

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk mendukung dalam penelitian ini, maka diperlukan tinjauan pustaka yang diambil dari beberapa jurnal terdahulu atau yang sudah ada dan berkaitan dengan judul penelitian serta pokok pembahasan dalam penelitian. Tinjauan pustaka disajikan pada tabel 2.1:

Tabel 2.1 Tinjauan Pustaka

1.	Judul	Sistem Informasi Pelayanan Pasien Studi Kasus Klinik Jati Husada Tangerang
	Penulis	(Herdiansah, Fitriawati and Hariri, 2021)
	Metode	Waterfall
	Permasalahan	Proses pendataan pasien dan rekam medis pasien masih dilakukan secara manual. Dokter yang bertugas di klinik tersebut melakukan pencatatan rekam medis pasien pada buku pasien
	Hasil Penelitian	Sistem yang dihasilkan telah dapat membantu petugas klinik dalam mengelola rekam medis, membantu dokter untuk lebih cepat mengetahui data mengelola rekam medis pasien, data obat, data dokter sehingga dapat meningkatkan pelayanan klinik bagi pasien.
2.	Judul	RealTime Pendaftaran Pasien Berbasis Web Service dan SMS Gateway : Pada Masa PSBB di Klinik Karisma Medika
	Penulis	(Hayat, 2021)
	Metode	Waterfall
	Permasalahan	Adanya pembatasan kegiatan yang dikenal dengan istilah 3M, mengharuskan Klinik Karisma Medika mengadaptasi alur kerja operasional dengan memanfaatkan teknologi informasi guna mengadaptasi pendaftaran pasien.
	Hasil Penelitian	Sistem dapat diaplikasikan secara realtime di klinik Karisma Medika pada masa PSBB dilaksanakan. SMS yang diterima pasien berupa nomor antrian dan

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

		waktu periksa konsultasi sehingga pasien bisa memprediksi waktu kunjungan ke klinik sesuai dengan jadwal yang diberikan.
3.	Judul	Sistem Informasi Pelayanan Medis Pada Pasien di Klinik Insani Citeureup Berbasis Java
	Penulis	(Alit, Aruan and Rahadyan, 2020)
	Metode	Waterfall
	Permasalahan	Pengelolaan pendaftaran yang dibuat menggunakan Microsoft excel, sehingga terjadi penumpukan data yang mengakibatkan sulitnya mencari data pendaftaran pasien.
	Hasil Penelitian	Sistem yang di implementasikan menjadikan proses manajemen pengolahan data dan informasi menjadi lebih efisien, menghasilkan output yang lebih cepat sehingga masalah-masalah yang terjadi pada Klinik Insani dapat terselesaikan.
4	Judul	Perancangan Sistem Informasi Pelayanan Klinik Alidza Kota Padang Berbasis Web
	Penulis	(Iskandar and Akhiyar, 2020)
	Metode	Waterfall
	Permasalahan	Sistem pelayanannya yang masih menggunakan sistem lama atau manual, tentu dapat menghambat proses penanganan penyembuhan penyakit yang diderita pasien.
	Hasil Penelitian	Adanya sistem informasi klinik ini proses pencatatan data-data pelayanan lebih efisien dan efektif karena menggunakan sistem yang terkomputerisasi.
5	Judul	Rancang Bangun Sistem Informasi Pendaftaran Pasien Rawat Jalan Pada Rumah Sakit Universitas Riau.
	Penulis	(Muhammad and Ananda, 2020)
	Metode	Waterfall
	Permasalahan	Pada saat ini pendaftaran pasien untuk melakukan pendaftaran rawat jalan pada rumah sakit Universitas Riau dapat dilakukan dengan cara pasien langsung datang kerumah sakit atau dengan menelpon bagian pelayanan. Hal ini akan memerlukan waktu lama bagi pasien, yang mana pasien harus antri atau menelpon berulang-ulang dikarenakan jaringan telepon sering sibuk.

