



BAB II
TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berikut ini beberapa penelitian yang berhubungan dengan penulisan skripsi adalah sebagai berikut :

Adapun perbedaan yang terdapat pada penelitian yang dilakukan oleh peneliti dari penelitian yang sudah ada sebelumnya oleh Ermayanti Astuti dan Nidia Enjelita Saragih (2020) dengan judul penelitian "Sistem Pendukung Keputusan Pemilihan Sekolah Terbaik dengan Metode Moora". Hasil dari penelitian ini yaitu sistem yang dibangun dapat membantu para siswa khususnya tingkat SMA. Dalam menentukan pilihan sekolah yang terjangkau dari segi ekonomis, kemampuan siswa dan siswi dalam minat belajar sesuai dengan bidangnya. Pembahasan tersebut hampir sama dengan penelitian yang peneliti lakukan, namun pada penelitian yang peneliti tulis lebih membahas mengenai penerapan metode moora untuk pemilihan sepatu *sport* terbaik.

Kemudian pada penelitian yang dilakukan Labuan Nababan dan Lamtiur Sinambela (2018) dengan judul penelitian "Sistem Pendukung Keputusan Penentuan Kelayakan Bedah Rumah Keluarga Miskin Menggunakan Metode Moora". Dengan hasil penelitian, penentuan keluarga miskin yang dapat kelayakan bedah rumahnya berdasarkan kriteria yang sudah di tentukan oleh pemerintah. Pada penelitian ini juga membahas mengenai penerapan metode moora dalam penentuan

kelayakan bedah rumah keluarga miskin sedangkan pada penelitian peneliti lebih membahas mengenai pemilihan sepatu *sport* terbaik.

Kemudian pada penelitian yang dilakukan oleh Ari Andini, et al. (2018) dengan judul penelitian "Penerapan Sistem Pendukung Keputusan Pemilihan Ban Sepeda Motor Honda dengan Metode *Multi Objective Optimizationon The Basic of Ratio Analysis (MOORA)*". Dengan hasil penelitian, dibangunnya sebuah sistem menggunakan metode *Multi Objective Optimizationon The Basic of Ratio Analysis (MOORA)* untuk memudahkan pemilik sepeda motor dalam memilih ban sepeda motor Honda yang cocok dari kriteria yang diinginkan pemilik sepeda motor seperti kriteria yang berpengaruh pada kenyamanan, keamanan dan juga kecepatan dalam berkendara. Kriteria ban sepeda motor meliputi standar kecepatan maximum yang dapat ditempuh ban, kelenturan ban dan lebar penampang ban. Perbedaan penelitian ini dengan yang peneliti lakukan terletak pada kasus yang diteliti. Hasil yang diinginkan pada penelitian ini yaitu dapat membantu konsumen Toko Gajah Mada *Fun shop* dalam pemilihan sepatu *sport* terbaik.

Kemudian pada penelitian yang dilakukan oleh Prasetyo Aji Wibawanto (2017) dengan judul penelitian "Sistem Pengambilan Keputusan Pemilihan Sepatu dengan Metode Promethee di Toko Sepatu *Stars*". Dengan hasil penelitian, sistem yang dibangun dengan menerapkan metode promethee dapat mempermudah supervisor Toko Sepatu *Stars* dalam memilih sepatu yang ingin di jual Toko Sepatu *Stars*. Pada penelitian ini juga membahas sepatu menggunakan metode *promothee*, sedangkan peneliti menggunakan metode *Moora* dalam pemilihan sepatu *sport* terbaik pada Toko Gajah Mada *Fun Shop*.

Kemudian pada penelitian yang dilakukan oleh Samuel Manurung (2018) yang berjudul “Sistem Pendukung Keputusan Pemilihan Guru Dan Pegawai Terbaik Menggunakan Metode Moora” Dimana penelitian ini untuk menentukan guru dan pegawai terbaik. Dengan menggunakan metode MOORA. Perbedaan penelitian ini dengan yang penulis lakukan terletak pada kasus yang diteliti. Hasil yang diinginkan pada penelitian ini yaitu dapat membantu Toko Gajah Mada *Fun Shop* untuk menentukan sepatu *sport* terbaik.

