

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Literatur

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

**Tabel 2.1** Tinjauan Pustaka

No	Penulis & Tahun	Topik	Keterangan
01	(Jorry Karim & Rifki Alfianto Muhtar, 2020)	Pengembangan Sistem Informasi Data Alumni Siswa Pada SMK Negeri 1 Gorontalo Berbasis Android	Sistem informasi yang dikembangkan dapat menjadi solusi permasalahan dari kesulitan data alumni serta menghemat ruang dalam penyimpanan arsip karena data alumni tersimpan dalam database.
02	(Mukrodin & Sugiamto, 2020)	Implementasi Metode Waterfall Dalam Membangun Tracer Study Dan Penerimaan Peserta Didik Baru Dengan Pengujian Black Box Testing	Sistem informasi sekolah dengan menerapkan metode waterfall untuk meningkatkan pelayanan yang selama ini masih lambat dalam pengolahan data pendaftaran dan tracer study.
03	(Carissa Serina Saraswati, Wiyono Sutari & Heriyono Lalu, 2018)	Perancangan Proses Bisnis Tracer Study Di Sekolah Menengah Kejuruan Telkom Bandung Menggunakan Metode Process Design Business Process Management	Tracer study atau penelusuran alumni diperlukan untuk dapat mengetahui sebaran para alumninya sehingga dapat menjadi dasar evaluasi maupun kebijakan.

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

04	(Dewi Rahayu M, Vivine Nurcahyawati & Martinus Sony Erstiawan, 2017)	Rancang Bangun Aplikasi Tracer Study Berbasis Web pada SMK Kesehatan Surabaya	<i>Tracer study</i> aplikasi untuk memudahkan pelaksanaan tracer study, sehingga pendataan alumni lebih banyak akurat dan lebih cepat dari sebelumnya, ada umpan balik dari pengguna lulusan dan alumni, serta terdapat report <i>tracer study</i> yang menggunakan tahapan <i>Software Development Life Cycle (SDLC)</i> yang mana meliputi tahapan analisis sistem, perancangan sistem dan implementasi sistem
05	(Elviza Diana & As'ad, 2017)	Analisis Dan Perancangan Sistem Informasi Tracer Study Berbasis Web	Analisis data tracer study nantinya akan memberikan informasi tentang relevansi pendidikan lulusan dengan pekerjaan yang digeluti saat ini, adanya sistem informasi tracer study yang dapat mempermudah dalam proses pelacakan alumni.

### 2.1.1 Tinjauan Literatur 01

Setiap institusi pendidikan setiap tahun meluluskan alumninya setelah menjalani masa sekolah 3 tahun. Dari tahun ke tahun setiap penerimaan siswa baru maka akan ada pelepasan alumni untuk melanjutkan ke jenjang yang lebih tinggi ataupun langsung terjun ke dunia kerja. Komunikasi dengan para alumni tidaklah mudah, biasanya setelah lulus dari sekolah, para alumni akan sibuk dengan aktifitas baru masing- masing. Sampai saat ini SMK Negeri 1 Gorontalo belum memanfaatkan penggunaan sistem informasi dalam pendataan dan penelusuran alumni. Metode yang digunakan dalam penelitian ini adalah metode *waterfall* yaitu model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Hasil penelitian disimpulkan bahwa sistem informasi yang kembangkan dapat menjadi

solusi permasalahan dari kesulitan data alumni serta menghemat ruang dalam penyimpanan arsip karena data alumni tersimpan dalam *database*.

### **2.1.2 Tinjauan Literatur 02**

SMK.S Al Habibatain Bumiayu merupakan salah satu SMK swasta di Kabupaten Brebes yang memiliki dua program studi yaitu Teknik Komputer Jaringan dan Teknik Sepeda Motor, selama ini aktifitas pengelolaan data siswa yang dilakukan di sekolah masih menggunakan pelayanan manual bahkan sekolah tersebut belum memiliki *website* termasuk dalam pendaftaran mahasiswa baru dan penelusuran alumni (*tracer Study*). Hal ini membuat pelayanan menjadi lambat dan sulit untuk menelusuri para alumni yang sudah bekerja ataupun yang sedang kuliah. Dari persoalan yang ada, maka peneliti berinisiatif melakukan penelitian untuk membangun dan mengembangkan sistem informasi sekolah dengan menerapkan metode *waterfall* untuk meningkatkan pelayanan yang selama ini masih lambat dalam pengolahan data pendaftaran dan *tracer study*. Metode *waterfall* adalah metode yang bersifat sistematis dan berurutan dalam membangun sebuah *software* yang menarik dan mudah digunakan, *waterfall* juga sering disebut dengan *classical life cycle*. Sehingga metode *waterfall* sangat cocok dalam membangun aplikasi yang baru dan juga dalam pengembangan aplikasi dengan tingkat resiko yang kecil serta waktu pengembangan yang relative cukup cepat sehingga mampu meningkatkan kecepatan dan ketepatan dalam pengolahan data pendaftaran dan *tracer study*.

