

## BAB II LANDASAN TEORI

### 2.1. Tinjauan Pustaka

Beberapa penelitian yang berkaitan dengan penerapan sistem pendukung bantuan operasional pendidikan dalam menentukan kelayakan berdasarkan jurnal penelitian terlihat pada Tabel 2.1:

Tabel 2.1 Tinjauan Pustaka

NO	Penulis (Tahun)	Judul	Metode	Hasil
1.	(Rahmadi <i>et al.</i> , 2021)	Pengembangan Manajemen Sekolah Terintegrasi Berbasis Sistem Informasi di Sumatera Utara	Metode Kualitatif	Hasil Penelitian menjelaskan bahwa penerapan sistem Informasi terintegrasi yang dikembangkan oleh PT. Metromatika Teknologi Rekayasa menggunakan aplikasi edulink. Aplikasi
2.	(Hambali, 2021)	Implementasi Sistem Informasi Manajemen (SIM) Dalam Meningkatkan Mutu Proses Pembelajaran	Metode Kualitatif	Sistem Informasi Manajemen yaitu dengan penyebaran angket/kuisisioner kepada seluruh siswa sebagai sasaran utama peningkatan mutu tersebut. Evaluasi
3	(Honkis, 2017)	Aplikasi Sistem Informasi Manajemen Sekolah Dasar Pilar Bangsa Untuk	Metode RAD ( <i>Rapid Application Development</i> )	Menghasilkan sebuah produk aplikasi Sistem Informasi Manajemen Pendidikan Sekolah Dasar Pilar Bangsa

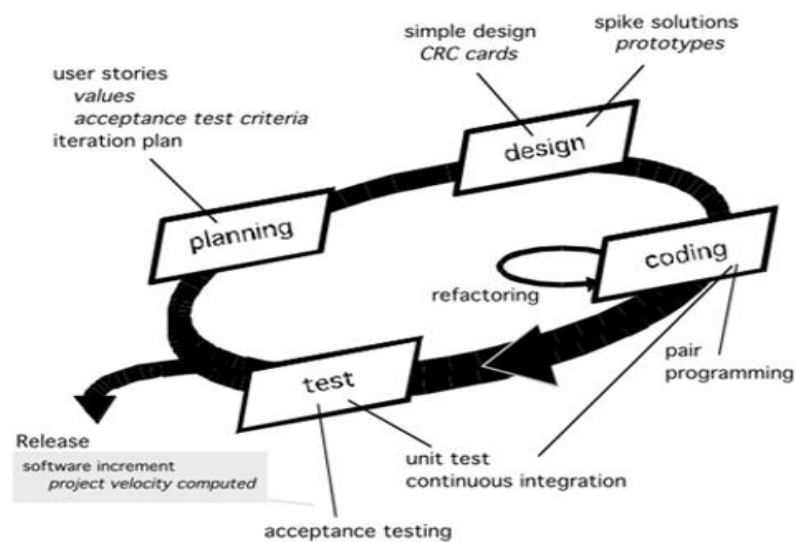
NO	Penulis (Tahun)	Judul	Metode	Hasil
		Meningkatkan Layanan Pendidikan		
4	(Shodiq, 2021)	Peran Sistem Informasi dan Teknologi Informasi terhadap Proses Pembelajaran di Masa Pandemi Covid-19	Metode <i>Prototype</i>	Hasil dari studi literasi menunjukkan bahwa sistem informasi manajemen dan teknologi informasi sangat berperan pada pembelajaran jarak jauh.
5	(Ratih Hendriawati, 2017)	Penerapan Sistem Informasi Manajemen Pendidikan Dalam Meningkatkan Mutu Pendidikan Di Sekolah	Metode <i>Waterfall</i>	Hasil dari penelitian yang dilakukan secara kualitatif dengan melihat berbagai sumber terkait dengan implementasi ataupun penerapan sistem informasi manajemen, dan tentunya juga mencermati berbagai fenomena dari perkembangan sistem informasi manajemen pendidikan dan pemanfaatannya didalam dunia pendidikan saat ini dan juga pihak-pihak terkait.

## 2.2. Sistem Informasi Manajemen Akademik

SIM AKADEMIK (Sistem Informasi Manajemen Akademik) adalah suatu Sistem Informasi Akademik yang dibangun untuk memberikan kemudahan kepada pengguna dalam kegiatan administrasi akademik secara online, seperti proses Penerimaan Mahasiswa Baru (PMB), pembuatan jadwal, pengisian nilai, perwalian, pengelolaan data dosen & mahasiswa. Sistem ini juga dapat berfungsi sebagai pendukung untuk analisis data dalam menentukan keputusan sekolah (Loryana and Syahidul, 2021).

## 2.3. Metode Pengembangan Sistem

*Extreme Programming* (XP) atau Pemrograman Ekstreme yaitu suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat (Pressman, 2012). Pada pengembangan ini terdapat beberapa konteks kegiatan kerangka kerja, perencanaan, perancangan, pengkodean, dan pengujian.



Gambar 2.1 Ilustrasi Proses *Extreme Prgamming*

Sumber: (Pressman, 2012)

Berikut ini adalah penjelasan tahapan *Extreme Programming* (Pressman, 2012), yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*ionalitas.

Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih.

Terdapat *simple design CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak

langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean.

Pada tahapan *pair programming* melakukan kerja sama untuk membuat code dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

#### 4. *Pengujian* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali.

Pada tahapan pengujian yaitu *unit test continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software realease*.

Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software realease* yang nantinya akan diterapkan dalam suatu sistem.





## 2.4. Bahasa Pemodelan Pengembangan Sistem (UML)

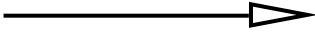

Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement* (kebutuhan), membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin,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

### 2.4.1. Use Case Diagram

*Use case diagram* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Rosa & Shalahuddin, 2018). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat menjelaskan 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
<p><i>Use Case</i></p> 	<p>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></p>
<p>Aktor/<i>actor</i></p> 	<p>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 <i>actor</i></p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i></p>
<p>Ekstensi/<i>extend</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan</p>
<p>&lt;&lt;<i>extend</i>&gt;&gt;</p> 	<p>dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman</p>



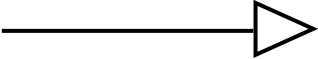

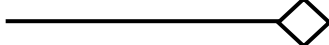
Simbol	Deskripsi
	berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
<p>Generalisasi/<i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan/<i>Include/uses</i></p> <p>&lt;&lt;<i>include</i>&gt;&gt;</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

#### 2.4.2. 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 (Rosa and Shalahudin, 2018). menjelaskan simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3.

Tabel 2.3 Simbol *Class Diagram*

Simbol	Deskripsi			
<p>Kelas/<i>Class</i></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><b>nama_kelas</b></td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table>	<b>nama_kelas</b>	+atribut	+operasi()	Kelas pada struktur sistem
<b>nama_kelas</b>				
+atribut				
+operasi()				
<p>Antarmuka/<i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek			





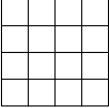


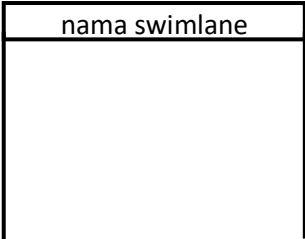
Simbol	Deskripsi
<b>nama_interface</b>	
Asosiasi/Asociation 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi Berarah/Directed Association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/Dependency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/Aggregation 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

### 2.4.3. Activity Diagram

Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Rosa & Shalahuddin, 2018), menjelaskan Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :



Tabel 2.4 Simbol *Activity Diagram*

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

## 2.5. Alat Implementasi

### 2.5.1. Xampp

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (sistem operasi apapun), *Apache*, *MySQL*, PHP dan *Perl*. Program ini tersedia dalam GNU, *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Kusbianto, Ardiansyah and Hamadi, 2017).

Berikut ini Penjelasan dari bagian-bagian *XAMPP*:

1. X , (*XAMPP*) bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah *Windows*, *Linux*, *Mac OS* dan *Solaris*.
2. A (*Apache*) merupakan aplikasi web server. *Apache* ini bersifat *open-source* yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama *Apache* adalah menghasilkan halaman web yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu *database* diakses terlebih dahulu (misalnya dalam *MySQL*) untuk mendukung halaman web yang dihasilkan.
3. M (*MySQL*), merupakan aplikasi *database server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL

merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. *MySQL* dapat digunakan untuk membuat dan mengelola *database* beserta isinya. Kita dapat memanfaatkan *MySQL* untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database*.

4. P (PHP), bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah *MySQL*. namun PHP juga mendukung sistem manajemen *database Oracle*, *Microsoft Access*, *Interbase*, *d-base*, *PostgreSQL*, dan sebagainya.
5. P (*Perl*), bahasa pemrograman, dua diantara karakteristik utama *perl* adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. *Perl* sangat populer di gunakan dalam program-program CGI (*Common Gateway Interface*) dan protokol internet lainnya.

### **2.5.2. Sublime Text**

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi *Phyton* API. Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

*Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan *add-ons* yang bisa diunduh sesuai kebutuhan user (Sadeli, 2015). Hasil pembuatan web yang dibangun menggunakan bahasa PHP dan aplikasi *sublime text* dengan tampilan *text editor* sebagai berikut :



Gambar 2.2 Text Editor Sublime Text

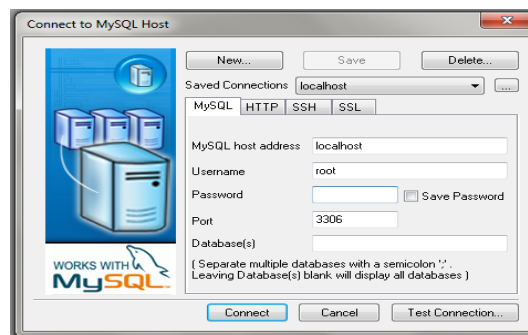
Sumber: (Sadeli, 2015)

### 2.5.3. MySQL

*MySQL (My Structured Query Language)* merupakan program manajemen basis data atau pabrikan dan sering disebut *Database Management System* (DBMS). Ini adalah sifat *open source* DBMS. MySQL juga merupakan basis data yang diakses jaringan program, sehingga dapat digunakan untuk *multi-user*

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 *et al.*, 2019).

