

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian yang pernah dilakukan terkait dengan penelitian yang dilakukan, dapat dilihat pada tabel 2.1

Tabel 2. 1 Tinjauan Pusataka

No	Judul	Nama (Tahun)	Metode	Deskripsi
1	Analisis tingkat kepuasan konsumen dengan menggunakan servqual (service quality) terhadap pelayanan jasa peminjaman dana	(Padhil <i>et al.</i> , 2023)	SERVQUAL	Tingkat kepuasan konsumen terhadap kualitas dari pelayanan PT. Finansia Multi Finance (Kredit Plus) nilai rata-rata tingkat kepuasan sebesar 94% dari harapan konsumen. Hasil pengukuran persentase tingkat kepuasan konsumen pada masing- masing dimensi yaitu dimensi empati sebesar 95,76%, dimensi daya tanggap 94,82%, dimensi keandalan 94,38%, dimensi bukti fisik 93,45%, dan persentase terendah pada dimensi jaminan dengan nilai sebesar 91,27%.
2	Analisis Tingkat Kepuasan Pelanggan Terhadap Kinerja Pelayanan Penjualan Dengan Menggunakan Metode Service Quality	(Wijaya <i>et al.</i> , 2023)	SERVQUAL	Hasil dari peneliti ini bisa disimpulkan pelayanan yang selama ini diberikan di PT.Pacific Indah Pratama terhadap pelanggan bisa dinilai baik, kemudian hasil tersebut bisa dijadikan acuan untuk meningkatkan proses ke tahap yang lebih baik lagi di PT.Pacific Indah Pratama.

No	Judul	Nama (Tahun)	Metode	Deskripsi
				Ciptaan
3	Analisis Kualitas Pelayanan Dengan Metode Servqual Guna Meningkatkan Kepuasan Pelanggan Pada Pelayanan Jasa Transportasi Terminal Maulana Malik Ibrahim	(Bachtiar, Ismiyah and Rizqi, 2022)	SERVQUAL	Hasil penelitian memperlihatkan bahwasanya kualitas pelayanan selalu bernilai negatif yang berarti kurangnya pelayanan yang diberikan oleh mitra. Pada servqual deviasi terletak pada atribut-atribut yaitu ketepatan waktu operasional dengan nilai gap -0,21, ketangkasan dalam menanggapi keluhan pelanggan dengan nilai gap -1 ,36, kendaraan dengan nilai gap -2,14, kepedulian terhadap pelanggan dengan nilai gap - 1,04, pemahaman kebutuhan pelanggan oleh karyawan dengan nilai gap - 2,24. Oleh karena itu, mitra perlu mengetahui bagaimana mengembangkan dan meningkatkan kualitas layanannya menjadi lebih baik
4	Kualitas Layanan X Dengan Metode Servqual Dan Analytical Hierarchy Process	(Eka <i>et al.</i> , 2020)	SERVQUAL	Hasil penelitian ini diketahui tingkat kepuasan pelanggan terhadap kualitas layanan X sudah puas dimata pelanggan dilihat dari nilai kepuasan pelanggan sebesar 4,025 artinya pelanggan sudah puas dengan pelayanan saat ini.
5	Dalam Membentuk Kepuasan Pelanggan	(Sinollah and Masruro, 2019)	SERVQUAL	Hasil uji analisis servqual menunjukkan bahwa kualitas pelayanan di

No	Judul	Nama (Tahun)	Metode	Deskripsi
	Sehingga tercipta Loyalitas Pelanggan (Studi Kasus pada Toko Mayang Collection cabang Kepanjen)			toko Mayang Collection cukup baik. Hal ini berarti kepuasan pelanggan dari segi kualitas pelayanan yang diberikan di Mayang Collection cukup baik

Berdasarkan penelitian terdahulu terdapat beberapa perbedaan penelitian

dengan peneliti yaitu :

1. Sistem dapat mencetak laporan hasil perhitungan
2. Sistem dapat dijalankan secara *online* sehingga pelanggan dapat melakukan penilaian dimana saja dan kapan saja.
3. Penelitian ini menggunakan metode pengembangan sistem *extreme programming*
4. Pengujian sistem menggunakan ISO 25010 dengan aspek *functionalit, usability, dan performance efficiency*

2.2. Kepuasan Pelanggan

Kepuasan Pelanggan merupakan suatu tingkatan dimana kebutuhan, keinginan dan harapan dari pelanggan akan dapat terpenuhi atau terlampaui melalui suatu transaksi yang akan mengakibatkan terjadinya pembelian ulang atau kesetiaan yang terus berlanjut. Kepuasan pelanggan sebagai respon pelanggan terhadap ketidaksesuaian antara tingkat kepentingan sebelumnya dan kinerja actual yang dirasakannya setelah pemakaian (Wijaya, 2017).

