

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Adapun perbedaan penelitian dengan penelitian terdahulu dapat dilihat pada

Tabel 2.1:

Tabel 2. 1 Tinjauan Pusataka

No. Literatur	Literatur 01
Penulis	(Syarif and Boksman, 2019)
Judul	Pengembangan Layanan Umpan Balik pada Fakultas Teknologi Informasi Universitas Atma Jaya Makassar
Metode	<i>Waterfall</i>
Hasil	Sistem umpan balik berbasis mobile yang dikembangkan telah dapat diisi oleh mahasiswa dan orang tua dan hasil dari umpan balik tersebut sudah dapat dilihat dan digunakan oleh dosen dan ketua prodi di FTI UAJM untuk mengevaluasi proses perkuliahan yang sedang berjalan
No. Literatur	Literatur 02
Penulis	(Nurul Permatasari et al., 2020)
Judul	Kajian Tentang Pengaruh Guest Comment Terhadap Perubahan Kebijakan Banquet Manager di Hotel Grand Keisha Yogyakarta
Metode	Kualitatif
Hasil	Pengaruh jumlah guest comment terhadap perubahan kebijakan manager dengan mempunyai target mendapatkan guest comment tiap harinya dan jumlah guest comment di dapatkan berdasarkan event yang berlangsung
No. Literatur	Literatur 03
Penulis	(Brata, 2021)
Judul	Analisis Dan Perancangan Sistem “Study Literasi Merancang Masukan Yang Efektif (Designing)
Metode	SDLC
Hasil	Sistem terkomputerisasi ini diharapkan dapat memudahkan kita dalam segala aspek kehidupan. Dengan kemajuan teknologi saat ini, sekarang semua orang berusaha untuk membuat suatu sistem

	komputerisasi yang baik dan mudah digunakan untuk membantu pekerjaan manusia. Analisis
No. Literatur	Literatur 04
Penulis	(Chandra et al., 2022)
Judul	Rancang Bangun Aplikasi Notifikasi Penjadwalan Dan Umpan Balik Serta Sertifikat Webinar Menggunakan Microsoft Power Automate
Metode	<i>Waterfall</i>
Hasil	Sistem ini dapat mengotomatisasi proses registrasi dan feedback dengan Microsoft Power Automate guna memberikan aktivitas link dan menghasilkan sertifikat para peserta dalam kegiatan yang dikirimkan melalui E-mail. Kesimpulan ini dapat dicapai berdasarkan pembahasan yang telah diuraikan. Baik
No. Literatur	Literatur 05
Penulis	(Supriyanto and Fitria, 2022)
Judul	Penyuluhan Dan Sosialisasi Sistem Layanan "Portal Emas" Pada Mahasiswa Universitas Panca Marga
Metode	SDLC
Hasil	Hasil dari kegiatan pengabdian ini adalah mahasiswa mendapatkan ilmu dan pemahaman terkait materi, pelayanan sektor publik dan metode penggunaan aplikasi Portal Emas. Output berbentuk e-book yang dapat dibaca oleh mitra dan masyarakat.

2.2. Rancang Bangun

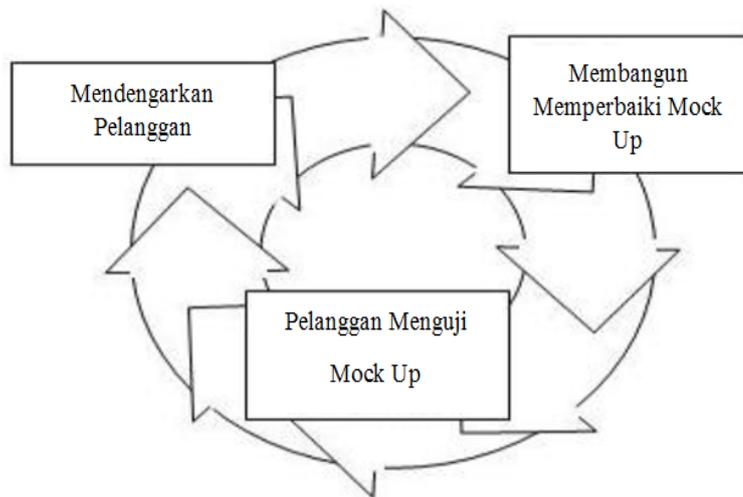
Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan (Solikin et al., 2018). Menurut (Mulyati and Hisyam, 2018) Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

2.3. Sistem Informasi

Menurut yang dikutip dari jurnal (Sallaby and Kanedi, 2020) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan–laporan yang diperlukan oleh pihak luar tertentu. Sistem Informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang, fasilitas, teknologi, media, prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian internal dan eksternal yang disini penulis menyimpulkan jika Sistem Informasi adalah suatu sistem yang yang terorganisir yang berfungsi untuk mengolah informasi yang bermanfaat yang mempunyai tujuan tertentu dan informasi yang akan disampaikan dapat diterima dengan baik oleh penerima sehingga tujuan tersebut dapat tercapai.

