

BAB II TINJAUAN PUSTAKA Dan LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada bagian ini terdapat beberapa penelitian yang memiliki keterkaitan dengan penelitian yang penulis lakukan, untuk dapat melihat lebih jelas *literasi-literasi* untuk tinjauan pustaka dapat dilihat pada Tabel 2.1.

Table 2.1 Daftar Literatur

No 1	Rayhan S (2022)
Judul	Aplikasi E-Booking pada Studio VIBESROOM menggunakan FRAMEWORK LARAVEL
Dokumen for web	https://eprints.amikompurwokerto.ac.id/id/eprint/1418/
Volume dan halaman	-
Tahun	2022
Penulis	Rayhan Shaquelle
Identifikasi Masalah	Tujuan penelitian ini yaitu untuk memberikan informasi lebih luas untuk memperkenalkan studio foto tersebut dan merancang proses sistem pemesanan menjadi lebih efektif
Metode/Tools	-Model UML -Sistem ini dikembangkan menggunakan metode <i>Extreme Programming (XP)</i> - Bahasa pemrograman php - DataBase Mysql
Hasil Penelitian	Hasil penelitian ini yaitu dimana dalam proses pembangunan sistem ini dapat memberikan informasi untuk customer lebih luas tentang studio tersebut dan membuat sistem lebih mudah untuk mengelola data pemesanan yang digunakan untuk memberikan data yang real time untuk customer.
No2	(Irvansyah, et al., 2020)
Judul	Aplikasi pemesanan Jasa Cukur Rambut berbasis <i>Android</i> .
Jurnal	Jurnal Ilmiah Infrastruktur Teknologi Informasi (JIITI)

Volume dan halaman	I dan 16
Tahun	2020
Penulis	Fajar Irvansyah, Setiawansyah dan Muhaqiqin
Identifikasi Masalah	Tujuan penelitian ini yaitu untuk merancang aplikasi untuk memberikan informasi melalui <i>Android</i> sebelum nya masih menggunakan sistem konvensional.
Metode/Tools	--Model UML -Sistem ini dikembangkan menggunakan metode <i>Extreme Programing (XP)</i> - Bahasa pemrograman php - DataBase Mysql
Hasil Penelitian	Hasil penelitian ini yaitu sistem ini bertujuan untuk mempermudah costumer melakukan pemesanan secara online melalui <i>Android</i> tidak lagi menggunakan sistem konvensional yang dimana costumer harus menunggu terlebih dahulu.
No 3	(Kurniadi, et al., 2020)
Judul	Rancangan bangun Sistem Informasi rekam medis pada Klinik menggunakan Metode <i>Extream Programing</i>
Jurnal	Jurnal Algoritma Sekolah Tinggi Teknologi Garut
Volume dan halaman	XVII dan 17
Tahun	2020
Penulis	Fajar Irvansyah, Setiawansyah dan Muhaqiqin
Identifikasi Masalah	Tujuan penelitian ini yaitu untuk mengelola data rekam medis yang masih menggunakan sistem konvensional yang di kembangkan menggunakan <i>website</i> .
Metode/Tools	-Model UML -Sistem ini dikembangkan menggunakan metode <i>Extreme Programing (XP)</i> - Bahasa pemrograman php - DataBase Mysql
Hasil Penelitian	Hasil penelitian ini yaitu memberikan kemudahan dimana dalam proses sebelum nya memakan waktu yang cukup lama dalam pengelolaan data rekam medis yang masih menggunakan sistem konvensional.

No 4	(Rizaldi, et al., 2021)
Judul	Penerapan Antrian dan Pemesanan Online di Aplikasi Pearl Salon And BarberShop Berbasis Mobile
Jurnal	Jurnal Algoritma Sekolah Tinggi Teknologi Garut
Volume dan halaman	I dan 17
Tahun	2021
Penulis	Alfian Rizaldi, Viktor Handrianus Pranatawijaya,Putu Bagus Adidyana Anugrah Putra
Identifikasi Masalah	Pada penelitian ini yaitu untuk penerapan antrian dan pemesana untuk bertujuan meningkatkan omset dan menjaga loyalitas pada customer untuk peningkatan pendapatan pada perusahaan.
Metode/Tools	-Model UML -Sistem ini dikembangkan menggunakan metode <i>Extreme Programing (XP)</i> - Bahasa pemrograman php - DataBase Mysql
Hasil Penelitian	Hasil penelitian ini membentuk sistem yang lebih tersusun menggunakan metode <i>extream programming</i> mampu menjaga kualitas pada perusahaan dan juga menaikkan omset pendapatan.
Nama (Tahun)	(Yudiyanto dan Birowo, 2021)
Judul	Implementasi Aplikasi Jasa pengiriman barang berbasis <i>Android</i> pada CV. EXPRESS TRI'YO MUJUR
Jurnal	Jurnal Informatika dan Bisnis
Volume dan halaman	X dan 18
Tahun	2021
Penulis	Yudiyanto dan Sigit Birowo
Identifikasi Masalah	Pada penelitian ini yaitu untuk menerapkan sistem pengolahan data dan memberikan informasi tentang harga pada pengiriman barang dari satu tempat ketempat lain.

