

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

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

No		Detail Jurnal
1	Judul	Aplikasi Tingkat Kepuasan Mahasiswa Terhadap Layanan Administrasi Fakultas Teknik Universitas Muhammadiyah Bengkulu
	Penulis	(Andialala&Marhalim, 2019)
	Metode Penelitian	Pengembangan Sistem
	Hasil	Hasil dari penelitian tersebut adalah Aplikasi Untuk mengukur Tingkat Kepuasan Mahasiswa terhadap Layanan Administrasi yang dapat dijadikan tolak ukur dalam tatakelola Perguruan Tinggi terutama tatakelola fakultas teknik dalam hal melakukan pelayanan terhadap mahasiswa
2	Judul	Aplikasi Evaluasi Layanan Mahasiswa Studi Kasus STIMIK Bandung
	Penulis	(Mina Ismu Rahayu et al., 2021)
	Metode Penelitian	Prototyping
	Hasil	Hasil dari penelitian tersebut adalah Aplikasi Evaluasi Layanan Mahasiswa terhadap pembelajaran maupun layanan administrasi sehingga dapat membantu perguruan tinggi dalam melakukan evaluasi layanan bagi mahasiswa
3	Judul	Perancangan Aplikasi Pengukuran Kepuasan Mahasiswa Terhadap Sarana Prasarana Kampus Universitas Dian Nuswantoro Menggunakan Metode Waterfall

	Penulis	(Nur Rokhman, 2019)
	Metode Penelitian	Metode Waterfall
	Hasil	Hasil Penelitian ini Aplikasi ini dirancang menggunakan metode waterfall yang memberikan output/keluaran berupa visual grafik yang menarik dan mudah dipahami mengenai data penilaian kepuasan civitas akademik terhadap sarana prasarana
4	Judul	Perancangan Digital Kuisisioner Kepuasan Mahasiswa Menggunakan Metode Waterfall
	Penulis	(Budiarto, 2019)
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil	Hasil Penelitian ini Rancangan dan Desain Digital Kuisisioner Kepuasan Mahasiswa menggunakan metode <i>waterfall</i> memberikan <i>output/keluaran</i> berupa grafik tingkat kepuasan mahasiswa dari aspek akademik dan sarana prasarana yang disediakan oleh STIKOM PGRI Banyuwangi
5	Judul	Pengembangan Aplikasi Sistem Informasi Layanan Akademik Berbasis CodeIgniter
	Penulis	(Mohammad Syahidul Haq & Khofidotur Rofiah, 2021)
	Metode Penelitian	Metode <i>Research and Development (R&amp;D)</i>
	Hasil	Hasil dari penelitian ini adalah sistem informasi manajemen yang dilakukan dengan memanfaatkan sistem informasi. Dengan adanya E-Layanan dapat membantu aktivitas layanan akademik terutama pada masa pandemi Covid-19. E-Layanan yang dikembangkan menggunakan sistem informasi manajemen yang berbasis Codeigniter oleh Fakultas Ilmu Pendidikan Unesa sangat berguna dan layak untuk digunakan

**Tabel 2. 1: Daftar Literatur**

### **2.1.1. Tinjauan Pada Literatur 1**

Pada Literatur 1 ini meneliti tentang sebuah Aplikasi Tingkat Kepuasan Mahasiswa Terhadap Layanan Administrasi Fakultas Teknik Universitas Muhammadiyah Bengkulu yang menggunakan Metode Pengembangan Sistem dalam penelitiannya. Tujuan dibuatnya aplikasi ini adalah untuk membuat aplikasi yang dapat dijadikan tolak ukur dalam tatakelola institusi terutama tatakelola fakultas teknik dalam hal melakukan pelayanan terhadap mahasiswa, objek dalam penelitian ini adalah mahasiswa aktif fakultas teknik yang terdiri dari program studi informatika dan program studi sistem informasi. Untuk mengetahui mutu pelayanan yang diberikan kepada mahasiswa adalah dengan cara melakukan evaluasi, salah satu cara evaluasi yang dapat dilakukan adalah dengan melakukan pengukuran tingkat kepuasan mahasiswa terhadap layanan yang diberikan oleh fakultas terutama oleh program studi dan tata usaha.

