

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Tinjauan pustaka yang serupa dan relevan dengan penelitian bertujuan untuk membuktikan keaslian penelitian, untuk tinjauan pustaka dapat dilihat pada tabel 2.1 dibawah ini:

**Tabel 2.1 Penelitian Terdahulu**

1.	Judul	Perancangan Sistem Pemesanan Makanan Berbasis Web menggunakan PHP dan MySQL
	Penulis, Tahun	(Kurniawan & Romzi, 2022)
	Metode	Rapid Application Development (RAD)
	Hasil Penelitian	Sistem informasi pemesanan makanan yang simple dengan menggunakan website yang berisi semua data tentang pemesanan makanan, minuman, status pesanan hingga pembayaran ke kasir.
2.	Judul	Penerapan Digital Marketing Pada E-Commerce Untuk Meningkatkan Penjualan UMKM Marning
	Penulis, Tahun	(Oktaviani et al., 2022)
	Metode	AIDA
	Hasil Penelitian	Implementasi kosep AIDA (Attention, Interest, Desire, Action) pada sistem ini yaitu menampilkan promosi yang sedang aktif ketika pelanggan pertama kali membuka sistem belanja online, penambahan detail produk, pada setiap produk yang sudah dibeli terdapat review komentar dengan rating yang diberikan konsumen serta adanya tombol checkout dengan warna yang cerah dan dominan menjadikan konsumen ingin tahu langkah selanjutnya sehingga terjadi transaksi.
3.	Judul	Sistem Informasi Manajemen Reservasi Restoran Dan Penyewaan Ruangan Berbasis Mobile (Studi Kasus : Begadang Resto)
	Penulis, Tahun	(Fathur Bahri et al., 2022)
	Metode	Extreme Programming (XP)
	Hasil Penelitian	Hasil penelitian yaitu sistem informasi manajemen reservasi restoran dan penyewaan ruangan yang

		<p>dapat diakses secara online menggunakan mobile menghasilkan aplikasi yang dapat digunakan oleh marketing dan bagian konsumen, untuk bagian mobile yang digunakan oleh konsumen memiliki fitur berupa informasi menu, ruangan, paket dan proses reservasi hingga pembayaran yang secara keseluruhan dilakukan secara online</p>
4.	Judul	E-Catalog Application For Food And Beverages At Café Ruang Seduh Based On Augmented Reality
	Penulis, Tahun	(Nurkholis et al., 2022)
	Metode	Prototype
	Hasil Penelitian	<p>Penelitian ini berhasil mengembangkan aplikasi e-katalog makanan dan minuman terintegrasi AR yang dapat membantu dalam mengelola data menu dan meningkatkan daya tarik pelanggan. Pelanggan dapat melihat bagaimana bentuk makanan/minuman yang akan disajikan secara 3D. Pelanggan juga dapat mengetahui penilaian kualitas makanan/minuman tersebut, dan pelanggan juga dapat mengetahui gambaran makanan/minuman tersebut.</p>
5.	Judul	Sistem Pemesanan Makanan Dan Minuman Berbasis Web (Studi Kasus Restoran Bumbu Desa Lampung)
	Penulis, Tahun	(Sidik, 2022)
	Metode	Waterfall
	Hasil Penelitian	<p>Hasil implementasi dari rancangan Sistem pemesanan makanan dan minuman yang sudah berbasis web digunakan untuk pramusaji/waiter dalam melakukan pelayanan untuk pelanggan pada Restoran Bumbu Desa Lampung, antara lain pemesanan makanan dan minuman, dashboard, menu makanan, pemesanan, dan user.</p>
6.	Judul	Sistem E-Marketplace Keripik Pada UMKM Keripik PU Bandar Lampung
	Penulis, Tahun	(Fuaedi, 2022)
	Metode	Waterfall
	Hasil Penelitian	<p>Sistem yang dibangun dapat mempermudah pihak ukm untuk mempromosikan hasil produknya kepada pembeli dengan luas dan dapat mempermudah pembeli untuk membeli atau melihat produk dari ukm tersebut serta ukm dapat melakukan mencetak laporan penjualan dengan mudah.</p>