Metode/Tools	- Sistem ini dikembangkan menggunakan metode <i>Incremental</i> - Bahasa pemrograman php - DataBase Mysql
Hasil Penelitian	Hasil penelitian ini membentuk sistem yang lebih tersusun dari pengolahan data dari admin pada perusahaan tersebut tentang klaim barang dan mengupdate setiap harga pada pengiriman barang dari satu tempat ketempat lainnya.

Kesimpulan dari Tinjauan pustaka ini dengan rumusan masalah nya yaitu untuk mempermudah *user/customer* dalam melakukan proses Transaksi secara *online* dan admin untuk sebagai ngelola data pemesanan yang telah di inputkan oleh *user/customer*. Disini menggunakan metode extreme programing yaitu merupakan sistem yang mempunyai rancangan sederhana dan metode ini termasuk dalam skala kategori aplikasi bersekala kecil dan proses untuk menghasilkan tidak butuh waktu lama sehingga lebih efektif dan efisien. Untuk penggunaan tools nya pada tinjauan pustakan diatas itu Bahasa pemrograman php dan Data Base nya menggunakan MySql.

2.2 *E-Booking*

E-Booking adalah pemesanan secara elektronik yang dapat dilakukan dimana saja dan kapanpun. Aplikasi *E-Booking* adalah sebuah aplikasi yang dibangun untuk pemesanan secara elektronik tersebut dengan memanfaatkan jasa internet dalam pengaplikasiannya. Aplikasi ini dapat digunakan pada perusahaan jasa penyewaan dan penjualan produk. Aplikasi ini mampu memberikan layanan kepada pelanggan dengan fitur-fitur yang mudah digunakan dan bagi pengguna jasa aplikasi ini juga diberikan kemudahan dalam informasi pemesanan dan pengontrolan pemesanan dengan laporan-laporan pemesanan yang lengkap (Kurnia, 2020).

2.3 Android

Android merupakan sebuah *Operating System* berbasis *linux* yang dipakai oleh perangkat seluler seperti Smartphone dan tablet. (Pradana, 2019)

Android menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasi. Artinya, satu aplikasi android yang dibangun dapat berjalan di berbagai perangkat yang menggunakan sistem operasi *android* baik itu smartphone, *smartwatch*, *tablet*, dan perangkat lainnya. Perkembangan teknologi *android* yang bergitu pesat juga tidak dapat dilepas dari peran *AOSP (Android Open Source Project)* yang bertanggung jawab dalam pengembangan sistem operasi android dan dipimpin langsung oleh *google* (Irvansyah, et al., 2020).

Android merupakan suatu *Operating System* atau *OS* yang sampai saat ini masih dalam tahap perkembangan, *OS* ini seperti *OS* lainnya seperti, *Symbian*, *IOS* di *I-Phone*, dan lain sebagainya (Pradana, 2019).

