

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Dalam penelitian ini akan digunakan lima referensi sebagai tinjauan pustaka. Berikut daftar literatur yang digunakan:

Tabel 2. 1 Tinjauan Literatur

Nomor Literatur	Penulis	Tahun	Judul
Literatur 001	Herdiansah, Arief & Handayani, Tuti & Yunita, Elsy.	2020	Rancang Bangun Sistem Pemasaran Properti Berbasis Web.
Literatur 002	Prima dan Nurahman	2019	Sistem Informasi Pemasaran dan Manajemen Pada PT.Citra Mandiri Dwi Pratama Berbasis <i>Web</i> .
Literatur 003	Rohadi, Pamenang dan Krisdiyanto	2020	Sistem Informasi Rekomendasi Pemilihan Perumahan di Kawasan Kabupaten Malang.
Literatur 004	Muhammad Fahrozi, Dedi Diartono, Dwi Agus	2015	Sistem Informasi Pemasaran Rumah di Graha Permata Surya Mijen Semarang Berbasis <i>Web</i> .

Literatur 005	Amirullah, Megawati	2016	Rancang Bangun Sistem Informasi Pemasaran Perumahan Permata Bening Berbasis <i>Web</i> .
------------------	------------------------	------	--

2.1.1. Tinjauan Terhadap Literatur 001

Dalam penelitian ini, dimana penulis membahas mengenai permasalahan di PT Akila Trijaya yang masih menggunakan promosi kepada konsumen secara manual atau konvensional melalui pameran di pusat perbelanjaan yaitu mall dan memerlukan biaya untuk melakukan promosi. Penulis melakukan penelitian dan bagaimana merancang bangun sistem pemasaran properti berbasis website menggunakan bahasa pemrograman PHP dengan database My SQL. Metode analisis pengembangan sistem menggunakan waterfall pengujian dengan metode *Blackbox*. Hasil dari penelitian ini adalah sebuah Rancang Bangun Sistem Pemasaran Properti di PT Akila Trijaya Berbasis *Website*.

2.1.2. Tinjauan Terhadap Literatur 002

Penelitian yang dilakukan penulis pada PT.Citra Mandiri Dwi Pratama masalah yang ada masih berjalan manual nasabah harus datang ke kantor untuk mencari info perumahan atau kavling dalam pemesanannya dan pengumpulan berkas syarat pemesanan nasabah harus mengantarnya ke kantor pada jam kerja. Penelitian ini menggunakan pengumpulan data dan bagaimana membangun suatu informasi terstruktur mengetahui pemesanan secara real time. Perancangan sistem ini menggunakan bahasa pemrograman PHP, Java, CSS. Hasil dari penelitian ini adalah Sistem Informasi untuk PT. Citra Mandiri Dwi Pratama.

2.1.3. Tinjauan Terhadap Literatur 003

Oleh Erfan Rohadi, Muhammad unggul Pemenang, Reynaldo Krisdayanto (2020) dalam rancang bangun sistem di penelitian ini

menggunakan metode *waterfall*. Metode perhitungan yang dipakai dalam penelitian ini adalah metode *Additive Ratio Assessment*(ARAS), dari penelitian yang dilakukan menunjukkan bahwa aplikasi tersebut dapat membantu *user* dalam memberikan informasi dan rekomendasi perumahan dengan nilai akurasi 100%. Hasil dari penelitian ini Sistem Informasi Rekomendasi Pemilihan Perumahan di Kawasan Kabupaten Malang yang dapat membantu pengguna dengan menghasilkan skor 95,75 masuk dalam kategori *Best Imaginable*.

2.1.4. Tinjauan Terhadap Literatur 004

Dalam penelitian ini, masalah dari sistem penjualan masih menggunakan brosur dan kelihaihan karyawan memikat konsumen. Jangkauan yang masih terbatas maka penulis ingin jangkauan yang luas memasarkan produk ke semua masyarakat tanpa ada batasan waktu atau tempat. Dari hasil penelitian Sistem mengatasi kelemahan yaitu Sistem informasi Pemasaran Rumah di graha Permata Surya Mijen Semarang berbasis Web sistem dibuat menggunakan bahasa pemrograman PHP dan database MySQL. Hasil akhir dari penelitian sistem dapat mengajukan pengambilan KPR dan transaksi dengan website dengan cara transfer rekening bank.