### **2.1.3 Tinjauan Literatur 03**

Dengan semakin banyaknya siswa yang lulus dari SMK Telkom Bandung, *tracer study* atau penelusuran alumni diperlukan untuk dapat mengetahui sebaran para alumninya sehingga dapat menjadi dasar evaluasi maupun kebijakan. Karena

SMK Telkom Bandung belum memiliki proses *tracer study* yang terstandar dan terdokumentasi, penelitian ini bertujuan untuk menghasilkan proses usulan *tracer study* dengan memperhatikan tujuan dari institusi tersebut. Perancangan dimulai dengan melakukan analisis gap berdasarkan kondisi aktual serta *requirement ISO 9001:2015* klausul 8.2.1 tentang komunikasi pelanggan dan PCF pada APQC poin 5.0 tentang Mengembangkan dan Mengelola Hubungan dan Layanan *Stakeholder*. Setelah analisis gap, dilakukan Perancangan proses dilakukan dengan membuat struktur proses terlebih dahulu agar mengetahui aktivitas apa saja yang terjadi sehingga menghasilkan usulan proses yang efisien. perancangan proses *tracer study* dengan metode *process design business process management*. Penelitian ini menghasilkan *SOP* yang diharapkan dapat bermanfaat bagi pihak SMK Telkom Bandung. Untuk mempermudah pelaksanaannya, maka dibuat aplikasi berbasis *web* yang terintegrasi mulai dari pendataan alumni, memperbarui data, serta mendokumentasikan hasil proses *tracer study* yang dilaksanakan.

#### **2.1.4 Tinjauan Literatur 04**

Proses *tracer study* di SMK Kesehatan Surabaya dimulai dari alumni mengisi kuesioner ketika alumni datang ke SMK untuk mengurus kelulusan atau untuk pengambilan ijazah, BK kemudian melakukan rekap data dengan mengolah jawaban kuesioner ke dalam *Microsoft Excel*. Selanjutnya data alumni tersebut akan diberikan kepada Kepala Sekolah untuk kemudian dilaporkan kepada Dinas Pendidikan ketika ada pendataan penyerapan lulusan. Selain itu data alumni tersebut juga digunakan untuk mengetahui kualitas alumni yang dihasilkan. Untuk mengatasi permasalahan tersebut diperlukan sebuah aplikasi *tracer study* berbasis *web*. Aplikasi ini dapat digunakan oleh banyak *user* dan dapat menjangkau alumni

lebih luas. Dalam aplikasi ini alumni dan pengguna lulusan dapat mengisi kuesioner. Kuesioner ini berisi data kegiatan alumni saat ini, data identitas perguruan tinggi tempat alumni melanjutkan *studi*, data identitas perusahaan tempat alumni bekerja (pengguna lulusan), data umpan balik proses pembelajaran, dan data penilaian kinerja alumni.

### **2.1.5 Tinjauan Literatur 05**

Penelitian ini berjudul Analisis dan perancangan sistem informasi *tracer study* berbasis *web*. Rumusan masalah yang diangkat dalam penelitian ini adalah mendapatkan informasi tentang kinerja lulusan Universitas Prof.DR. Hazairin,SH Bengkulu, menganalisis data lulusan untuk memperoleh relevansi kompetensi lulusan Universitas Prof.DR. Hazairin,SH Bengkulu, mengubah sistem lama ke sistem yang baru. Tujuan dari penelitian adalah menyediakan informasi dan laporan hasil pendataan *tracer study* dari para alumni Universitas Prof.DR. Hazairin,SH Bengkulu. Metode pengumpulan data yang dipakai adalah studi pustaka, studi lapangan (wawancara, kuesioner, observasi) dan lab *research*. Metode pengembangan sistem adalah menggunakan metode *waterfall*. Hasil analisis data *tracer study* nantinya akan memberikan informasi tentang relevansi pendidikan lulusan dengan pekerjaan yang digeluti saat ini, adanya sistem informasi *tracer study* yang dapat mempermudah dalam proses pelacakan alumni. Manfaat penelitian adalah Perangkat lunak *tracer study* ini dapat dipergunakan di Universitas Prof.DR. Hazairin,SH Bengkulu, untuk mendapatkan informasi tentang lulusan dari Universitas.

