

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada penelitian ini, penulis melakukan tinjauan pustaka pada penelitian sebelumnya dan serupa. Sebagai pendukung penelitian yang dilakukan oleh penulis. Dibawah ini merupakan tinjauan pustaka yang sudah di teliti sebelumnya dan serupa :

Tabel 2. 1 Tinjauan Pustaka

Nomor	Detai Jurnal	
1	Judul	Aplikasi Penanganan Keluhan Mahasiswa Menggunakan Metode Prototipe
	Tahun Terbit	2018
	Penulis	Muhammad Ravi Sofyan, Emi Iryanti
	Metode Penelitian	Waterfall
	Hasil	Hasil dari penelitian tersebut adalah sebuah aplikasi penanganan keluhan mahasiswa terhadap kampus, dengan tujuan dapat membantu mahasiswa dan mempermudah mahasiswa dalam menyampaikan keluhan terhadap pihak kampus.

Tabel 2. 2 Tinjauan Pustaka (Lanjutan)

Nomor	Detail Jurnal	
2	Judul	Sistem Pengaduan Layanan Akademik Menggunakan <i>Responsive Web Design</i>
	Tahun Terbit	2019
	Penulis	Yayuk Ike Melani
	Metode Penelitian	Metode Waterfall
	Hasil	Hasil dari penelitian tersebut adalah sebuah website yang menampung keluhan mahasiswa terhadap pelayanan yang diberikan oleh BAAK, Keuangan, dan Sarpras.
3	Judul	Perancangan Aplikasi Pelaporan Kekerasan Perempuan dan Anak pada DPPA Prov Kalbar Berbasis Android
	Tahun Terbit	2019
	Penulis	Syarifah Putri Agustini Alkadri, Rachmat Wahid Saleh Insani
	Metode Penelitian	Metode Prototyping
	Hasil	Hasil dari penelitian tersebut adalah sebuah prototipe aplikasi pelaporan kekerasan berbasis android yang dapat digunakan oleh masyarakat untuk melakukan pengaduan kekerasan dengan melampirkan bukti kekerasan berupa foto atau gambar

Tabel 2. 3 Tinjauan Pustaka (Lanjutan)

Nomor		Detail Jurnal
4	Judul	Sistem Lapor Dini Bencana Kebakaran di Kota Bitung
	Tahun Terbit	2020
	Penulis	Abdul F. Duke, Made Krisnanda, Quido C. Kainde
	Metode Penelitian	Metode Waterfall
	Hasil	Hasil dari penelitian tersebut adalah sebuah sistem lapor dini bencana kebakaran berbasis mobile dengan beberapa fitur untuk mengatasi masalah masyarakat dalam melaporkan adanya bencana kebakaran dan membantu petugas pemadam kebakaran untuk mendapatkan lokasi yang akurat dan rute tercepat ke lokasi kebakaran.
5	Judul	Analisa dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web (Studi Kasus : Universitas Mercu Buana Kranggan)
	Tahun Terbit	2019
	Penulis	Melati Naomi, Handrie Noprisson
	Metode Penelitian	Metode Pieces
	Hasil	Hasil dari penelitian tersebut adalah sebuah rancangan sistem berbasis web yang memudahkan mahasiswa dalam membuat pengaduan dan melihat sudah sejauh mana pengaduannya di proses oleh pihak kampus.

2.1.1. Tinjauan Pada Literatur 1

Pada literature 1 ini meneliti tentang sebuah aplikasi penanganan keluhan mahasiswa berbasis website yang menggunakan metode prototipe sebagai metode penelitiannya. Tujuan dari dibuatnya aplikasi ini adalah untuk menangani pengaduan yang disampaikan oleh mahasiswa terhadap kampus berdasarkan dari bidang kerja staf, dan dapat juga menjadi penghubung antara staf dan mahasiswa yang dapat meningkatkan kualitas kampus.

Perbedaan penelitian yang akan diteliti dengan literatur 1 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti penulis menggunakan metode *Extreme Programming (XP)* sebagai metode penelitiannya, karena selain fleksibel dan adaptif metode ini juga dapat menyederhanakan proses saat pengembangan sistem.