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

	Hasil Penelitian	Hasil dari penelitian ini adalah sebuah sistem pendaftaran rawat jalan berbasis web yang dapat mempermudah pasien didalam melakukan proses pendaftaran rawat jalan yang mana sistem ini dapat diakses oleh dosen, karyawan, mahasiswa dan masyarakat melalui perangkat elektronik dan terhubung dengan akses internet.
--	------------------	--

Berikut perbedaan penelitian ini dengan 5 (lima) penelitian sebelumnya antara lain:

1. Terdapat kebaruaran fitur pada penelitian ini yaitu hasil rekam medis disajikan dengan menggunakan barcode sehingga dapat discan menggunakan alat komunikasi genggam (*smartphone*).
2. Penelitian ini memiliki persamaan tujuan yang di lakukan oleh Alit, Aruan dan Rahadyan (2020) yaitu mempercepat pendaftaran pasien dan mempercepat penemuan kembali data rekam medis pasien, namun sistem yang dirancang dan dibangun masih berbasis dekstop menggunakan Java, sedangkan penelitian ini menggunakan *Web Programming* dengan *Framework Laravel*.
3. Kelima penelitian sebelumnya menggunakan metode peneltian *Waterfall* sedangkan pada penelitian ini menggunakan metode penelitian *Extreme Programming (XP)*.

2.2 Perancangan

Perancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari perancangan adalah untuk memberi gambaran yang jelas serta lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan harus berguna dan mudah dipahami sehingga dapat dengan mudah digunakan.

Perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem tersebut dapat di implementasikan (Novitasari, Adrian dan Kurnia, 2021).

2.3 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama- sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Informasi sendiri merupakan sekumpulan data yang telah diolah sehingga mempunyai nilai dan makna bagi yang menerimanya. Sistem informasi merupakan gabungan yang terdiri dari manusia, teknologi informasi dan prosedur kerja yang memproses, menyimpan, menganalisis dan menyebarkan informasi untuk mencapai suatu tujuan (Alit, Aruan and Rahadyan, 2020).

2.4 Pelayanan

Pelayanan adalah pemberian jasa baik oleh pemerintah, pihak swasta atas nama pemerintah ataupun pihak swasta kepada masyarakat, dengan atau tanpa pembayaran guna memenuhi kebutuhan dan kepentingan masyarakat. Pelayanan juga merupakan suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik, dan menyediakan kepuasan pelanggan (Amalia dan Huda, 2020).

2.5 Mutu Pelayanan Kesehatan

Mutu adalah ukuran yang dibuat oleh konsumen terhadap produk atau jasa yang dilihat dari segala dimensi atau karakteristik untuk memenuhi tuntutan

kebutuhan, keamanan, dan kenyamanan konsumen. Pelayanan kesehatan yang bermutu adalah pelayanan kesehatan yang dapat memuaskan setiap pemakai jasa pelayanan kesehatan sesuai tingkat kepuasan rata-rata penduduk. Penyelenggaraannya juga harus sesuai dengan standar dan kode etik profesi yang telah ditetapkan. Pasien cenderung memilih atau menetapkan prioritas indikator kualitas pelayanan kesehatan, sebagai dasar untuk memutuskan tingkat kepuasannya (Suyanti, 2019).