Berdasarkan penelitian diatas menunjukkan bahwa dengan adanya sistem akan mempermudah dalam penelusuran alumni menggunakan aplikasi *tracer study* pada

SMK Negeri 1 Talang Padang. Penelitian yang dilakukan akan dirancang sebuah sistem berbasis *web*. Dalam proses pembuatan sistem nantinya akan menggunakan beberapa *software* seperti *PHP*, *HTML*, *MySql* dan *CodeIgniter* untuk mempermudah dalam pembangunan sistem tersebut.

## **2.2 Tracer Study**

*Tracer study* merupakan sebuah pendekatan yang memungkinkan lembaga pendidikan untuk memperoleh informasi. Tentang jejak lulusan alumni SMK dalam proses mencari atau pun yang sudah bekerja dan dapat membentuk dasar perencanaan kegiatan untuk peningkatan di masa depan dalam menyusun program atau kegiatan dibidang bursa kerja. Kegiatan penelusuran ini dimaksudkan untuk mengetahui keberadaan dan keberhasilan para lulusan suatu lembaga pendidikan yaitu sekolah khususnya SMK di dunia kerja yang sesungguhnya. Keberadaan dan keberhasilan para lulusan tersebut dapat dijadikan bahan acuan dalam menentukan tingkat keberhasilan suatu lembaga pendidikan dalam melaksanakan program-programnya. Melalui informasi dari data lulusan yang diperoleh, dapat diketahui sejauh mana keberhasilan suatu lembaga pendidikan dalam menyiapkan para lulusannya untuk bekerja pada bidang pekerjaan yang relevan (Sagala, Permana, Enda, 2019).

*Tracer study* bertujuan untuk mengetahui *outcome* pendidikan dalam bentuk transisi dari dunia pendidikan ke dunia kerja, output pendidikan yaitu penilaian diri terhadap penguasaan dan pemerolehan kompetensi, proses pendidikan berupa evaluasi proses pembelajaran dan kontribusi pendidikan terhadap pemerolehan kompetensi serta *input* pendidikan berupa penggalan lebih lanjut terhadap informasi sosiobiografis lulusan (Rachmatullah, Gunawan. 2016).

### 2.3 Alumni

Alumni merupakan salah satu bagian dari keberadaan sebuah lembaga pendidikan. Keberadaan alumni pada saat ini menjadi salah satu bagian dalam penilaian akreditasi sebuah lembaga pendidikan. Dengan mengetahui keberadaan alumni maka lembaga pendidikan dapat mengukur sejauh mana keberhasilan dalam mendidik hingga siap bekerja. Untuk mengetahui keberadaan para alumni dapat menggunakan metode *survei* yang sering digunakan atau dengan menggunakan kuisisioner yang akan dibagikan ketika seseorang lulus, atau dengan menggunakan media lain yang dapat digunakan untuk melacak keberadaan alumni (Sodik, 2017).

Untuk kepentingan tersebut maka diperlukan suatu sistem informasi yang berisi tentang lowongan pekerjaan diberbagai perusahaan, melacak beradaan alumni dan sebagai media komunikasi antara dunia industri dan dunia pendidikan. Keberadaan alumni sangatlah berpengaruh pada peningkatan kualitas dari suatu instansi. Ketika alumni bekerja disuatu perusahaan alumni akan membawa nama baik dari sekolah alumni berasal. Selain itu alumni juga dapat membantu almamater mereka dengan memberikan masukan yang bermanfaat (Kirana, Wahdaniyah, 2018).

### 2.4 Analisis *PIECES*

Metode *PIECES* adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi dan pelayanan. Analisis ini disebut analisis *PIECES* (*Performance, Information, Economic, Control, Efficiency and Service*).

Analisis *PIECES* ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi (Supriyatna, Maria, 2017).

Istilah *PIECES* yang dapat diterjemahkan hurufnya sebagai berikut:

1. Analisis Kinerja Sistem (*Performance*)

Kinerja adalah kemampuan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat dengan cepat tercapai. Kinerja diukur dengan jumlah produksi dan waktu yang digunakan untuk menyesuaikan dengan perpindahan kerja.

2. Analisis Informasi (*Information*)

Informasi merupakan hal penting karena dengan informasi tersebut pihak manajemen dan user dapat melakukan langkah selanjutnya.