Perbedaan penelitian yang akan diteliti dengan Literatur 1 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti oleh penulis menggunakan metode Extream Programming sebagai metode penelitian, karena metode extream programming merupakan sebuah metode pengembangan perangkat lunak yang ringan dengan mengimplementasikan agile yang lebih cenderung menggunakan pendekatan berorientasi objek serta lebih mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan pengguna.

### **2.1.2. Tinjauan Pada Literatur 2**

Pada Literatur 2 ini meneliti tentang sebuah aplikasi Evaluasi Layanan Mahasiswa Studi Kasus STIMIK Bandung yang menggunakan metode *Prototyping* dalam penelitiannya. Tujuan dari penelitian ini untuk memenuhi kebutuhan pengolahan data evaluasi kuisisioner, dan untuk mempermudah dalam melakukan

analisis terhadap kepuasan layanan suatu institusi perguruan tinggi bagi mahasiswa.

Perbedaan penelitian yang akan diteliti dengan Literatur 2 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti oleh penulis menggunakan metode Extream Programming sebagai metode penelitian, karena selain ringan dan mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan pengguna, Extream Programming merupakan metode yang cepat, efisien, beresiko rendah, fleksibel, scientific, dan menyenangkan, selain itu model ini juga dapat meminimalisir biaya yang diperlukan jika terdapat perubahan kebutuhan fitur dalam pengembangan perangkat lunak.

### **2.1.3. Tinjauan Pada Literatur 3**

Pada Literatur 3 ini meneliti tentang sebuah Perancangan Aplikasi Pengukuran Kepuasan Mahasiswa Terhadap Sarana Prasarana Kampus Universitas Dian Nuswantoro Menggunakan Metode Waterfall yang menggunakan metode waterfall dalam penelitiannya. Tujuan dari penelitian ini yakni merancang prototype aplikasi untuk mengetahui seberapa besar tingkat kepuasan mahasiswa pada sarana prasarana yang ada pada tiap prodi dan fakultas di Universitas Dian Nuswantoro.

Perbedaan penelitian yang akan diteliti dengan Literatur 2 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti oleh penulis menggunakan metode Extream Programming sebagai metode penelitian, karena selain ringan dan mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan pengguna, Extream Programming merupakan metode yang cepat, efisien, beresiko rendah, fleksibel, scientific, dan menyenangkan, selain itu model ini juga dapat meminimalisir biaya yang diperlukan jika terdapat perubahan kebutuhan fitur dalam pengembangan perangkat lunak.

#### **2.1.4. Tinjauan Pada Literatur 4**

Pada Literatur 4 ini meneliti tentang sebuah Perancangan Digital Kuisisioner Kepuasan Mahasiswa yang menggunakan Metode Waterfall dalam penelitiannya. Tujuan dari penelitian ini adalah menjadikan salah satu alternative untuk mengukur kualitas layanan sehingga akan diperoleh nilai gap (kesenjangan) yang menjadi perbandingan antara persepsi mahasiswa terhadap layanan yang diterima dengan harapan mahasiswa untuk perbaikan yang akan datang.

Perbedaan penelitian yang akan diteliti dengan literatur 4 terdapat pada studi kasus dan permasalahan yang ada. Permasalahan yang akan diteliti tentang bagaimana cara membangun Aplikasi Penilaian Mutu Pelayanan berbasis website menggunakan di Universitas XYZ.

#### **2.1.5. Tinjauan Pada Literatur 5**

Pada Literatur 5 ini meneliti tentang Pengembangan Aplikasi Sistem Informasi Layanan Akademik Berbasis CodeIgniter, yang menggunakan metode *Research and Development* (R&D), dimana metode tersebut merupakan metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tertentu. Tujuan dari penelitian ini ialah untuk mengembangkan suatu layanan pendidikan untuk memudahkan mahasiswa dalam mengakses segala kebutuhan mengenai surat menyurat dan dinamakan dengan sistem informasi layanan akademik Fakultas Ilmu Pendidikan Unesa.