2.1.2. Tinjauan Pada Literatur 2

Pada literature 2 ini meneliti tentang Sistem Pengaduan Layanan Akademik Menggunakan *Responsive Web Design*. *Responsive Web Design* merupakan suatu metode web desain yang bisa menyesuaikan masalah pada perangkat keras dan web browser yang mengoperasikannya untuk menimbulkan respon terhadap kebutuhan. Peneliti pada literature ini menggunakan *Responsive Web Design* untuk memudahkan pengguna karena web tersebut dapat diakses melalui laptop ataupun smartphone, peneliti menggunakan suatu metode yang sering digunakan pengembang lainnya yaitu metode *waterfall* sebagai metode pengembangan sistemnya. Tujuan dari sistem ini adalah agar mahasiswa bisa lebih mudah dalam menyampaikan keluhan-keluhannya kepada layanan akademik tanpa harus menunggu satu bulan untuk menyampaikan keluhan-keluhannya.

Perbedaan literatur 2 dengan penelitian yang akan diteliti adalah terdapat pada metode yang dipakai, pada penelitian yang akan diteliti penulis menggunakan metode *Extreme Programming* (XP) sebagai metode penelitiannya, karena selain fleksibel dan adaptif metode ini juga dapat menyederhanakan proses saat pengembangan sistem, sedangkan pada literatur 2 menggunakan metode *waterfall* sebagai metode pengembangan sistemnya.

2.1.3. Tinjauan Pada Literatur 3

Pada literatur 3 ini meneliti tentang Perancangan Aplikasi Pelaporan Kekerasan Perempuan Dan Anak Pada DPPA Prov Kalbar Berbasis Android. Dengan adanya penelitian ini bertujuan untuk mempermudah masyarakat melaporkan tindakan kekerasan sehingga dapat tertangani dengan cepat oleh pihak terkait. Dalam penelitian ini metode yang digunakan adalah *prototyping*. Digunakannya *prototyping* sebagai metode pengembangan sistem dengan tahapan pengerjaan sistem meliputi tahap *planning*, *analysis*, desain, dan implementasi secara cepat dengan waktu yang relatif lebih pendek untuk segera mendapatkan evaluasi/*feedback* dari pengguna. Uji coba prototipe dilakukan dengan menggunakan metode pengujian *blackbox*. Hasil uji coba menunjukkan bahwa aplikasi kekerasan berbasis android dapat digunakan untuk melaporkan kekerasan dan menampilkan titik lokasi kekerasan di wilayah Kalimantan Barat.

Perbedaan penelitian yang akan diteliti dengan literatur 3 terdapat pada metode yang dipakai, dan pada pengujian sistem. Pada penelitian yang akan diteliti penulis menggunakan metode *Extreme Programming* (XP) sebagai metode penelitiannya, karena selain fleksibel dan adaptif metode ini juga dapat

menyederhanakan proses saat pengembangan sistem, sedangkan pengujian sistemnya menggunakan pengujian ISO 25010.

2.1.4. Tinjauan Pada Literatur 4

Pada literatur 4 ini meneliti tentang Analisis Dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web. Sistem pengaduan mahasiswa ini merupakan sebuah sistem berbasis web yang digunakan untuk membantu mahasiswa dalam mengajukan pengaduan kepada pihak kampus. Proses pengumpulan data dalam merancang sistem ini menggunakan metode PIECES untuk menganalisa penilaian elemen analisis dan metode perancangan Unified Modeling Language (UML) untuk membuat rancangan desain. Hasil dari penelitian ini berupa rancangan sistem berbasis web yang memudahkan mahasiswa dalam membuat pengaduan dan melihat sudah sejauh mana pengaduannya di proses oleh pihak kampus.