3. Analisis Ekonomi (*Economic*)

Pemanfaatan biaya yang digunakan dari pemanfaatan informasi, peningkatan terhadap kebutuhan ekonomis mempengaruhi pengendalian biaya dan peningkatan manfaat.

4. Analisis Pengendalian (*Control*)

Analisis ini digunakan untuk membandingkan sistem yang dianalisis berdasarkan pada segi ketepatan waktu, kemudahan akses, dan ketelitian data yang diproses.



#### 5. Analisis Efisiensi (*Effeciency*)

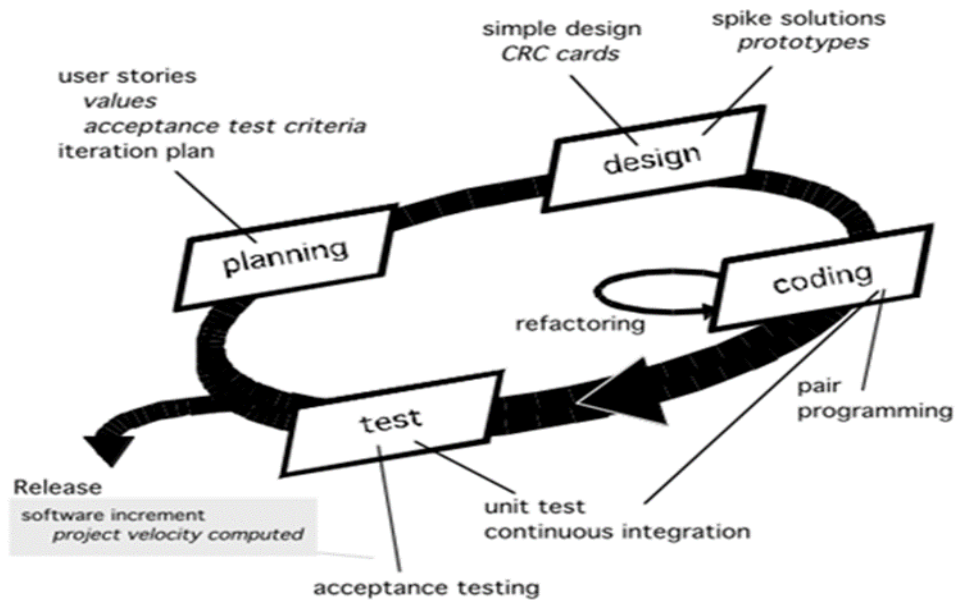
Efisiensi berhubungan dengan bagaimana sumber tersebut dapat digunakan secara optimal. Operasi pada suatu perusahaan dapat dikatakan efisien atau tidak biasanya didasarkan pada tugas dan tanggung jawab dalam melaksanakan tugas.

#### 6. Analisis Pelayanan (*Service*)

Perangkat pelayanan memperlihatkan kategori yang beragam. Proyek yang dipilih merupakan peningkatan pelayanan yang lebih baik manajemen, user dan bagian lain yang merupakan simbol kualitas dari suatu sistem informasi.

### **2.5 Metode Pengembangan Sistem**

Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan requirement yang sangat cepat. Extreme Programming (XP) dikenal dengan metode atau “technical how to” bagaimana suatu tim teknis mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik praktis pengembangan perangkat lunak. XP menjadi dasar bagaimana tim bekerja sehari-hari (Carolina, 2019).



**Gambar 2.1** *Extreme Programming (XP)*

Terdapat empat tahapan yang harus dikerjakan pada metode *Extreme Programmin (XP)* yaitu:

1. *Planning* (Perencanaan).

Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

2. *Design* (Perancangan).

Tahapan berikutnya adalah perancangan dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data. Pemodelan sistem dan arsitektur menggunakan diagram *Unified Modelling Language (UML)* sedangkan pemodelan basis data menggunakan *Entity Relationship Diagram (ERD)*.

### 3. *Coding* (Pengkodean).

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemrograman. Adapun bahasa pemrograman yang digunakan adalah *PHP* dengan metode terstruktur. Untuk sistem manajemen basis data menggunakan piranti lunak *MySQL*.

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

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahapan ini adalah standar *ISO 25010*, dimana pengujian yang dilakukan terhadap *form* beberapa masukkan apakah sudah berjalan sesuai dengan fungsinya masingmasing.