Kemudian pada penelitian yang dilakukan oleh Ade Septi Rezeki Anggreani Binjori, et al. (2018) yang berjudul “Sistem Pendukung Keputusan Penentuan Handphone Bekas Terbaik Menggunakan Metode *Multi-Objective Optimization on The Basis of Ratio Analysis (MOORA)*” Hasil penelitian ini menggunakan Metode MOORA untuk menentukan Handphone Bekas Terbaik. Perbedaan penelitian yang penulis lakukan dengan penelitian sebelumnya adalah dengan metode MOORA penulis akan menentukan sepatu *sport* terbaik di Toko Gajah Mada *Fun Shop* dengan sebuah aplikasi berbasis *web*.

II.2. Landasan Teori

Landasan teori sangat penting dalam sebuah penelitian terutama dalam penulisan skripsi, peneliti tidak bisa mengembangkan masalah yang mungkin di temui di tempat penelitian jika tidak memiliki acuan landasan teori yang mendukungnya.

II.2.1. Sistem

Kata Sistem berasal dari bahasa Yunani (*systema*) dan bahasa Latin (*sistema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. (Rini Asmara, 2018). Sedangkan menurut Asep Muhidin (2017) Pengertian sistem adalah seperangkat komponen yang saling berhubungan dan saling berkerjasama untuk mencapai beberapa tujuan. Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Suatu sistem mempunyai karakteristik atau sifat tertentu yang mempunyai komponen-komponen (*Components*), batas sistem (*Boundary*), lingkungan luar sistem (*Environments*), penghubung (*Interface*), masukan (*Input*), keluaran (*Output*), pengolah (*Process*) dan sasaran (*Objectives*) atau tujuan (*Goal*).

II.2.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sebuah sistem yang dimaksudkan untuk mendukung dan memberikan kemampuan pemecahan masalah, sehingga membantu dan mendukung kerja seseorang maupun kelompok dengan memberikan informasi atau usulan yang menuju pada keputusan tertentu (Alfiasea, R. dkk, 2014).

Sistem pendukung keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan (Fithry, 2012).

Sistem pendukung keputusan (SPK) bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik.

II.2.2.1. Tahapan Sistem Pendukung Keputusan

Tahap-tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut (Dyna M Khairina, dkk, 2016).

1. Tahap Pemahaman (*Intelligence Phase*) Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.
2. Tahap Perancangan (*Design Phase*) Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan atau solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.
3. Tahap Pemilihan (*Choice Phase*) Pada tahap ini dilakukan pemilihan terhadap berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan atau dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai.
4. Tahap Implementasi (*Implementation Phase*) Pada tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap

perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.

II.2.2.2. Karakteristik Sistem Pendukung Keputusan

Sistem pendukung keputusan mempunyai karakteristik sebagai berikut :

1. Mendukung pengambilan keputusan secara cepat dan tepat.
2. Menggunakan model matematis yang sesuai, model tersebut merupakan salah satu cara dalam ilmu manajemen yang digunakan untuk memecahkan masalah dengan memakai notasi dan persamaan matematika yang kemudian direpresentasikan menjadi sebuah sistem.
3. Adanya *interface* manusia dan mesin dimana manusia yang mengontrol.
4. Mempunyai kemampuan dialog.

II.2.2.3. Komponen-Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan terdiri dari 4 komponen utama yaitu sebagai berikut:

1. Subsistem manajemen data berfungsi sebagai memasukkan suatu *database* yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen *database* (DBMS). *Knowledge Base* berisi semua fakta, ide, hubungan dan interaksi suatu domain tertentu.
2. Subsistem manajemen basis pengetahuan bertugas untuk mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen. Ia memberikan intelegensi untuk memperbesar pengetahuan pengambil keputusan.

3. Subsistem manajemen model Merupakan paket perangkat lunak yang memasukkan model keuangan statistik, ilmu manajemen atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.
4. Subsistem antar muka pengguna (dialog) untuk mengimplementasikan sistem kedalam program aplikasi sehingga pengguna atau pemakai dapat bersama dengan sistem yang dirancang.