7.	Judul	Implementasi Teknologi Sistem Customer Relationship Management (CRM) Guna Meningkatkan Pelayanan Penjualan
	Penulis, Tahun	(Oktarina, 2022)
	Metode	Waterfall
	Hasil Penelitian	Adanya sistem informasi membangun Customer Relationship Management pada PT Cipta Niaga Semesta dapat memudahkan sales untuk mengInputkan data penjualan yang dilakukan toko serta memudahkan dalam pencatatan data penjualan makanan dan minuman karena sistem sudah menggunakan aplikasi berbasis android yang databasenya terpusat dalam online sehingga bisa diakses oleh admin yang berada dikantor pusat.
8.	Judul	Perancangan Sistem Pemesanan Makanan Dan Minuman Berbasis Dekstop Pada Steam Cafe
	Penulis, Tahun	(Yahya et al., 2021)
	Metode	Waterfall
	Hasil Penelitian	Perancangan sistem pemesanan makanan dan minuman membantu pegawai, admin, dan pimpinan serta mempermudah pekerjaan.
9.	Judul	Perancangan Sistem Informasi Pemesanan Makanan Berbasis Web Di Foodcourt Rskia Bandung
	Penulis, Tahun	(Indriani, Saeful & Taryanto 2021)
	Metode	Prototype
	Hasil Penelitian	Sistem yang di rancang dapat mempercepat pelayanan terhadap konsumen khususnya pada saat melakukan transaksi penjualan karena sistem ini dilengkapi dengan pemilihan makanan berupa gambar dan memuat harga dari setiap makanan serta ditunjang dengan tampilan yang user friendly.
10.	Judul	Rancang Bangun Aplikasi Penjualan Dimsum Berbasis Web (Studi Kasus: Kedai Dimsum Soraya)
	Penulis, Tahun	(Soraya & Wahyudi, 2021)
	Metode	Waterfall
	Hasil Penelitian	Sistem yang dibangun mempermudah masyarakat dalam mendapatkan informasi seputar penjualan produk dimsum yang ditawarkan Kedai Dimsum Soraya secara cepat yang dapat dilakukan dimana saja dan kapan saja tanpa harus membuang waktu untuk datang langsung ke Kedai Dimsum Soraya.

11.	Judul	Pengembangan Aplikasi Online Delivery Sebagai Upaya Untuk Membantu Peningkatan Pendapatan
	Penulis, Tahun	(Ahmad et al., 2021)
	Metode	Prototype
	Hasil Penelitian	Pengembangan aplikasi yang dilakukan dapat meningkatkan pendapatan dan mempermudah dalam proses bisnis mitra. Selama lima bulan penggunaan aplikasi terdapat peningkatan transaksi sebanyak 28,13% dan peningkatan pendapatan sebanyak 21%. Kata
12	Judul	Sistem Pemesanan Makanan Di Mini Resto Aa Fried Chicken
	Penulis	(Mallisza et al., 2020)
	Tahun	2020
	Metode	RAD (Rapid Application Development)
	Hasil Penelitian	Telah dibuatnya Sistem Pemesanan Makanan Pada Mini Resto AA Fried Chicken yang dapat mengelola sistem pelayanan, menentukan nomor meja, checking pesanan pada bagian dapur dan laporan penjualan.
13.	Judul	Perancangan Aplikasi Pemesanan Makanan dan Minuman pada Rumah Makan Cepat Saji D'besto
	Penulis	(Setiawan et al., 2020)
	Tahun	2020
	Metode	Waterfall
	Hasil Penelitian	Aplikasi dapat memberikan kemudahan kasir dalam pemesanan, pengolahan data dan laporan penerimaan agar lebih tepat dan akurat.
14.	Judul	E-Delivery Makanan Berbasis Mobile (Studi Kasus : Okonomix Kedaton Bandar Lampung)
	Penulis, Tahun	(Kumala et al., 2020)
	Metode	Prototype
	Hasil Penelitian	Sistem yang dibangun mempermudah admin untuk melakukan pengelolaan data order penjualan menu makanan secara cepat dan mudah dikarenakan data otomatis tersimpan kedalam menu order admin.
15	Judul	Penerapan Algoritma A Star Untuk Penentuan Jarak Terdekat Wisata Kuliner di Kota Bandarlampung.

	Penulis	(Purnama et al., 2018)
	Metode	Waterfall
	Hasil Penelitian	Hasil implemntasi program aplikasi yang dilakukan dapat memberikan rekomendasi kuliner apa saja di sekitar pengguna kota bandarlampung dan perlu di adakan penelitian lanjut untuk memaksimalkan fungsi-fungsi aplikasi seperti menambahkan navigasi jalur ke tujuan dan sebaiknya ditambahkan beberapa pilihan jalur yang akan dipilih oleh pengguna.

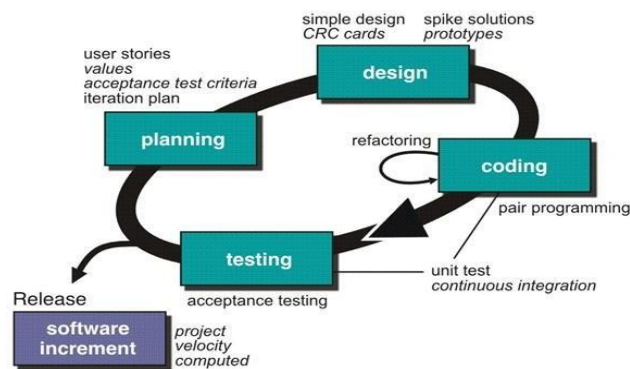
## 2.2 Digital Marketing

*Digital marketing* merupakan aktivitas perusahaan dalam mengelola kegiatan promosi dan melakukan transaksi baik barang maupun jasa melalui *internet*. *Digital marketing* mencakup semua kegiatan bisnis yang dilakukan melalui internet di seluruh dunia dengan tujuan menarik calon pealnggan baru maupun pelanggan lama untuk mempertahankan citra perusahaan (Nugraha, 2018).