Perbedaan penelitian yang akan diteliti dengan literatur 4 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti penulis menggunakan dua metode, yaitu metode *Extreme Programming* (XP) sebagai metode penelitiannya, karena selain fleksibel dan adaptif metode ini juga dapat menyederhanakan proses saat pengembangan sistem, dan menggunakan metode PIECES sebagai metode untuk menganalisis sebuah masalah dan mencari solusinya untuk sistem yang akan dibuat.

2.1.5. Tinjauan Pada Literatur 5

Pada literatur 5 ini meneliti tentang Sistem Laporan Dini Bencana Kebakaran Berbasis *Mobile* di Kota Bitung. Dalam melakukan penelitian ini metode yang digunakan adalah metode *waterfall*. Sedangkan untuk pemodelan

didokumentasikan menggunakan Diagram Alir Data (DFD). Hasil penelitian ini adalah sebuah sistem lapor dini bencana kebakaran berbasis mobile dengan beberapa fitur untuk mengatasi masalah masyarakat dalam melaporkan adanya bencana kebakaran dan membantu petugas pemadam kebakaran untuk mendapatkan lokasi yang akurat dan rute tercepat ke lokasi kebakaran. Sistem yang dibuat terbagi dua yaitu berbasis *mobile android* untuk masyarakat dan berbasis web untuk admin atau petugas pemadam kebakaran.

Perbedaan penelitian yang akan diteliti dengan literatur 5 terdapat pada metode yang dipakai dan pemodelan, untuk metode pada penelitian yang akan diteliti penulis menggunakan metode *Extreme Programming (XP)* sebagai metode penelitiannya, karena selain fleksibel dan adaptif metode ini juga dapat menyederhanakan proses saat pengembangan sistem, sedangkan untuk pemodelannya penulis menggunakan jenis pemodelan UML.

2.2. Flutter

Flutter merupakan salah satu *framework* atau SDK yang dibuat dan dikembangkan oleh Google untuk pengembangan aplikasi mobile. *Framework* flutter ini dapat digunakan dalam membuat dan mengembangkan sebuah aplikasi mobile yang bisa dijalankan menggunakan android dan ios. Bahasa yang digunakan pada *framework* flutter ini bermacam-macam misalnya seperti C, C++, Dart, dan Skia. Dengan tanpa adanya *intrepreter* pada proses *compile*, membuat proses *compile* menjadi lebih cepat karena semua kode di *compile* dalam kode native. Pada flutter terdapat dua widget yang mempunyai peran sebagai *container* dari seluruh layar, dua widget tersebut ialah *stateful widget* dan *stateless widget*.



Gambar 2. 1 Logo Flutter

Sumber : Wikipedia

Stateful widget merupakan sebuah widget pada flutter yang bersifat dinamis atau berubah-ubah, yang dimaksud dari berubah-ubah disini adalah pada saat pengguna menggunakan aplikasi ini komponen pada stateful widget ini berubah-ubah secara dinamis.

Stateless widget merupakan kebalikan dari stateful widget, yang mana widget ini bersifat statis atau tidak dapat diubah-ubah, karena biasanya widget ini digunakan untuk komponen sekunder yang terdapat pada stateful widget yang dibuat pengguna yang berisi sebuah komponen kustomisasi.

2.3. Dart

Menurut (Raharjo, 2019), dart adalah sebuah bahasa yang diproduksi oleh Google, yang dirancang oleh Lars Bak dan Kasper Lund, dan diperkenalkan pada 10 Oktober 2011. Dart dapat digunakan untuk membuat sebuah aplikasi server (berbentuk *command-line interfaces*), web, maupun mobile. Dart dapat digunakan di beberapa platform salah satunya yaitu flutter yang menjadi framework utama yang digunakan oleh penulis untuk membuat sistem.

2.4. Visual Studio Code

Visual studio code merupakan sebuah teks editor yang dibuat oleh Microsoft yang dapat digunakan pada sistem operasi multiplatform, yang didukung dengan beberapa Bahasa pemrograman seperti *JavaScript*, *TypeScript*, dan *Node.js*. Adapun beberapa bahasa pemrograman yang dapat digunakan sebagai bantuan *plugin* yang bisa dipasang melalui marketplace pada *visual studio code* misalnya seperti *C#*, *C++*, *Python*, *Go*, *Java*, dan lain-lain.