2.1.5. Tinjauan Terhadap Literatur 005

Oleh Amirullah, Megawati (2016) Jurusan Sistem Informasi, Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Riau, dengan judul Rancang Bangun Sistem Informasi Pemasaran Perumahan Permata Bening Berbasis Web. Dalam penelitian ini membahas masalah masih konvensionalnya sistem dalam pemasaran perumahan calon pembeli harus datang ke kantor belum lagi, pemasaran tidak dilakukan setiap saat menyesuaikan jadwal kerja pegawai ataupun ketika stan dibuka. Penulis melakukan analisa dengan metode analisa *PIECES* dan untuk metode perancangan dalam penelitian ini menggunakan *UML*, perancangan *interface* hingga pengkodean dan

database menggunakan *syntax PHP, MySQL, dan JQuery*, hingga pengujian fungsional sistem menggunakan *Blackbox Testing*. Hasil dari penelitian sesuai dengan yang diharapkan telah memenuhi kebutuhan admin pemasaran dan pelanggan dalam memberikan informasi unit rumah dan menerima informasi unit rumah yang dipasarkan.

2.2. Sistem Informasi

Sistem Informasi adalah kerangka kerja yang mengkoordinasikan sumber daya manusia, komputer untuk mengubah masukan menjadi keluaran untuk mencapai tujuan perusahaan. Guna mencapai sasaran-sasaran perusahaan sistem informasi didalam suatu organisasi yang merupakan rangkaian orang, data, proses, *interface*, teknologi, prosedur-prosedur yang mengeluarkan informasi untuk mencapai tujuan pemecahan masalah atau pengambilan keputusan (Ariska & Jazman, 2016).

2.3. Perumahan

Perumahan merupakan salah satu kebutuhan hidup manusia. Menurut UU No 1 Tahun 2011, Perumahan adalah kawasan pemukiman sebagai tempat tinggal yang layak huni bagi masyarakat baik di perkotaan maupun di pedesaan. Salah satu jenis perumahan yaitu perumahan subsidi dan nonsubsidi, perumahan subsidi yaitu rumah yang diselenggarakan untuk memenuhi kebutuhan masyarakat berpenghasilan rendah sedangkan perumahan non subsidi atau komersil adalah sebaliknya untuk tujuan mendapatkan keuntungan.

2.4. PT Rumah Kita Lampung

PT Rumah Kita Lampung merupakan salah satu perusahaan yang bergerak dalam bidang Developer, Consultan, Agency Marketing, dan produk

interior eksterior.yang membangun serta memasarkan beberapa type perumahan yang ada di kota Lampung. Perusahaan ini didirikan pada tanggal 21 agustus 2020 dengan Akte Notaris Nomor 39 dihadapan Notaris Illanur, S.H., M.M., M.Kn yang berkedudukan di Bandar Lampung. PT Rumah Kita Lampung beralamat jalan Pulau Buru No. 31 Wayhalim Bandar Lampung.

Komitmen PT Rumah Kita Lampung senantiasa berusaha mengembangkan kegiatan usaha serta memberikan pelayanan kepada pembeli atau rekanan kerja dengan lebih baik dan berorientasi pada perkembangan aset jaringan yang luas.

2.5. Website

Website adalah sekumpulan informasi yang dibuat perseorangan maupun kelompok atau organisasi, di dalamnya merupakan kumpulan halaman yang berisi informasi yang disimpan melalui jaringan internet pada perangkat - perangkat yang bisa mengakses internet itu sendiri seperti komputer. Definisi kata Web adalah penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *WORLD WIDE WEB* yang merupakan bagian dari teknologi Internet (Hastanti, Purnama dan Wardati, 2015).

Secara umum website mempunyai fungsi sebagai berikut:

1. Fungsi Komunikasi
Beberapa fasilitas yang memberikan fungsi komunikasi, seperti : chatting, web base email dan lain-lain.
2. Fungsi Informasi
Fungsi informasi website seperti : News, Profile, Library, referensi dan lain-lain.
3. Fungsi Intertainment
Website mempunyai fungsi hiburan. Misalnya web-web yang menyediakan game online, music on-line dan lain-lain.

