

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Adapun perbedaan penelitian dengan penelitian terdahulu dapat dilihat dibawah ini:

2.1.1. Tinjauan Literatur 1

Feizal (2018) meneliti tentang Analisa dan Pengembangan Aplikasi Sentralisasi Kas *Cash Deposit Machine*. CDM atau *Cash Deposit Machine* adalah sebuah perangkat keras yang disediakan oleh bank untuk digunakan oleh nasabah sebagai wahana atau stasiun penyetoran uang tunai. Sistem pencatatan *Cash Deposit Machine* yang terjadi selama ini, dilakukan dan dimaintain oleh pihak cabang, sehingga monitoring terhadap pencatatan tersebut menjadi lebih sulit dan membutuhkan sdm yang lebih banyak. Dilihat dari latar belakang dan permasalahan diatas, maka perlu dibuat sentralisasi Kas CDM agar mempermudah pencatatan dan penggunaan transaksi yang masuk melalui *Cash Deposit Machine*. Selain itu diharapkan sistem sentralisasi Kas CDM ini dapat memperkecil *effort* yang dikeluarkan oleh Bagian Operasional dalam mendukung operasional *Cash Deposit Machine*. Metode yang digunakan adalah studi *referensi* dan perancangan untuk mempelajari informasi yang didapat dari berbagai sumber serta menjadikannya dasar dalam pengembangan aplikasi. Dengan adanya inisiatif dan aplikasi yang dapat melakukan sentralisasi Kas CDM ini dapat memberikan efisiensi dan efektifitas dalam melakukan pengelolaan *Cash Deposit Machine*. Kelebihan dalam penelitian ini mempermudah setiap pusat bank mengambil laporan yang valid. Kekurangan dari penelitian ini adalah membutuhkan biaya

yang besar, karena dalam sistem ini membutuhkan analisis yang matang dan memperhatikan setiap detail dalam sistem Kas CDM untuk menghasilkan setiap laporan yang di perlukan.

2.1.2. Tinjauan Literatur 2

Mulyadi (2021) meneliti tentang Analisis Implementasi Deposito Berjangka Pada Bank Syariah Berdasarkan Hukum Ekonomi Syariah. Deposito berjangka adalah salah satu produk perbankan Syariah yang termasuk dalam kategori pendanaan menggunakan perjanjian Mudharabah. Investor mengambil keuntungan dari dana yang mereka investasikan dalam metode bagi hasil yang disepakati sebelumnya dengan bank. Namun dalam praktiknya rasio yang diberikan oleh bank syariah masih mengacu pada suku bunga dasar yang membuat beberapa investor mengamankan rasio halal. Meskipun haluan atau bukan kontrak mudharabah bebas dari bunga. Jadi perlu mengetahui cara menghitung rasio atau persentase keuntungan yang dibuat oleh bank syariah sehingga jelas posisi hukum setoran dengan perjanjian mudharabah. Kelebihan penelitian ini adalah dapat mengetahui cara menghitung rasio atau persentase keuntungan yang dibuat oleh bank syariah sehingga jelas posisi hukum setoran dengan perjanjian mudharabah. Kelemahan penelitian ini adalah tidak membangun sistem sehingga perhitungan dan pengelolaan masih dilakukan secara manual.

2.1.3. Tinjauan Literatur 3

Ricki and Putri (2020) meneliti tentang Sistem Pakar Simulasi Kredit Dan *Deposito* Berbasis Android PT.BPR Kintamas Mitra Dana belum memiliki

simulasi *kredit* dan *deposito* yang dapat diakses oleh calon debitur sebelum mengajukan pinjaman ke bank sehingga banyak pinjaman yg ditolak, Proses analisa penghitungan simulasi kredit dan *deposito* pada Analisa *kredit* di bank masih belum efisien, serta belum adanya aplikasi khusus untuk melakukan simulasi kredit dan *deposito* yang dapat diakses oleh calon debitur. Sehingga dibutuhkan suatu aplikasi yang dapat menghasilkan pengetahuan untuk menangani masalah-masalah yang timbul dari simulasi kredit dan deposito. Maka dibuatlah suatu aplikasi berbasis android untuk menangani masalah- masalah pada simulasi *kredit* dan *deposito*. Perancangan menggunakan android studio dengan versi android *jelly bean* ke atas agar mudah digunakan oleh kalangan pengguna *smartphone*. Kelebihan penelitian ini adalah sistem yang dibangun dapat membantu dalam pengambilan keputusan *deposito* secara cepat tanpa harus melakukan perhitungan secara manual. Kelemahan penelitian ini adalah hanya menampilkan keputusan tidak mengelola data dan membuat laporan deposito.