II.2.3. Metode MOORA

Metode MOORA pertama kali diperkenalkan oleh Brauers dan Zavadkas (2006). Metode ini diterapkan untuk memecahkan berbagai jenis masalah dengan perhitungan matematika kompleks. Metode yang masih baru pada Sistem Pendukung Keputusan ini memiliki tingkat fleksibilitas dan kemudahan untuk dipahami dalam memisahkan bagian subjektif dari suatu proses evaluasi ke dalam kriteria bobot keputusan dengan beberapa atribut pengambilan keputusan.

Langkah-langkah pada metode MOORA yaitu:

- **Langkah 1** : Menentukan tujuan untuk mengidentifikasi atribut evaluasi yang bersangkutan.
- **Langkah 2** : Dimulai dengan menentukan matriks keputusan dengan alternatif sebagai baris, dan kriteria sebagai kolom.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{1n} \\ x_{21} & x_{22} & x_{2n} \\ x_{m1} & x_{m2} & x_{mn} \end{bmatrix}$$

Keterangan:

X_{ij} = Nilai dari alternatif i pada kriteria j .

I = 1, 2, ..., m sebagai banyaknya alternatif.

j = 1, 2, ..., n sebagai banyaknya kriteria.

- **Langkah 3** : menentukan matriks normalisasi yang ditunjukkan pada persamaan berikut.

$$X_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Keterangan:

X_{ij} = Nilai dari alternatif i pada kriteria j .

i = 1, 2, ..., m sebagai banyaknya alternatif.

j = 1, 2, ..., n sebagai banyaknya kriteria.

x_{ij} = bilangan tidak berdimensi yang termasuk dalam interval $[0, 1]$

mewakili nilai normalisasi dari alternatif i pada kriteria j .

- **Langkah 4** : menentukan matriks normalisasi terbobot.

$$y_1 = \sum_{j=1}^g x_{ij} - \sum_{i=g+1}^n x_{ij}$$

Keterangan:

J = 1, 2, ..., g adalah jumlah tipe kriteria yang dimaksimalkan.

i = $g+1$, $g+2$, ..., n adalah jumlah tipe kriteria yang diminimalkan.

y_i = nilai dari penilaian yang telah dinormalisasi dari alternatif i terhadap semua kriteria.

x_{ij} = Nilai dari alternatif i pada kriteria j .

- **Langkah 5** : menentukan nilai preferensi atau perankingan yang dilakukan dengan cara mengurutkan nilai optimasi setiap alternatif dari nilai tertinggi ke nilai terendah. Alternatif dengan nilai optimasi tertinggi merupakan alternatif terbaik.

II.2.4. Sepatu *Sport*

Sepatu adalah salah satu jenis alas kaki yang biasanya digunakan untuk menutupi semua bagian mulai jari kaki, punggung kaki hingga bagian tumit yang difungsikan untuk melindungi kaki dari kotoran berupa debu, kerikil atau bahkan lumpur¹.

Sport atau dalam Bahasa Indonesia ialah olahraga merupakan suatu kegiatan jasmani yang dilakukan untuk memelihara Kesehatan dan memperkuat otot-otot tubuh.

Dari penjelasan di atas sepatu *sport* adalah sepatu yang fungsi utamanya dirancang untuk kegiatan olahraga atau kegiatan membentuk tubuh fisik lainnya, guna untuk kesehatan manusia.

II.2.5. PHP (*Hypertext Preprocessor*)

PHP merupakan bahasa standar yang digunakan dalam dunia *website*. PHP adalah bahasa program yang berbentuk *script* yang diletakkan di dalam *web server* (Nugroho, 2004). PHP merupakan bahasa *server-side scripting* yang menyatu dengan tag-tag HTML. Penggunaan PHP memungkinkan *Web* dapat dibuat dinamis sehingga *maintenance* situs *Web* tersebut menjadi lebih mudah dan efisien.