Pada dasarnya pengertian kepuasan pelanggan mencakup perbedaan antara tingkat kepentingan dengan kinerja atau hasil yang dirasakan. Tentang jaminan pelayanan yang dapat memberikan manfaat dalam hal sebagai berikut:

- a. Fokus pada pelanggan. Perusahaan mengidentifikasi kebutuhan pelanggan dan memulihkannya ketika terjadi penyimpangan, untuk itu selalu ada survei untuk mengetahui bagaimana kebutuhan yang telah diidentifikasi dapat terus-menerus sampai ke pelanggan.
- b. Menyusun standar yang jelas. Kebutuhan yang spesifik membuat standar yang jelas dan tanggungjawab dari penyedia.
- c. Menjamin umpan balik. Pelanggan yang tidak puas akan merasa ada insentif untuk menyampaikannya kepada perusahaan karena perusahaan peduli dengan mereka.
- d. Membangun loyalitas pelanggan. Jaminan akan mengurangi risiko, membuat harapan secara eksplisit, dan membangun pangsa pasar dengan menahan pelanggan yang tidak puas yang akan meninggalkan perusahaan.

2.3. SERVQUAL

Menurut Tjiptono and Chandra (2016) mengartikan jika kualitas jasa sebagai tolak ukur bagusnya pelayanan yang diberikan sesuai dengan keinginan. Kualitas tersebut bisa direalisasikan dengan pemenuhan apa yang dibutuhkan pemakai jasa. Menurut Tjiptono and Chandra (2016) mengidentifikasikan 5 GAP kesenjangan pada kualitas pelayanan, hal tersebut adalah:

1. Kesenjangan diantara apa yang diharapkan dengan persepsi manajemen (*Knowledge Gap*)
2. Kesenjangan antara anggapan manajemen dengan yang dipikirkan oleh pemakai jasa yang menyangkut mutunya (*Standard Gap*)
3. Kesenjangan mutu dengan yang disampaikan jasa (*Delivery Gap*)

4. Kesenjangan antara yang disampaikan dengan relasi eksternal
(*Communication Gap*)

Menurut Parasuraman, Zeithamal dan Berry (2016) menyebutkan dimensi serta atribut model SERVQUAL antara lain :

1. *Tangible* (Bukti Fisik), terkait dengan ketertarikan dari pelayanan ataupun kelengkapan dari sebuah organisasi, serta tampilan dari pekerjanya.
2. *Emphaty* (Empati), artinya jika sebuah organisasi paham dengan permasalahan yang dirasakan oleh pemakai jasa, kemudian memberikan pelayanan yang berhubungan dengan kepentingan pemakai jasa, serta pelayanan agar mereka merasa nyaman.
3. *Responsiveness* (Daya Tanggap), diartikan sebagai sikap mampu pekerja untuk membantu pemakai jasa dari yang mereka inginkan, dengan estimasi waktu yang cepat.
4. *Reliability* (Reliabilitas), terkait dengan mampunya organisasi pada pemberian layanan yang baik tanpa adanya kesalahan dan sesuai dengan kesepakatan awal.
5. *Assurance*

Dimensi berikutnya adalah *assurance*. *Assurance* berkaitan dengan kepastian, tepatnya kepastian yang didapatkan pelanggan dari perilaku pelaku usaha. *Assurance* ini bisa didapat, misalnya, dari komunikasi yang baik, pengetahuan yang luas, hingga sikap sopan dan santun kepada pelanggan.