2.4 Studio

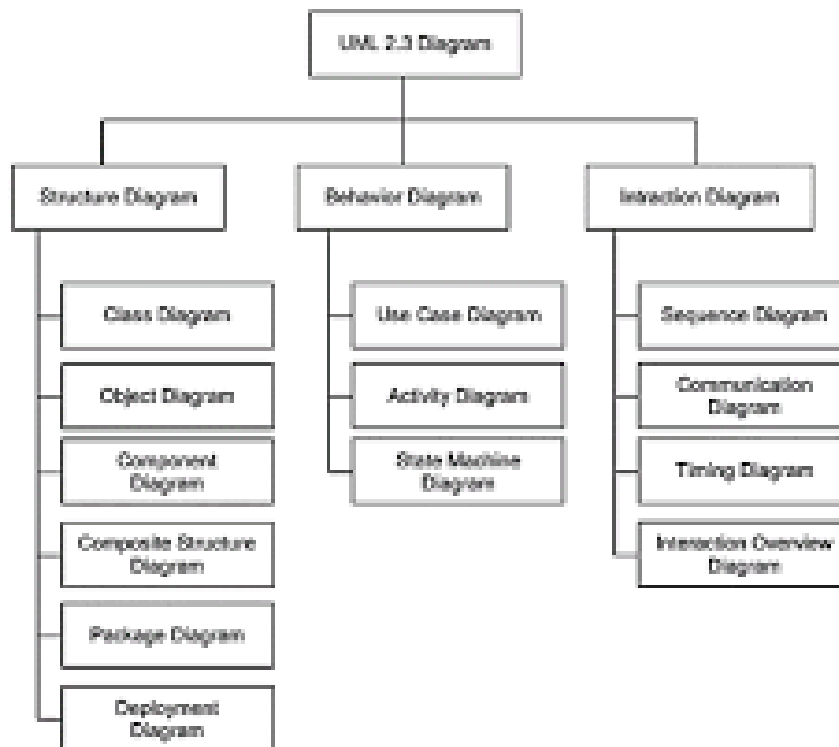
Studio dapat diartikan sebagai ruang bengkel atau tempat di mana seseorang aktif menghasilkan karya. Studio adalah tempat yang digunakan untuk memproduksi film, video, foto dan karya seni lainnya. Secara umum sebuah studio perlu menyediakan berbagai peralatan untuk dapat mencakup semuanya mulai dari tahap awal hingga tahap penyelesaian akhir dari sebuah produksi *audiovisual*. Saat membangun studio ada beberapa persyaratan yang perlu diperhatikan yaitu persyaratan suara pencahayaan, AC, dan lain-lain (Oriwardanu, 2022).

Jadi dapat di simpulkan dari literatur diatas studio merupakan tempat berkarya nya *photographer* atau *videographer* untuk memproduksi foto,video,atau karya lainnya.

2.5 Unified Modeling Language (UML)

UML (unified Modelling Language) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*)(Rosa dan Shalahuddin,2019). Unified Model Language (UML) merupakan bahasa universal untuk :

- a. memvisualisasikan grafis model yang tepat.
- b. menetapkan model yang tepat, lengkap, serta tidak bermakna lebih dari satu untuk dapat mengambil dari semua keputusan penting didalam suatu analisis, desain serta implementasi.
- c. membangun suatu model yang bisa dihubungkan secara langsung dengan bahasa program.
- d. mendokumentasikan semua data informasi yang sudah dikumpulkan oleh team sehingga dapat memungkinkan untuk menshare informasi.



a. *Class Diagram*

Class Diagram adalah menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan *method* atau operasi:

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau method adalah fungsi-fungsi yang dimiliki oleh suatu kelas

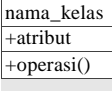
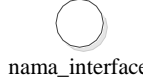


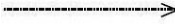

Adapun sifat pada atribut dan *method*, yaitu:

1. *Private*, suatu sifat yang berarti tidak dapat dipanggil dari luar *class* yang bersangkutan dan dinotasikan dalam bentuk (-) pada *class diagram*.
2. *Protected*, suatu sifat yang berarti hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya. Sifat *protected* dinotasikan dalam bentuk (*) pada *class diagram*.

3. *Public*, suatu sifat yang dapat dipanggil oleh siapa saja dan dinotasikan dalam bentuk (+) pada *class diagram*.

4. *Package*, suatu sifat yang hanya dapat dipanggil oleh instance sebuah *class* pada paket yang sama dan dinotasikan dalam bentuk (~) pada *class diagram*.

Table 2.2 Simbol Diagram Kelas (Rosa dan Shalahuddin,2019).


No	Nama	Simbol	Keterangan
1	Operasi		Kelas pada struktur sistem.
2	Antarmuka/ <i>interface</i>		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi berarah		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4	Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
5	Kebergantungan		Relasi antar tabel dengan makna kebergantungan antar kelas.
6	Agresi		Relasi antar kelas dengan makna semua bagian.

b. *Usecase Diagram*

Usecase Diagram merupakan rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian Use Case Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan.