## 2.6 *PHP*

Pengertian *PHP* adalah bahasa pemrograman *script server side* yang didesain untuk pengembangan *web*. Dimana *PHP* ini merupakan singkatan dari *Hypertext preprocessor* yang digunakan sebagai *script* untuk memperindah tampilan *website*. Selain itu *PHP* juga digunakan bersamaan dengan bahasa pemrograman lainnya seperti bahasa pemrograman *HTML* dan *java script*. (Yan Watequlis Syaifudin, 2018). *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada server (*server side HTML embedded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini

menyebabkan informasi yang diterima client selalu yang terbaru/*up to date*. Semua *script PHP* dieksekusi pada server dimana *script* tersebut dijalankan. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. *PHP* adalah bahasa *server-side-scripting* yang menyatukan *HTML* untuk membuat halaman *web* yang dinamis (Arief, 2015).

Salah satu keunggulan yang dimiliki *PHP* adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data atau *database management sistem (DBMS)*, sehingga dapat menciptakan suatu halaman *web* dinamis. *PHP* mempunyai koneksitas yang baik dengan beberapa *DBMS* seperti *Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm*, dan tidak terkecuali semua *database* ber-*interface ODBC*.

## **2.7 CodeIgniter (CI)**

*Codeigniter* adalah kumpulan intruksi-intruksi yang di kumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar terciptanya *source code* yang bersih dan terstruktur. *MVC* adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, *view* dan *controller*.

*CodeIgniter* adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *web*. Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai

macam *library* yang dapat mempermudah dalam pengembangan. *CodeIgniter* sendiri dibangun menggunakan konsep *ModelView-Controller development pattern* (Syarifudin, 2018).

## **2.8 Hypertext Markup Language (HTML)**

*HTML* atau singkatan dari *Hypertext Markup Language* ini merupakan sebuah Bahasa markup pada *internet* khususnya *web* berupa kode dan simbol. Kode dan simbol tersebut nantinya akan ditempatkan kedalam sebuah *file* dengan tujuan untuk dimunculkan pada sebuah *website*. Fungsi dari *HTML* sendiri adalah digunakan untuk membuat sebuah *website*. *HTML* ini diibaratkan sebagai sebuah pondasi dari sebuah *website*. Karena jika sebuah *website* tidak memiliki *HTML* maka para pembuat *website* tidak dapat untuk menggunakan bahasa markup lainnya seperti *CSS* yang digunakan untuk mendesain *Website*, lalu *Javascript*, ataupun *PHP* (Arida, 2018).

## **2.9 MySql**

*MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan *structure* data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan basis data. *MySQL* adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan (Enterprise, 2016). *MySQL* merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

## 2.10 Konsep OOP (*Object Oriented Programming*)

*OOP* merupakan cara berpikir, pandangan atau paradigma baru untuk membuat program atau merancang sistem dengan memperhatikan objek, ciri objek dan perilakunya. *OOP* ini sangat berbeda dengan membuat program prosedural yang memperhatikan aspek *input*, *proses* dan *output*. (Rossa, 2013).

Terdapat beberapa konsep mengenai *OOP* yaitu:

### 1. Kelas Abstrak (*Class Abstraksi*)

Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data. Kelas dapat diilustrasikan sebagai suatu cetak biru (*blueprint*) atau *prototipe* yang digunakan untuk menciptakan objek. Kelas digunakan untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

### 2. Enkapsulasi (*Encapsulation*)

Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

### 3. Pewarisan (*Inheritance*)

Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada. Penurunan sifat ini bisa dilakukan secara bertingkat-

tingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kode-kode.

#### 4. Polimorfisme (*Polymorphism*)

Polimorfisme merupakan kemampuan objek-objek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama. *Polimorfisme* juga dapat dikatakan kemampuan sebuah objek untuk memutuskan *method* mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.


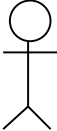

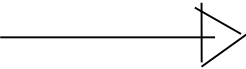
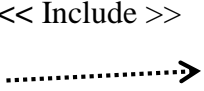
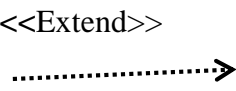
### 2.11 UML (*unified Modelling Language*)

Menurut (Rosa & Salahuddin, 2013), *UML (unified Modelling Language)* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan *teks-teks* pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *UML (Unified Modelling Language)*.

#### 2.11.1 Use Case Diagram

Menurut (Rosa & Salahuddin, 2014) *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.2 berikut ini :

**Tabel 2.2** Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		Usecase Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi ( <i>generalization</i> ) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi ( <i>extend</i> ) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.