4. Fungsi Transaksi
Sebuah web dapat dijadikan sarana untuk melakukan transaksi dan lain-lain.
5. Media Pemasaran
6. Pada toko online atau sistem afiliasi, website merupakan media pemasaran yang cukup baik, karena dibandingkan dengan toko sebagaimana di dunia nyata, untuk membangun toko online diperlukan modal yang relatif lebih kecil, dan dapat beroperasi 24 jam walaupun pemilik website tersebut sedang istirahat atau sedang tidak ditempat, serta dapat diakses dari mana saja.
7. Media Pendidikan
8. Ada komunitas yang membangun website khusus berisi informasi atau artikel yang sarat dengan informasi ilmiah misalnya wikipedia.

2.6. Javascript

Javascript adalah bahasa pemrograman skrip pertama untuk web, yang berbentuk kumpulan skrip fungsinya berjalan di dokumen HTML. Bahasa ini adalah Bahasa pemrograman yang memberikan kemampuan lebih di HTML dengan izin pengeksekusian perintah di sisi user, yang artinya di browser bukan sisi server web (A Alamsyah, 2003). Javascript tidak membutuhkan kompilator atau penterjemah untuk menjalankannya (kompilator sendiri sudah dalam browser tersebut).

Javascript diperkenalkan pertama kali oleh *Netscape* pada tahun 1995. Pada awalnya bahasa ini dinamakan *Livescore* yang berfungsi sebagai Bahasa sederhana untuk browser Netscape Navigator 2. Kemudian sejalan dengan giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa

pemrograman *java*) pada masa itu maka *Netscape* memberikan nama *Javascript* kepada Bahasa tersebut pada tanggal 4 desember 1995. Pada saat bersamaan Microsoft mencoba untuk mengadaptasikan teknologi ini yang mereka sebut *Jscript* di browser Internet Explorer 3.

2.7. Basis Data (*Database*)

Basis data terdiri dari dua kata, yaitu basis dan data (Fathansyah, 2007). Basis kurang lebih dapat diartikan sebagai markas atau Gudang, tempat berkumpulnya. Sedangkan data adalah representasi fakta dunia nyata yang mewakili sebuah objek seperti manusia, barang konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar atau kombinasinya. Basis data sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsipnya adalah pengaturan data/arsip gunanya mudah dan cepat pengambilan data. Namun perbedaannya hanya terletak pada media penyimpanan yang digunakan.

Basis data juga dapat didefinisikan dalam sejumlah sudut pandang, yaitu;

- a. Kumpulan data yang saling berhubungan dan disimpan secara Bersama sedemikian rupa tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- b. Kumpulan file/table/arsip saling berhubungan yang disimpan dalam media penyimpanan.
- c. Himpunan kelompok data yang saling terhubung di organisasi sedemikian rupa agar kelak dapat dimanfaatkan Kembali dengan cepat dan mudah.

2.8. MongoDB

MongoDB diklasifikasikan sebagai basis data NoSQL yaitu menghindari struktur basis data relasional, membuat kesatuan data dalam beberapa jenis aplikasi mudah dan cepat. Nama *MongoDB* diambil dari kata humongous yaitu artinya gagasan support sejumlah besar data. *MongoDB* adalah sistem manajemen basis data berbasis open source yang kegunaannya dalam aplikasi data besar dan pemrosesan lainnya. *MongoDB* menggunakan arsitektur dari koleksi dan dokumen. Keunggulan dari *MongoDB* mendukung data yang menjalankan program big data bagi perusahaan yang juga memiliki dokumen yang disematkan akan penggabungan basis data yang dapat menekan biaya, sehingga meningkatkan efisiensi.

2.9. React Js

Menurut situs resmi reactjs.org, merupakan library javascript yang bersifat open source yang dibuat oleh facebook. Untuk membangun *user interface* sebuah *web application*, dimanakita dapat membuat beberapa komponen user interface pada aplikasi yang akan dikembangkan secara kompleks.

2.10. MongoDB Compass

Mongodb Compass merupakan tool *GUI* resmi untuk MongoDB yang dikelola oleh MongoDB itu sendiri. MongoDB compass dapat diakses tanpa menggunakan kueri dengan kelebihan *GUI* didalamnya memudahkan pengguna membuat keputusan cerdas untuk struktur data. Untuk menjelajahi, menganalisis, dan berinteraksi dengan konten yg disimpan dalam database MongoDB yang big data di perusahaan. Secara keseluruhan MongoDB Compass dapat diartikan alternatif untuk mengakses data di MongoDB dengan mudah dan cepat tanpa menggunakan kueri. Ini merupakan tool gratis yang

digunakan untuk semua orang dan akan membuat kita mampu bekerja mudah dengan MongoDB apabila dengan tool lainnya.