Table 2. 3 Simbol Diagram Use Case (Rosa dan Shalahuddin, 2019).



No	Nama	Simbol	Keterangan
1	<i>Use case</i>	nama use case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.



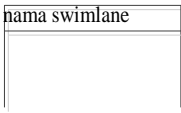
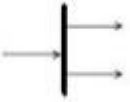

2	Aktor/ <i>actor</i>	 nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3	Asosiasi/ <i>association</i>	—————	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<i>Generalisasi</i>	—————▶	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
5	<i>Extend</i>	<<extend>> ←-----	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.
6	<i>Include</i>	<<include>> ----->	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

c. *Activity Diagram*

Activity Diagram adalah menggambarkan tentang workflow proses dan urutan aktifitas dalam sebuah proses yang sangat mirip dengan flowchart. Membuat activity diagram sangat menguntungkan karena membantu memahami seluru proses pada beberapa use case.

Table 2. 4 Simbol Diagram Aktivitas (Rosa dan Shalahuddin, 2019)




No	Nama	Simbol	Keterangan
1	Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas	aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	<i>Decision</i>		Pilihan untuk pengambilan keputusan.

4	<i>Join</i>		Aktivitas untuk menunjukan kegiatan yang dilakukan secara paralel atau bercabang.
5	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
7	Percabangan		Menunjukan adanya dekomposisi
8	Penggabungan		Tanda penerima

d. *Sequence Diagram*

Diagram ini menggambarkan tentang interaksi antara sejumlah objek dalam urutan waktu untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem..

Table 2. 5 Simbol Diagram Sequence (Rosa dan Shalahuddin, 2019)

No	Nama	Simbol	Keterangan
1	Aktor	 nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
2	Garis hidup/ <i>lifeline</i>		Menyatakan kehidupan suatu objek
3	Objek	nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
5	Pesan tipe <i>send</i>	1 : masukan	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

6	Pesan tipe <i>return</i>	1 : keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
---	--------------------------	--------------	---

2.6 Extreme Programming

Metode *Extreme programming* yaitu metodologi pengembangan perangkat lunak yang dipecah menjadi sprint kerja . Kerangka kerja tangkas mengikuti proses berulang menyelesaikan dan meninjau kerangka kerja setelah setiap sprint, menyempurnakannya untuk efisiensi maksimum , dan menyesuaikan dengan perubahan persyaratan (Anon., 2022). Berikut tahapan yang terdapat pada metode *Extreme Programming* (XP) yaitu:

2.6.1 Perencanaan (*Planning*)

Pada tahap perencanaan XP, menentukan apakah proyek tersebut layak dan paling cocok untuk XP. Untuk melakukan ini dapat di melihat:

- Cerita pengguna untuk melihat apakah cocok dengan nilai kesederhanaan dan periksa untuk memastikan bahwa pelanggan tersedia untuk proses tersebut. Jika cerita pengguna lebih kompleks, atau dibuat oleh pelanggan anonim, kemungkinan besar cerita tersebut tidak akan berfungsi di XP.
- Nilai bisnis dan prioritas proyek untuk memastikan bahwa hal ini sejalan dengan “menyelesaikan pekerjaan yang paling penting terlebih dahulu.”
- Pada tahap pengembangan XP paling baik untuk pengembangan tahap awal, dan tidak akan berfungsi dengan baik untuk iterasi selanjutnya.

Setelah itu buatlah jadwal rilis—namun perlu diingat bahwa harus merilisnya lebih awal dan sering untuk mendapatkan masukan. Untuk melakukan

ini:

- Bagi proyek menjadi beberapa iterasi dan buat rencana untuk masing-masing iterasi.
- Tetapkan tenggat waktu yang realistis dan kecepatan yang berkelanjutan.
- Bagikan pembaruan yang terjadi, yang memberdayakan tim Anda untuk bersikap jujur dan transparan.
- Bagikan pembaruan waktu nyata yang membantu tim mengidentifikasi, beradaptasi, dan membuat perubahan dengan lebih cepat.
- Gunakan alat manajemen proyek untuk membuat papan Kanban atau garis waktu untuk melacak kemajuan Anda secara real-time.