¹ *Wikipedia*, "Sepatu" (<https://id.wikipedia.org/wiki/Sepatu>, di akses pada 15 November 2020, 13:15)

II.2.5.1. Kelebihan PHP (*Hypertext Preprocessor*)

Adapun kelebihan PHP (*Hypertext Preprocessor*) yaitu sebagai berikut :

1. Bisa membuat *web* menjadi dinamis.
2. PHP bersifat *opensource* yang berarti dapat digunakan oleh siapa saja secara gratis.
3. Program yang dibuat dengan PHP bisa dijalankan oleh semua sistem operasi karena PHP berjalan secara *web base* yang artinya semua sistem operasi bahkan Smartphone yang mempunyai *Web Browser* dapat menggunakan program PHP.
4. Aplikasi PHP lebih cepat dibandingkan dengan ASP maupun Java.
5. Mendukung banyak paket *database* seperti *MySQL*, *Oracle*, *PostgreSQL*, dan lain-lain.
6. Bahasa pemrograman PHP tidak memerlukan Kompilasi / *Compile* dalam penggunaannya.
7. Banyak *Web Server* yang mendukung PHP seperti *Apache*, *Lighttpd*, *IIS* dan lain-lain.
8. Pengembangan Aplikasi PHP mudah karena banyak Dokumentasi, Refrensi dan *Developer* yang membantu dalam pengembangannya.

II.2.5.2. Kekurangan PHP (*Hypertext Preprocessor*)

Adapun kekurangan PHP (*Hypertext Preprocessor*) yaitu sebagai berikut :

1. PHP tidak mengenal *Package*.
2. Jika tidak di *encoding*, maka kode PHP dapat dibaca semua orang dan untuk *encodingnya* dibutuhkan *tool* dari *Zend* yang mahal sekali biayanya.

3. PHP memiliki kelemahan keamanan. Jadi Programmer harus jeli dan berhati-hati dalam melakukan pemrograman & Konfigurasi PHP.

II.2.6. Database (Basis Data)

Database atau basis data adalah kumpulan data yang saling berhubungan. Istilah tersebut bisa digunakan pada sistem yang berelasi dan terkomputerisasi. Dalam pengertian umum, *database* diartikan sebagai gabungan dari elemen-elemen data yang berhubungan dengan terorganisir (Rita, dkk, 2018).

Saat ini pengolahan *database* banyak digunakan untuk memasukkan data (*input*), keluaran data (*output*), ubah data (*edit*), pencarian data serta penyajian laporan data yang cepat dan akurat dibutuhkan suatu sistem komputerisasi pendidikan maupun dunia non-pendidikan (Effrida Manalu, dkk, 2017).

II.2.7. AppServ

Appserv salah satu *Server Web* dalam membangun *Website*. *Appserv* adalah sebuah aplikasi *Web server* lokal yang terdiri dari *Apache*, *My SQL*, *PHP*, dan *PHP My Admin*. *Appserv* merupakan sebuah aplikasi *Open Source* yang mendukung sebagai aplikasi untuk dijadikan *Web Server*. *Appserv* merupakan *Web server* yang mudah di gunakan yang dapat melayani halaman dinamis. Untuk membangun sebuah *Web server*, salah satu program yang handal dan gratis Menurut Yuhendra, M.T, Dr. Eng (2015).

II.2.8. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan cepat, *multiuser* serta menggunakan perintah standar *SQL*. *MySQL* memiliki dua bentuk lisensi, yaitu *FreeSoftware* dan

Shareware. *MySQL* yang biasa digunakan adalah *MySQL Free Software* yang berada di bawah lisensi *GNU/GPL (General Public License)*. Sebagai *database server* yang *free*, artinya *MySQL* dapat secara bebas digunakan untuk kepentingan pribadi atau usaha. Selain sebagai *server*, *MySQL* dapat juga berperan sebagai *client* sehingga sering disebut *database client/server* (Abdul dkk, 2017).

II.2.9. UML (*Unified Modeling Language*)


Unified Modeling Language (UML) adalah metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Gellysa Urva, Helmi Fauzi Siregar, 2015).

