

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada penelitian ini penulis melakukan tinjauan pustaka pada beberapa penelitian sebelumnya untuk mendukung penelitian yang sedang dilakukan. Berikut adalah tinjauan pustaka yang digunakan penulis yang dapat dilihat pada Tabel 2.1. :

Tabel 2. 1 Tinjauan Pustaka

Nomor Literatur	Penulis dan Tahun	Judul
Literatur 1	(Hantoro, Zaidiah, and Sarika 2021)	Perancangan Sistem Informasi Pengelolaan Keuangan Pembangunan Desa Berbasis Website Menggunakan Framework Codeigniter pada Desa Kedung Pomahan Wetan.
Literatur 2	(Yani and Lusyi 2021)	Sistem Informasi Pengelolaan dan Realisasi Dana APBDES Berbasis Website Pada Kantor Desa Pekuncen.
Literatur 3	(Irmayani et al. 2019)	Sistem Pengolahan Data APBDes (SIPDABDES) Pada Kantor Desa Wajok Hilir Kabupaten Mempawah.
Literatur 4	(Suswandi, Hatta, and Kartika 2021)	Sistem Informasi Pengelolaan Anggaran Pendapatan dan Belanja Desa Berbasis Web (Studi Kasus : Desa Sinarancang).
Literatur 5	(Rasminto et al. 2019)	Sistem Administrasi Pengelolaan Anggaran Dana Desa Terpadu Menggunakan Metode Berorientasi Objek Berbasis Visual.

Penelitian Hantoro, Zaidiah, and Sarika(2021) yang berjudul Perancangan Sistem Informasi Pengelolaan Keuangan Pembangunan Desa Berbasis Website

Menggunakan Framework Codeigniter pada Desa Kedung Pomahan Wetan. Penelitian ini bertujuan untuk membandingkan penggunaan metode pengembangan sistem, metode yang dipakai oleh judul diatas adalah SDLC (*software development life cycle*). Penelitian ini menggunakan framework codeigniter, yang dimana memakai framework tersebut dapat memudahkan untuk membuat website. Selain itu juga *framework codeigniter* ini juga sangat mudah untuk digunakan.

Penelitian Yani and Lusyi (2021) yang berjudul Sistem Informasi Pengelolaan dan Realisasi Dana APBDES Berbasis Website Pada Kantor Desa Pekuncen. Penelitian ini memakai metode pengembangan sistem waterfall. Semakin majunya zaman, kantor desa juga harus mengikuti zaman yang maju ini, contohnya kantor desa pekuncen membuat sistem website untuk APBDes. Ini adalah hal yang menarik untuk di contoh kantor desa lain nya, agar banyak juga yang membuat website. Maka dari itu penelitian ini sangat bagus untuk di contoh kantor desa lain nya agar kantor desa lain nya bisa membuat sistem APBDes berbasis website.

Penelitian Irmayani, Yulia, and Erni (2019) yang berjudul Sistem Pengolahan Data APBDes (SIPDABDES) Pada Kantor Desa Wajok Hilir Kabupaten Mempawah. Penelitian ini bertujuan untuk membantu proses pengolahan data APBDes Desa Wajok Hilir Kabupaten Mempawah supaya desa tersebut dapat mengolah data APBDes dengan efisien dan teratur dalam peg-inputan data. Maka dari itu penelitian ini dibuat untuk mempermudah dan membantu dalam proses pengolahan data APBDes. Dengan adanya sistem pengolahan data APBDes pada Kantor Desa Wajok Hilir Kabupaten Mempawah ini, diharapkan dapat membantu

Kaur Keuangan, Sekretaris dan Kepala Desa dalam mengolah data APBDes lebih mudah, cepat dan Akurat

Penelitian Suswandi, Hatta, and Kartika (2021) yang berjudul Sistem Informasi Pengelolaan Anggaran Pendapatan dan Belanja Desa Berbasis Web (Studi Kasus : Desa Sinarancang). Penelitian ini bertujuan Sebagai bentuk upaya mengatasi masalah, maka perlu dirancang software aplikasi laporan realisasi penggunaan anggaran desa berbasis web. Proses transparansi ini tidak membawa dampak positif kepada tata pemerintahan yang ada di Desa Sinarancang, serta keterbukaan pemerintah dalam membuat kebijakan-kebijakan tidak diketahui oleh masyarakat. Pelaksanaan Program APBDes di Desa Tandu tidak menerapkan prinsip akuntabilitas, walaupun penerapan prinsip akuntabilitas pada tahap ini pertanggung jawaban hanya diberikan kepada pemerintah daerah. Sedangkan kepada masyarakat proses pertanggung jawaban tidak dilakukan sampai sekarang sehingga sampai saat ini respon dari masyarakat untuk menunjang program pemerintah kurang. Masyarakat hanya menginginkan laporan pertanggung dari pemerintah kepada masyarakat sebelum melaksanakan program pemerintah selanjutnya. Dari itulah penelitian ini dibuat untuk membantu proses APBDes agar dapat lebih efisien dan mempercepat pekerjaan aparatur desa tersebut.

