

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	Implementasi Customer Relationship Management (CRM) pada Sistem Informasi Pemasaran dengan Menggunakan Framework React.JS Berbasis Website.
	Penulis	Fauzan, Muhammad Fitri, Iskandar dan Nuraini, Rini
	Tahun	2022
	Metode	Agile
	Hasil Penelitian	Penerapan CRM membuat hubungan antara konsumen dengan pelaku bisnis menjadi lebih dekat. Pengisian form feedback bertujuan agar dapat mengetahui tingkat kepuasan atas produk dan service dari Reswara Digital printing.
2.	Judul	Sistem Antrian Cuci Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML
	Penulis	Voutama, Apriade
	Tahun	2022
	Metode	Software Development Life Cycle (SDLC)
	Hasil Penelitian	Penerapan pemodelan UML memberikan kemudahan perancangan dalam membatu proses pengkodean menjadi sebuah Aplikasi Sistem Antrian Cuci Mobil Berbasis Website. Dengan adanya Aplikasi ini maka dapat memberikan kemudahan baik dari owner maupun kostumer dari transaksi berlangsung secara online dan terkomputerisasikan dengan baik.

3.	Judul	Sistem Informasi Pemesanan Tiket Berbasis Web Pada Pasadena Buana Travel Bandarlampung
	Penulis	Iqbal, Syauqoni Santoso, Angga Bayu
	Tahun	2022
	Metode	Waterfall
	Hasil Penelitian	Adanya bantuan sistem informasi mampu memperluas jaringan pemasaran yang terdapat di Pasadena Buana Travel Bandarlampung dan mempermudah customer dalam melakukan pemesanan serta pembayaran secara online.
4	Judul	Rancang Bangun Sistem Informasi Pemesanan Paket Wisata Berbasis Web (Studi Kasus pada Jams Tour)
	Penulis	Sutanto, Prasajo Herdy Lidwan, Nanang Ridwan, Wawan Lahat, Amas dan Atas, Mahdi
	Tahun	2021
	Metode	FAST
	Hasil Penelitian	Model sistem yang dikembangkan dapat mencegah kesalahan transaksi data pesanan, dapat memantau informasi pelanggan dan memudahkan untuk pencarian data juga promosi paket wisata yang baru.
5	Judul	Sistem Informasi Order Jasa Pariwisata (Study Kasus: Musa Tour Lampung)
	Penulis	Andre dan Prastowo, Agung Tri
	Tahun	2020
	Metode	Waterfall
	Hasil Penelitian	Sebuah sistem jasa tour pariwisata untuk meningkatkan jangkauan pemesanan perusahaan, maka Musa Tour Lampung ingin membangun sarana pemesanan berbasis web, sistem ini di fokuskan untuk mengelola data pemesanan sehingga dapat menghasilkan laporan yang dibutuhkan oleh perusahaan.

Berdasarkan tabel penelitian terdahulu diatas dapat dijelaskan kembali dibawah ini :

Fauzan, Fitri & Irsandi (2022), melakukan penelitian tentang Implementasi *Customer Relationship Management* (CRM) pada Sistem Informasi Pemasaran dengan Menggunakan Framework React.JS Berbasis Website. Permasalahan dalam penelitian ini adalah CV. Reswara Arvin Jaya belum menggunakan fasilitas sistem pemasaran berbasis web untuk meningkatkan kualitas dan kuantitas pemasaran produk. Sehingga timbul persoalan dalam memasarkan jasa dan produk kepada para konsumen yang berdomisili jauh dari lokasi. Tujuan penelitian ini adalah terciptanya sebuah fasilitas sistem informasi pemasaran dengan penerapan CRM menggunakan *framework* React.JS berbasis web. Metode yang digunakan untuk membangun sistem ini adalah metode Agile. Hasil dari penelitian ini adalah sistem pemasaran berbasis website dengan penerapan Customer Relationship Management (CRM) pada Reswara Digital printing menghasilkan penyimpanan data yang tersimpan secara terstruktur dan terperinci serta penyampaian informasi produk terhadap konsumen dapat dilangsungkan optimal.