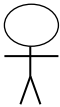


Dalam skripsi ini, peneliti menggunakan 4 diagram UML sebagai desain sistem yaitu sebagai berikut :

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol *Use Case Diagram*

| Gambar | Keterangan |
|---|---|
|  | <i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar |

| | |
|---|---|
| | <p>pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p> |
|  | <p>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 control terhadap <i>use case</i>.</p> |
|  | <p>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.</p> |
|  | <p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p> |

| | |
|--------|--|
| -----> | <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 : Gellysa Urva, dkk : 2015)

2. Class Diagram

Class Diagram Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang merupakan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), *Relasi*, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.2. Multiplicity Class Diagram






| Multiplicity | Penjelasan |
|---------------------|-----------------------------------|
| 1 | Satu dan hanya satu |
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |

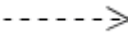

| | |
|------|---|
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4 |

(Sumber : Gellysa Urva, dkk : 2015)

Simbol-simbol yang terdapat pada *Class Diagram* dapat dilihat pada table II.3

Tabel II.3. Simbol *Class Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|-------------------------|---|
| 1 |  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 2 |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor. |
| 5 |  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek. |




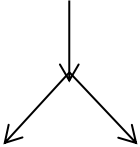
| | | | |
|---|---|--------------------|---|
| 6 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya |


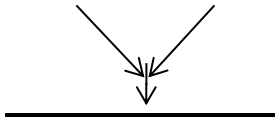
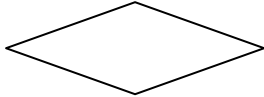
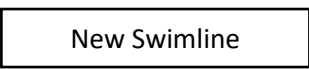
(Sumber : Aprianti, 2016)

3. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.4. 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 parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu. |

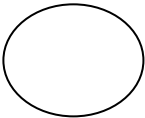
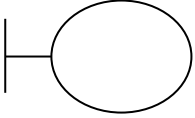
| | |
|---|--|
|  | |
|  | <i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi. |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> . |
|  | <i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa. |

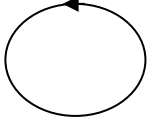

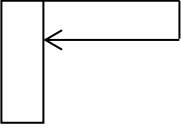
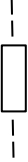

(Sumber : Gellysa Urva, dkk : 2015)

4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu

Tabel II.5. 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 formentry dan <i>form</i> cetak. |

| | |
|---|--|
|  | <i>Control class</i> , suatu objek yang berisi logika aplikasi 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 : Gellysa Urva, dkk : 2015)

II.2.10. Normalisasi Basis Data

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). (Dwi dkk, 2016)

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. Gambaran proses normalisasi adalah sebagai berikut :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

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 / 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*).

Berdasarkan tahapan normalisasi, terdapat lima bentuk normal yaitu :

1. Bentuk Normal Tahap Pertama (*1st Normal Form / 1NF*)

Bentuk Normal Pertama / *1NF*, memiliki aturan :

- Tidak adanya atribut multi-value, atribut komposit atau kombinasinya.
- Mendefinisikan atribut kunci.

- Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).

2. Bentuk Normal Tahap Kedua (*2nd Normal Form / 2NF*)

Bentuk Normal Kedua / *2NF*, memiliki aturan :

- Sudah memenuhi dalam bentuk normal kesatu (*1NF*)
- Semua atribut bukan kunci hanya boleh tergantung (*functional dependency*) pada atribut kunci.
- Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.
- Perlu ada tabel penghubung ataupun kehadiran *foreign key* bagi atribut-atribut yang telah dipisah tadi.

3. Bentuk Normal Tahap (*3rd Normal Form / 3NF*)

Bentuk Normal Ketiga / *3NF*, memiliki aturan :

- Sudah memenuhi dalam bentuk normal kedua (*2NF*).
- Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

4. Bentuk Normal Tahap (*4th Normal Form / 4NF*)

Pada bentuk normal tahap ini digunakan untuk menghilangkan anomali-anomali yang tersisa dan tidak ketergantungan Multivalued.

5. Bentuk Normal Tahap (*5th Normal Form / 5NF*)

Disebut juga *PJNF (Projection Join Normal Form)* dan *4NF* dilakukan dengan menghilangkan ketergantungan join yang bukan merupakan kunci kandidat.