Penelitian Rasminto et al (2019), yang berjudul Sistem Administrasi Pengelolaan Anggaran Dana Desa Terpadu Menggunakan Metode Berorientasi Objek Berbasis Visual. Penyusunan Anggaran Pendapatan dan Belanja Desa (APBDes) di Kelurahan Kaliwungu dilakukan dengan menggunakan Microsoft Excel yaitu dengan memasukkan data dari buku Anggaran Pendapatan dan Belanja Desa (APBDes) ke dalam Microsoft Excel. Hal ini akan menambah lama

waktu pengerjaan karena setiap penyusunan Anggaran Pendapatan dan Belanja Desa yang baru harus membuat Form baru di Microsoft Excel. Maka dari itu peneliti membuat aplikasi berorientasi objek visual agar memudahkan desa keluarahan kaliwungu untuk meng-input data APBDes dengan baik dan juga lebih cepat.

Berdasarkan penelitian yang dilakukan terdapat beberapa perbedaan dalam peneliti yaitu :

1. Metode yang digunakan dalam peneltiian yaitu metode *extreme programming*
2. Sistem yang dibangun menghasilkan laporan secara periode (perbulan, tanggal, dan tahun).
3. Sistem yang dibangun menampilkan grafik anggaran dan pendapatan desa

2.2. Anggaran Pendapatan dan Belanja Desa (APBDes)

2.2.1. Definisi APBDes

APBDes adalah suatu rencana tahunan keuangan desa yang ditetapkan berdasarkan peraturan desa yang mengandung prakiraan sumber pendapatan belanja untuk mendukung kebutuhan program pembangunan desa yang bersangkutan (Yani and Lusyi 2021).

2.2.2. Prosedur APBDes

Berikut ini adalah prosedur dalam pengelolaan dana APBDes yaitu :

1. Serketaris Desa melakukan penyusunan rancangan peraturan desa tentang APBDes
2. Kepala Desa melakukan penyampaian kepada PBD untuk pembahasan

3. BPD menyepakati penyusunan yang direncanakan oleh Desa

2.2.3. Dokumen Terkait APBDes

Berikut ini adalah dokumen dalam pengelolaan dana APBDes yaitu :

1. Dokumen berita acara
2. Penyusunan Dokumen Keuangan (APBDes, APBDes perubahan, LPJ APBDes)
3. Kuwitansi pembelanjaan.

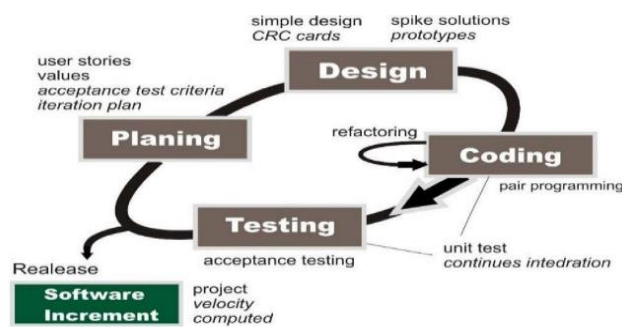
2.2.4. Aturan Pengelolaan APBDes

1. Bahwa Anggaran Pendapatan dan Belanja Desa sebagai wujud dari pengelolaan keuangan Desa dilaksanakan secara terbuka dan bertanggung jawab untuk sebesar-besarnya kemakmuran masyarakat Desa;
2. Bahwa Anggaran Pendapatan dan Belanja Desa Tahun Anggaran 2022 yang termuat dalam rancangan
3. Peraturan Desa tentang Anggaran Pendapatan dan Belanja Desa Tahun Anggaran 2022 yang disusun sesuai dengan kebutuhan penyelenggaraan pemerintahan Desa berdasarkan prinsip kebersamaan, efisiensi, berkeadilan, berkelanjutan, berwawasan lingkungan dan kemandirian sehingga menciptakan landasan kuat dalam melaksanakan pemerintahan dan pembangunan menuju masyarakat yang adil, makmur, dan sejahtera;

2.3. Metode Pengembangan Sistem

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

yang dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan - tahapan dari *Extreme Programming* terdiri dari planning seperti memahami kriteria pengguna dan perencanaan pengembangan, designing seperti perancangan prototype dan tampilan, coding termasuk pengintegrasian, dan yang terakhir adalah testing (Ependi, Usman, and Widayati 2013).