Perbedaan penelitian yang akan diteliti dengan Literatur 5 terdapat pada metode yang dipakai, pada penelitian yang akan diteliti oleh penulis menggunakan metode Extream Programming sebagai metode penelitian, karena metode extream programming merupakan sebuah metode pengembangan perangkat lunak yang ringan dengan mengimplementasikan agile yang lebih cenderung menggunakan

pendekatan berorientasi objek serta lebih mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan pengguna.

## **2.2. Aplikasi Web**

Aplikasi web adalah sebuah program atau sistem informasi melalui antarmuka berbasis web yang dijalankan melalui browser seperti Internet Explorer, Mozilla, Opera dan aplikasi browser lainnya (Agus Suheri, 2022). Dapat disimpulkan bahwa Aplikasi web adalah sebuah media penyampaian informasi yang dapat diakses melalui browser.

## **2.3. Pengertian Kuisisioner**

Kuisisioner adalah salah satu teknik yang dipakai untuk media pengumpulan informasi menggunakan analisis yang mempelajari sikap-sikap, keyakinan, perilaku dan karakteristik beberapa orang utama di dalam organisasi yang dapat terpengaruh oleh sistem yang diajukan atau oleh sistem yang sudah ada sehingga analisis dapat mengukur apa yang ditemukan dalam wawancara dan menentukan seberapa luas atau terbatasnya sentiment yang diekspresikan dalam suatu wawancara (Mina Ismu Rahayu et al., 2021).

## **2.4. Pengertian Evaluasi**

Evaluasi merupakan sebuah prosedur yang digunakan sebagai penentuan subjek apakah memenuhi kualifikasi sesuai dengan kriteria yang telah ditetapkan. Dapat diartikan bahwa Evaluasi adalah proses pengukuran serta perbaikan yang digunakan sebagai bahan untuk membandingkan berbagai hasil akhir dari kegiatan yang dibuat (Mina Ismu Rahayu et al., 2021).

## **2.5. Pengertian Mutu Layanan**

Mutu layanan adalah suatu aktivitas pemberian layanan(jasa) berupa pemenuhan segala kebutuhan akademik yang mampu mencukupi harapan mahasiswa sebagai pelanggan di perguruan tinggi (KhoirulAzan, 2016).

Dapat disimpulkan bahwa Mutu Layanan adalah kualitas dari setiap tindakan atau kinerja pemberian layanan berupa pemenuhan segala kebutuhan baik akademik maupun non-akademik yang mencakup harapan mahasiswa.

## **2.6. Layanan Mahasiswa**

Layanan mahasiswa memiliki tujuan untuk mengetahui tingkat kepuasan mahasiswa terhadap layanan kemahasiswaan dan fasilitas yang dimiliki oleh perguruan tinggi yang digunakan sebagai tolak ukur penunjang perbaikan mutu layanan. Pada pelaksanaannya, kuisioner layanan mahasiswa dilakukan setiap akhir semester atau setiap satu tahun sekali, dan data yang diperoleh dapat dijadikan sebagai acuan perguruan tinggi untuk memperbaiki kualitas pelayanan mahasiswa dari sisi internal yang nantinya dijadikan sebagai kontribusi dalam mempertahankan serta memperbaiki akreditasi perguruan tinggi (Mina Ismu Rahayu et al., 2021).

## **2.7. Agile Software Development Methods**

*Agile Software Development Methods* atau *Agile Methodology* merupakan sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iterative, dimana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir (Pressman, 2010). Berikut beberapa metode Agile yang berkembang saat ini:

- a. Extreme Programming
- b. SCRUM

- c. Adaptive Software Development (ASD)
- d. Dynamic Systems Development Method (DSDM)
- e. Feature Driven Development (FDD)
- f. Crystal Light Methodology Family
- g. Pragmatic Programming
- h. Open Source Software Development (OSSD)

## 2.8. Metode Extreme Programming (XP)

Menurut (Supriyatna, 2018), *Extreme Programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek.