Visual studio code ini merupakan jenis teks editor yang bersifat *open source*, yang artinya dimana kode sumbernya dapat dilihat dan dapat dikembangkan untuk bahan pengembangannya. *Visual studio code* dapat digunakan langsung tanpa membutuhkan sebuah ekstensi yang mempunyai ketentuan alur program yang sudah didukung langsung. Kegunaan ekstensi pada *visual studio code* ini adalah agar dapat menambah kemampuan dukungan alur pada program yang diinginkan.

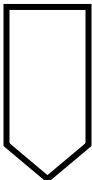
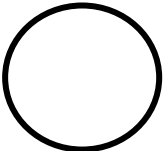


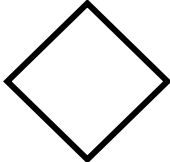
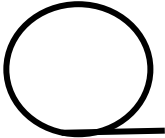
2.5. My SQL

My SQL adalah sebuah tool sistem manajemen basis data yang menggunakan perintah *SQL* sebagai perintah dasarnya. *My SQL* ini termasuk ke dalam tool berjenis RDBMS (*Relational Database Management System*), istilah-istilah yang terdapat pada *My SQL* misalnya kolom, baris, dan tabel. Jadi, *My SQL* dapat diartikan sebagai sebuah *tool* sistem manajemen basis data yang menggunakan bahasa penghubung yaitu bahasa *SQL*.



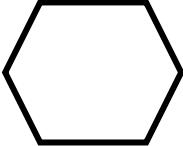
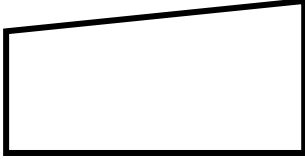


2.6. Flowchart

Menurut (Pahlevy, 2010), *flowchart* merupakan suatu gambaran berbentuk diagram alir dari algoritma-algoritma dalam suatu program, dan yang menyatakan suatu arah alur program tersebut. Jadi, *flowchart* adalah sebuah diagram alir yang menggambarkan sebuah alur jalannya program dari algoritma yang ada pada program tersebut. Gambaran pada *flowchart* dinyatakan dalam bentuk simbol-simbol yang menggambarkan suatu proses tertentu dan diantara proses tersebut dihubungkan dengan garis penghubung. *Flowchart* sangat berguna untuk melakukan pengecekan ulang pada bagian-bagian yang terlewatkan dalam tahap analisis masalah. Dibawah ini merupakan simbol-simbol yang ada pada *flowchart* :





Tabel 2. 4 Simbol-Simbol Flowchart

Simbol	Keterangan
	<p><i>Connector Off Page</i> : simbol yang digunakan untuk menghubungkan proses pada halaman yang berbeda.</p>
	<p><i>Connector On Page</i> : simbol yang digunakan untuk menghubungkan proses pada halaman yang sama.</p>
	<p><i>Terminal Point Symbol</i> : simbol yang digunakan untuk memulai atau mengakhiri suatu proses.</p>
	<p><i>Processing Symbol</i> : simbol yang digunakan untuk menunjukkan sebuah proses yang terdapat pada program.</p>
	<p><i>Decision Symbol</i> : simbol yang digunakan untuk memilih atau membuat keputusan berdasarkan kondisi yang ada pada program.</p>
	<p><i>Magnetic-Tape Unit Symbol</i> : simbol yang digunakan untuk menyatakan inputan berasal dari pita magnetic atau output yang disimpan ke pita magnetic.</p>

Tabel 2. 5 Simbol-Simbol Flowchart (Lanjutan)