2.6.2 Requirement Analysis/Mengelola

Tahap Requirement Analysis/Mengelola fokus nya adalah mempertahankan kemajuan pekerjaan, mengukur kecepatan, mengubah anggota tim untuk mengatasi bug atau masalah yang muncul, atau mengubah proses XP agar sesuai dengan proyek dan iterasi. Ingat, tujuan XP adalah menjadi fleksibel dan mengambil tindakan, sehingga pekerjaan akan sangat fokus pada pekerjaan tim saat ini dan reaktif terhadap perubahan apa pun.

2.6.2 Desain (*Design*)

Desain yang paling sederhana, karena mengetahui bahwa iterasi selanjutnya akan membuatnya lebih kompleks. Jangan menambahkan fungsionalitas awal pada tahap ini untuk menjaganya tetap sederhana. Tim metodologi XP sering kali menggunakan kartu kolaborasi tanggung jawab kelas (CRC) untuk menunjukkan bagaimana setiap objek dalam desain berinteraksi. Dengan mengisi setiap kolom di kartu dan akan mendapatkan interaksi visual dari semua fungsi yang terkait dan

berinteraksi. Kartu CRC meliputi:

- Kelas (kumpulan benda serupa)
- Tanggung jawab (terkait dengan kelas)
- Kolaborator (kelas yang berinteraksi dengan yang satu ini)

CRC berguna untuk merangsang proses dan menemukan potensi masalah adapun mengurangi potensi kemacetan. Untuk melakukan ini, pastikan secara proaktif mencari risiko. Segera setelah potensi ancaman muncul, tugaskan satu hingga dua anggota tim untuk mencari solusi jika ancaman itu terjadi.

2.6.3 Coding

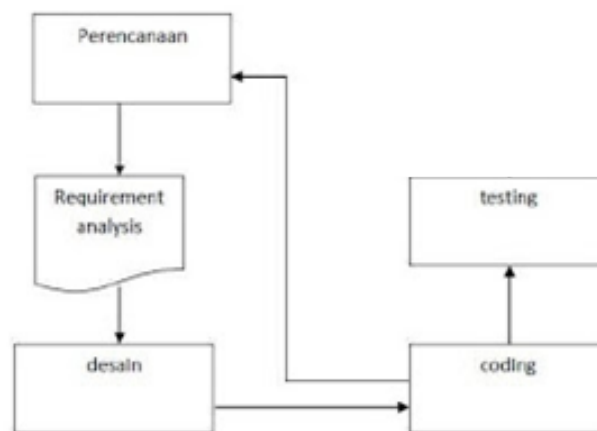
Salah satu aspek unik XP adalah akan terus berhubungan dengan pelanggan selama proses pengkodean. Kemitraan ini memungkinkan menguji dan memasukkan umpan balik dalam setiap iterasi, alih-alih menunggu hingga akhir sprint. Namun aturan pengkodean cukup ketat di XP. Beberapa aturan tersebut antara lain:

- Semua kode harus memenuhi standar pengkodean.
- Menggunakan pengujian unit untuk memenuhi persyaratan dan mengembangkan semua aspek proyek.
- Pemrograman berpasangan—dua pengembang bekerja sama secara bersamaan di komputer yang sama. Ini tidak menambah waktu, melainkan menggunakan fokus ganda untuk menghasilkan hasil kualitas tertinggi.
- Gunakan integrasi berkelanjutan untuk menambahkan kode baru dan segera mengujinya.
- Hanya satu pasangan yang dapat memperbarui kode pada waktu tertentu untuk mengurangi kesalahan.

- Kepemilikan kode kolektif—setiap anggota tim dapat mengubah kode Anda kapan saja.

2.6.4 Testing

Dalam menguji seluruh proses pemrograman ekstrim. Semua kode harus lulus pengujian unit sebelum dirilis. Jika Anda menemukan bug selama pengujian ini, harus membuat pengujian tambahan baru untuk memperbaikinya. Nanti, akan mengonfigurasi cerita pengguna yang sama yang sedang di kerjakan ke dalam tes penerimaan. Selama pengujian ini, pelanggan meninjau hasilnya untuk melihat seberapa baik dalam menerjemahkan cerita pengguna ke dalam produk.