Tahapan-tahapan dari *Extreme Programming* terdiri dari

### ➤ *Planning* (Perencanaan)

Tahap ini merupakan langkah awal dalam pembangunan sistem dimana pada tahapan ini terdapat beberapa kegiatan yang berupa perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

### ➤ *Designing* (Perancangan)

Pada tahap kedua ini yaitu perancangan dimana pada tahap 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). Adapun alat yang sering digunakan pada tahap ini adalah *Class Responsibility Collaborator* (CRC). CRC dapat digunakan untuk



proses membangun kelas yang akan digunakan pada *use case diagram*, *class diagram* dan objek.

➤ *Coding* (Pengkodean)

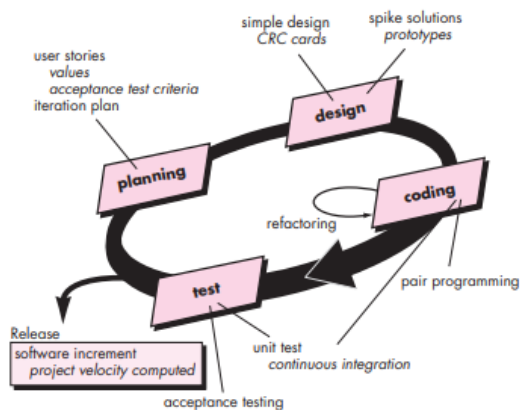
Pada Tahap ketiga ini yang merupakan tahap utama dalam pengembangan aplikasi dengan XP yaitu *pair programming*.

➤ *Testing* (Pengujian)

Tahap keempat yang merupakan tahap terakhir berfokus pada pengujian fitur untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna.

## 2.9. Tahapan Metode Extreme Programming (XP)

Dalam metode *Extreme Programming* (XP), pengembang harus mengikuti langkah-langkah yang ada pada metode ini, terdapat empat langkah yang harus diikuti untuk dapat menyelesaikan proses pengembangan :



**Gambar 2. 1 : Tahapan Metode Extreme Programming**

### a. *Planning* (Perencanaan)

Pada tahap *planning* atau perencanaan merupakan langkah awal yang diperlukan dalam pembangunan sistem, dimulai dengan melakukan perencanaan atau analisa kebutuhan pengguna. Pengembang harus dapat menyimpulkan hasil dari *story* oleh

pengguna yang mendeskripsikan output, fitur, dan fungsi dari aplikasi yang akan dibangun, adapun *story-story* yang dilakukan dengan cara:

1. Pengguna menceritakan permasalahan yang dihadapi dan aplikasi seperti apa yang akan dibangun.
2. Berdasarkan hasil yang didapat dari cerita pengguna maka peneliti menentukan poin pada bagian *value* untuk selanjutnya dapat memutuskan sistem seperti apa dan apa saja yang akan dibangun.
3. Selanjutnya dari hasil tersebut maka peneliti menentukan *acceptance test criteria* atau kebutuhan apa saja yang nantinya menjadi acuan terhadap sistem yang akan diuji.
4. Sehingga peneliti dapat menyimpulkan *release* akan dilakukan berapa kali

**b. Design (Perancangan)**

Setelah tahap pertama selesai, tahap selanjutnya yaitu *design* atau perancangan. Pada tahap ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data

1. *Simple Design* adalah pengembang membuat desain yang sederhana, dimulai dengan menggunakan UML (*Unified Modelling Language*) seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.
2. CRC (*Class Responsibility Collaborator*) digunakan untuk mengenali dan mengatur *object oriented class* yang sesuai, jika terdapat kesalahan atau ketidaksesuaian maupun perbaikan pada saat perancangan maka akan dilakukan.
3. *Prototype* adalah perancangan seperti *user interface* dalam bentuk *wireframing* untuk mempermudah pengguna melihat desain dari sistem yang akan dibangun.

**c. Coding (Pengkodean)**

Pada tahap pengkodean ini peneliti menyesuaikan dari *story* yang sebelumnya sudah diberikan oleh pengguna sehingga sistem yang dibangun sesuai, proses yang dilakukan antara lain:

1. *Pair Programming* yaitu tahap sistem dibangun dengan menggunakan bahasa pemrograman dan media penyimpanan yang sudah direncanakan.
2. *Refactory* yaitu tahap yang akan dilakukan jika terdapat ketidak sesuaian pada kode program yang digunakan untuk perbaikan guna mendapatkan hasil yang sesuai.

