

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Dian Jingga Permana, 2018, dengan judul penelitian “Tingkat Kepuasan Pelanggan melalui Metode *importance Performance Analysis* di CV Meat Ball Foods Bekasi”. Penelitian ini menghasilkan di mana pelanggan masih belum puas terhadap persepsi dimensi kualitas, hal ini dapat dibuktikan berdasarkan nilai rata-rata tingkat kepuasan pelanggan sebesar $0.97 < 1$ dengan nilai gap kesenjangan kualitas sebesar $-0,13$ terhadap 30 atribut dimensi kualitas. Prioritas utama yang harus diperbaiki terutama pada atribut kepuasan harga (B1), penulisan keterangan PT (B6), logo standar SNI (B9) dan keterangan komposisi (B11).

Dina Lusianti, 2017, dengan judul penelitian “Pengukuran kepuasan peserta JKN melalui pendekatan *Importance Performance Analysis*”. Penelitian ini menghasilkan nilai indeks kepuasan Peserta JKN Cabang Pati sebesar 88.85 persen dan kemampuan petugas Kantor BPJS Kesehatan dalam memberikan kepastian waktu saat sistem mengalami kendala menjadi hal yang penting sebagai upaya peningkatan kepuasan Peserta JKN Cabang Pati.

Azizatul Hamidiyah, dkk, 2018, dengan judul penelitian “Analisis Kepuasan Ibu Hamil Terhadap Pelayanan Antenatal Menggunakan Metode *Importance Performance Analysis*”. Penelitian ini menghasilkan perhitungan gap antara rata-rata harapan dan realita menunjukkan masih adanya gap walau tidak terlalu signifikan yaitu -0.40. Hal ini menunjukkan ibu hamil belum puas terhadap pelayanan yang diberikan pada pelayanan ANC di Puskesmas Banyuputih. Hasil analisis IPA menunjukkan secara keseluruhan pada 5 dimensi *reliability, responsiveness, assurance, empathy, tangible* (43 indikator pelayanan) terdapat 7 pelayanan yang perlu ditingkatkan dan 15 kualitas pelayanan ANC yang perlu dipertahankan.

Hertiana Ikasari, 2017, dengan judul penelitian “Peningkatan Kepuasan Wisatawan Terhadap Pariwisata Kota Semarang Dengan *Importance Performance Analysis*”. Penelitian ini menghasilkan meningkatkan pelayanan di tempat wisata dan penanganan keluhan wisatawan dengan cara perbaikan kualitas SDM, salah satunya memberikan pelatihan *service excellent* kepada pengelola obyek wisata, melakukan promosi di berbagai kota tentang pariwisata kota Semarang dan *event-event* pariwisata di Kota Semarang melalui web, baliho, dll. Meningkatkan pertunjukan kesenian tradisional yang mencerminkan budaya lokal Kota Semarang untuk skala nasional dan internasional. Peningkatan pemeliharaan dan manajemen wisata budaya asli sebagai daya tarik Kota Semarang yang melibatkan seluruh *stakeholder*.

Gracecilla Aprillia Immanuel, 2020, dengan judul penelitian “Implementasi metode *Importance Performance Analysis* Untuk Pengukuran kualitas Sistem

Informasi Akademik”. Penelitian ini menghasilkan Universitas Ma Chung dapat melakukan pengembangan terhadap sistem informasi akademik yang saat ini digunakan yaitu MAC IS. Selain itu tetap mempertahankan kinerja indikator yang dianggap sudah cukup baik dalam memberikan kinerja dan sesuai dengan kepentingannya. Untuk penelitian selanjutnya untuk menambahkan atau mengembangkan variabel dan instrumen kuesioner dengan metode yang lain yang sekiranya tidak ada dalam penelitian ini. Selain itu untuk penelitian selanjutnya untuk melakukan pengukuran Sistem Informasi Akademik (SIKAD) di Universitas Ma Chung pada lingkup atau domain MAC IS lainnya.

II.2. Landasan Teori

II.2.1. Kepuasan Konsumen

Kepuasan konsumen dihasilkan dari proses perbandingan antara kinerja yang dirasakan dengan harapannya. Menurut Lovelock dan Wirtz yang dikutip oleh Fandy Tjiptono (2016:74), kepuasan konsumen adalah suatu sikap yang diputuskan berdasarkan pengalaman yang didapatkan. Sangat dibutuhkan penelitian untuk membuktikan ada atau tidaknya harapan sebelumnya yang merupakan bagian terpenting dalam kepuasan. Seorang konsumen puas atau tidak, sangat tergantung pada kinerja produk (*Perceived Performance*) dibandingkan ekspektasi konsumen bersangkutan dan apakah sang konsumen menginterpretasikan adanya deviasi atau gap di antara kinerja dan ekspektasi tersebut. Apabila kinerja lebih rendah dibandingkan ekspektasi, maka konsumen bersangkutan akan merasa tidak puas. Apabila kinerja sama dengan ekspektasi, maka ia akan puas. Sedangkan jika kinerja

melampaui ekspektasi, maka konsumen itu akan merasa sangat puas atau bahkan bahagia (*delighted*). Windasuri & Hyacintha (2016:64) menambahkan kepuasan konsumen adalah suatu tanggapan emosional pada evaluasi terhadap pengalaman konsumsi suatu produk atau jasa.