Pengukuran kualitas jasa dalam model SERVQUAL didasarkan pada skala multi-item yang dirancang untuk mengukur harapan dan persepsi pelanggan, serta gap di antara keduanya pada lima dimensi kualitas jasa. Skor SERVQUAL untuk

setiap pasang pernyataan, bagi masing-masing pelanggan dapat dihitung berdasarkan rumus berikut:

$$\mathbf{SKOR\ SERVQUAL = SKOR\ PERSEPSI - SKOR\ HARAPAN.....(6)}$$

Dalam suatu prinsip, terdapat data yang diperoleh melalui instrumen SERVQUAL dapat dipergunakan untuk menghitung skor gap kualitas jasa pada berbagai level secara rinci (Tjiptono and Chandra, 2016).

1. *Item by item analysis*, P1 – H1, P2 – H2, dan seterusnya.
2. *Dimension by dimension analysis*, dengan contoh:

$$\left(\frac{P1+P2+P3+P4+P5}{5} - \frac{H1+H2+H3+H4+H5}{5} \right) \text{dimana P1 sampai P5 dan H1 sampai}$$

H5 mencerminkan lima pernyataan persepsi dan harapan dalam satu dimensi tertentu, dan perhitungan dilakukan untuk masing-masing dimensi selanjutnya.

3. Perhitungan ukuran tunggal kualitas jasa atau gap SERVQUAL

$$\left(\frac{P1+P2+P3+P4+P5}{5} - \frac{H1+H2+H3+H4+H5}{5} \right) \text{dimana P1 sampai PK dan H1 sampai}$$

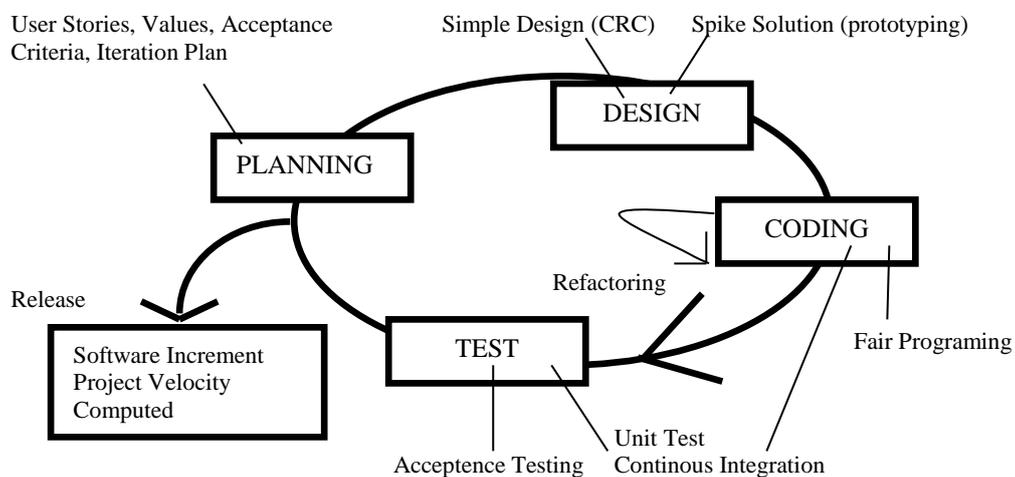
HK mencerminkan seluruh pernyataan persepsi dan harapan yaitu sebesar K pernyataan.

2.4. Metode Pengembangan Sistem *Extreme Programming*

Menurut Armanda and Putra (2020) yang mengutip buku Pressman (2015) Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang bertujuan untuk meningkatkan kualitas perangkat lunak dan daya tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi.

Tahapan Extreme Programming terdiri dari perencanaan seperti pemahaman kriteria pengguna dan perencanaan pengembangan, perancangan seperti perancangan prototipe dan tampilan pengkodean termasuk integrasi, dan terakhir pengujian. Elemen lain dari Pemrograman Ekstrim termasuk pemrograman berpasangan pada tahap pengkodean, pengujian unit semua kode, penghindaran fitur pemrograman kecuali benar-benar diperlukan, struktur manajemen datar, kode sederhana dan jelas, dan komunikasi yang sering antara pemrogram dan pelanggan ketika hal itu terjadi. mengubah kebutuhan pelanggan seiring berjalannya waktu.