*MySQL* adalah nama *database* server. *Database* server adalah server yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. *MySQL* tergolong sebagai *database* relasional. pada model ini, data dinyatakan dalam bentuk dua dimensi yaitu secara khusus dinamakan tabel, tabel tersusun atas baris dan kolom seperti gambar berikut:



Gambar 2.3 login koneksi *SQLyog*

Sumber: (Pasha *et al.*, 2020)

Berikut informasi yang harus diisi untuk melakukan koneksi *database server MySQL*:

1. *MySQL Host Address* diisi dengan nama *host* atau *IP address* dari *MySQL server* yang akan dihubungkan. Nilai *defaultnya* adalah *localhost*.
2. *Username* diisi dengan nama pemakai untuk melakukan koneksi ke *MySQL server*. Nama pemakai yang dimasukkan adalah nama pemakai *MySQL*, bukan nama pemakai *FTP* atau *Web Server*. *User default-nya* adalah *root*.
3. *Password* diisi dengan *password* untuk nama pemakai *MySQL*.

4. *Port* diisi dengan nomor port *TCP/IP* untuk melakukan koneksi ke *MySQL server*. Nomor *port defaultnya* adalah 3306.

Setelah koneksi ke *database server MySQL* berhasil, maka akan muncul jendela utama *SQLyog* seperti gambar berikut :



Gambar 2.4 Jendela Utama *SQLyog*

Sumber: (Pasha *et al.*, 2020)

Jendela utama *SQLyog* dibagi menjadi 3 *panel*, yaitu *object browser*, *SQL window*, dan *result window*:

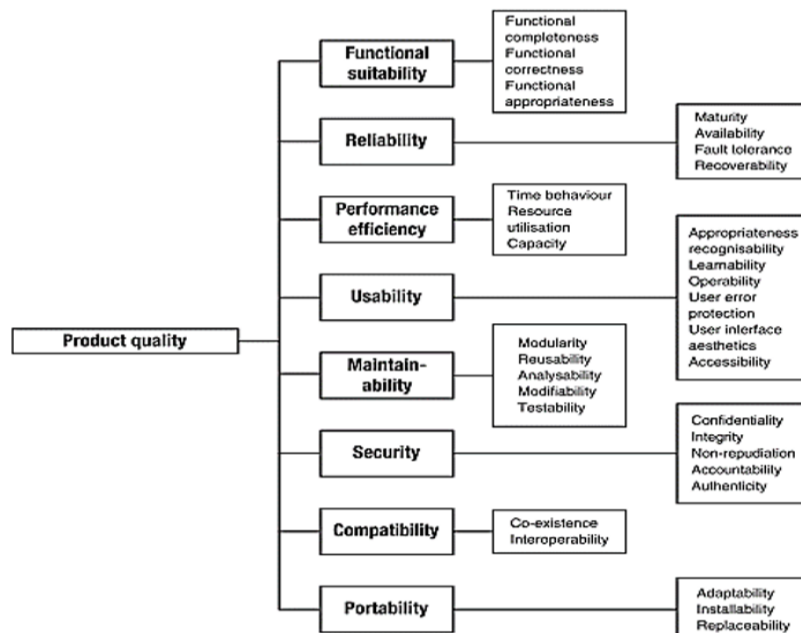
1. *Object browser* menampilkan secara detail semua *database*, tabel, kolom, dan *indeks* yang terdapat pada *MySQL server*.
2. *SQL Window* merupakan jendela tempat mengetikkan perintah *SQL* yang ingin dieksekusi.
3. *Result window* menampilkan hasil *query* yang dieksekusi. Jendela ini juga menampilkan informasi tentang *database*, tabel, kolom, dan *indeks* yang dipilih pada *object browser*.

## 2.6. Pengujian Sistem 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* (Alfian, 2017). *Product quality* ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas



Gambar 2.5 Model kualitas produk ISO/IEC 25010

Sumber: (Alfian, 2017)

Adapun untuk mengetahui gambaran kualitas model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan *dimensi quality in use*. Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) Fungsionalitas merupakan 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 serta 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) Kegunaan, 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. Peningkatan, sejauh mana produk atau sistem berhasil diinstal atau dihapus di lingkungan tertentu.
  - c. Dapat diganti yaitu 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) Pemeliharaan 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. Se jauh 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. Dapat dimodifikasi sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa mengurangi kualitas produk yang ada.
- e. Testabilitas, tingkat keefektifan dan efisiensi untuk membentuk kriteria pengujian suatu produk, sistem atau komponen dan pengujian dapat dilakukan untuk mengetahui apakah kriteria tersebut telah terpenuhi.

## **2.7. 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

(Sugiyono, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada 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

$$Persentase = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\% \dots \dots \dots (2)$$

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
≥ 70%- <90%	Baik
≥ 60%- <80%	Cukup Baik
≥ 50%- <70%	Kurang Baik
0%-60%	Tidak Baik

Sumber : (Sugiyono, 2017)