*Digital marketing* memudahkan pebisnis memantau dan menyediakan segala kebutuhan dan keinginan calon konsumen, di sisi lain calon konsumen juga bisa mencari dan mendapatkan informasi produk, memesan hingga melakukan transaksi hanya dengan menggunakan media internet sehingga mempermudah konsumen dalam proses tersebut. Fasilitas-fasilitas yang terdapat pada penerapan *digital marketing* ,menurut (Sulaksono & Zakaria, 2020) yaitu ketersediaan informasi produk dan panduan produk, ketersediaan gambar-gambar seperti foto atau ilustrasi produk, ketersediaan alat transaksi, pemesanan dan variasi media pembayaran, ketersediaan tampilan testimonial, kemudahan pencarian produk, kemampuan mengidentifikasi dan menarik pelanggan baru.

### 2.3 Metode Pengembangan Sistem

*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. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing* (Pressman, 2012).



**Gambar 2.1 Model *Extreme Programming* (XP)**

Sumber : (Pressman, 2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

#### 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 memnggunakan rekayasa perangkat lunak.

## 2. *Design*

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

## 3. *Coding*

Tahap pengkodean perangkat lunak yang menggunakan tim kecil yang bekerja secara bertahap dengan panduan alur sistem yang sudah dirancang 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.

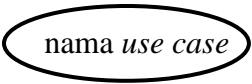


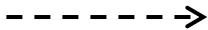
### **2.4 *Unified Modeling Language (UML)***

Perancangan sistem dalam penelitian ini menggunakan *Unified 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 & Shalahudin, 2018).


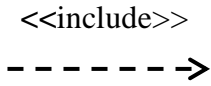
### 2.4.1 Use Case Diagram

*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat (A.S & Shalahudin, 2018).

**Tabel 2.2 Simbol Use Case Diagram**

No.	Simbol	Deskripsi
1	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2	Aktor / <i>actor</i> 	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	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan kesebuah <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





No.	Simbol	Deskripsi
		(umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i>  	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.

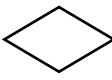


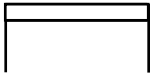
Sumber: (A.S & Shalahudin, 2018).

#### 2.4.2 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. Hal yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (A.S & Shalahudin, 2018).

**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.

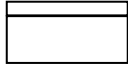
No.	Simbol	Deskripsi
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem sebuah diagram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



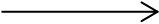
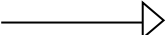
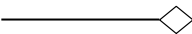
Sumber: (A.S & Shalahudin, 2018).

### 2.4.3 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 & Shalahudin, 2018)..

**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

No.	Simbol	Deskripsi
		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 dan Shalahudin, 2018).

## 2.5 Perangkat Lunak (*Software*)

*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 et al., 2018)

### 2.5.1 *Website*

*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 dihubungkan dengan jaringan-jaringan halaman (Kurniawan & Romzi, 2022).

### **2.1.1 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 et al., 2021).

### **2.1.2 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).

### **2.1.3 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).

#### **2.1.4 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 & Priandika, 2021).

#### **2.5.2 Visual Code Studio**

*Visual Studio Code (VS Code)* ini adalah sebuah *teks editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux, Mac, dan Windows*. *Teks editor* ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript, dan Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang *via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java)* (Wirawan & Nugraha, 2020).

#### **2.6 Pengujian ISO 25010**

Standar *ISO/IEC 25010* pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. Dokumen standard *ISO/IEC 25010* sangat panjang. Hal ini dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak (Suparto & Dai, 2021).

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* (Suparto dan Dai, 2021)

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

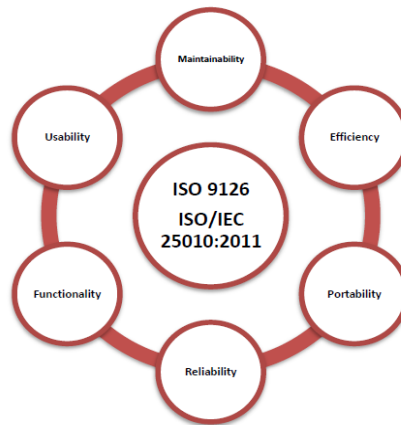
dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

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



No	Karakteristik	Indikator	Deskripsi
			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 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

No	Karakteristik	Indikator	Deskripsi
			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
4.	<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal.

No	Karakteristik	Indikator	Deskripsi
		<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

No	Karakteristik	Indikator	Deskripsi
			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

No	Karakteristik	Indikator	Deskripsi
			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)