2.4. Metode Pengembangan *Prototype*

Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Shalahuddin and Rosa, 2018).



Gambar 2. 1 Ilustrasi Model Prototipe.

Menurut (Shalahuddin and Rosa, 2018) terdapat tahapan dalam proses prototipe yaitu:

1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Merancang dan Membuat Prototipe

Pada tahap ini, dilakukan perancangan dan pembuatan prototipe sistem yang disesuaikan dengan kebutuhan pengguna.

3. Uji Coba

Pada tahap ini, dilakukan pengujian prototipe sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan prototipe, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih belum sesuai kembali ke tahap pertama.

2.5. Perancangan Sistem *Unified Modeling Language* (UML)

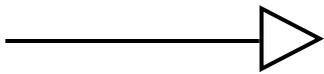
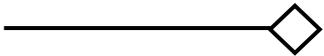
Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Shalahuddin and Rosa, 2018).

1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2 di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>asociation</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 yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>

No	Simbol	Deskripsi
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

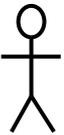
(Shalahuddin and Rosa, 2018)

2. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Shalahuddin and Rosa, 2018). simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.3 di bawah ini:

Tabel 2. 3 Simbol diagram *use case*

No	Simbol	Deskripsi
1.	<i>Use Case</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>

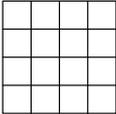
No	Simbol	Deskripsi
2.	Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
3.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
4.	Ekstensi/ <i>extend</i> <i><<extend>></i> 	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 mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
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.	Menggunakan/ <i>Include/uses</i> <i><<include>></i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

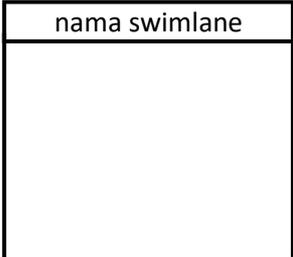
(Shalahuddin and Rosa, 2018)

3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem *bukan* apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 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.	Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
6.	Dokumen 	Menunjukkan dokumen sumber atau laporan

No	Simbol	Deskripsi
7.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
8.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Shalahuddin and Rosa, 2018)

2.6. MySQL

Menurut (Cahyono and Jayanti, 2022) *MySQL* merupakan *Database* yang menghubungkan *scriptphp* menggunakan perintah *query* dan *escaps character* yang sama dengan *php*. *MySQL* mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengizinkan proses yang bisa anda lakukan. Kelebihan *MySQL* adalah sebagai berikut:

1. Gratis karena bersifat *open source* dibawah licensi *GPL*
2. Dapat berjalan dalam berbagai *platformOS* seperti *Linux*, *Windows*, *MacOS* dan lain-lain
3. Mendukung penggunaan *multi-user*, yang artinya beberapa user dapat menggunakan secara bersamaan.
4. Kecepatan yang sangat baik ketika menangani kode-kode perintah *SQL*.
5. Mendukung sepenuhnya standar bahasa *SQL* untuk *database*.
6. Mendukung banyak tipe variabel seperti *integer*, *float*, *double*, *char*, *text*, *date* dan *datetime*.

7. Mendukung penggunaan fungsi dan operator dalam perintah *SQL*.
8. Memiliki keamanan yang berlapis seperti *namahost*, *subnetmask*, ijin dan akses user serta penggunaan kata sandi ter-enkripsi.
9. Dapat menangani basis data dalam skala besar lebih dari 50juta *record* dan 60ribu tabel serta 5 miliar baris.

Sedangkan *phpMyAdmin* adalah sebuah perangkat lunak untuk mengelola *databaseMySQL* dengan antarmuka grafis (GUI). Penggunaan antarmuka grafis lebih memudahkan daripada antarmuka pengelolaan asli *MySQL* yang berbasis text. Jadi yang pada *MySQL* semuanya dilakukan dengan *text*, pada *phpMyAdmin* dibuat dalam model grafis. *PhpMyAdmin* dibuat dengan kode program *PHP* dan berjalan pada antarmuka *web*. Boleh dikatakan kalau *phpMyAdmin* adalah sebuah aplikasi grafis berbasis *web* yang dipakai untuk pengelolaan *databaseMySQL*. *phpMyAdmin* sendiri sepertinya berasal dari kata *php*, *MySQL* dan *Administrator*.