Simbol	Keterangan
	<p><i>Input-Output Symbol</i> : simbol yang berguna untuk menunjukkan proses input-output yang terdapat di dalam program.</p>
	<p><i>Predefined Symbol</i> : simbol yang digunakan untuk melaksanakan suatu bagian dari proses.</p>
	<p><i>Preparation Symbol</i> : simbol yang digunakan untuk persiapan penyimpanan yang terdapat pada <i>storage</i>.</p>
	<p><i>Manual Input Symbol</i> : simbol yang digunakan untuk melakukan pengintputan secara manual.</p>
	<p><i>Manual Operation Symbol</i> : simbol yang digunakan untuk melakukan pemrosesan secara manual.</p>
	<p><i>Punch Card Symbol</i> : simbol yang digunakan untuk menyatakan inputan berasal dari kartu atau output yang ditulis dikartu.</p>

Tabel 2. 6 Simbol-Simbol Flowchart (Lanjutan)

Simbol	Keterangan
	<p><i>Disk and On-line Storage Symbol</i> : simbol yang digunakan untuk menyatakan bahwa input yang disimpan ke disk.</p>
	<p><i>Document Symbol</i> : simbol yang digunakan untuk menyatakan bahwa inputan berasal dari dokumen berbentuk kertas atau output yang dicetak ke kertas.</p>
	<p><i>Flow Direction Symbol</i> : simbol yang digunakan untuk menghubungkan simbol satu dengan simbol yang lainnya.</p>
	<p><i>Display Symbol</i> : simbol yang digunakan untuk menyatakan peralatan output, misalnya printer, layar, plotter, dan lain-lain.</p>

2.7. Unified Modelling Language (UML)

Menurut (Rosa dan Shalahuddin, 2013:133), UML adalah salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek.

Menurut (Nugroho, 2010:6), *Unified Modelling Language (UML)* adalah bahasa permodelan khusus untuk sistem atau perangkat lunak dengan paradigma “berorientasi objek”. Pemodelan ini sebenarnya digunakan khusus untuk menyederhanakan berbagai kendala yang kompleks sehingga nantinya mudah untuk dipelajari.



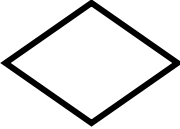

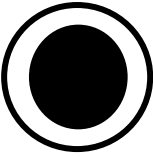
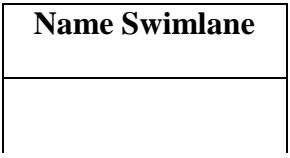
UML terdapat tiga diagram yang mempunyai fungsi masing-masing diantaranya yaitu :

1. *Activity Diagram*

Menurut (Rosa dan Shalahuddin, 2013:161), menyatakan bahwa diagram atau *activity* ini digunakan untuk menggambarkan aliran (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Berikut simbol-simbol yang ada pada *activity diagram* :

Tabel 2. 7 Simbol-Simbol *Activity Diagram*

Simbol	Keterangan
	Status awal, sebuah diagram aktivitas yang memiliki status awal.
	Aktivitas, yang dilakukan oleh sistem, yang biasanya diawali dengan kata kerja.
	<i>Decision</i> , asosiasi percabangan, dimana jika ada pilihan aktivitas lebih dari satu
	<i>Join</i> , asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i> , memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

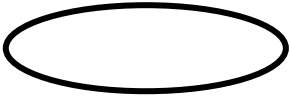


Sumber : Rosa dan Shalahuddin (2013:162)

2. Use Case Diagram




Menurut (Rosa dan Shalahudiin, 2013:155), *Use Case Diagram* adalah sebuah pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat.

Berikut dibawah ini simbol-simbol yang digunakan pada *Use Case Diagram* dapat dilihat pada Tabel 2.4

Tabel 2. 8 Simbol – Simbol *Use Case Diagram*

Simbol	Keterangan
	<p><i>Use Case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit atau actor, yang biasanya dinyatakan dengan kata kerja diawal frase nama <i>use case</i>.</p>
	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, yang biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
	<p>Asosiasi, adalah 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>

Tabel 2. 9 Simbol – Simbol *Use Case Diagram* (Lanjutan)

Simbol	Keterangan
	<< <i>Extend</i> >>, simbol ini menunjukkan bahwa <i>use case</i> tambahan ke <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan tersebut.
	Generalisasi, adalah hubungan umum-khusus antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum.
	<< <i>Include</i> >>, simbol ini menunjukkan bahwa <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.