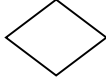

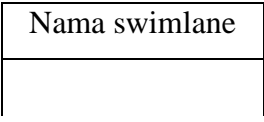

Sumber : (Rosa & Salahuddin, 2014)

### 2.11.2 Activity Diagram

Menurut (Rosa & Salahuddin, 2013) *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini :



**Tabel 2.3** Simbol *Activity Diagram*

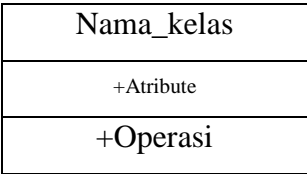


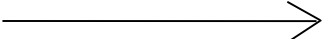
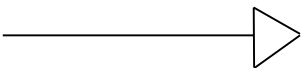
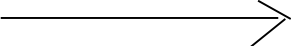
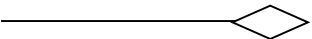
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		Penggabungan ( <i>Join</i> ) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		<i>Swimlane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber : (Rosa & Salahuddin, 2014)

### 2.11.3 *Class Diagram*

Menurut (Rosa & Salahuddin, 2013) *Class Diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4 berikut ini :

Tabel 2.4 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Asociation</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / dependency</p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / aggregation</p> 	Relasi antar kelas dengan maksna semua bagian ( <i>whole-part</i> )

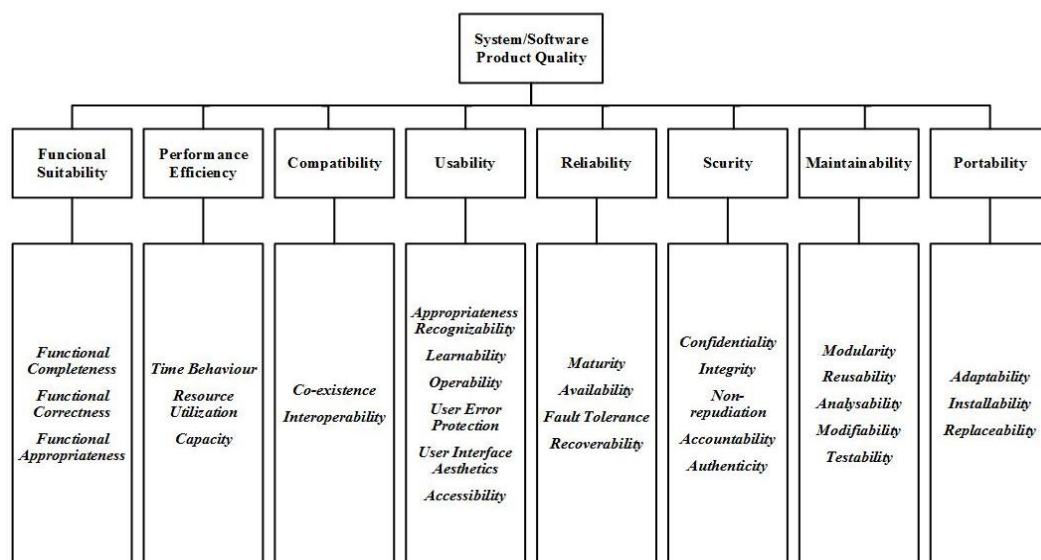
Sumber: (Rosa & Salahuddin, 2014)

## 2.12 Kualitas Perangkat Lunak

Rosa, A. S dan Shalahuddin (2014) menyatakan bahwa sebuah perangkat lunak perlu dijaga kualitasnya. Kualitas ini sangat mempengaruhi kepuasan pelanggan. Selain itu kualitas perangkat lunak perlu dijaga agar dapat bertahan hidup, dapat bersaing dengan perangkat lunak lain, dapat bersaing dalam hal pemasaran, efektif dalam biaya pengembangan, dan mempertahankan pelanggan.

Sehingga, dengan terjaganya kualitas aplikasi maka dapat meningkatkan keuntungan. Pengujian perangkat lunak adalah elemen penting dari jaminan kualitas dan mempresentasikan spesifikasi, desain dan pengkodean (Pressman, 2010). Sering perangkat lunak mengandung kesalahan (*error*) pada proses-proses tertentu pada saat perangkat lunak sudah berada ditangan pengguna. Kesalahan-kesalahan pada perangkat lunak ini sering disebut dengan *bug*. Untuk menghindari banyaknya *bug* maka diperlukan pengujian perangkat lunak sebelum perangkat lunak didistribusikan ke pelanggan atau selama perangkat lunak masih dalam pengembangan (Rosa, A. S & Shalahuddin, 2014).