Gambar 2. 1 Tahapan pada *Extreme Programming (XP)*.

Sumber : (Ependi, Usman, and Widayati 2013)

Dibawah ini adalah penjelasan tahapan Extreme Programming 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 fungsionalitas. Pada perencanaan terdapat user stories values yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, acceptance test criteria integration 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 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 16 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 kode 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 release. Selanjutnya terdapat

tahapan software increment project velocity computed yaitu tahapan yang telah diimplementasikan dari software release yang nantinya akan diterapkan dalam suatu sistem.

2.4. E-Government

Menurut Abdurahman and Prasetyo (2018) E-Government adalah penggunaan teknologi informasi oleh instansi pemerintah seperti wide area Networks (WAN) internet, mobile computing, yang dapat digunakan untuk membangun hubungan dengan masyarakat, dunia usaha dan instansi pemerintah lainnya. Terdapat 4 (empat) tipe aplikasi relasi *e-government* yang disebutkan Wahyuno and Sopiandi (2020), yaitu sebagai berikut:

1. Government to Citizens (G2C)

Aplikasi e-Government bertipe G2C adalah untuk mendekatkan pemerintah dengan rakyatnya melalui kanal-kanal akses yang beragam agar masyarakat dapat dengan mudah menjangkau pemerintahnya untuk pemenuhan berbagai kebutuhan pelayanan sehari-hari.

2. Government to Business (G2B)

Dalam melakukan aktivitas bisnis, perusahaan-perusahaan swasta membutuhkan banyak sekali data dan informasi yang mendukung business process perusahaan tersebut yang dimiliki oleh pemerintah. Diperlukannya relasi yang baik antara pemerintah dengan kalangan bisnis tidak saja bertujuan untuk memperlancar para praktisi bisnis dalam menjalankan roda perusahaannya, namun lebih jauh lagi banyak hal yang dapat menguntungkan pemerintah jika terjadi relasi interaksi yang baik dan efektif dengan industri swasta.

3. *Government to Governments (G2G)*

G2G adalah Kebutuhan untuk berinteraksi antar pemerintah setiap harinya tidak hanya berkisar pada hal-hal yang berbau diplomasi semata, namun lebih jauh lagi untuk memperlancar kerjasama antar negara dan kerjasama antar entiti-entiti negara (masyarakat, industri, perusahaan, dan lain-lain) dalam melakukan hal-hal yang berkaitan dengan administrasi perdagangan, proses-proses politik, mekanisme hubungan sosial dan budaya, dan lain sebagainya yang diwujudkan dalam Aplikasi *Government to Government*(G2C).

4. *Government to Employees (G2E)*

Aplikasi e-Government G2E juga bertujuan untuk meningkatkan kinerja dan kesejahteraan para pegawai negeri atau karyawan pemerintahan yang bekerja di sejumlah institusi sebagai pelayan publik agar memudahkan warga negara dalam mendapatkan pelayanan.

2.5. *Unified Modeling Language (UML)*

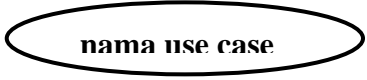
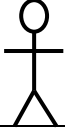




UML merupakan pengembangan dari teknik pemrograman berorientasi objek, menghasilkan bahasa pemodelan yang terstandarisasi untuk pengembangan perangkat lunak untuk membuat analisis dan perancangan, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahudin, 2018)

2.5.1. *Use Case Diagram*

Use Case adalah model untuk perilaku aplikasi yang akan dibuat. Use case mendeskripsikan interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi-fungsi apa

saja yang ada dalam suatu sistem informasi dan siapa yang berhak menggunakan fungsi tersebut (Rosa and Shalahudin 2018).