**d. Testing (Pengujian)**




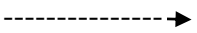
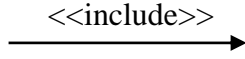
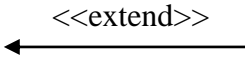
Tahap yang terakhir *Tesing* atau pengujian yang dilakukan oleh pengguna sebagai *user* dengan melakukan pengujian sesuai dengan yang telah ditentukan dan disetujui. Unit test yang dibuat focus pada fitur dan fungsional sistem. Sehingga sistem dapat disimpulkan telah sesuai dan dapat dirilis.

**2.10. Unifed Modelling Language (UML)**

UML (*Unifed Modelling Language*) adalah metode pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ secara visual sebagai sarana untuk merancang dan membuat software berorientasi objek. UML adalah salah satu tool/model yang digunakan untuk merancang pengembangan software yang berbasis objek oriented. UML sendiri memberikan standar penulisan blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software. Terdapat beberapa diagram pada UML antara lain sebagai berikut :

### 2.10.1. Use Case Diagram




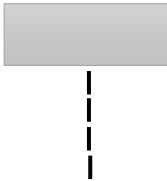



*User Case Diagram* merupakan pemodelan untuk perilaku sistem informasi yang akan dibuat, *use case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. *Use case diagram* dekat kaitannya dengan kejadian-kejadian atau *scenario* (apa yang terjadi ketika seseorang berinteraksi dengan sistem).


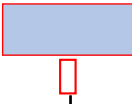
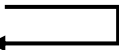
No	Nama	Symbol	Keterangan
1	Actor		Actor merupakan pengguna sistem, sistem yang berhubungan dengan sistem lain atau <i>time/waktu</i>
2	<i>Use Case</i>		<i>Use Case</i> merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya yang disediakan sistem
3	Asosiasi		Menghubungkan antara actor dengan <i>use case</i>
4	Generalisasi		Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i>
5	<i>Include</i>		Memungkinkan 1 <i>use case</i> menggunakan fungsionalitas yang disediakan oleh <i>use case</i> lain
6	<i>Extend</i>		Memungkinkan 1 <i>use case</i> secara optional menggunakan fungsionalitas yang disediakan oleh <i>use case</i> lain

**Tabel 2. 2: Simbol-simbol Use Case Diagram**

### 2.10.2. Sequence Diagram

*Sequence diagram* merupakan salah satu diagram interaction yang menjelaskan bagaimana suatu operasi itu dilakukan, diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.


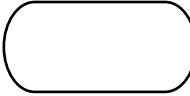



No	Nama	Symbol	Keterangan
1	Actor		Actor mewakili peran orang atau sistem yang dapat berinteraksi dengan object.
2	<i>Activation Box</i>		Menyatakan objek dalam keadaan aktif dan beriteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
3	Objek		Menyatakan objek yang berinteraksi oleh pesan.
4	<i>Lifelines</i>		Objek, <i>entity</i> , antarmuka yang menunjukkan kejadian yang terjadi secara berurutan pada sebuah objek.
5	<i>Synchronous Message</i>		Memberi isyarat bila pengirim harus menunggu respon pesan sebelum melanjutkan aktivitas.
6	<i>Asynchronous Message</i>		Memberi isyarat bila pengirim tidak harus menunggu respon pesan sebelum melanjutkan aktivitas.
7	<i>Reply Message</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi menghasilkan suatu kembalian/balasan.

8	<i>Delete Message</i>		Menghapus suatu objek
9	<i>General</i>		Menunjukkan entitas tunggal dalam sebuah <i>sequence diagram</i> , berupa <i>instance (class)</i> .
10	<i>Message Return</i>		Merupakan penggambaran hasil dari pengiriman pesan

**Tabel 2. 3: Simbol-simbol Sequence Diagram**

### 2.10.3. Activity Diagram

Activity diagram merupakan diagram yang memberikan gambaran-gambaran aktivitas yang terjadi dalam sebuah sistem secara lengkap. *Activity diagram* mendefinisikan bagaimana alur kerja dimulai, berakhir dan aktifitas apa yang terjadi serta bagaimana urutan kejadian dalam alur kerja tersebut. *Activity diagram* juga menyediakan pendekatan proses pemodelan paralel.