Gambar 2.1 Tahapan Metode Extreme programming

2.7 Kodular

Kodular adalah sebuah situs web, yang menyediakan tools yang untuk membuat aplikasi *android* dengan konsep *drag-drop block programming*. *Blok programming* adalah fitur andalan *Kodular*, dengan fitur ini kita tidak perlu lagi menyetik kode *program* secara manual untuk membuat *aplikasi Android*. Karena kemudahan inilah *pemrograman Android* menggunakan *Kodular* dapat diajarkan sejak usia dini. (Kholifah & Imansari, 2022)

2.8 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program *MySQL* database, *Apache* HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl (Purba dan Rahmat, 2021).

2.9 Java

Java API (Application Programming Interface) adalah sebuah layer pemrograman *Java* yang berisi *class-class* yang sudah didefinisikan dan antar muka pemrograman yang akan membantu pada pengembangan aplikasi dalam perancangan aplikasi (Wahyudi,2019).

Berbeda dengan bahasa pemrograman yang lain yang hanya perlu di-*compile* atau hanya di-*interpret* sudah dapat menjalankan program di komputer, *Java* membutuhkan kedua hal tersebut (*compile dan interpret*) baru dapat menjalankan program yang telah dibuat (Wahyudi,2019).

2.10 CSS

CSS merupakan singkatan dari *Cascading Style Sheet*. Tujuannya adalah untuk mengatur tampilan dokumen *HTML* misalnya dengan mengatur spasi baris teks warna dan format atas bahkan tampilan file gambar baris. *CSS* dikembangkan oleh *W3C*. Organisasi pengembangan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman (Setiawan dan Rostianingsih, 2020).

2.11 MySQL

MySQL merupakan sebuah perangkat lunak atau software sistem manajemen basis data *SQL* atau *DBMS Multithread* dan *multi user*. *MySQL* sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis (Purba dan Rahmat, 2021).

2.12 Black Box Testing

Dapat dikatakan black-box testing merupakan aktivitas pengujian program yang dilakukan dengan mengamati keluaran program yang telah atau dengan kata lain merupakan sebuah aktivitas untuk menguji tombol fitur yang ada pada program yang dibuat. (Kurniadi, et al., 2020).

2.13 ISO 25010

Kualitas suatu sistem adalah sejauh mana sistem tersebut memenuhi kebutuhan yang dinyatakan dan tersirat dari berbagai pemangku kepentingan, dan dengan demikian memberikan nilai. Kebutuhan para pemangku kepentingan tersebut (fungsionalitas, kinerja, keamanan, pemeliharaan, dll.) persis seperti yang direpresentasikan dalam model kualitas, yang mengkategorikan kualitas produk ke dalam karakteristik dan sub-karakteristik. (25010, 2023)

Model kualitas produk yang didefinisikan dalam ISO/IEC 25010 terdiri dari delapan karakteristik kualitas yang ditunjukkan pada gambar berikut:



Gambar 2. 2 Tahapan ISO 25010

a. Functional Suitability (Kesesuaian Fungsional)

Karakteristik ini mewakili sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Kelengkapan fungsional** - Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
- **Ketepatan fungsional** - Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang diperlukan.
- **Kesesuaian fungsional** - Sejauh mana fungsi-fungsi tersebut memfasilitasi pencapaian tugas dan tujuan tertentu.

b. Performance efficiency (Efisiensi kinerja)

Karakteristik ini mewakili kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Perilaku waktu** - Sejauh mana respons dan waktu pemrosesan serta tingkat keluaran suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.
- **Pemanfaatan sumber daya** - Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.
- **Kapasitas** - Sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.

c. Compatibility (Kesesuaian)

Sejauh mana suatu produk, sistem, atau komponen dapat bertukar informasi dengan produk, sistem, atau komponen lain, dan/atau menjalankan fungsi yang diperlukan sambil berbagi lingkungan perangkat keras atau perangkat lunak yang sama. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Ko-eksistensi** - Sejauh mana suatu produk dapat menjalankan fungsi-fungsi yang diperlukan secara efisien sambil berbagi lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak merugikan pada produk lain.
- **Interoperabilitas** - Sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.

d. Usability (Kegunaan)

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Kesesuaian yang dapat dikenali** – Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.
- **Learnability** - Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran tertentu untuk menggunakan produk atau sistem dengan efektif, efisien, bebas dari risiko dan kepuasan dalam konteks penggunaan tertentu.
- **Operabilitas** - Sejauh mana suatu produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.