Voutama (2022), melakukan penelitian tentang Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. Masalah yang terjadi adalah sistem penjadwalan cuci mobil yang ada masih dilakukan dengan cara melakukan pendaftaran di tempat, waktu tunggu yang tidak menentu serta pelanggan tidak bisa mengetahui banyaknya kendaraan yang antri kalau tidak pergi melihat secara langsung sehingga sistem manual tersebut perlu dibenahi. Tujuan penelitian untuk merancang sebuah sistem aplikasi Antrian cuci

mobil dengan Konsep *Customer Relationship Management* (CRM). Metode penelitian yang digunakan adalah *Software Development Life Cycle* (SDLC). Perancangan sistem menggunakan UML (*Unified Modelling Language*) yang terdiri dari *Usecase Diagram*, *Activity Diagram* dan *Class Diagram*. Hasil penelitian ini penerapan pemodelan UML memberikan kemudahan perancangan dalam membatu proses pengkodean menjadi sebuah Aplikasi Sistem Antrian Cucian Mobil Berbasis Website. Dengan adanya Aplikasi ini maka dapat memberikan kemudahan baik dari owner maupun kostumer dari transaksi berlangsung secara online dan terkomputerisasikan dengan baik..

Iqbal & Santoso (2022), melakukan penelitian tentang Sistem Informasi Pemesanan Tiket Berbasis Web Pada Pasadena Buana Travel Bandarlampung. Permasalahan dalam penelitian ini sistem pemasaran yang terjadi pada Pasadena Buana Travel yang masih memasang banner dan menyebar brosur di beberapa tempat yang ada di wilayah Bandarlampung, sehingga customer masih harus datang langsung ke kantor untuk memperoleh informasi secara lengkap maupun melakukan pemesanan tiket. Tujuan pada penelitian ini membangun sistem informasi pemesanan tiket berbasis web. Penelitian ini menggunakan metode *Waterfall*. Perancangan sistem menggunakan UML (*Unified Modelling Language*) yang terdiri dari *Usecase Diagram*, *Activity Diagram* dan *Class Diagram* dan pengujian menggunakan *Blackbox*. Hasil dari penelitian ini adanya bantuan sistem informasi mampu memperluas jaringan pemasaran yang terdapat di Pasadena Buana Travel Bandarlampung dan mempermudah customer dalam melakukan pemesanan serta pembayaran secara online.

Sutanto, dkk (2021), melakukan penelitian tentang Rancang Bangun Sistem Informasi Pemesanan Paket Wisata Berbasis Web (Studi Kasus pada Jams Tour). Permasalahan dalam penelitian ini adalah pemesanan paket wisata bagi konsumen melalui telepon atau melalui kunjungan langsung ke perusahaan, memperkenalkan paket wisata melalui brosur kertas dan spanduk, dan mencatat kegiatan administrasi menggunakan media kertas. Tujuan penelitian ini adalah membuat perancangan sistem informasi tour berbasis web sebagai solusi yang lebih efektif dan efisien. Penelitian ini dilakukan dengan pola penjabaran serta pembuatan aplikasi memanfaatkan model FAST (*Framework for the Application System Thinking*). Pengembangan sistem melalui beberapa tahapan yang mencakup analisis kebutuhan, pembuatan model berbasis UML, pemrograman berbasis WEB yaitu PHP dan database MySql, serta Pengujian Black Box dan Alpha. Hasil penelitian ini adalah model sistem yang dikembangkan dapat mencegah kesalahan transaksi data pesanan, dapat memantau informasi pelanggan dan memudahkan untuk pencarian data juga promosi paket wisata yang baru..