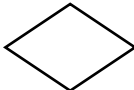
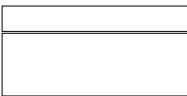


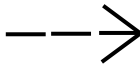
No	Nama	Symbol	Keterangan
1	<i>Activity</i>		Merupakan sebuah aktivitas yang dilakukan oleh sistem, biasanya diawali dengan kata kerja.
2	<i>Action</i>		keadaan dalam sebuah sistem yang memberikan informasi mengenai tindakan dari <i>activity</i>
3	<i>Initial Node</i>		Menunjukkan dimana aliran kerja dimulai
4	<i>Activity Final Node</i>		Menunjukkan dimana aliran kerja selesai/berakhir
5	<i>extend</i>		Untuk menunjukkan kegiatan yang dilakukan secara paralel

**Tabel 2. 4: Simbol-simbol Activity Diagram**

### 2.10.4. Class Diagram

*Class* merupakan sebuah penekanan ketika diinisialisasikan membuat objek dan merupakan pusat pengembangan dan desain berorientasi objek.

*Class Diagram* merupakan diagram yang menggambarkan struktur sistem dan mendefinisikan *class*, *package*, dan objek serta hubungan satu sama lain yang diperlukan untuk membangun sistem. *Class diagram* memiliki tiga area pokok seperti nama, atribut dan metode, adapun sifat dari atribut dan metode yaitu *private*, *protected* dan *public*. Berikut merupakan simbol-simbol yang ada pada *calss diagram*

No	Nama	Symbol	Keterangan
1	<i>Association</i>		Relasi yang menghubungkan antar objek satu dengan objek lainnya
2	<i>Nary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari dua objek
3	<i>Class</i>		Merupakan himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
4	<i>Collaboration</i>		Mendeskripsikan langkah-langkah kegiatan yang dilakukan oleh sistem, yang memberikan hasil yang terukur bagi aktor
5	<i>Realization</i>		Operasi yang sebenarnya dilakukan oleh objek
6	<i>Dependency</i>		Hubungan perubahan yang terjadi pada elemen independen akan mempengaruhi elemen yang bergantung pada elemen yang tidak independen

**Tabel 2 4 Simbol-simbol Class Diagram**

## 2.11. CodeIgniter

CodeIgniter merupakan salah satu *framework* PHP, CodeIgniter dikembangkan oleh Rick Ellis. CodeIgniter menerapkan lingkungan pengembangan dengan metode MVC (Model View Controller). MVC memisahkan antara logika pembuatan kode dengan pembuatan *template* atau pembuatan situs web.

Penggunaan metode MVC membuat proses pembuatan web menjadi lebih terstruktur dan lebih sederhana (Amelia Belinda Silviana, 2018).

## **2.12. Bootstrap**

*Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah *template* desain *web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai *HTML* dan *CSS* (Andi Christian et al, 2018)