2.6 Covid-19

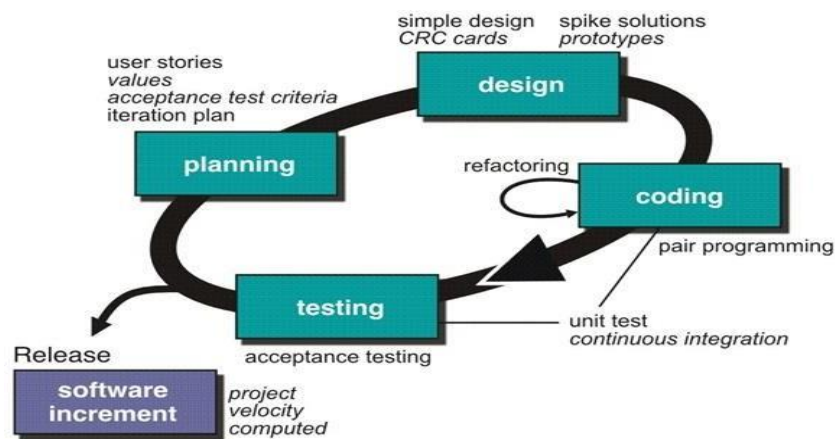
Virus Covid-19 sudah dikenal sejak tahun 1930-an dan diketahui terdapat pada hewan. Pada tahun 2002, muncul penyakit baru golongan Virus Corona yang menyebabkan penyakit Severe Acute Respiratory Syndrome(SARS). Pada tahun 2012, timbul lagi kalangan Virus Corona ini yang menimbulkan penyakit Middle East Respiratory Syndrome(MERS) di Timur Tengah, khususnya negara-negara Arab. Pada bulan Desember 2019, di Kota Wuhan, Tiongkok, terjadi kejadian luar biasa (KLB) kasus radang paru-paru (pneumonia) yang disebabkan oleh virus dari keluarga besar Virus Corona, tetapi virus ini belum pernah dikenal sebelumnya, sehingga disebut sebagai Corona jenis baru atau Novel Coronavirus (= novel, paling baru). Pada 11 Februari 2020, WHO secara resmi mengumumkan penamaan baru virus penyebab pneumonia misterius itu dengan nama Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2) dan nama penyakit yang ditimbulkannya adalah Coronavirus Disease 2019 (Arpiansah, Fernando dan Fakhrurozi, 2021).

2.7 Teori Metode Pengembangan Sistem

2.7.1 Extreme Programming (XP)

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. (Pressman, 2012).

Ada beberapa tahapan yang ada pada Extreme Programming yaitu terdiri dari perencanaan seperti memahami kriteria pengguna dan perencanaan pengembangan, desain seperti perancangan *prototype* dan tampilan, pengkodean juga termasuk dalam pengintegrasian, terakhir adalah pengujian (Ariyanti, Satria and Alita, 2020).



Gambar 2.1 Model *Extreme Programming* (XP)
Sumber : (Pressman, 2012)

2.7.2 Tahapan *Extreme Programming* (XP)

Terdapat empat tahap proses yang dilakukan dalam *Extreme Programming* (Pressman, 2012)

1. *Planning*

Tahapan perancangan yang digunakan untuk memahami konsep bisnis, pengumpulan kebutuhan sistem, menggambarkan output yang di perlukan, fitur-fitur dan fungsionalitas yang akan dibangun menggunakan rekayasa perangkat lunak.

2. *Design*

Metode ini menekankan desain aplikasi yang sederhana mempunyai manfaat untuk pemodelan sistem yang sudah memiliki standae. Tahap ini digunakan untuk dapat memastikan perangkat lunak yang dibangun selesai dengan tepat waktu, sesuai anggaran dan sesuai dengan spesifikasi yang di tetapkan pada tahapan *planning*.

3. *Coding*

Tahap pengkodean perangkat lunak yang menggunakan tim kecil yang bekerja secara bertahap dengan panduan alur sistem yang sudah di rancang pada tahap desain modul permodul. Dengan menggunakan refactoring agar dapat mudah dibaca dan dimodifikasi yang merupakan ciri khas dari metode ini sehingga hasil yang diharapkan dengan pengembangan perangkat lunak menjadi cepat.

4. *Testing*

Tahap akhir dari metode ini yang akan dilakukan dengan cara mendapatkan feedback dari aktor pengguna sistem dengan melakukan pengujian fungsional perangkat lunak.

2.8 Teori *Unifed Modeling Language* (UML)

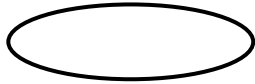


Perancangan sistem dalam penelitian ini menggunakan *Unifed Modeling Language* (UML). UML adalah salah satu standar bahasa yang banyak

digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S dan Shalahudin, 2018).