Andre & Prastowo (2020), melakukan penelitian tentang Sistem Informasi Order Jasa Pariwisata (Study Kasus: Musa Tour Lampung). Masalah pada penelitian ini adalah dalam mengelola data pemesanan jasa tour masih dilakukan secara manual yaitu dicatat kedalam buku besar dan akan direkap menggunakan aplikasi ms excel, dan tidak adanya informasi jika terjadi kesamaan data, serta tidak adanya laporan pemesanan secara perperiode sesuai dengan yang dibutuhkan. Tujuan penelitian ini membangun sebuah sistem jasa tour pariwisata yang diharapkan dapat meningkatkan jangkauan pemesanan berbasis web. Metode pengembangan sistem yang digunakan adalah metode waterfall, dan perancangan

sistem menggunakan perancangan sistem UML. Pengamatan, wawancara, dokumentasi dalam pengolahan data pemesanan pariwisata juga digunakan untuk menambah kedekatan keakuratan data. Hasil yang dicapai dalam tulisan ini adalah sebuah sistem jasa tour pariwisata untuk meningkatkan jangkauan pemesanan perusahaan, maka Musa Tour Lampung ingin membangun sarana pemesanan berbasis web, sistem ini di fokuskan untuk mengelola data pemesanan sehingga dapat menghasilkan laporan yang dibutuhkan oleh perusahaan.

Berdasarkan lima *literature* yang telah dipaparkan, maka penelitian ini memiliki perbedaan dari penelitian-penelitian terdahulu, diantaranya:

1. Pada sistem yang dikembangkan terdapat fitur yang dapat digunakan pelanggan untuk mengetahui jenis armada atau transportasi serta jumlah kursi yang masih tersedia, berbeda dengan penelitian sebelumnya yang belum ada fitur tersebut.
2. Pada penelitian sebelumnya belum ada yang menggunakan metode penelitian *Extreme Programming* (XP) dalam proses pengembangan sistemnya.

2.2 Pengertian Sistem

Sistem sebagai urutan operasi klerikal, biasanya melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi. Adapun pendekatan yang lebih menekankan pada elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Hakim, Sukron dan Awaludin, 2019). Sistem mempunyai beberapa karakteristik menurut Samsudin, Abdurahman dan Abdullah (2019) sebagai berikut:

1. Batasan (*Boundary*)

Batas sistem merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya. Batas sistem ini bagi umat manusia sangat relatif dan tergantung kepada tingkat pengetahuan dan situasi kondisi yang dirasakan oleh orang yang melihat sistem tersebut.

2. Lingkungan (*Environment*)

Bentuk apapun yang ada diluar rung lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

3. Masukan (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal. Contoh, di dalam suatu unit sisten komputer, “program” adalah *maintenance* input yang digunakan untuk mengoperasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

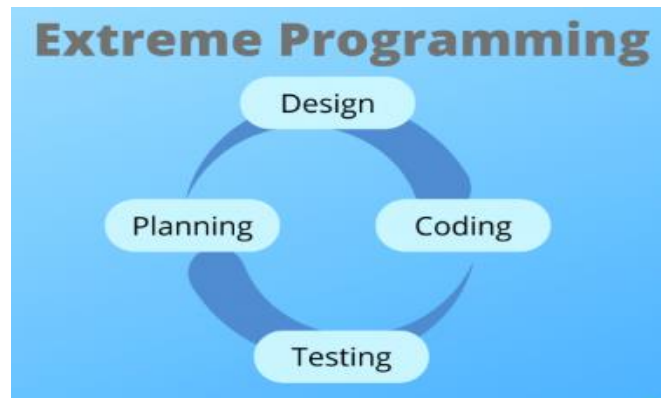
4. Keluaran (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan utau hal-hal yang lain yang menjadi input bagi sub sistem lain

2.3 Metode Pengembangan Sistem

2.3.1 *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). Pendekatan yang digunakan dalam *Extreme Programming* adalah *object-oriented* sebagai paradigma pengembangan dan mencakup seperangkat aturan.