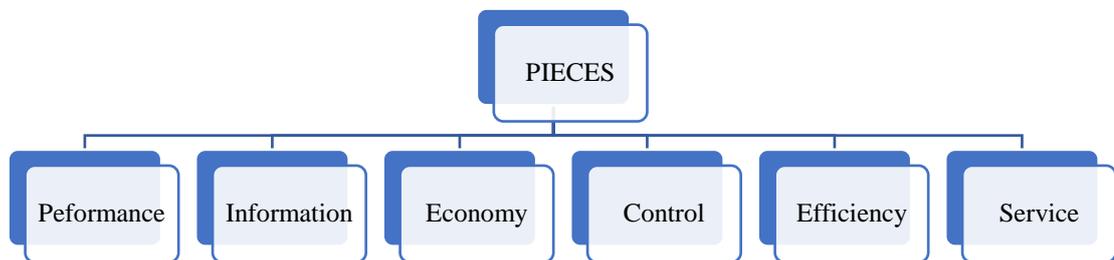
Kelebihan *phpMyAdmin* adalah sebagai berikut :

1. Memiliki antarmuka grafis berbasis *web* sehingga bisa diakses secara luas.
2. Mendukung banyak fitur *MySQL*
3. Bisa *import* data dari *SQL* dan *CSV*
4. Menyediakan *ekspor* data dalam berbagai format seperti *SQL*, *XLS*, *CSV*, *XML*, *PDF* dan *Text*.
5. Terdapat monitoring proses *server* seperti koneksi, prosesor, memory dan sebagainya.

2.7. Analisis PIECES

Menurut dari jurnal (Muliansah and Budihartanti, 2020) metode PIECES merupakan suatu teknik analisis yang digunakan sebagai landasan untuk

mengidentifikasi aspek-aspek permasalahan yang lebih rinci. Ketika kita menganalisis suatu sistem, kita umumnya melakukan analisis terhadap beberapa faktor, seperti kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Proses analisis PIECES ini sangat penting sebelum kita mulai mengembangkan sistem informasi, karena melalui analisis ini kita dapat mengidentifikasi masalah inti serta masalah yang mungkin merupakan gejala dari masalah inti tersebut.



Gambar 2. 2 Diagram PIECES

1. *Performance*

Adalah kemampuan menyelesaikan tugas bisnis dengan cepat sehingga sasaran segera tercapai. Kinerja diukur dengan jumlah produksi dan waktu tanggap dari suatu sistem.

2. *Information*

Laporan-laporan yang sudah selesai diproses digunakan untuk menggunakan hasil informasi yang dibutuhkan oleh manajemen di dalam pengambilan keputusan.

3. *Economy*

Adalah penilain sistem dalam pengurangan dan keuntungan yang akan didapatkan dari sistem yang dikembangkan. Sistem ini akan memberikan penghematan operasional dan meningkatkan keuntungan perusahaan.

4. *Security*

Sistem keamanan yang digunakan harus dapat mengamankan data dari kerusakan, misalnya dengan membuat *back up* data. Selain itu sistem keamanan juga harus dapat mengamankan data dari akses yang tidak diijinkan, biasanya dilakukan dengan password terutama pada form aplikasi dan database-nya.

5. *Efficiency*

Berhubungan dengan sumber daya yang ada guna meminimalkan pemborosan. Efisiensi dari sistem yang dikembangkan adalah pemakaian secara maksimal atas sumber daya yang tersedia yang meliputi manusia, informasi, waktu, uang, peralatan, ruang, dan keterlambatan pengolahan data.

6. *Service*

Adalah suatu pelayanan yang lebih baik. Peningkatan pelayanan terhadap sistem yang sedang dikembangkan akan memberikan:

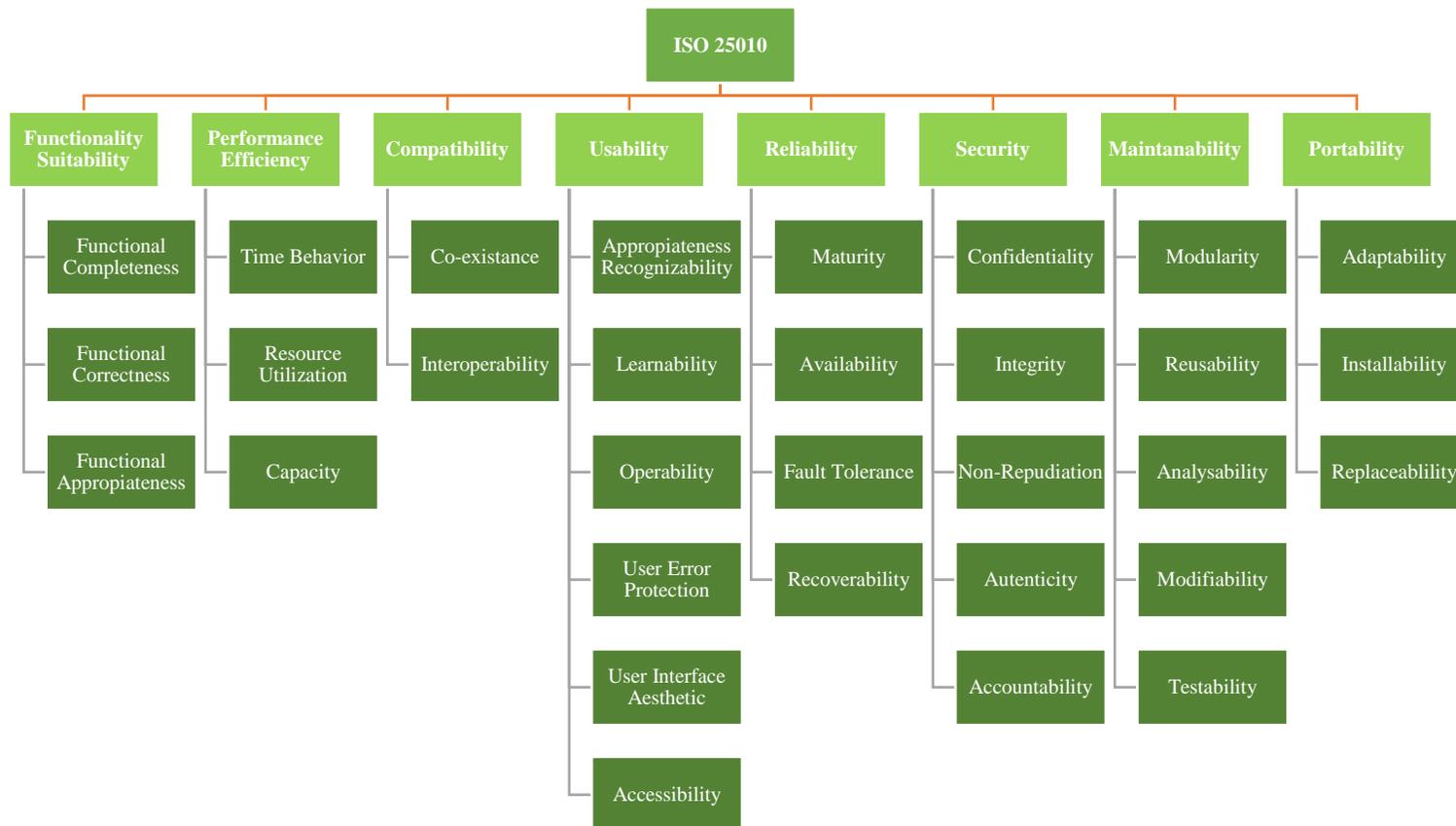
- a. Akurasi dalam pengolahan data
- b. Keandalan terhadap konsistensi dalam pengolahan input dan outputnya serta keandalan dalam menangani pengecualian.
- c. Sistem mudah pakai
- d. Kemampuan menangani masalah yang di luar kondisi normal.
- e. Mampu mengkoordinasi aktifitas untuk mencapai tujuan dan sasaran.

2.8. *Framework Laravel*

Laravel adalah framework bahasa pemrograman *Hypertext Preprocessor* (PHP) yang ditujukan untuk pengembangan aplikasi berbasis web dengan menerapkan konsep Model *View Controller* (MVC). Framework ini dibuat oleh Taylor Otwell dan pertama kali dirilis pada tanggal 9 Juni 2011. Laravel berlisensi *opensource* yang artinya bebas digunakan tanpa harus melakukan pembayaran. Laravel bertujuan untuk membuat proses pengembangan menyenangkan bagi pengembang tanpa mengorbankan fungsionalitas aplikasi. Laravel mudah diakses, namun kuat, menyediakan alat canggih yang dibutuhkan untuk aplikasi besar dan tangguh. Alamat website resmi dari framework Laravel adalah <https://laravel.com>. Fitur-fitur modern Laravel yang sangat membantu developer dalam membuat aplikasi adalah *Bundles, Eloquent ORM (Object-Relational Mapping), Query Builder, Application Logic, Reverse Routing, Resource Controller, Class Auto Loading, View Composers, Blade, IoC Containers, Migration, Database Seeding, Unit Testing, Automatic Pagination, Form request*, dan *Middleware* (Adieb, 2021). Framework Laravel juga memiliki beberapa keunggulan sebagai berikut.