2.1.4. Tinjauan Literatur 4

Wahyudi *et al* (2018) meneliti tentang Analisis Kebutuhan Sistem *e-Deposit* (Sistem Serah Simpan Karya *Elektronik*). Undang-undang nomor 4 tahun 1990 tentang serah simpan karya cetak dan karya rekam merupakan undang-undang deposit yang mengatur kewajiban bagi para penerbit atau pengusaha rekaman untuk menyerahkan karyanya, baik itu karya cetak maupun karya rekam kepada Perpustakaan Nasional Republik Indonesia dan perpustakaan daerah. Bidang yang bertugas untuk melaksanakan amanat undang-undang ini adalah Direktorat *Deposit* Bahan Pustaka. Kajian ini bertujuan untuk mengetahui kesiapan dan

kebutuhan sistem *e-deposit* bagi Perpustakaan Nasional Republik Indonesia sampai dengan tahap permodelan bisnis proses (*requirement development*) perangkat lunak *e-deposit* yang meliputi penghimpunan, pengelolaan dan pelaporan. Data dihimpun melalui observasi lapangan, wawancara, FGD (*Focus Group Discussion*) dan penyebaran kuesioner ke penerbit. Data diolah dengan menggunakan analisis SWOT (*Strengths, Weaknesses, Opportunities, Threats*) dan VORD (*Viewpoints Oriented Requirements Definition*) untuk pembuatan aplikasi *e-deposit*. Dari kajian ini diketahui bahwa Perpustakaan Nasional Republik Indonesia khususnya Direktorat *Deposit* Bahan Pustaka secara umum siap menjalankan sistem *e-deposit*. Hal ini didukung dengan dukungan infrastruktur teknologi informasi komunikasi yang dimiliki oleh Perpustakaan Nasional Republik Indonesia, dukungan peraturan dan penganggaran namun tetap memerlukan penambahan sarana perangkat keras seperti UPS dan server. Hasil analisis VORD dijabarkan berupa suatu uraian yang menyangkut perilaku dari sistem (aplikasi) yang akan dikembangkan. Kelebihan penelitian ini adalah menggambarkan alur deposito dengan terperinci. Kelemahan penelitian ini adalah tidak membangun sistem hanya menggambarkan alur pembuatan sistem *e-deposit*.

2.1.5. Tinjauan Literatur 5

Sjafruddin (2019) meneliti tentang Sistem Penjaminan Simpanan (*Deposit Insurance System*) Pada Industri Perbankan Indonesia. Perbankan memegang peran yang sangat penting dalam perekonomian seiring dengan fungsinya untuk menyalurkan dana dari pihak yang mempunyai kelebihan dana (*surplus of funds*) kepada pihak-pihak yang membutuhkan dana (*lack of funds*). Apabila industri

perbankan tidak bekerja dengan baik, maka perekonomian menjadi tidak *efisien* dan pertumbuhan ekonomi yang diharapkan tidak akan tercapai. Risiko yang selalu melekat dalam sektor keuangan dan perbankan, dapat memicu terjadinya krisis sewaktu-waktu dan berakibat lumpuhnya ekonomi negara. Untuk menanggulangi dampak krisis tersebut, pemerintah harus mengeluarkan biaya publik cukup besar. Artikel ini menganalisis beberapa konsep penting, yaitu risiko bank dan efek penularan (*Contagion Effect*), penyelenggaraan sistem penjaminan simpanan yang telah di implementasikan berbagai negara setelah terjadi krisis keuangan dan bagaimana implementasi program penjaminan simpanan di Indonesia. Hasilnya menunjukkan bahwa *Deposit Insurance System (DIS)* dapat diimplementasikan melalui sistem penegakan hukum yang kuat, disiplin pasar, kebebasan politik dan ekonomi, tingkat korupsi yang rendah, regulasi khususnya di bidang perbankan yang kuat, penetapan premi penjaminan simpanan yang memadai dan berdasarkan tingkat risiko bank, serta pemberian jaminan simpanan yang selektif. Kelebihan penelitian ini adalah menerapkan informasi peraturand alam pengelolaan dan pengambilan dana *deposito*. Kelemahan penelitian ini adalah sistem sehingga perhitungan dan pengelolaan masih dilakukan secara manual.