Compass dapat melakukan operasi termasuk:

- a. Visualisasikan dan jelajahi data yang disimpan dalam database
- b. Buat database dan sisipkan, perbarui, dan hapus database
- c. Dapatkan statistik langsung server waktu nyata
- d. Pahami masalah kinerja dengan rencana penjelasan visual
- e. Kelola indeks
- f. Validasi data dengan aturan validasi skema JSON
- g. Dapat diperpanjang melalui plugin.

2.11. Visual Studio Code

Visual Studio Code adalah software yang sangat ringan, namun kuat editor kode sumbernya berjalan dari desktop. Muncul dengan *built-in* dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki *array* beragam ekstensi yang tersedia untuk Bahasa lain, termasuk C++, C #, Python, dan PHP. Berdasarkan JavaScript dan HTML5, editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi *cloud* yang terbuka *Microsoft*. *Visual Studio Code* menggunakan *open source NET* perkakas untuk memberi dukungan untuk ASP.NET C # kode, membangun alat pengembang *Omnisharp* NET dan *Compiler Roslyn*. Antarmuka yang mudah untuk bekerja dengan didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder anda memiliki akses ke panel editor di sebelah kanan, menunjukkan isi dari file yang telah dibuka.

Visual Studio Code benar-benar ditargetkan pada pengembang JavaScript yang ingin alat pengembangnya lengkap untuk *scripting server-side* mereka dan yang mungkin ingin usaha dari *Node.js* untuk kerangka

berbasis NET. *Visual Studio Code*, adalah belum solid, lintas platform kode editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web.

2.12. Mockup

Mockup adalah sebuah media visual atau preview dari sebuah konsep desain "datar" yang diberikan efek visual sehingga hasilnya sangat tampak atau menyerupai wujud nyata, mockup dapat memberikan gambaran nyata dari sebuah konsep desain bagaimana konsep itu akan terlihat nantinya jika sudah diaplikasikan menjadi atau ke dalam benda nyata, apakah terlihat bagus atau kurang sesuai. Mockup juga dapat menambahkan nilai visual dari sebuah konsep desain, misalnya sebuah desain dibuat dalam versi datar maka desain itu akan terlihat bisa-bisa saja, namun hanya dengan menggunakan mockup untuk membuat sebuah konsep desain maka desain tersebut akan terlihat lebih menarik dan mengundang apresiasi(Sabila, Rosely dan Nugroho, 2018).


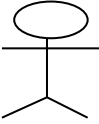

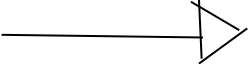
2.13. Unified Modelling Language (UML)

Menurut Rosa dan Shalahuddin (2013), mendefinisikan UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek".UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Diagram-diagram dari UML yaitu *Usecase diagram*, *Activity diagram*. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *UML(Unified Modelling Language)*.

2.14. Use Case Diagram

Menurut (Rosa & Salahuddin, 2014) *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 berikut:

Tabel 2. 2 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 frase nama <i>use case</i> .
2.		Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/ <i>association</i> merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< Include >>>	Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.


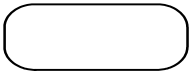
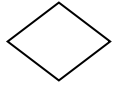

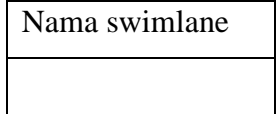
6.	<<Extend>>>	Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
----	----------------------	--


Sumber :(Rosa & Salahuddin, 2014).