1. Menggunakan Command Line Interface (CLI) Artisan.
2. Menggunakan package manager PHP Composer.
3. Penulisan kode program lebih singkat, mudah dimengerti, dan ekspresif.

2.9. ISO 25010



Gambar 2. 3 Model kualitas produk ISO/IEC 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability, reliability, operability, performance efficiency, security, compatibility, maintainability* dan *transferability*. *Quality in use* dan *product quality*. Sedangkan pada *dimensionality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use, Flexibility in use, dan Safety* (Riswanda and Priandika, 2021). Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika

dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.

- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.
- 4) *Operability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna, perangkat lunak dapat memberikan bantuan ketika pengguna membutuhkan panduan, perangkat lunak dapat menarik perhatian pengguna, perangkat lunak memenuhi kebutuhan pengguna yang memiliki keterbatasan dan perangkat memungkinkan untuk dianalisis oleh pengguna apakah perangkat lunak sudah memenuhi kebutuhan mereka.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.

- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan , kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.
- 8) *Portability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.10. Skala Pengukuran

Skala pengukuran adalah kesepakatan yang berfungsi sebagai acuan untuk menentukan panjang pendeknya interval pada alat ukur. Hal ini memastikan bahwa alat ukur dapat menghasilkan data kuantitatif yang akurat. Dalam penelitian ini, penulis menggunakan skala pengukuran dengan metode skala Likert. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi individu atau kelompok terhadap fenomena sosial. Fenomena sosial ini ditentukan secara spesifik oleh peneliti dan disebut sebagai variabel penelitian. Dengan menggunakan skala Likert, peneliti dapat mengumpulkan data yang menggambarkan sejauh mana responden

setuju atau tidak setuju dengan berbagai pernyataan terkait variabel penelitian. Skala ini membantu dalam mengkonversi sikap subyektif menjadi data numerik yang bisa dianalisis secara statistik. Oleh karena itu, penggunaan skala Likert sangat efektif dalam penelitian yang bertujuan untuk memahami pandangan atau sikap masyarakat terhadap isu-isu sosial yang telah ditetapkan oleh peneliti. Dengan demikian, alat ukur ini memungkinkan pengumpulan data yang konsisten dan valid, sehingga dapat mendukung kesimpulan yang dapat diandalkan dalam penelitian sosial (Sugiyono, 2022). yang akan dijelaskan sebagai berikut:

- a. Pertanyaan positif
 - a) Skor 5 untuk Sangat (Setuju/Baik)
 - b) Skor 4 untuk (Setuju/Baik)
 - c) Skor 3 untuk (Netral/Cukup)
 - d) Skor 2 untuk Tidak (Setuju/Baik)
 - e) Skor 1 untuk Sangat Tidak (Setuju/Baik)
- b. Pertanyaan negative
 - a. Skor 5 untuk Sangat Tidak (Setuju/Baik)
 - b. Skor 4 untuk Tidak (Setuju/Baik)
 - c. Skor 3 untuk (Netral/Cukup)
 - d. Skor 2 untuk (Setuju/Baik)
 - e. Skor 1 untuk Sangat (Setuju/Baik)

Total skor didapat dari masing-masing dari kategori penilaian tersebut dan akan dihitung dengan rumus untuk mendapatkan persentase. Terdapat kategori persentase yang menunjukkan sebagai indikator dari hasil persentase untuk sebuah pertanyaan yang diberikan. Berikut adalah indikator hasil persentase:

- a. Untuk persentase 0% - 19,999% maka keteranganya Sangat Tidak (Setuju/Baik)
- b. Untuk persentase 20% - 39,999% maka keteranganya Tidak (Setuju/Baik)
- c. Untuk persentase 40% - 59,999% maka keteranganya (Cukup/Netral)
- d. Untuk persentase 60% - 79,999% maka keteranganya (Setuju/Baik)

Untuk persentase 80% - 100% maka keteranganya Sangat Tidak (Setuju/Baik)