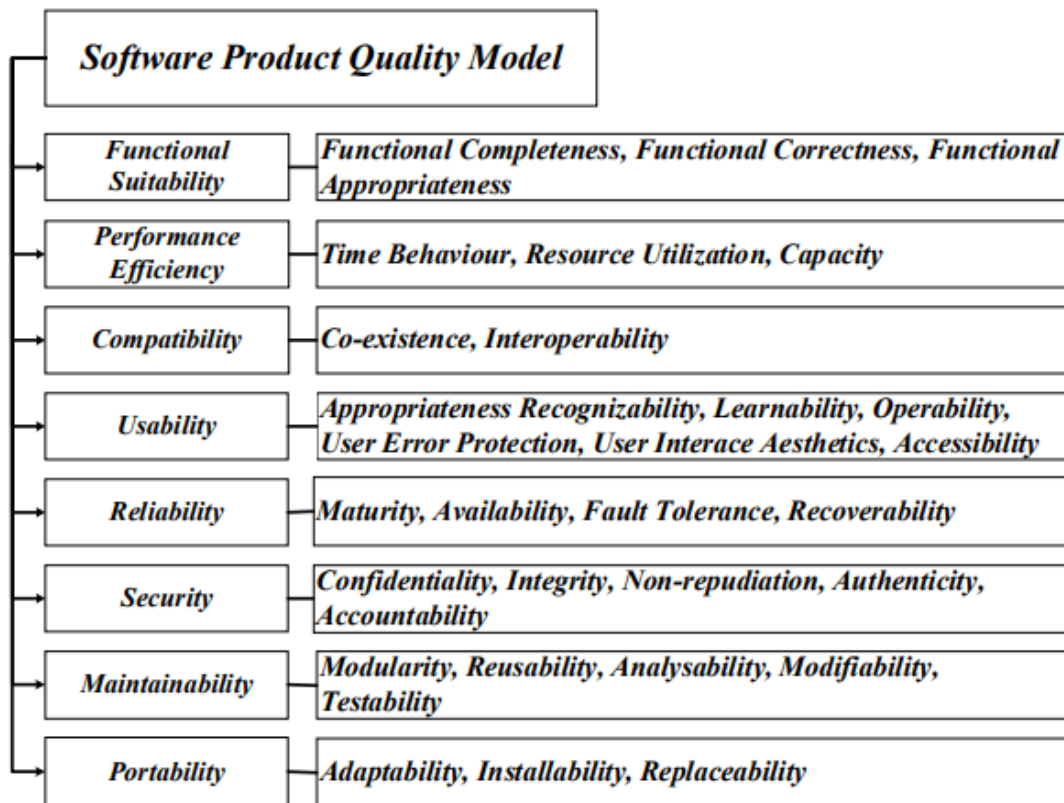
## **2.13. MySQL**

*MySQL* merupakan *software RDBMS (server database)* yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi-user)* dan dapat melakukan suatu poses secara sinkron atau berbarengan (*multi-thhreaded*) (Andi Christian et al, 2018)

## **2.14. Pegujian ISO 25010**

ISO 25010 merupakan pedoman yang digunakan untuk melakukan evaluasi perangkat lunak yang dikeluarkan oleh *Canadian Standards Association* pada tahun 2011. ISO 25010 adalah model baru dari seri ISO 250n yang merupakan pengembangan dari versi ISO 9126. Model ISO 15010 memiliki karakteristik kualitas yang dapat dipertimbangkan ketika melakukan evaluasi perangkat lunak. Pada ISO 25010 terdapat 2 model yang dapat digunakan untuk mengukur kualitas sistem, yaitu *quality in use model* dan *software product quality model* (Made Dwi Mulyawan, 2021)





**Gambar 2. 2: Karakteristik ISO 25010**

Berikut ini adalah penjelasan dari gambar diatas, delapan karakteristik ISO 25010.

Adapun penjelasannya yaitu:

1. *Functionality Suitability*, menggunakan alat penelitian seperti *test case* dengan Guttman scale. Guttman scale/skala guttman digunakan untuk memberikan jawaban unik untuk pertanyaan yang dicoba ujikan.
2. *Performance Efficiency*, dilakukan untuk menguji kinerja aplikasi yang dikembangkan.
3. *Compatibility*, merupakan kemampuan suatu komponen atau sistem untuk bertukar informasi.
4. *Usability*, untuk menganalisis umpan balik pengguna menggunakan skala 5 pilihan.
5. *Reliability*, dilakukan untuk menguji keamdalan atau keterpercayaan sistem.

6. *Security*, untuk menguji bagaimana sistem atau produk menyediakan layanan melindungi dari akses, penggunaan, modifikasi, gangguan, atau pengungkapan bahaya.
7. *Maintainability*, menggunakan alat ukur yang telah diuji oleh peneliti langsung di lapangan kegiatan, sesuai dengan alat uji yang disebutkan, pengujian ini terdiri dari 3 aspek bagian yaitu, alat ukur, perhitungan konsistensi, dan kesederhanaan.
8. *Portability*, pengujian sistem sejauh mana suatu sistem atau produk dapat dipindahkan dari suatu ruangan ke ruangan lainnya.