II.2.2. Metode *Importance Performance Analysis*

Importance Performance Analysis (IPA) digunakan untuk mengukur tingkat kepuasan seseorang atas kinerja pihak lain. Kepuasan seseorang tersebut diukur dengan cara membandingkan tingkat harapannya dengan kinerja yang dilakukan pihak lain. Seringkali IPA digunakan oleh perusahaan untuk mengukur kepuasan konsumennya.

Rumus

1. Menghitung nilai kinerja dan harapan

$$\text{Kinerja (X)} = ((1 \cdot \text{STS}) + (2 \cdot \text{TS}) + (3 \cdot \text{N}) + (4 \cdot \text{S}) + (5 \cdot \text{SS})) / \text{jumlah responden}$$

$$\text{Harapan (Y)} = ((1 \cdot \text{STS}) + (2 \cdot \text{TS}) + (3 \cdot \text{N}) + (4 \cdot \text{S}) + (5 \cdot \text{SS})) / \text{jumlah responden}$$

2. Tingkat Kepuasan

$$\text{Tingkat Kepuasan} = (\text{Kinerja (X)} / \text{Harapan (Y)}) * 100\%$$

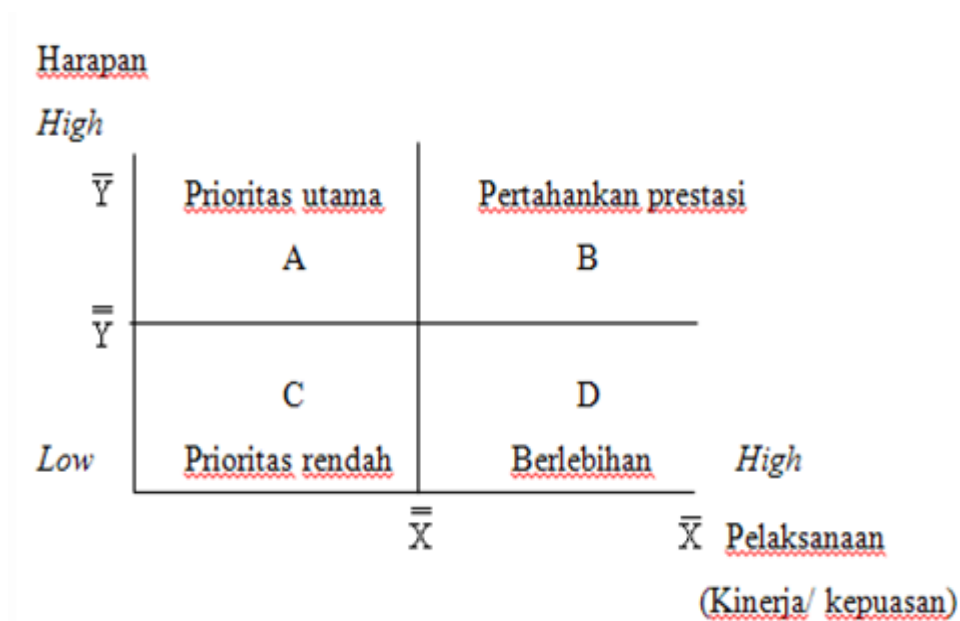
Keterangan :

Tidak Puas : Tingkat kepuasan < 1

Puas : Tingkat Kepuasan >= 1

Menurut Algifari (2016) tujuan utama *Importance Performance Analysis* (IPA) adalah sebagai alat diagnosis untuk memudahkan mengidentifikasi atribut-

atribut yang didasarkan pada kepentingan masing-masing, apakah produk atau jasa tersebut berkinerja buruk atau berkinerja berlebih. *Important Performance Analysis* ditampilkan dalam sebuah grafik (derajat kartesius) yang memiliki empat kuadran.



