliharaan *(maintenance)*

   Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemelihara dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai dengan alurnya karena sebab sebagai berikut:

1. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
2. Sangat sulit bagi siswa untuk mendefinisikan semua spesifikasi diawal alur pengembangan. Siswa sering sekali butuh contoh (*ptototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
3. Siswa tidak mungkin bersabar mengakomodasi perubahan yang diperlukan diakhir alur pengembangan.

Dengan beberapa kelemahan yang dimiliki model air terjun tetapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikkan model pengembangan perangkat lunak. model air terjun sangat cocok digunakan kebutuhan siswa sudah sangat dipahami dan kemungkinan terjadi perubahan kebutuhan selama pengembangan perangkat lunak sangat kecil. Hal positive dari

model air terjun adalah struktur tahap pengembangan sistem jelas, dokumentasi di hasilkan disetiap tahap pengembangan ,dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih saat pelaksanaan).

1.5.2 Siklus Metode Waterfall

Alasan peneliti menggunakan Waterfall adalah, karena sesuai dengan kegiatan penulis, yaitu:

1. Peneliti melakukan analisis terlebih dahulu ke sekolah, seperti wawancara dan pengamatan langsung ke sekolah
2. Kemudian peneliti melakukan desain yang paling mudah yaitu memakai UML (Unified Modeling Language) seperti *usecase diagram*, *class diagram* dan *activity diagram*.
3. Setelah itu peneliti melakukan pengodean menggunakan bahasa pemrograman PHP Hypertext Processor dan database seperti MySQL,
4. Yang terakhir penulis melakukan pengujian sistem, dalam penelitian ini menggunakan pengujian ISO 25010. Lebih rinci akan di jelaskan pada bab 3 penelitian ini.

1.6 Pemodelan Unified Modeling Language (UML)

1.6.1 Pengenalan UML

Sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek,yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML *(Unified*

Modeling Language) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan Shalahuddin. M, 2019).

1.6.2 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal denghan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang. Konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* sehingga menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML) Sehingga pada tahun 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait dalamnya. Secara fisik, UMl adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UMLyang terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu Diagram Interchange Specification,,. UML Infrastructure, UML Superstructure, dan Object Constraint Language (OCL). (Rosa & Shalahuddin., 2019).

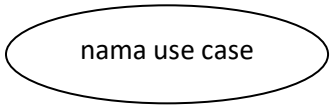
1.6.3 Use Case Diagram

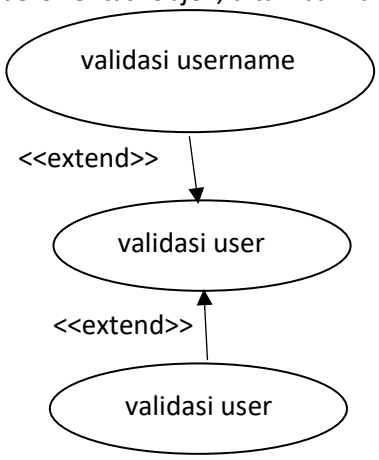
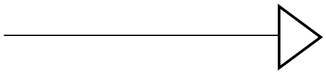
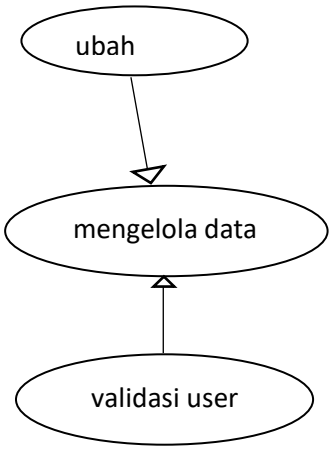
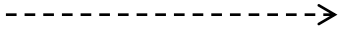
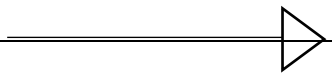
Use case atau diagram *use case* yaitu pemodelan untuk *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan interaksi antara suatu atau lebih dari actor sehingga sistem informasi yang akan dibuat. Penamaan *use case* adalah nama yang didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case diagram* yaitu pendefinisian apa yang disebut aktor dan *use case*.

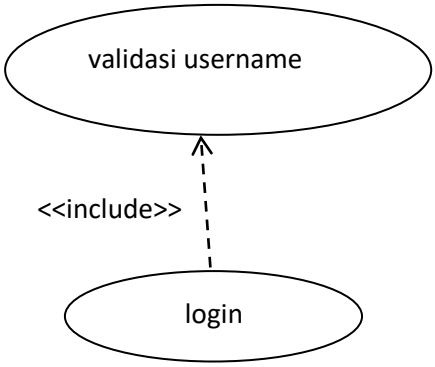
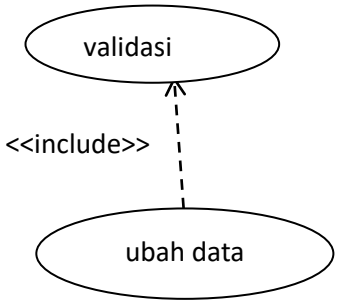
1. Aktor yaitu orang atau sistem yang berinteraksi dengan sistem informasi yang dibuat, sehingga walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang.
2. *Use case* yaitu fungsionalitas sistem yang disediakan sistem untuk unit-unit yang saling mengirim pesan antar unit atau aktor.

Berikut ini adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 1.1 Komponen Use Case Diagram

Simbol	Deskipsi
Use case 	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 use case.
Aktor / *actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 actor.
Asosiasi / *association*	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.