2.15. Activity diagram

Menurut (Rosa & Salahuddin, 2013) *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini

Tabel 2. 3 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.		<i>Swimlane</i>

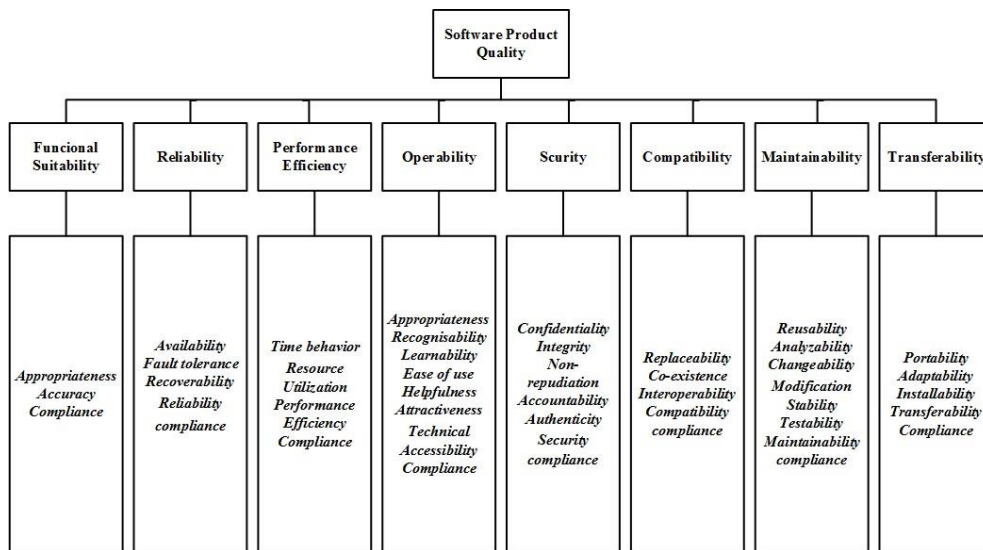
		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber : (Rosa & Salahuddin, 2014).

2.16. Pengujian Sistem (ISO 25010)

Pengujian perangkat lunak adalah elemen penting dari jaminan kualitas dan mempresentasikan spesifikasi, desain dan pengkodean (Pressman, 2010). Sering perangkat lunak mengandung kesalahan (*error*) pada proses-proses tertentu pada saat perangkat lunak sudah berada ditangan pengguna. Kesalahan- kesalahan pada perangkat lunak ini sering disebut dengan *bug*. Untuk menghindari banyaknya *bug* maka diperlukan pengujian sistem perangkat lunak sebelum perangkat lunak didistribusikan ke pelanggan atau selama perangkat lunak masih dalam pengembangan (Rosa, A. S & Shalahuddin, 2014). Dalam penelitian ini dilakukan pengujian sistem, dengan menggunakan *ISO 25010*.

Salah satu tolak ukur kualitas perangkat lunak menggunakan *ISO 25010*, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)* (Wagner, 2013: 2). *ISO 25010* hadir untuk memperbarui versi sebelumnya yaitu standar *ISO/IEC 9126* (ISO, 2011). Model kualitas produk (*ISO, 2011*) mengategorikan sifat kualitas produk menjadi delapan karakteristik yaitu *Functional Suitability, Reliability, Performance Efficiency, Usability, Security, Compatibility, Maintainability, dan Portability*.



Gambar 2.1 Karakteristik ISO 25010
(Sumber : BS ISO/IEC 25010, 2011)

1. *Functional Suitability*

Functional Suitability merupakan tingkat dimana produk perangkat lunak menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika perangkat lunak digunakan dalam kondisi tertentu. Subkarakteristik *Functional Suitability* meliputi *appropriateness*, *accuracy*, dan *compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Functional Suitability* yang dapat dilihat pada Tabel 2.4.

Tabel 2.4 Subkarakteristik *Functional Suitability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Functional Suitability</i>	<i>Appropriateness</i>	Sejauh mana produk perangkat lunak menyediakan fungsi yang tepat untuk tugas-tugas tertentu dan tujuan pengguna?

	<i>Accuracy</i>	Sejauh mana produk perangkat lunak memberikan hasil yang tepat atau spesifik dengan tingkat presisi yang diperlukan?
--	-----------------	--

Karakteristik	Subkarakteristik	Penjelasan
	<i>Compliance</i>	Tingkat dimana perangkat lunak mematuhi standar, konvensi, atau peraturan dalam undang-undang dan peraturan serupa yang berkaitan dengan kesesuaian fungsional?