Salah satu tolak ukur kualitas perangkat lunak menggunakan *ISO 25010*, yang dibuat oleh *International Organization for Standarization (ISO)* dan *International Electrotechnical Commission (IEC)* (Wagner, 2013: 2). *ISO 25010* hadir untuk memperbarui versi sebelumnya yaitu standar *ISO/IEC 9126* (ISO, 2011). Model kualitas produk (*ISO, 2011*) mengkategorikan sifat kualitas produk menjadi delapan karakteristik yaitu *Functional Suitability, Reliability, Performance Efficiency, Usability, Security, Compatibility, Maintainability, dan Portability*.



**Gambar 2.2** Karakteristik ISO 25010

### **1. *Functional Suitability***

*Functional Suitability* merepresentasikan sudah sejauh mana fungsi-fungsi pada perangkat lunak atau sistem telah memenuhi kebutuhan pengguna yang direncanakan dan diimplementasikan ketika dipergunakan pada suatu keadaan tertentu. Pada karakteristik *Functional Suitability* sendiri dijabarkan menjadi 3 (tiga) subkarakteristik yaitu :

- 1) *Functional Completeness* adalah karakteristik yang mengatur fungsi-fungsi tersebut mencakup seluruh task yang telah ditentukan.
- 2) *Functional Correctness* adalah karakteristik perangkat lunak atau sistem yang dikembangkan mampu memberikan hasil yang presisi.
- 3) *Functional Appropriateness* adalah tingkat dimana fungsi dapat memberikan fasilitas untuk mencapai tugas dan tujuan tertentu.

### **2. *Performance Efficiency***

*Performance Efficiency* merepresentasikan performa relatif terhadap kuantitas sumber daya atau resources yang dipergunakan pada suatu keadaan. *Performance Efficiency* sendiri dijabarkan dalam 3 (tiga) subkarakteristik atau tingkatan sebagai berikut :

- 1) *Time Behaviour*, adalah tingkat dimana respon dan waktu proses serta tingkat output suatu perangkat lunak atau sistem ketika fungsinya berjalan dapat memenuhi persyaratan.
- 2) *Resource Utilization* (Pemanfaat Sumber Daya), yaitu tingkat kuantitas serta dan sumber daya (*resources*) yang dipergunakan oleh suatu perangkat lunak atau sistem ketika menjalankan fungsinya dapat memenuhi persyaratan.

- 3) *Capacity*, adalah tingkatan dimana batasan maksimum produk atau parameter sistem dapat menyesuaikan persyaratan.

### 3. *Compatibility*

Yaitu tingkatan perangkat lunak, sistem atau komponen mampu saling berbagi informasi terhadap perangkat lunak, sistem maupun komponen lainnya, dan atau menjalankan fungsinya yang dibutuhkan, sedangkan sistem tetap dapat saling berbagi informasi pada hardware maupun software di suatu lingkungan yang sama.

- 1) *Co-existence*, tingkatan perangkat lunak mampu menjalankan kebutuhan fungsi dengan efisien, sementara sistem sedang berbagi *resources* atau sumberdaya terhadap produk lainnya, tanpa memberikan dampak yang memberikan kerugian pada produk lainnya.
- 2) *Interoperability*, tingkatan dimana dua atau lebih suatu sistem, perangkat lunak atau komponen dapat berbagi informasi satu sama lain dan dapat memakai kembali informasi yang telah ditukar sebelumnya.

### 4. *Usability*

Yaitu tingkatan dimana perangkat lunak atau sistem memungkinkan untuk digunakan oleh suatu pengguna/*user* tertentu untuk menggapai sebuah tujuan tertentu dengan efektivitas, efisiensi, serta kepuasan dalam *konteks* penggunaan tertentu. Karakteristik ini dijabarkan menjadi beberapa subkarakteristik sebagai berikut :

- 1) *Appropriateness Recognizability*, sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.

- 2) *Learnability*, tingkat suatu perangkat lunak atau sistem memungkinkan untuk dioperasikan oleh pengguna tertentu untuk mencapai tujuan dari pembelajaran.
- 3) *Operability*, adalah tingkatan suatu produk atau sistem mempunyai atribut yang membuat mudah dioperasikan.
- 4) *User Error Protection*, yaitu tingkat dimana sistem mampu memberikan perlindungan kepada pengguna dari kesalahan atau *error*.
- 5) *User Interface Aesthetics*, adalah tingkat dimana *user interface* atau antarmuka pengguna mampu memberikan interaksi yang memberikan kesenangan dan memberi kepuasan bagi pengguna.
- 6) *Accessibility*, yaitu tingkat suatu perangkat lunak atau sistem dapat dioperasikan oleh pengguna dalam jangkauan karakteristik dan kemampuan terluas untuk dicapai pada tujuan dan dalam *konteks* penggunaan tertentu.