Simbol	Deskipsi
Ekstensi / *extend* *<<extend>>*	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; ditambahkan, misal  arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.
Generalisasi /*generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. misalnya: 
<<include>>  Menggunakan / *include* / *uses* <<uses>> 	Relasi use case tambahan ke sebuah use case dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

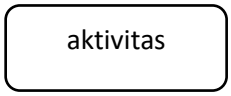
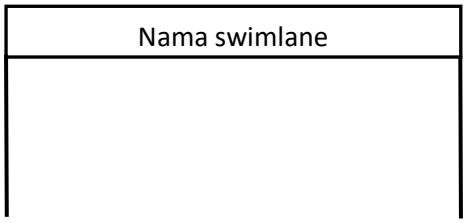
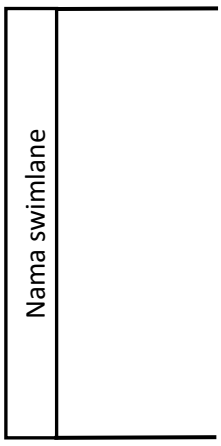
Simbol	Deskipsi
	Ada dua sudut pandang yang cukup besar mengenai include di *use case*: Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut:  Include berarti *use case* yang ditambahkan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:  Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

1.6.4 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja)atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Berikut ini adalah simbol-simbol yang ada pada diagram activity :

Tabel 1.2 Komponen Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuahdiagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasai bisnis yang bertanggungjawab terhadap aktivitas yang terjadi 

1.6.5 Class Diagram

Diagram kategori service atau *class diagram* menggambarkan struktur sistem

dari segi pendefinisian kategori service-kategori service yang akan dibuat untuk membangun sistem". Kategori service memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kategori service.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kategori service.

Susunan struktur kategori service yang baik pada diagram kategori service sebaiknya memiliki jenis-jenis kategori service berikut:

1. Kategori service main

   Kategori service yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kategori service yang menangani tampilan sistem *(view)*

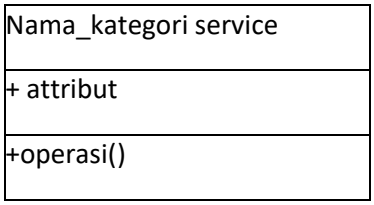
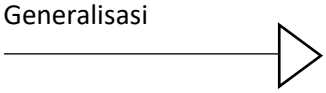
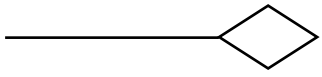
   Kategori service yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kategori service yang diambil dari pendefinisian *use case (controller)*

   Kategori service yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kategori service ini biasanya disebut dengan kategori service proses yang menangani proses bisnis pada perangkat lunak.
4. Kategori service yang diambil dari pendefinisian data *(model)*

   Kategori service yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

   Berikut adalah simbol-simbol yang ada pada class diagram :

Tabel 1.3 Class Diagram

Simbol	Deskripsi
Kategori service 	Kategori service pada struktur system
Antarmuka / *interface*  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / *association* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Asosiasi berarah / *directed association* 	Relasi antarkategori service dengan makna kategori service yang satu digunakan oleh kategori service yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antarkategori service dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antarkategori service dengan makna kebergantungan antarkategori service.
Agregasi / *aggregation* 	Relasi antarkategori service dengan makna semua- bagian (whole-part).

1.7 Kualitas Perangkat Lunak ISO/IEC 25010

Menurut Wattiheluw (2019), ISO/IEC merupakan standar yang digunakan oleh dunia internasional untuk melakukan evaluasi atau penguku- ran kualitas dari

perangkat lunak. ISO/IEC yang digunakan dalam penelitian ini adalah versi 25010 yang merupa- kan versi lanjutan dari ISO/IEC 25010 dengan penambahan beberapa struktur dan bagian dari standar model kualitas. Secara keseluruhan ISO/IEC 25010 memiliki 8 karakteristik untuk mengukur kualitas perangkat lunak secara me- nyeluruh, antara lain portability, performance efficiency, reliability, security usability, maintainability, compatibil- ity, dan functional suitability. Adapun beberapa definisi karakteristik ISO/IEC 25010 adalah sebagai berikut :

1. Functional suitability adalah produk aplikasi yang memberikan fungsional untuk memenuhi kebutuhan saat menggunakan produk dalam keadaan tertentu.
2. Reliability adalah tingkat dimana produk aplikasi dapat mempertahankan kinerja pada level tertentu ketika digunakan dalam keadaan tertentu.
3. Performance efficiency adalah tingkat dimana produk aplikasi menyediakan performa yang baik dengan jumlah resource yang digunakan.
4. Usability adalah dimana produk aplikasi mudah dimengerti, dipakai dan menarik untuk digunakan.
5. Security adalah tingkat produk aplikasi menyedikan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, atau pengungkapan yang berbahaya.
6. Compatibility adalah kemampuan dari suatu komponen aplikasi atau lebih untuk bertukar informasi.
7. Maintainability adalah tingkat dimana produk aplikasi dapat dimodifikasi. Modifikasi yang dilakukan dapat meliputi perbaikan, pengembangan atau adaptasi perangkat lunak untuk menyesuaikan dengan lingkungan, serta

modifikasi pada kriteria dan spesifikasi fungsi.

8. Portability adalah tingkat dimana produk aplikasi dapat dipindahkan dari satu ruang ke ruang lain.