(Sumber : BS ISO/IEC 25010, 2011)

2. *Reliability*

Reliability didefinisikan sejauh mana perangkat lunak dapat mempertahankan tingkat kinerja dalam kondisi tertentu. Subkarakteristik *Reliability* meliputi *availability*, *fault tolerance*, *recoverability*, *reliability compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Reliability* yang dapat dilihat pada Tabel 2.5.

Tabel 2.5 Subkarakteristik *Reliability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Reliability</i>	<i>Availability</i>	Sejauh mana komponen perangkat lunak beroperasi dan tersedia saat diperlukan untuk digunakan.
	<i>Fault tolerance</i>	Sejauh mana produk perangkat lunak dapat mempertahankan tingkat kinerja tertentu dalam kasus kesalahan perangkat lunak atau pelanggaran antarmuka yang ditentukan.

	<i>Recoverability</i>	Tingkat di mana produk perangkat lunak dapat menetapkan kembali tingkat kinerja yang ditentukan dan memulihkan data yang secara langsung terpengaruh dalam kasus kegagalan.
--	-----------------------	---

Tabel 2.6 Subkarakteristik *Reliability* (Lanjutan)

	<i>Reliability compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keandalan.
--	-------------------------------	---

(Sumber : BS ISO/IEC 25010, 2011)

3. *Performance efficiency*

Performance efficiency merupakan sejauh mana perangkat lunak memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Subkarakteristik *Performance efficiency* meliputi *time behaviour*, *resource utilization*, *performance efficiency compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Performance efficiency* yang dapat dilihat pada Tabel 2.7.

Tabel 2.7 Subkarakteristik *Performance efficiency*

Karakteristik	Subkarakteristik	Penjelasan
<i>Performan efficiency</i>	<i>Time behavior</i>	Sejauh mana perangkat lunak memberikan respons yang tepat, waktu pemrosesan dan laju keluaran ketika menjalankan fungsinya dalam kondisi yang ditentukan.

	<i>Resource utilization</i>	Sejauh mana perangkat lunak menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak menjalankan fungsinya dalam kondisi yang ditentukan.
	<i>Performance Efficiency Compliance</i>	Sejauh mana perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan efisiensi kinerja.

(Sumber : BS ISO/IEC 25010, 2011)

4. *Operability*

Operability merupakan sejauh mana produk perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik bagi pengguna bila digunakan dalam kondisi tertentu. Subkarakteristik *Operability* meliputi *appropriateness recognisability*, *learnability*, *ease of use*, *helpfulness*, *attractiveness*, *technical accessibility*, dan *compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Operability* yang dapat dilihat pada Tabel 2.8.

Tabel 2.8 Subkarakteristik *Operability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Operability</i>	<i>Appropriateness Recognisability</i>	Sejauh mana perangkat lunak memungkinkan pengguna untuk mengenali apakah perangkat lunak sesuai dengan kebutuhan pengguna?
	<i>Learnability</i>	Sejauh mana perangkat lunak memungkinkan pengguna untuk mempelajari aplikasinya?

<i>Ease of use</i>	Sejauh mana perangkat lunak memudahkan pengguna untuk mengoperasikan dan mengendalikannya?
<i>Helpfulness</i>	Sejauh mana perangkat lunak dapat membantu pengguna?
<i>Attractiveness</i>	Apakah antarmuka terlihat baik?
<i>Technical Accessibility</i>	Tingkat pengoperasian perangkat lunak untuk pengguna dengan cacat tertentu?
<i>Compliance</i>	Apakah perangkat lunak sudah mematuhi standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan pengoperasian?

(Sumber : BS ISO/IEC 25010, 2011)

5. *Security*

Security merupakan perlindungan item sistem dari akses yang tidak disengaja atau berbahaya, penggunaan, modifikasi, perusakan dan pengungkapan. Subkarakteristik *Security* meliputi *confidentiality*, *integrity*, *non-repudiation*, *accountability*, *authenticity*, *security compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *security* yang dapat dilihat pada Tabel 2.9.

Tabel 2.9 Subkarakteristik *Security*

Karakteristik	Subkarakteristik	Penjelasan
<i>Security</i>	<i>Confidentiality</i>	Sejauh mana perangkat lunak memberikan perlindungan dari

		pengungkapan data atau informasi yang tidak sah, baik disengaja atau disengaja.
	<i>Integrity</i>	Sejauh mana ketepatan dan kelengkapan aset dijaga.
	<i>Non-repudiation</i>	Sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak kemudian.
	<i>Accountability</i>	Sejauh mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
	<i>Authenticity</i>	Sejauh mana identitas suatu subjek sumber daya dapat dibuktikan sebagai yang diklaim.
	<i>Security compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keamanan.