Metode ini mengambil unsur-unsur yang bermanfaat dari praktek rekayasa perangkat lunak tradisional ke tingkat "ekstrim", oleh karena itu dinamakan Pemrograman Ekstrim. Unsur-unsur yang menjadi ciri metodologi adalah kesederhanaan, komunikasi, umpan balik, dan keberanian. Gambar tahapan XP dapat dilihat pada gambar 2.1 :



Gambar 2.1 Tahapan *Extreme Programming*
Sumber : Pressman (2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning*

Perencanaan berfokus untuk mendapatkan gambaran umum tentang fitur dan fungsi perangkat lunak yang akan dibangun. Kegiatan perencanaan dimulai dengan membuat kumpulan gambar atau cerita yang telah diberikan oleh klien yang akan menjadi gambaran dasar dari perangkat lunak tersebut. Kumpulan gambar atau cerita akan dikumpulkan dalam sebuah indeks yang tiap poinnya memiliki prioritas tersendiri. Tim pengembang aplikasi juga akan menentukan perkiraan waktu dan biaya yang dibutuhkan untuk setiap indeks. Setelah semua persyaratan terpenuhi, tim XP akan menentukan alur pengembangan aplikasi sebelum memulai tugas pengembangan.

2. *Design*

Kegiatan perancangan dalam pembangunan aplikasi ini bertujuan untuk mengatur pola-pola logika dalam sistem. Desain aplikasi yang baik adalah desain yang dapat mengurangi ketergantungan antar proses pada suatu sistem. Jika salah satu fitur pada sistem rusak, hal itu tidak akan mempengaruhi sistem secara keseluruhan.

Tahapan Design pada model proses Extreme Programming merupakan pedoman dalam membangun perangkat lunak berdasarkan client story sebelumnya yang telah dikumpulkan pada tahap perencanaan. Di XP, proses desain terjadi sebelum dan sesudah aktivitas pengkodean berlangsung. Artinya aktivitas desain terjadi terus menerus selama proses pengembangan aplikasi. Kartu CRC desain sederhana adalah deskripsi untuk mengidentifikasi dan mengelola kelas berorientasi objek sesuai dengan peningkatan perangkat lunak dan solusi lonjakan. Prototipe melakukan solusi spesifikasi kelas berorientasi objek.

3. *Coding*

Setelah melengkapi deskripsi dasar perangkat lunak dan melengkapi desain aplikasi secara keseluruhan, XP merekomendasikan agar tim terlebih dahulu membuat modul uji unit yang bertujuan untuk menguji setiap cerita dan deskripsi yang diberikan oleh klien. Setelah berbagai unit test selesai dilakukan, tim melanjutkan kegiatan penulisan koding aplikasi. XP menerapkan konsep Pair Programming di mana setiap tugas dari sebuah modul dikembangkan oleh dua programmer. XP berpikir, 2 orang akan lebih cepat dan lebih baik dalam memecahkan suatu masalah. Selanjutnya modul aplikasi yang telah dibangun digabungkan dengan aplikasi utama.

4. *Testing*

Meskipun tahap pengujian telah dilakukan pada tahap pengkodean, XP juga akan menguji sistem yang sempurna. Pada tahap coding, XP akan terus memeriksa dan memperbaiki semua masalah yang terjadi meskipun hanya masalah kecil. Setiap modul yang akan dikembangkan akan diuji terlebih dahulu dengan modul uji unit yang telah dibuat sebelumnya.

Setelah semua modul diselesaikan dan dikumpulkan menjadi sistem yang sempurna, tim XP akan melakukan tes penerimaan. Pada tahap ini aplikasi akan langsung diujicobakan oleh pengguna dan klien agar mendapatkan tanggapan langsung terkait penerapan gambar dan story yang telah dilakukan sebelumnya.

2.5. *Unified Modeling Language (UML)*

UML merupakan pengembangan dari teknik pemrograman berorientasi objek, menghasilkan bahasa pemodelan yang terstandarisasi untuk pengembangan

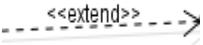
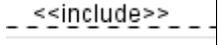
perangkat lunak untuk membuat analisis dan perancangan, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahudin, 2018)

1) *Usecase Diagram*

Menurut Rosa dan Shalahuddin (2018) mengungkapkan *Use Case Diagram* pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use Case* diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang ada pada *Use Case* diagram dapat dilihat pada tabel 2.2 berikut :

Tabel 2. 2 Simbol *Usecase Diagram*

No	Simbol	Keterangan
1.	<i>UseCase</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	<i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<i>Association</i> 	Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada usecase atau <i>Use Case</i> memiliki interaksi dengan aktor.