Gambar 2.3 Diagram Kartesius
(Sumber : Algifari : 2016)

Penjelasan mengenai keempat kuadran tersebut menurut Algifari (2016), yaitu:

1. Kuadran A menggambarkan bahwa pelanggan menganggap atribut tersebut penting, sehingga pelanggan memiliki harapan yang tinggi pada atribut tersebut, namun perusahaan tidak memberikan pelayanan terbaik pada atribut ini. Kuadran A mengisyaratkan perusahaan harus berkonsentrasi untuk memperbaiki segala kinerjanya pada atribut ini (*concentrate here*).
2. Kuadran B menggambarkan atribut yang dianggap penting bagi pelanggan sementara perusahaan sudah memberikan pelayanan yang baik kepada

pelanggan untuk atribut ini. Dengan demikian, pelayanan yang diberikan pada atribut di Kuadran B ini perlu dipertahankan (*keep up the good work*).

3. Kuadran C pada model *Importance Performance Analysis* (IPA) menggambarkan atribut di Kuadran B yang dianggap tidak penting oleh pelanggan sehingga dalam atribut Kuadran ini perusahaan perlu memberikan pelayanan dengan prioritas rendah (*low priority*).
4. Kuadran D menggambarkan wilayah di mana atribut memiliki kepentingan rendah bagi pelanggan, akan tetapi perusahaan memberikan pelayanan yang baik kepada pelanggan. Oleh karena itu, maka daerah ini disebut daerah berlebih (*possible overkill*).

II.2.3. PHP (*Hypertext Preprocessor*)

Menurut Jubille Enterprise (2017:1) PHP adalah bahasa pemrograman yang digunakan untuk membuat sistem berbasis *website*. Sebagai sebuah sistem, *website* tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis, artinya *website* tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya, menampilkan produk yang berbeda-beda untuk setiap pengunjung). Artinya interaktif *website* tersebut dapat memberi *feedback* bagi *user* (misalnya, menampilkan hasil pencarian produk).

PHP merupakan bahasa pemrograman berjenis *service-side*. Dengan demikian, PHP akan diproses oleh *server* yang hasil olahannya akan dikirim kembali ke *browser*. Oleh karena itu, salah satu *tool* yang harus tersedia sebelum memulai pemrograman PHP adalah *server*.

II.2.4. Sublime Text

Sublime text merupakan perangkat lunak *text* editor yang digunakan untuk membuat atau meng-edit suatu aplikasi. *Sublime text* mempunyai fitur *plugin* tambahan yang memudahkan *programmer* (Supono dan Putratama, 2016:14).

II.2.5. Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi. *Database* mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah:

1. Redudansi dan inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Isolasi data untuk standarisasi
4. *Multiuser*
5. Keamanan data
6. Integritas data
7. Kebebasan data (Urva dan Siregar, 2016:93).

II.2.6. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data relasional yang mengelompokkan atribut

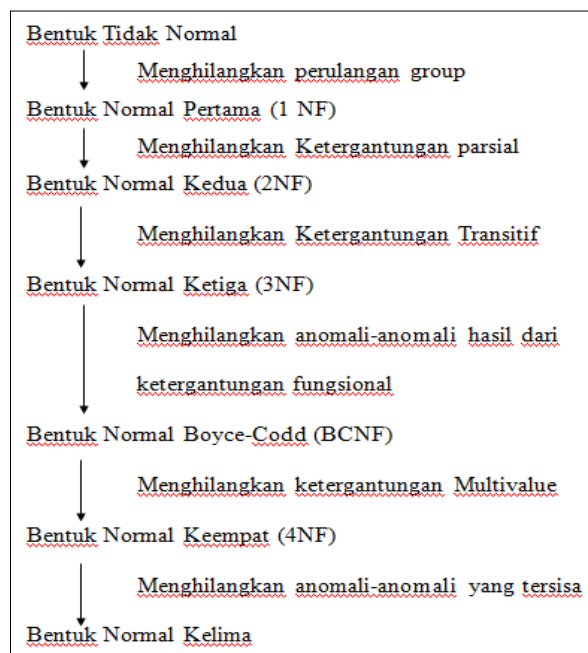
dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk:

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations(file)*.
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Adapun proses normalisasi sebagai berikut:

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat kemudian.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan. Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi tampak seperti gambar II.2.



Gambar II.1. Tahapan Normalisasi
(Sumber : Dwi Puspitasari, et al ; 2016)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan atau didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd* Normal Form (BCNF). Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF) (Dwi Puspitasari, et al : 2016).

II.2.7. MySQL

Menurut Solichin (2016:138), MySQL merupakan salah satu perangkat lunak basis data yang sangat populer. Saat ini tersedia versi MySQL yang berbayar (MySQL *Enterprise Edition*), namun tetap tersedia versi MySQL yang gratis (MySQL *Community Edition*). MySQL *Community Edition* dapat diunduh secara gratis dan bebas digunakan dalam berbagai keperluan. Walaupun demikian, secara fitur dan kemampuan, MySQL versi tidak terlalu jauh berbeda dengan versi berbayar.

Kelebihan MySQL adalah sebagai berikut:

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
3. *Super performance* dan *realible*, tidak bisa diragukan, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*easy of use*).
5. Memiliki dukungan *support (group)* penggunaan MySQL.
6. *Multiuser*, di mana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik.