(Sumber : BS ISO/IEC 25010, 2011)

6. *Compatibility*

Compatibility merupakan kemampuan dua atau lebih komponen perangkat lunak untuk bertukar informasi dan untuk melakukan fungsi yang diperlukan saat berbagi perangkat keras atau perangkat lunak yang sama. Subkarakteristik *compatibility* meliputi *replaceability*, *co-existence*, *interoperability*, *compatibility compliance*. Berikut ini penjelasan untuk masing-masing subkarakteristik *compatibility* yang dapat dilihat pada Tabel 2.10.

Tabel 2.10 Subkarakteristik *Compatibility*

Karakteristik	Subkarakteristik	Penjelasan
<i>Compatibility</i>	<i>Replaceability</i>	Sejauh mana perangkat lunak dapat digunakan ditempat perangkat lunak lain yang ditentukan untuk tujuan yang sama dilingkungan yang sama.
	<i>Co-existence</i>	Sejauh mana perangkat lunak dapat bekerja sama dengan perangkat lunak independen lainnya dalam lingkungan umum berbagi sumber daya umum tanpa ada dampak yang merugikan.
	<i>Interoperability</i>	Sejauh mana perangkat lunak dapat dioperasikan secara kooperatif dengan satu atau lebih perangkat lunak lainnya.
	<i>Compatibility compliance</i>	Sejauh mana perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan kompatibilitas.

(Sumber : BS ISO/IEC 25010, 2011)

7. *Maintainability*

Maintainability merupakan sejauh mana perangkat lunak dapat dimodifikasi. Modifikasi dapat mencakup koreksi, peningkatan atau adaptasi perangkat lunak terhadap perubahan lingkungan, dan persyaratan serta spesifikasi fungsional. Subkarakteristik *Maintainability* meliputi *modularity*, *reusability*, *analyzability*,

changeability, modification stability, testability, maintainability compliance. Berikut ini penjelasan untuk masing-masing subkarakteristik *maintainability* yang dapat dilihat pada Tabel 2.11.

Tabel 2.11 Subkarakteristik *Maintainability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Maintainability</i>	<i>Reusability</i>	Sejauh mana aset dapat digunakan lebih dari satu sistem perangkat lunak, atau dalam membangun aset lainnya.
	<i>Analyzability</i>	Tingkat dimana perangkat lunak dapat didiagnosis untuk kekurangan atau penyebab kegagalan dalam perangkat lunak.
	<i>Changeability</i>	Sejauh mana perangkat lunak memungkinkan modifikasi tertentu untuk diimplementasikan.
	<i>Modification stability</i>	Sejauh mana perangkat lunak dapat menghindari efek tak terduga dari modifikasi perangkat lunak.
	<i>Testability</i>	Sejauh mana perangkat lunak yang dimodifikasi untuk divalidasi.
	<i>Maintainability compliance</i>	Sejauh mana perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan pemeliharaan.

(Sumber : BS ISO/IEC 25010, 2011)

8. *Transferability*

Merupakan sejauh mana perangkat lunak dapat ditransfer dari satu lingkungan ke lingkungan lain. Subkarakteristik *transferability* meliputi *portability, adaptability, installability, transferability*

compliance. Berikut ini penjelasan untuk masing-masing subkarakteristik *transferability* yang dapat dilihat pada Tabel 2.12.

Tabel 2.12 Subkarakteristik *Transferability*

Karakteristik	Subkarakteristik	Penjelasan
<i>Transferability</i>	<i>Portability</i>	Kemudahan sistem atau komponen Yang dapat ditransfer dari satu perangkat keras atau perangkat lunak ke perangkat lain
	<i>Adaptability</i>	Apakah perangkat lunak dapat disesuaikan dengan lingkungan tertentu Yang Berbeda
	<i>Installability</i>	Apakah perangkat lunak dapat diinstal dan dihapus pada lingkungan tertentu
	<i>Transferability Compliance</i>	Apakah perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan portabilitas.

(Sumber : BS ISO/IEC 25010, 2011)