Dari semua penelitian diatas sehingga dapat disimpulkan bahwa hubungan dari penelitian ini yaitu terkait pengelolaan data yang terstruktur sehingga penelitian diatas bisa dijadikan *referensi* untuk penelitian ini. Dan terdapat perbedaan dalam penelitian yaitu :

- a. Sistem yang dibangun dapat mengelola data *deposit* dan pengambilan *deposit*.
- b. Sistem yang dibangun menambahkan menu transaksi *booking golf*

- c. Sistem yang dibangun dapat terhitung secara otomatis jumlah *deposit*
- d. Sistem yang dibangun dapat mencetak laporan secara periode sesuai kebutuhan perusahaan

2.2. Sistem Informasi

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2017).

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018).

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 (Kristanto, 2018).

2.3. Deposit

Deposito adalah tabungan simpanan berjangka yang menjaminkan deposan mendapatkan imbalan berupa bunga dengan ketentuan jangka waktu penempatan yang telah disepakati oleh kedua belah pihak. Dalam bentuk simpanan *deposito* untuk menginvestasikan kelebihan likuiditasnya (Ricki and Putri, 2020).

2.4. Booking

Booking yaitu pemesanan fasilitas yang diantaranya akomodasi, *meal*, *seat* pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club*, *discoutegue* dan sebagainya (Ramadijanti and Badriyah, 2017)

Pemesanan dibagi menjadi dua jenis yaitu pemesanan online dan pemesanan offline :

1) Pemesanan *Online*.

Kemajuan teknologi saat ini mengakibatkan sistem pemesanan juga mengalami perkembangan kearah sistem pemesanan *online*. Pemesanan online bisa diakses oleh siapapun dan dimanapun mereka berada yang memiliki akses internet.

2) Pemesanan *Offline*

Sistem pemesanan yang menggunakan pengiriman pemesanan langsung ke tempat dengan media pemesanan seperti telepon, *fax*, *e-mail*, dan *walk in*.

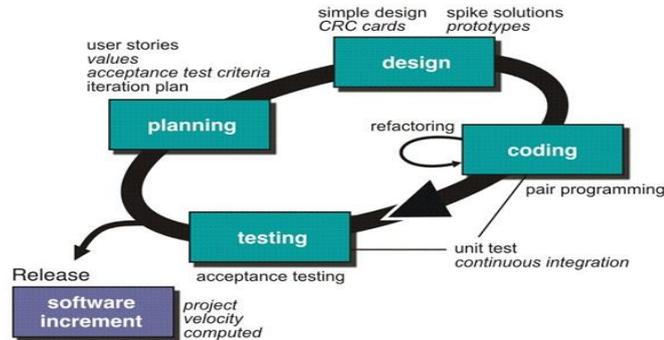
2.5. Metode Pengembangan *Extreme Programming*

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 di mana persyaratan pelanggan baru dapat diadopsi (Pressman, 2012).

Metode ini membawa unsur-unsur yang menguntungkan dari praktek rekayasa perangkat lunak tradisional ke tingkat "*ekstrem*", sehingga metode ini dinamai *Extreme Programming*. Unsur-unsur yang menjadi karakteristik

metodologi adalah kesederhanaan, komunikasi, umpan balik, dan keberanian.

Gambar tahapan XP dapat dilihat pada gambar 2.1 :



Gambar 2.1 Tahapan Extreme Programming

Tahapan-tahapan dari *Extreme Programming* terdiri dari :

1. *Planning*

Pada *Planning* berfokus untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Aktivitas *planning* dimulai dengan membuat kumpulan gambaran atau cerita yang telah diberikan oleh klien yang akan menjadi gambaran dasar dari perangkat lunak tersebut. Kumpulan gambaran atau cerita tersebut akan dikumpulkan dalam sebuah indeks dimana setiap poin memiliki prioritasnya masing-masing. Tim pengembang aplikasi juga akan menentukan perkiraan waktu serta biaya yang dibutuhkan untuk masing-masing indeks. Setelah semua kebutuhan terpenuhi, tim XP akan menentukan alur dari pengembangan aplikasi sebelum memulai pengembangan tugas.