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></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>Aktor/<i>actor</i></p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang.
<p>Asosiasi/<i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Ekstensi/<i>extend</i></p> <p><<<i>extend</i>>></p> 	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> 	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><<<i>include</i>>></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

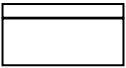



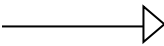
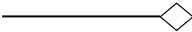
Sumber : Rosa and Shalahudin (2018)

2.5.2. *Class Diagram*

Class diagram adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem.

Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan (Rosa and Shalahudin 2018).

Tabel 2. 3 Simbol *Class Diagram*





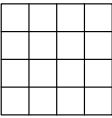


No.	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: Rosa and Shalahudin (2018)

2.5.3. Activity Diagram

Activity diagram atau diagram aktivitas ini menggambarkan rangkaian aliran dari aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa and Shalahudin 2018).

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.

<p><i>Swimlane</i></p> <table border="1" data-bbox="343 257 635 488"> <tr> <td data-bbox="343 257 635 295">nama swimlane</td> </tr> <tr> <td data-bbox="343 295 635 488"></td> </tr> </table>	nama swimlane		<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
nama swimlane			

Sumber : (Rosa dan Shalahiddin, 2018)

2.6. Pengertian PHP

PHP (*Hypertext apareprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. PHP termasuk dalam *Open Source Product*, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis di situs resmi PHP: <http://www.php.net>. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas platform. Artinya PHP dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway*

Interface). PHP dapat mengirim HTTP header, dapat mengatur cookies, mengatur *authentication* dan *redirect user* (Gilmore 2015).

2.7. 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 Oracel, DB2, dan SQL Server. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk menentukan data yang disimpan.

2.8. XAMPP

Menurut Sheldon and Moes (2015) XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan MySQL.

Keunggulan 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.

2.9. Analisis PIECES

Untuk mengidentifikasi suatu masalah, harus dilakukan analisis terhadap kinerja, informasi, keamanan, aplikasi, efisiensi, dan pelayanan. Panduan ini

dikenal dengan analisis PIECES (*performance, information, economy, control, efficiency, dan service*). Berikut penjelasan singkat dari analisis PIECES (Al-Fatta 2012)

1. *Performance*

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

2. *Information*

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

3. *Economy*

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

4. *Security*

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

5. *Efficiency*

Berhubungan dengan sumber daya yang ada guna meminimalkan pemborosan. Efisiensi dari sistem yang dikembangkan adalah pemakaian

secara maksimal atas sumber daya yang tersedia yang meliputi manusia, informasi, waktu, uang, peralatan, ruang, dan keterlambatan pengolahan data.

6. *Service*

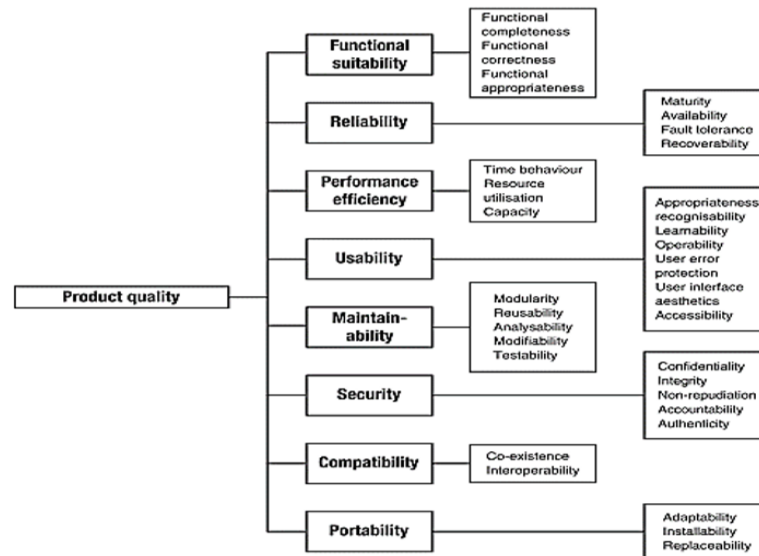
Adalah suatu pelayanan yang lebih baik. Peningkatan pelayanan terhadap sistem yang sedang dikembangkan akan memberikan:

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

2.10. 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 (Tangkudung *et al.* 2019).



Gambar 2. 2 Model kualitas produk ISO/IEC 25010
Sumber : (Tangkudung *et al.* 2019).

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.

2.11. Skala Pengukuran

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 (Sugiyono 2017). 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

$$\text{Persentase} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\% \dots \dots \dots (1)$$

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

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

Sumber : (Sugiyono 2017)

Keterangan: x = persentase hasil pengujian.