2.8.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat A.S dan Shalahudin (2018). Simbol-simbol pada *use case diagram* disajikan pada tabel 2.2 dibawah ini:

Tabel 2.2 Simbol Use Case Diagram

No.	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2	<p>Aktor / <i>actor</i></p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	<p>Asosiasi / <i>association</i></p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)



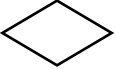


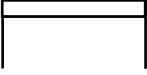
4	Ekstensi / <i>extend</i> <<extend>> ----->	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	Generalisasi / <i>generalization</i> —————>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i> <<include>> ----->	Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini menjalankan fungsionalnya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: (A.S dan Shalahudin, 2018)

2.8.2 *Activity Diagram*

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (A.S and Shalahudin, 2018). Simbol-simbol yang terdapat dalam *Activity Diagram* disajikan pada tabel 2.3 dibawah ini:

Tabel 2.3 Simbol *Activity Diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

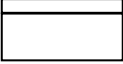


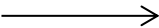
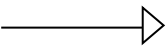
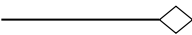
Sumber: (A.S and Shalahudin, 2018)

2.8.3 *Class Diagram*

Class diagram adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem. *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk

mencapai sebuah tujuan (A.S and Shalahudin, 2018). Simbol-simbol yang terdapat dalam *Class Diagram* disajikan pada tabel 2.4 dibawah ini:

Tabel 2.4 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: (A.S and Shalahudin, 2018)

2.9 Teori Perangkat Lunak

Perangkat lunak atau yang sering disebut dengan *software* adalah sekumpulan data elektronik yang disimpan dan disesuaikan oleh komputer. Data elektronik yang disimpan oleh komputer dapat berupa program atau instruksi yang menjalankan perintah. (Handayani, Wijianto dan Anggoro, 2018)

2.9.1 Website

Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet (Amalia and Huda, 2020). Untuk menyediakan sebuah *website*, maka harus menyediakan unsur-unsur penunjangnya (Sari, Damayanti dan Savitri, 2021), yaitu:

1. Nama *Domain* (*URL*)

Nama *domain* atau biasa disebut dengan *domain name* atau *URL* adalah alamat unik di dunia *internet* yang digunakan untuk mengidentifikasi sebuah *website*.

2. Rumah Tempat *Website* (*Web hosting*)

Pengertian *web hosting* dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* yang disewa atau dipunyai.

3. Bahasa Program (*Scripts Program*)

Bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*.

4. Desain *Website*

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*.

5. Publikasi *Website*

Keberadaan situs tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh masyarakat atau pengunjung *internet*. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi *website*.

2.9.2 PHP

PHP adalah kepanjangan dari *Hypertext Preprocessor* merupakan bahasa pemrograman yang dibuat oleh Rasmus Lerdorf pada tahun 1994 dan dapat bekerja bersama ke dalam HTML maupun javascript. Untuk membangun sebuah CMS saat ini PHP banyak digunakan. PHP hanya melakukan eksekusi kode sesuai dengan batas penulisan sintak “<?php” sebagai pembuka dan “?” sebagai penutup (Dewi dkk, 2021).

2.9.3 MySQL

MySQL merupakan 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 (Kharisma, Saniati dan Neneng, 2022).

2.9.4 XAMPP

XAMPP merupakan salah satu *software web server* yang dapat digunakan untuk mengatasi masalah PHP dan MySQL jika dirasa sulit. PHP, MySQL, Apache tergabung dalam satu *software* yaitu XAMPP. Fungsi XAMPP adalah sebagai server yang berdiri sendiri (localhost) yang terdiri dari program Apache HTTP Server, MySQL database, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Per (Puspitasari, Setiawansyah dan Budiman, 2021).