Sumber : Rosa dan Shalahuddin (2013:155)


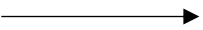
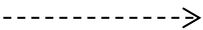
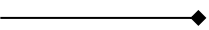
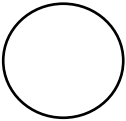

3. *Class Diagram*

Menurut (Rosa dan Shalahuddin, 2013:141), *Class Diagram* adalah suatu pemodelan yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut dibawah ini merupakan simbol-simbol yang ada pada *Class Diagram* :

Tabel 2. 10 Simbol-Simbol *Class Diagram*

Simbol	Keterangan			
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Nama_Kelas</td> </tr> <tr> <td>+ atribut</td> </tr> <tr> <td>- operasi</td> </tr> </table>	Nama_Kelas	+ atribut	- operasi	Kelas, kelas yang terdapat pada struktur
Nama_Kelas				
+ atribut				
- operasi				

Tabel 2. 11 Simbol-Simbol *Class Diagram* (Lanjutan)

Simbol	Keterangan
	Asosiasi berarah, relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
	Generalisasi, relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
	<i>Dependency</i> , relasi antarkelas dengan makna ketergantungan antarkelas.
	Agregasi, relasi antarkelas dengan makna semua bagian
	<i>Interface</i> , sama dengan konsep <i>interface</i> yang ada pada pemrograman objek
	Asosiasi, relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.

Sumber : Rosa dan Shalahuddin (2013:141)

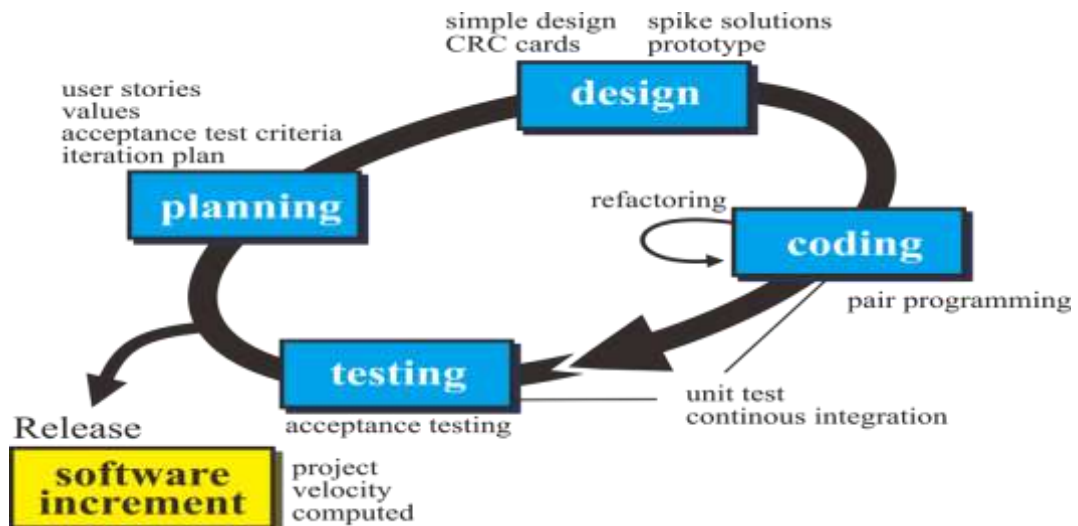
2.8. Metode Pengembangan Sistem

2.8.1. Metode Extreme Programming (XP)

Extreme Programming adalah suatu metode pengembangan perangkat lunak yang digunakan untuk menyederhanakan tahapan saat proses pengembangan, sehingga menjadi lebih fleksibel, adaptif, dan dapat dikerjakan oleh satu atau dua orang. Metode *Extreme Programming* ini dikembangkan oleh Kent Beck pada bulan Maret tahun 1996, menurutnya metode ini adalah suatu metode pengembangan perangkat lunak yang cepat, efisien, beresiko rendah, fleksibel, terprediksi, scientific, dan menyenangkan.