## 5. **Reliability**

Yaitu tingkatan dimana suatu perangkat lunak/sistem atau komponen mampu melakukan suatu fungsi tertentu dalam situasi tertentu serta untuk jangka waktu tertentu. Karakteristik ini terdiri dari 3 (tiga) sub-karakteristik berikut :

- 1) *Maturity*, yaitu tingkat dimana perangkat lunak atau sistem dapat memenuhi kebutuhan akan *reliabilitas* di bawah operasi normal.
- 2) *Availability*, yaitu tingkat dimana produk atau sistem dapat beroperasi dan diakses ketika diperlukan untuk digunakan.
- 3) *Fault Tolerance* Sejauh mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun terdapat kesalahan pada perangkat keras atau perangkat lunak

- 4) *Recoverability*, yaitu tingkat dimana produk atau sistem ketika interupsi atau kegagalan terjadi, perangkat lunak atau sistem mampu mengembalikan data yang terkena dampak dan membangun kembali seperti semula.

## **6. Security**

Yaitu tingkatan perangkat lunak atau sistem yang mampu memberikan perlindungan informasi dan data karakteristik ini terdiri dari sub karakteristik berikut :

- 1) *Confidentiality*, tingkatan perangkat lunak atau sistem mampu memberikan kepastian bahwa data hanya dapat diakses kepada pengguna yang memiliki wewenang untuk menggunakannya.
- 2) *Integrity*, tingkatan sistem, perangkat lunak atau komponen mampu memberikan pencegahan terhadap akses ilegal untuk mengakses suatu data rahasia.
- 3) *Non-repudiation*, tingkat tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan pada peristiwa atau tindakan.
- 4) *Accountability*, tingkatan tindakan dari suatu entitas dapat dilacak secara unik pada suatu entitas tersebut.
- 5) *Authenticity*, tingkat identitas subjek atau sumber daya dapat dibuktikan menjadi salah satu yang dapat diklaim.

## **7. Maintainability**

Karakteristik ini mewakili tingkatan efektifitas dan efisiensi suatu sistem atau perangkat lunak agar dapat dilakukan *modifikasi* dalam rangka untuk memperbaikinya, serta menyesuaikannya apabila terjadi perubahan lingkungan,

dan dalam kebutuhan. Karakteristik ini terdiri dari beberapa sub-karakteristik berikut:

- 1) *Modularity* yaitu tingkatan dimana sistem atau perangkat lunak terdiri dari sekumpulan komponen yang terpisah sehingga apabila terjadi perubahan pada suatu komponen, akan meminimalisir dampak pada komponen lain.
- 2) *Reusability* yaitu tingkat penggunaan kembali suatu aset pada satu sistem, atau penggunaan untuk membangun properti lainnya.
- 3) *Analysability* yaitu tingkatan keefektifan serta efisiensi yang memberikan kemungkinan untuk mengukur dampak pada perubahan satu atau lebih bagian perangkat lunak atau sistem, mencari penyebab *error* pada suatu produk, serta mengidentifikasi suatu bagian yang akan diubah.
- 4) *Modifiability* yaitu tingkat dimana suatu perangkat lunak atau sistem dapat dilakukan modifikasi dengan efektif serta efisien tanpa memberikan suatu cacat maupun mengurangi kualitas produk yang dikembangkan.
- 5) *Testability* yaitu tingkat efektivitas dan efisiensi pada kriteria pengujian dapat dibangun untuk perangkat lunak, serta pengujian dapat dilakukan untuk menentukan apakah kebutuhan atau kriteria tersebut telah terpenuhi.

#### **8. *Portability***

Tingkatan efektivitas serta efisiensi dimana suatu sistem atau perangkat lunak dapat ditransfer dari suatu perangkat keras, perangkat lunak, atau lingkungan operasional yang berbeda.

- 1) *Adaptability* yaitu tingkat dimana suatu perangkat lunak atau sistem mampu beradaptasi secara efektif dan efisien untuk perangkat keras, perangkat



lunak, atau lingkungan operasional serta penggunaan lain yang berbeda atau sedang dikembangkan.

- 2) *Installability* yaitu tingkatan efektivitas dan efisiensi dimana perangkat lunak atau keberhasilan sistem dapat dilakukan pemasangan dan atau pelepasan pada sebuah lingkungan tertentu.
- 3) *Replaceability* yaitu tingkat di mana suatu perangkat lunak mampu menggantikan perangkat lunak lain yang telah ditentukan untuk tujuan yang sama dalam lingkungan yang sama.