4.	Ekstensi/ <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 walaupun tanpa <i>Use Case</i> tambahan itu.
5.	<i>Generalization</i> 	Hubungan generalisasi dan spesialisasi 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 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.

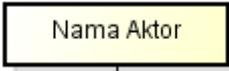
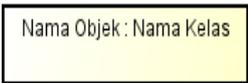
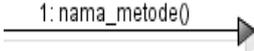
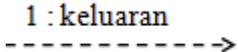
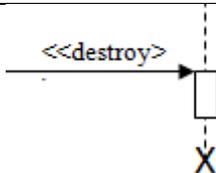
Sumber : Rosa dan Shalahuddin (2018)

2) *Sequence Diagram*

Menurut Rosa dan Shalahudin (2018) *sequence Diagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas. Simbol – simbol yang ada pada *sequence diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Keterangan
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu

Simbol	Keterangan
atau  Tanpa waktu aktif	merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.
Garis hidup	Menyatakan kehidupan suatu objek
Objek 	Menyatukan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe <i>create</i> <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirim data /masukan/informasi ke objek lainnya , arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Rosa dan Shalahuddin (2018)

3) Class Diagram

Menurut Rosa dan Shalahudin (2018) *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. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol – simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.4 berikut ini :

Tabel 2. 4 Simbol Class Diagram

Simbol	Keterangan
Kelas/Class 	Kelas pada struktur system
Antar muka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/ <i>directed</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan atau <i>Dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.

(Sumber: Rosa danShalahuddin, 2018)

2.6. Pengertian *MySQL*

Gilmore (2015) *MySQL* adalah server database relasional yang menawarkan berbagai mekanisme untuk memproses data yang dikenal sebagai mesin penyimpanan.

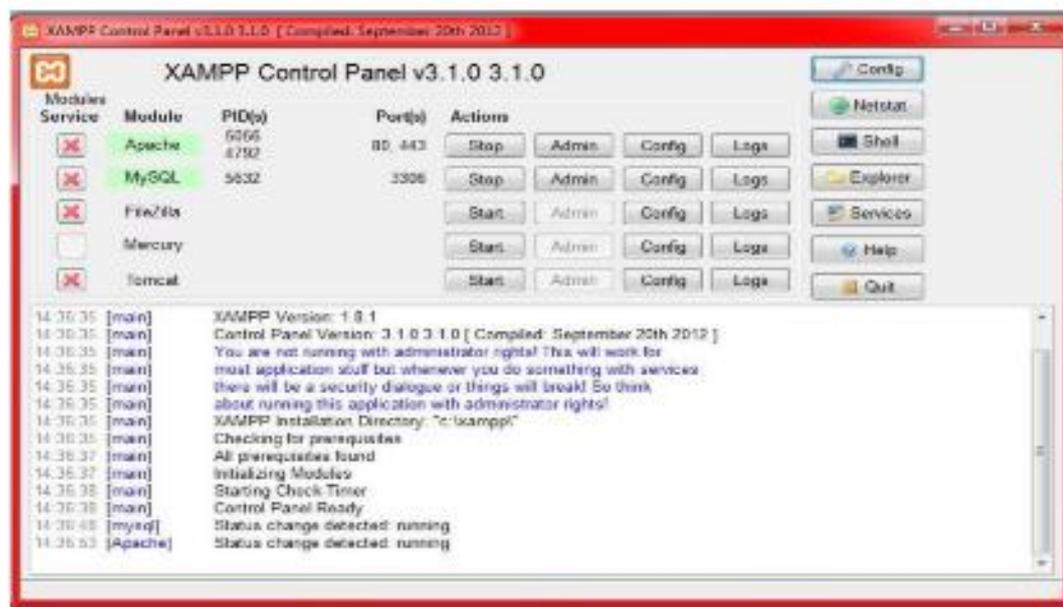
Sheldon and Moes (2015) *MySQL* sama seperti sistem DBMS lainnya, seperti Oracle, DB2, dan SQL Server. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk menentukan data yang disimpan.