Extreme Programming 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 prosedur pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi (Pressman, 2009).

Pada metode ini terdapat empat tahapan yang harus dilakukan oleh pengembang sebelum mengerjakan sebuah perangkat lunak, empat tahapan tersebut adalah sebagai berikut :



Gambar 2. 2 Tahapan-Tahapan Metode *Extreme Programming* (XP)

1. Planning (Perencanaan)

Pada tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem, tahapan ini penting karena dalam membuat sebuah sistem harus direncanakan atau dianalisis kebutuhan-kebutuhan yang diperlukan user. Dengan cara mengidentifikasi permasalahannya, kemudian menganalisis kebutuhan yang diperlukan, dan menetapkan jadwal untuk melaksanakan pembuatan sistem.

2. Design (Perancangan)

Setelah pada tahapan perencanaan selesai, maka tahapan selanjutnya adalah perancangan. Pada tahapan ini pengembang melakukan perancangan dengan membuat sebuah pemodelan, yang dimulai dari pemodelan sistem, kemudian pemodelan arsitektur, dan yang terakhir adalah pemodelan basis data.

3. Coding (Pengkodean)

Setelah tahapan perancangan selesai, maka tahapan selanjutnya yaitu pengkodean. Tahapan ini merupakan tahapan untuk menerapkan pemodelan

yang sudah dirancang di tahapan perancangan yang sudah dibuat ke dalam bentuk *user interface* dan menggunakan bahasa pemrograman.

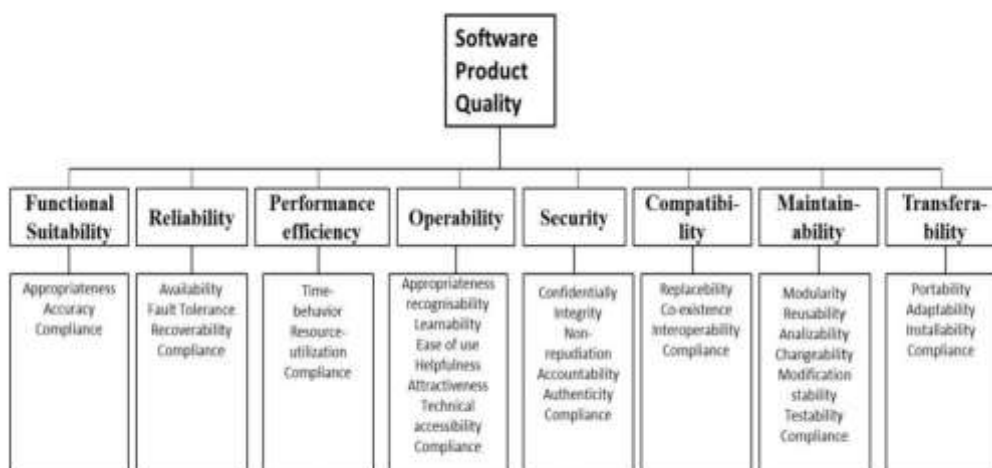
4. Testing (Pengujian)

Tahapan ini merupakan tahapan terakhir setelah melakukan tahapan pengkodean. Pada tahapan ini pengembang sistem melakukan pengujian terhadap sistem yang sudah dibuat untuk mengetahui apakah ada kesalahan yang terdapat pada sistem saat sistem tersebut dijalankan, dan untuk mengecek sistem tersebut apakah sistem yang telah dibuat sudah sesuai dengan kebutuhan pengguna atau belum.

2.9. Pengujian ISO 25010

Menurut (Wattiheluw, 2019), pengujian ISO 25010 merupakan bagian dari *Systems and Software Quality Requirements and Evaluation* (SQuaRE) yang merupakan versi lanjutan dari ISO 91261, yang telah direvisi secara teknis dengan menambahkan beberapa struktur dan bagian dari standar model kualitas. Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko.