II.2.8. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) digunakan sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem

perangkat lunak. Dengan adanya bahasa yang bersifat standar, komunikasi perancang dengan pemrogram (komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus. Adapun pengertian UML menurut para ahli dapat dipaparkan sebagai berikut:

Menurut Shofwan Hanief & Dian Pramana (2018 : 166) menyatakan bahwa *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak.





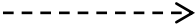
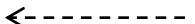
Menurut Eng RH. Sianipar (2016 :75) menyatakan bahwa UML merupakan singkatan *Unifed Modelling Language* yang telah menjadi notasi populer untuk mempresentasikan perancangan atas sebuah program berorientasi objek.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

Menurut Sri Mulyani (2016 : 245) menyatakan bahwa *Use Case Diagram* yaitu *diagram* yang menggambarkan dan merepresentasikan aktor, *use case* dan *dependencies* suatu proyek di mana tujuan dari diagram ini adalah menjelaskan konsep hubungan antara sistem dengan dunia luar. Jadi, dapat disimpulkan *use case* adalah langkah-langkah atau urutan kegiatan yang dilakukan aktor dan sistem informasi yang akan dibuat. Secara singkat, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan.

Tabel II.1. Simbol *Use Case*

| Gambar | Keterangan |
|---|--|
|  | <i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> . |
|  | Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> . |
|  | Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data. |
|  | Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem. |
|  | <i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program. |
|  | <i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |

(Sumber: Sri Mulyani: 2016: 245)

2. Diagram Aktivitas (*Activity Diagram*)




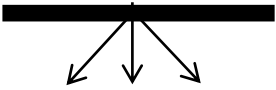

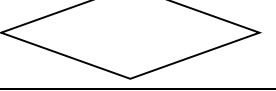

Activity diagram menggambarkan rangkaian aliran dari aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Menurut Sri Mulyani (2016: 249) *activity diagram* adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari suatu proses. Menurut Muhammad Muslihudin dan Oktafianto (2016:63) mengungkapkan diagram

aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2. Simbol Activity Diagram

| Gambar | Keterangan |
|---|---|
|  | <i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas. |
|  | <i>End point</i> , akhir aktifitas. |
|  | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | <i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi. |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> . |
|  | <i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa. |

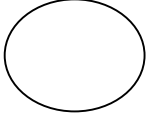
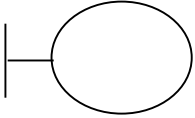
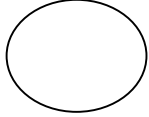

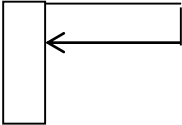


(Sumber: Sri Mulyani: 2016: 249)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Menurut (Irmayani & Susyatih, 2017), *Sequence Diagram* menggambarkan bagaimana sistem merespon kegiatan *user*. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

| Gambar | Keterangan |
|---|---|
|  | <i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
|  | <i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak. |
|  | <i>Control class</i> , suatu objek yang berisi logika sistem yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> . |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
|  | <i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | <i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |

(Sumber: Irmayani & Susyati, 2017)







4. Diagram Kelas (*Class Diagram*)

Diagram kelas (*Class Diagram*) adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang kita kembangkan. Dan berikut ini merupakan penjelasan mengenai *class diagram*:

Menurut Sri Mulyani (2016: 247) menyatakan bahwa *class diagram* adalah diagram yang digunakan untuk mempresentasikan kelas, komponen-komponen

kelas dan hubungan antara masing-masing kelas. Simbol *class diagram* dan *multiplicity class diagram* dapat dilihat pada Tabel II.4 dan Tabel II.5.:

Tabel II.4. Simbol Class Diagram

| Gambar | Keterangan | | | |
|--|--|----------|----------|--|
|  | <i>Generalization</i> , untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini. | | | |
| <table border="1" data-bbox="352 734 564 860"> <tr> <td>Nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi</td> </tr> </table> | Nama_kelas | +atribut | +operasi | <i>Class</i> , untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi. |
| Nama_kelas | | | | |
| +atribut | | | | |
| +operasi | | | | |
|  | <i>Interface</i> , untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP). | | | |
|  | <i>Association</i> , digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum. | | | |
|  | <i>Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. | | | |
|  | <i>Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian. | | | |
|  | <i>Dependency</i> , adalah relasi antar kelas dengan makna kebergantungan antar kelas. | | | |

(Sumber: Sri Mulyani: 2016: 247)

Tabel II.5. Multiplicity Class Diagram

| Multiplicity | Penjelasan |
|--------------|---|
| 1 | Satu dan hanya satu |
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4 |

(Sumber: Sri Mulyani: 2016: 247)