Jadi dapat disimpulkan bahwa *MySQL* adalah sebuah relasional *database server* suatu bahasa yang digunakan untuk mengelola data pada *Relation DBMS*.

2.7. XAMPP

Menurut Sheldon and Moes (2015) XAMPP merupakan pengembangan dari LAMP (Linux, Apache, MySQL, PHP and PERL), XAMPP ini merupakan project nonprofit yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server.

XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan MySQL (Sheldon and Moes, 2015) . Berikut ini adalah tampilan XAMPP dapat dilihat pada Gambar 2.2



Gambar 2.2 Tampilan XAMPP

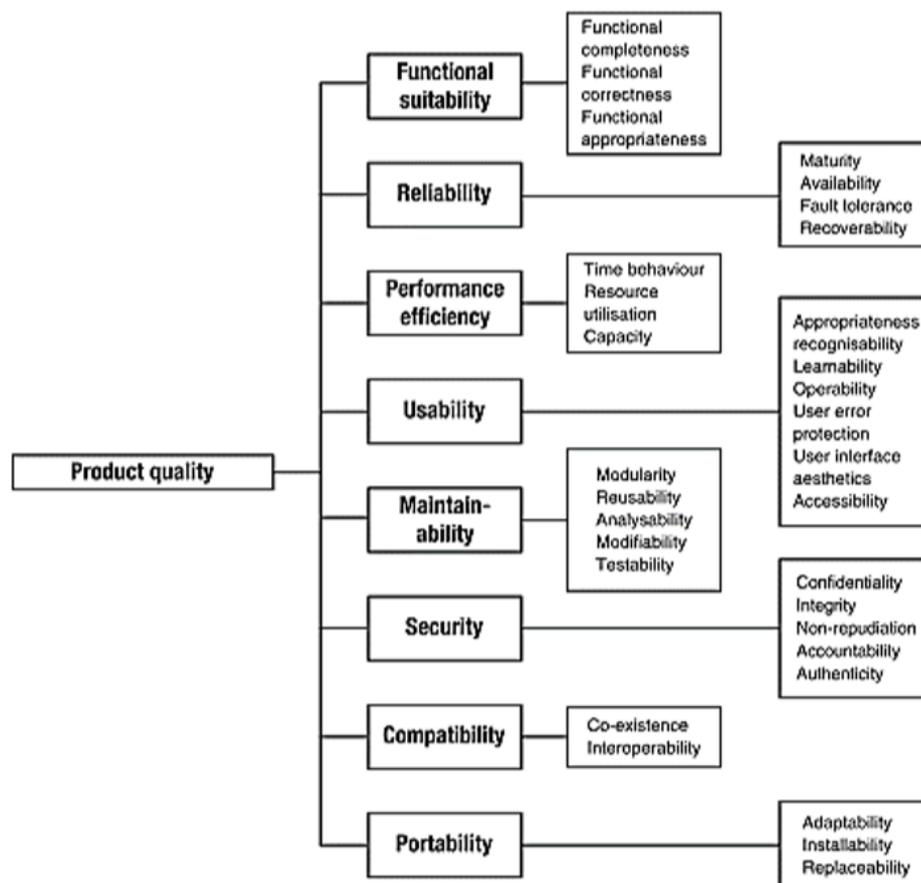
Kelebihan Xampp adalah performa tinggi, stabil, banyak fitur, mudah dikonfigurasi, hanya menggunakan sedikit resource di server, tidak bergantung pada thread untuk melayani klien.

Kekurangan Xampp adalah tidak support IPV6, update / patch terbaru sudah lama, Fast-CGI-nya tidak berfungsi maksimal, penggunaanya tidak sebanyak aktivitas paket program lainnya.