Gambar 2.1 Fase *ExtremeProgramming*
Sumber : (Pressman, 2012)

2.3.2 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 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)*

Unified Modeling Language (UML) digunakan untuk memvisualisasikan, menentukan, membangun, dan membuat dokumen artefak dari sebuah sistem *software* yang intensif (Pressman, 2012). UML merupakan suatu kumpulan teknik dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Terdapat 3 kategori dalam UML yaitu :

1. *Structure Diagram*

Structur diagram merupakan sekumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behaviour Diagram*

Behaviour diagram adalah kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem.


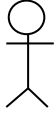

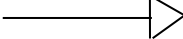


3. *Interaction Diagram*

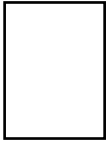


Interaction diagram adalah kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain.

2.4.1 *Usecase Diagram*

Usecase Diagram menggambarkan bagaimana *user* berinteraksi dengan sistem dengan cara mendefinisikan langkah-langkah yang dibutuhkan untuk menyelesaikan tujuan tertentu (Pressman, 2012). Sebuah format yang mudah untuk membuat sebuah *use case* adalah dengan menjelaskan skenario utamanya sebagai sebuah urutan langkah-langkah dan alternatif langkah-langkah sebagai variasi dari urutan tersebut. *Usecase* juga dapat digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak atas perintah-perintah yang ada pada sistem tersebut. *Usecase* harus dengan penamaan yang singkat dan mudah dimengerti. Simbol-Simbol *Usecase* diagram dapat dilihat di tabel 2.2 berikut.

Tabel 2.2 Simbol-Simbol *Usecase Diagram*

No.	Simbol	Keterangan
1.	<i>Use Case</i> 	<i>Use Case</i> merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya menggunakan kata kerja.
2.	Aktor / <i>actor</i> 	Aktor merupakan seseorang/sesuatu yang berinteraksi dengan yang akan dibuat diluar sistem informasi.
3.	Asosiasi / <i>association</i> 	Asosiasi/ <i>associatio</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> 	Generalisasi (<i>generalization</i>) merupakan <i>hupromosin</i> (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.	<i>Include</i> << <i>include</i> >> 	Include berarti <i>Use Case</i> yang ditambahkan akan dipanggil saat <i>Use Case</i> tambahan dijalankan.
6.	Ekstensi / <i>extend</i> << <i>extend</i> >> 	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.

No.	Simbol	Keterangan
7.	<i>System</i> 	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.	<i>Collaboration</i> 	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah elemen-elemennya (sinergi).
9.	<i>Note</i> 	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.


Sumber: (Pressman, 2012)

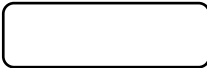

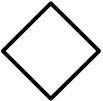


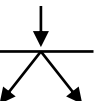

2.4.2 Activity Diagram

Activity Diagram menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem melalui aliran kontrol antara tindakan bahwa sistem melakukan kegiatan tersebut. Hal ini mirip dengan *flowchart* kecuali bahwa suatu diagram aktivitas dapat menunjukkan aliran secara bersamaan tertentu (Pressman, 2012).

Simbol-simbol *activity diagram* dapat dilihat pada tabel 2.3:

Tabel 2.3 Simbol Activity Diagram

No	Notasi	Simbol	Keterangan
1.	Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

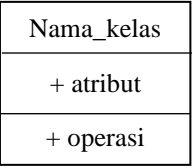


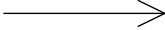
No	Notasi	Simbol	Keterangan
2.	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir.
4.	Percabangan / <i>Decision</i>		Asosiasi percabangan Dimana jika ada pilihan aktifitas lebih dari satu
5.	Penggabungan / <i>Join</i>		Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu.
6.	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi .
7.	<i>Fork</i>		Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
8.	<i>Join</i>		Digunakan untuk menunjukkan kegiatan yang digabungkan.