- **Perlindungan kesalahan Pengguna** - Sejauh mana suatu sistem melindungi pengguna dari kesalahan.
- **Estetika antarmuka pengguna** - Sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
- **Aksesibilitas** - Sejauh mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu.

e. **Reliability (Keandalan)**

Sejauh mana suatu sistem, produk, atau komponen menjalankan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Kematangan** - Sejauh mana suatu sistem, produk atau komponen memenuhi kebutuhan keandalan dalam operasi normal.
- **Ketersediaan** - Sejauh mana suatu sistem, produk atau komponen dapat beroperasi dan dapat diakses ketika diperlukan untuk digunakan.
- **Toleransi kesalahan** - Sejauh mana sistem, produk, atau komponen beroperasi sebagaimana mestinya meskipun terdapat kesalahan perangkat keras atau perangkat lunak.
- **Pemulihan** - Sejauh mana, jika terjadi gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terkena dampak langsung dan memulihkan kondisi sistem yang diinginkan.

f. **Security (Keamanan)**

Sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga orang atau produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasinya. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Kerahasiaan** - Sejauh mana suatu produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
- **Integritas** - Sejauh mana sistem, produk, atau komponen mencegah akses tidak sah, atau modifikasi, program atau data komputer.
- **Non-repudiation** - Sejauh mana suatu tindakan atau peristiwa dapat dibuktikan telah terjadi sehingga peristiwa atau tindakan tersebut tidak dapat disangkal lagi di kemudian hari.
- **Akuntabilitas** - Sejauh mana tindakan suatu entitas dapat ditelusuri secara unik ke entitas tersebut.
- **Keaslian** - Sejauh mana identitas subjek atau sumber daya dapat dibuktikan sesuai dengan yang diklaim.

g. Maintainability (Pemeliharaan)

Karakteristik ini mewakili tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat dimodifikasi untuk memperbaikinya, memperbaikinya atau menyesuaikannya dengan perubahan lingkungan, dan persyaratan. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Modularitas** - Sejauh mana suatu sistem atau program komputer terdiri dari komponen-komponen terpisah sedemikian rupa sehingga perubahan pada satu komponen mempunyai dampak minimal pada komponen lainnya.

- **Reusability** - Sejauh mana suatu aset dapat digunakan di lebih dari satu sistem, atau dalam membangun aset lain.
- **Analysability** - Tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari perubahan yang direncanakan pada satu atau lebih bagian-bagiannya, atau untuk mendiagnosis kekurangan atau penyebab kegagalan produk, atau untuk mengidentifikasi bagian-bagian yang perlu dianalisa dimodifikasi.
- **Modifiability** - Sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan cacat atau menurunkan kualitas produk yang ada.
- **Testabilitas** - Tingkat efektivitas dan efisiensi kriteria pengujian yang dapat ditetapkan untuk suatu sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah dipenuhi.

h. Portability (Portabilitas)

Tingkat efektivitas dan efisiensi sistem, produk, atau komponen yang dapat ditransfer dari satu perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya ke lingkungan lain. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- **Kemampuan beradaptasi** - Sejauh mana suatu produk atau sistem dapat diadaptasi secara efektif dan efisien untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya yang berbeda atau terus berkembang.

- **Kemampuan untuk diinstal** - Tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat berhasil diinstal dan/atau dihapus instalasinya dalam lingkungan tertentu.
- **Replaceability** - Sejauh mana suatu produk dapat menggantikan produk perangkat lunak lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.

2.14 Skala Likert

Likert scale atau skala likert adalah skala penelitian yang digunakan untuk mengukur sikap dan pendapat. Skala ini digunakan untuk mengisi kuesioner yang meminta responden untuk menunjukkan tingkat persetujuan mereka dengan serangkaian pertanyaan. Biasanya pertanyaan yang digunakan untuk penelitian ini berdasarkan variabel penelitian yang di definisikan secara spesifik. Penelitian ini memiliki karakteristik yaitu berupa pertanyaan dari sangat positif hingga sangat negatif, seperti sangat baik, baik, cukup, tidak baik, dan sangat tidak baik (Farid et al., 2023). Persentasi kelayakan yang diperoleh berpedoman pada acuan konversi nilai pada Table 2.6 Sebagai berikut :

Table 2. 6 Skala Konversi Nilai

No	Kategori	Skor Maks
1	Sangat Baik	80%-100%
2	Baik	60%-80%
3	Cukup	40%-60%
4	Tidak Baik	20%-40%
5	Sangat Tidak Baik	0%-20%