2.8. Pengujian ISO 25010

ISO/IEC 25010 merupakan bagian dari *Systems and software engineering*, *Systems and software Quality Requirements and Evaluation (SQuARE)*, *Systems and software quality models*. ISO/IEC 25010 adalah pengembangan dari ISO/IEC 9126, serta secara resmi membatalkan dan menggantikan ISO / IEC 9126:2001. Edisi pertama ISO/IEC 25010:2011 yang telah direvisi secara teknis, menjadi standar internasional terbaru dan relevan untuk menguji sistem informasi yang akan dikembangkan. ISO/IEC 25010 menjadi standar tolak ukur analisis kualitas

perangkat lunak yang digunakan oleh perusahaan, instansi, ataupun organisasi. Dengan ISO/IEC 25010 evaluasi kualitas sistem perangkat lunak dapat dilakukan secara spesifik berdasarkan dimensi product quality yang terdiri dari 8 karakteristik yaitu functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, dan portability. (*International Organization for Standardization and International Electrotechnical Commission, 2011*).



Gambar 2. 3 Model kualitas produk ISO/IEC 25010

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya

:

- 1) **Fungsionalitas (Fungsionalitas).** Kemampuan perangkat lunak berada pada level dimana perangkat lunak dapat menyediakan fungsionalitas yang

dibutuhkan ketika perangkat lunak digunakan pada kondisi tertentu, dalam hal ini perangkat lunak dapat memenuhi kelayakan suatu fungsi untuk melakukan pekerjaan tertentu bagi pengguna dan dapat memberikan hasil yang tepat. dan akurasi untuk kebutuhan pengguna tingkat. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu.

- a. Kelengkapan fungsional, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna tertentu.
 - b. Ketepatan fungsional, sejauh mana produk atau sistem memberikan hasil yang benar sesuai kebutuhan.
 - c. Kesesuaian fungsional, sejauh mana fungsi yang diberikan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.
- 2) Kompatibilitas, sejauh mana produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen atau melakukan fungsi lain yang diperlukan secara bersamaan ketika berbagi lingkungan perangkat keras dan perangkat lunak yang sama. Ciri ini terbagi menjadi 2 ciri yaitu.
- a. Koeksistensi, sejauh mana produk atau sistem dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi sumber daya dengan produk atau sistem lain tanpa merusak produk atau sistem.
 - b. Interoperabilitas, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.
- 3) Usability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan secara efektif, efisiensi, dan

kepuasan tertentu dalam konteks penggunaan. Ciri ini terbagi menjadi beberapa ciri yaitu.

- a. Ketepatan pengenalan, sejauh mana pengguna dapat mengetahui apakah sistem atau produk memenuhi kebutuhan mereka.
 - b. Learnability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan belajar menggunakan sistem atau produk tersebut secara efisien, efektif, bebas dari risiko dan kepuasan dalam konteks tertentu.
 - c. Operabilitas, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.
 - d. Perlindungan kesalahan pengguna, sejauh mana produk atau sistem melindungi pengguna dari membuat kesalahan.
 - e. Estetika antarmuka pengguna, sejauh mana antarmuka pengguna suatu produk atau sistem memungkinkan interaksi yang ramah pengguna dan memuaskan.
 - f. Aksesibilitas, sejauh mana suatu produk atau sistem dapat digunakan oleh semua kelompok untuk mencapai tujuan tertentu sesuai dengan konteks penggunaannya.
- 4) Reliabilitas Merupakan tingkat dimana perangkat lunak dapat bertahan pada tingkat tertentu ketika digunakan oleh pengguna dalam kondisi tertentu, dalam hal ini perangkat lunak dapat beroperasi dan siap pada saat dibutuhkan untuk digunakan dan juga dapat menahan tingkat kemampuan tertentu. terhadap kegagalan, kesalahan dan pengembalian perangkat lunak sampai tingkat tertentu dalam memulihkan pengembalian data karena

kegagalan atau kesalahan perangkat lunak. Ciri-ciri tersebut terbagi menjadi beberapa sub ciri, yaitu:

- a. Kematangan, sejauh mana suatu produk atau sistem dapat memenuhi kebutuhan dengan andal dalam keadaan normal.
- b. Ketersediaan, sejauh mana produk atau sistem siap untuk dioperasikan dan dapat diakses pada saat perlu digunakan.
- c. Toleransi kesalahan, sejauh mana produk atau sistem tetap berjalan sebagaimana mestinya meskipun ada kesalahan pada perangkat keras atau perangkat lunaknya.
- d. Dapat dipulihkan, sejauh mana produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan mengatur ulang kondisi sistem sesuai keinginan ketika terjadi gangguan.