2. *Design*

Aktivitas *design* dalam pengembangan aplikasi ini, bertujuan untuk mengatur pola logika dalam sistem. Sebuah desain aplikasi yang baik adalah desain yang dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Jika salah satu

fitur pada sistem mengalami kerusakan, maka hal tersebut tidak akan mempengaruhi sistem secara keseluruhan.

Tahap Design pada model proses *Extreme Programming* merupakan panduan dalam membangun perangkat lunak yang didasari dari cerita klien sebelumnya yang telah dikumpulkan pada tahap *planning*. Dalam XP, proses design terjadi sebelum dan sesudah aktivitas coding berlangsung. Artinya, aktivitas design terjadi secara terus-menerus selama proses pengembangan aplikasi berlangsung.

3. *Coding*

Setelah menyelesaikan gambaran dasar perangkat lunak dan menyelesaikan design untuk aplikasi secara keseluruhan, XP lebih merekomendasikan tim untuk membuat modul unit tes terlebih dahulu yang bertujuan untuk melakukan uji coba setiap cerita dan gambaran yang diberikan oleh klien.

Setelah berbagai unit tes selesai dibangun, tim barulah melanjutkan aktivitasnya kepenulisan coding aplikasi. XP menerapkan konsep *Pair Programming* dimana setiap tugas sebuah modul dikembangkan oleh dua orang programmer. XP beranggapan, 2 orang akan lebih cepat dan baik dalam menyelesaikan sebuah masalah. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.

4. *Testing*

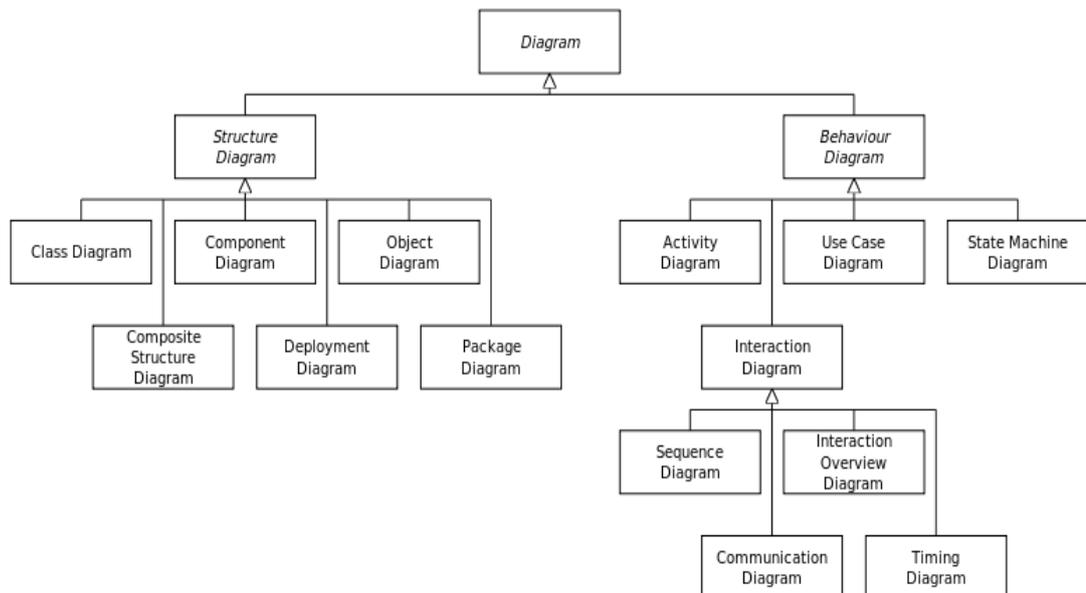
Walaupun tahapan uji coba sudah dilakukan pada tahapan coding, XP juga akan melakukan pengujian sistem yang sudah sempurna. Pada tahap coding, XP akan terus mengecek dan memperbaiki semua masalah-masalah yang terjadi walaupun

hanya masalah kecil. Setiap modul yang sedang dikembangkan, akan diuji terlebih dahulu dengan modul unit tes yang telah dibuat sebelumnya.

Setelah semua modul selesai dan dikumpulkan ke dalam sebuah sistem yang sempurna, maka tim XP akan melakukan pengujian penerimaan atau *acceptance test*. Pada tahap ini, aplikasi akan langsung diuji coba oleh user dan klien agar mendapat tanggapan langsung mengenai penerapan gambaran dan cerita yang telah dilakukan sebelumnya.