Menurut (Harun, 2018), ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer, yang dapat ditunjukkan pada gambar dibawah ini :



Gambar 2. 3 Model ISO 25010

Berdasarkan gambar diatas, dapat dijelaskan mengenai delapan karakteristik tersebut, sebagai berikut :

1. *Functional Suitability*, merupakan sistem atau produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.
2. *Reliability*, merupakan tingkat dimana suatu sistem atau produk dapat mempertahankan kinerjanya pada level tertentu ketika digunakan pada keadaan tertentu.
3. *Performance Efficiency*, merupakan tingkat dimana sistem atau produk menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.
4. *Usability*, merupakan tingkat dimana pada suatu sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.
5. *Security*, merupakan tingkat dimana pada suatu sistem atau produk menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, ataupun pengungkapan yang berbahaya.

6. *Compatibility*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.
7. *Maintainability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
8. *Portability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dipindahkan dari satu ruang ke ruang lainnya.

2.10. Skala Likert

Skala likert merupakan sebuah skala yang dapat digunakan untuk mengukur pendapat, persepsi, ataupun sikap seseorang maupun sekelompok orang tentang suatu fenomena sosial. Untuk mengukur sikap pada suatu objek, subjek, atau kejadian tertentu pada skala likert terdapat dua pernyataan yaitu setuju dan tidak setuju. Pada skala likert ini biasanya juga menggunakan beberapa pernyataan, misalnya seperti sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju.

2.11. PIECES

Menurut (Ragil, 2010:17), metode *PIECES* merupakan sebuah metode analisis yang digunakan sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Analisis ini disebut dengan *PIECES* Analisis (*Performance, Information, Economy, Control, Efficiency, and Service*).

2.12. Algoritma Naïve Bayes Classifier

Menurut (Widianto, 2019), *Naïve Bayes Classifier* adalah sebuah metode klasifikasi yang berakar pada teorema *Bayes*. Metode pengklasifikasian yang menggunakan probabilitas dan statistika ini dikemukakan oleh ilmuwan Inggris Thomas Bayes, dengan berdasarkan prediksi peluang di masa yang akan datang berdasarkan pengalaman di masa yang sebelumnya itulah yang dinamakan Teorema *Bayes*. Keuntungan menggunakan metode ini adalah pada banyaknya data, jumlah data yang kecil menentukan perkiraan parameter yang diperlukan pada proses pengklasifikasian.

Metode bayesian classifier menggunakan pendekatan teori peluang untuk melakukan klasifikasi. Acuan yang digunakan dalam metode ini adalah Teorema Bayes. Dimana teorema bayes`menjelaskan mengenai suatu peluang kejadian berdasarkan kombinasi antara pengetahuan sebelumnya tentang kejadian tersebut dan bukti-bukti baru yang dikumpulkan dari data. Dimana secara sistematis Teorema Bayes dapat diekspresikan sebagai berikut:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Keterangan :

$P(A|B)$ = peluang terjadinya kejadian A dengan syarat kejadian B telah terjadi.

$P(B|A)$ = peluang terjadinya kejadian B dengan syarat kejadian telah terjadi.

$P(A)$ = peluang terjadinya kejadian A, tanpa pengaruh kejadian yang lain.

$P(B)$ = peluang terjadinya kejadian B, tanpa pengaruh kejadian yang lain.

Persamaan teorema bayes diatas dapat digunakan untuk melakukan klasifikasi. Pada permasalahan klasifikasi, variabel B pada persamaan Teorema Bayes menyatakan sekumpulan variabel-variabel pada data A, sedangkan A menyatakan kelas pada data.

Selama proses pelatihan, probabilitas $P(A/B)$ dicermati dari setiap kombinasi A dan B dari data latih. Berdasarkan nilai probabilitas posterior data diklasifikasikan kedalam kelas yang memiliki nilai probabilitas Posterior $P(A|B)$ tertinggi.

Terdapat tahapan-tahapan pada saat memproses algoritma *Naïve Bayes Classifier*, sebagai berikut :

1. Menghitung jumlah kelas atau label.
2. Menghitung jumlah kasus per kelas.
3. Kalikan semua variabel kelas.
4. Bandingkan hasil per kelas.