5) Keamanan, sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan tingkat otorisasi yang dimilikinya. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:

- a. Kerahasiaan, sejauh mana produk atau perangkat lunak memastikan data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
- b. Integritas, sejauh mana produk atau perangkat lunak dapat mencegah akses tidak sah untuk mengubah data.

- c. Non repudiation, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan atas peristiwa atau tindakan tersebut.
 - d. Akuntabilitas, sejauh mana tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
 - e. Keaslian, sejauh mana identitas subjek atau sumber dapat dibuktikan menjadi yang diklaim.
- 6) Portabilitas, sejauh mana efektivitas dan efisiensi suatu sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan dalam lingkungan yang berbeda. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Adaptabilitas, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi ke perangkat lunak, perangkat keras, dan lingkungan yang berbeda.
 - b. Penginstalan, sejauh mana produk atau sistem berhasil diinstal atau dihapus di lingkungan tertentu.
 - c. Replaceability, sejauh mana suatu produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.
- 7) Kinerja terhadap sejumlah sumber daya yang digunakan dalam kondisi tertentu, dalam hal ini efisiensi kinerja dapat memberikan reaksi dan waktu yang diperlukan ketika melakukan suatu tindakan dari suatu fungsi dan perangkat lunak dapat menggunakan jumlah sumber daya saat melakukan

tindakan. dari suatu fungsi. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Ciri ini terbagi menjadi beberapa sub ciri yaitu :

- a. Perilaku waktu, sejauh mana respon dan waktu pemrosesan produk atau sistem dapat memenuhi persyaratan saat menjalankan suatu fungsi.
 - b. Pemanfaatan sumber daya, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem dapat memenuhi persyaratan saat menjalankan fungsinya.
 - c. Kapasitas, sejauh mana batas maksimum produk atau parameter sistem dapat memenuhi persyaratan.
- 8) Maintainability Adalah tingkat dimana perangkat lunak dapat dimodifikasi. Dalam hal ini, modifikasi adalah perbaikan, perubahan atau penyesuaian pada perangkat lunak untuk dapat mengubah lingkungan, persyaratan dan fungsionalitas tertentu. Selain itu, perangkat lunak dapat dianalisis untuk mengetahui penyebab kegagalan perangkat lunak dalam mengidentifikasi bagian-bagian yang dapat dimodifikasi. Sejauh mana efektivitas dan efisiensi suatu produk atau sistem dapat dicapai diobati. Ciri ini terbagi menjadi beberapa sub ciri yaitu.
- a. Modularitas, sejauh mana sistem terdiri dari komponen yang terpisah sehingga perubahan atau modifikasi salah satu komponen tersebut berdampak kecil pada komponen lainnya.
 - b. Dapat digunakan kembali, sejauh mana aset dapat digunakan lebih banyak oleh satu sistem atau digunakan untuk membangun aset lain.

- c. Analisis, tingkat efektivitas dan efisiensi untuk menilai dampak perubahan pada satu atau lebih bagian dari suatu produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. Modifiability, sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa mengurangi kualitas produk yang ada.
- e. Testability, tingkat keefektifan dan efisiensi untuk membentuk kriteria pengujian suatu produk, sistem atau komponen dan pengujian dapat dilakukan untuk mengetahui apakah kriteria tersebut telah terpenuhi.

Dalam penelitian ini menggunakan pengujian ISO 25010 dengan menggunakan 3 aspek yaitu aspek *functionality*, *performance efficiency*, dan *usability*.

2.9. Skala Pengukuran

Menurut Budiman, Wahyuni and Bantun (2019) yang mengutip dari buku Sugiyono (2016) Skala pengukuran yang digunakan adalah skala likert, yaitu skala yang didasarkan pada penjumlahan sikap responden dalam menanggapi pernyataan terkait dengan indikator suatu konsep atau variabel yang diukur. Skala likert pada umumnya menggunakan lima titik dengan label netral di posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.5

Tabel 2. 5 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4

Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Sumber : (Sugiyono, 2017)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini

$$\text{Persentase} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\% \dots \dots \dots (7)$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
90 – 100	Sangat Baik
80 - 90	Baik
70 - 80	Cukup
60 - 70	Kurang
< 60	Sangat Kurang

Keterangan: x = persentase hasil pengujian.