2.9.5 Framework Laravel

Laravel adalah sebuah *MVC web development framework* yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi. Laravel adalah framework PHP opensource yang ditulis oleh Taylor Otwell di bawah lisensi MIT. Laravel dibuat untuk membantu para developer dalam membuat sebuah web dengan sintaks yang sederhana, mudah, elegan, dan menyenangkan (Ramadhanu dan Priandika, 2021).

2.10 Pengujian ISO 25010

Standar *ISO/IEC 25010* merupakan model kualitas sistem dan perangkat lunak yang menggantikan *ISO/IEC 9126* tentang *software engineering* (Ardian dan Fernando, 2020). Hal ini dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak. Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko (Permana dan Puspaningrum, 2021). Secara keseluruhan, *ISO/IEC 25010* memiliki 6 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain *Functional Suitability*, *Reliability*, *Performance Efficiency*, *Usability*, *Maintainability*, dan *Portability*. Berikut ini adalah penjelasan masing-masing karakteristik *ISO/IEC 25010* :

1. *Functional Suitability*

Pengujian *functional suitability* merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

2. *Reliability*

Pengujian *reliability* ini merupakan kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

3. *Performance efficiency*

Pengujian *performance* ini dilakukan untuk mengukur karakteristik performa dari komponen aplikasi. Aspek ini merupakan aspek untuk mengukur keandalan sistem informasi yang digunakan pengembangan aplikasi sistem informasi. yang sama.

4. *Usability*

Pengujian *usability* ini merupakan kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

5. *Maintainability*

Pengujian *maintainability* dilakukan untuk menguji efektifitas dan efisiensi perangkat lunak untuk dimodifikasi atau dikembangkan.

6. *Portability*

Pengujian *portability* ini merupakan kemampuan perangkat lunak untuk di transfer dari satu lingkungan ke lingkungan lain.



Gambar 2.2 Model Pengujian Perangkat Lunak ISO 25010
Sumber : (Suparto dan Dai, 2021)

Masing-masing karakteristik model *ISO 25010* dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik model *ISO 25010* dapat dilihat pada tabel 2.5:

Tabel 2.5 Karakteristik *ISO 25010*

No.	Karakteristik	Indikator	Deskripsi
1.	<i>Functional Suitability</i>	<i>Functional Completeness</i>	Kemampuan perangkat lunak dalam menyediakan fungsi dapat mencakup semua tugas dan tujuan pengguna secara spesifik.
		<i>Functional Correctness</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan.
		<i>Functional Appropriateness</i>	Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik.
2.	<i>Performance Efficiency</i>	<i>Resource Utilization</i>	Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria
		<i>Capacity</i>	Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria.
		<i>Time Behaviour</i>	Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

			menjalankan fungsinya sesuai dengan kriteria.
3.	<i>Usability</i>	<i>Appropriateness</i> <i>Recognizability</i>	Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka.
		<i>Learnability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan.
		<i>Operability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem sistem mampu memenuhi kriteria dioperasikan dan dikontrol.
		<i>User Error Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.
		<i>User Interface Aesthetics</i>	Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna.
		<i>Accessibility</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks penggunaan

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

4.	<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal.
		<i>Availability</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen dapat beroperasi ketika diperlukan untuk digunakan.
		<i>Fault Tolerance</i>	Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen.
		<i>Recoverability</i>	Kemampuan perangkat lunak ketika produk atau sistem mengalami kejadian atau kegagalan dapat mengembalikan data dan menjalankan kondisi sistem yang diharapkan.
5.	<i>Maintainability</i>	<i>Modularity</i>	Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain
		<i>Reusability</i>	Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.
		<i>Analyzability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada

			perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.
		<i>Modifiability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.
		<i>Testability</i>	Kemampuan perangkat lunak dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.
6.	<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.
		<i>Installability</i>	Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.
		<i>Replaceability</i>	Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada ketergantungan kepada perangkat lunak lain digunakan.

Sumber : (Suparto dan Dai, 2021)