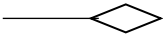
Sumber: (Pressman, 2012)

2.4.3 Class Diagram

Class diagram merupakan bangunan utama dalam pemodelan berorientasi objek. Diagram ini menggambarkan sebuah pandangan dari satu aspek tertentu dari model atau keseluruhan, menggambarkan struktur elemen beserta hubungan mereka. *Class Diagram* terutama digunakan untuk membangun sebuah arsitektur sistem dengan menangkap dan mendefinisikan *class-class* dan *interface* dan hubungan antara mereka. Sebuah *class diagram* menggambarkan hubungan antar kelas daripada hubungan antar objek (Pressman, 2012).

Tabel 2.4 Simbol Class Diagram

No	Notasi	Simbol	Keterangan
1.	Kelas		Kelas pada struktur sistem
2.	Antar muka / <i>Interface</i>		Sama dengan konsep interface dalam pemograman berorientasi objek
3.	Asosiasi / <i>Association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Isosiasi berarah / <i>directed association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang

No	Notasi	Simbol	Keterangan
			lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Kebergantungan <i>/ dependency</i>		Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agresi / <i>aggregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber : (Pressman, 2012)

2.5 Sistem Pemesanan Online

Sistem pemesanan onlinen atau sering disebut aplikasi *E-Booking* adalah aplikasi pemesanan secara elektronik yang digunakan dalam proses pemesanan suatu jasa atau produk memanfaatkan jasa internet dalam pengaplikasiannya. Aplikasi ini dapat digunakan pada perusahaan jasa penyewaan dan penjualan produk. Penggunaan aplikasi *E-Booking*, pelanggan cukup membuka situs web kemudian memesan layanan sesuai dengan tanggal yang mereka inginkan, mengisi formulir, dan melakukan *online payment*. Sistem juga dapat memperbarui data

ataupun informasi secara real time sehingga pelanggan tidak perlu menghubungi secara manual untuk menanyakan ketersediaan layanan untuk tanggal yang mereka pesan (Kurnia, 2020)

2.6 Website

Website adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan- jaringan halaman (*hyperlink*) (Ismatullah dan Adrian, 2021).

2.7 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. *MySQL* adalah nama sebuah *database server* yang menangani akses *database* yang selalu dalam bentuk pernyataan *SQL (Structured Query Language)* yaitu suatu bahasa yang digunakan untuk mengakses *database relational* (Ardian & Fernando, 2020).

2.8 XAMPP

XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan XAMPP kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah

disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah *Apache*, *MySQL*, *Php*, *Filezila* dan *Phpmyadmin* (Putera dan Ibrahim, 2018).

2.9 PHP

PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan *Software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya. PHP ditulis menggunakan bahasa C (Hutagalung dan Arif, 2018).

2.10 CPanel

Cpanel adalah salah satu *control panel* berbasis *linux* yang paling banyak digunakan di akun *web hosting*. Melalui panel ini, bisa dengan mudah mengelola semua layanan dalam satu tempat. Saat ini, *cpanel* merupakan panel standar yang paling banyak digunakan dan sebagian besar *web developer* sudah sangat familiar dengan tool ini. Selain intuitif dan mudah digunakan, *cpanel* juga memungkinkan untuk mengelola akun *web hosting* dengan maksimal. Membuat *FTP user* dan alamat *email* baru, memonitor *resource*, membuat *subdomain*, dan *install software* merupakan beberapa kegunaan *cpanel* (Lidar, 2020).

2.11 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).

2.12 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*. Berikut ini penjelasan 6 karakteristik ISO 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 & 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

No.	Karakteristik	Indikator	Deskripsi
			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</i> <i>Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.
		<i>User Interface</i> <i>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

No.	Karakteristik	Indikator	Deskripsi
			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.
		<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.

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

No.	Karakteristik	Indikator	Deskripsi
		<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 & Dai, 2021)