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

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek seperti pada Gambar 2.2 di bawah ini: (Rosa dan Shalahuddin, 2018)

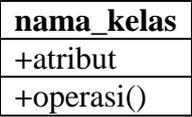


Gambar 2. 2 Tahapan Unified Modeling Language

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 2.1 (Rosa, A.S, 2018)

Tabel 2. 1 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
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>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

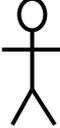
2. Use Case Diagram

Menurut Rosa, A.S. (2018)

“*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. simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.2 di bawah ini:

Tabel 2. 2 Simbol Diagram Use Case

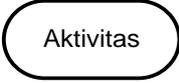
Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan 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
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
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 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

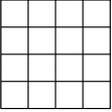
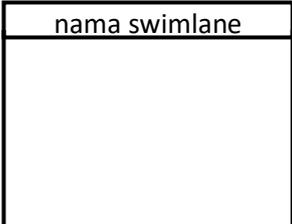
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan/<i>Include/uses</i></p> <p><<<i>include</i>>></p> 	<p>Relasi <i>use case</i> tambahan 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</p>

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 2.3 (Rosa, A.S, 2018)

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Deskripsi
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan/<i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p>Penggabungan/<i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Tabel</p>	<p>Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis</p>

	
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

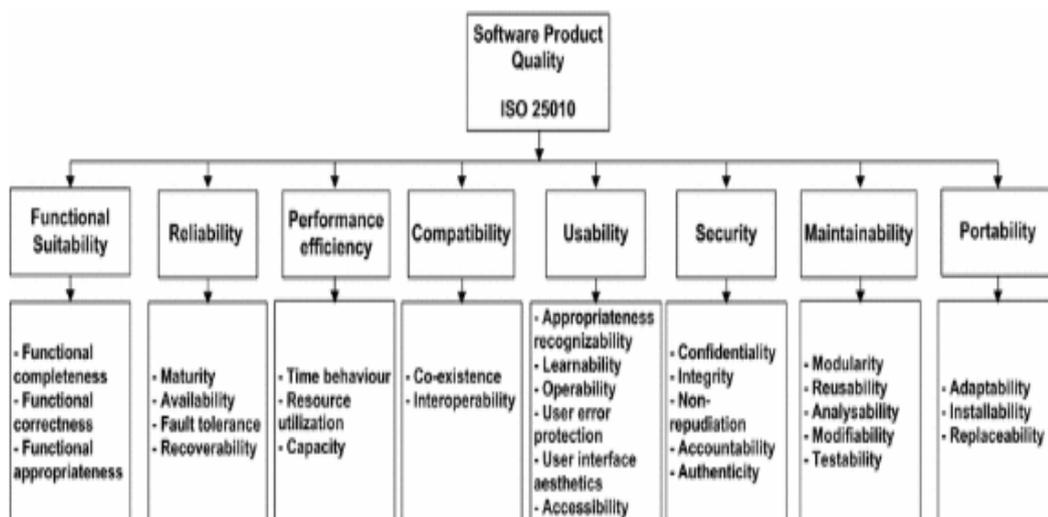
2.7. MySQL

MySQL merupakan *My Structured Query Language* (MySQL) adalah program manajemen basis data atau pabrikan dan sering disebut Manajemen *Database Sistem* (DBMS). Ini adalah sifat *open source* DBMS. MySQL juga merupakan basis data yang diakses jaringan program, sehingga dapat digunakan untuk *multiuser* aplikasi (banyak pengguna). Keuntungan lain MySQL menggunakan bahasa query (permintaan) SQL standar. SQL adalah bahasa *query* terstruktur, SQL telah distandarisasi untuk semua program yang diakses *database* (Wadisman and Nozomi, 2019).

2.8. Pengujian ISO 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 *dimensi quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety* Adapun untuk mengetahui gambaran kualitas *system*, penulis melakukan analisis berdasarkan model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan dimensi *quality in use* (Abran et al., 2008) Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :



Gambar 2. 3 Model kualitas produk ISO/IEC 25010

Sumber: (Kurniawan, Arifianto and Muharom, 2018)

- 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) *Transferability*

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.

2.9. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala *Likert*, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau *variable* yang sedang diukur (Sanusi, 2012). Skala *Likert* umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala *Likert* dapat dilihat pada Tabel 2.4 (Sanusi, 2012)

Tabel 2. 4 *Skala Likert*

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

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

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.5 (Sanusi, 2012)

Tabel 2. 5 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Keterangan: x = persentase hasil pengujian.