

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 (Abdul Jalil, Ika Purwanti Ningrum, Mutmainnah Muchtar, 2017:173) dengan judul penelitian “Spk Pemberian Kredit Menggunakan Metode WP (Weighted Product) Pada Bmt Mu’amalah Sejahtera Kendari”. Hasil dari penelitian ini yaitu sistem yang dibangun dapat membantu Pimpinan BMT Mu’amalah Sejahtera Kendari menentukan calon debitur yang layak menerima kredit. Pembahasan tersebut hampir sama dengan penelitian yang peneliti lakukan, namun pada penelitian yang peneliti tulis lebih membahas mengenai penerapan metode *Weighted Product* untuk pemilihan mitra kerja pengantar barang. (Dyna Marisa Khairina, Dio Ivando, Septya Maharani, 2016:16) dengan judul penelitian “Implementasi Metode Weighted Product Untuk Aplikasi Pemilihan Smartphone Android”. Dengan hasil penelitian, pemilihan smartphone terbaik berdasarkan masukan dari tingkat kepentingan kriteria oleh user. Pada penelitian ini juga membahas mengenai penerapan metode *Weighted Product* untuk pemilihan *smartphone* android sedangkan pada penelitian peneliti lebih membahas mengenai pemilihan mitra kerja pengantar barang. (Lisa Septian Putri, Nurul Hidayat, Suprpto,

2018:1219) dengan judul penelitian “Sistem Pendukung Keputusan Pemilihan Mitra Jasa Pengiriman Barang menggunakan Metode *Simple Additive Weighting* (SAW) – *Technique for Other Reference by Similarity to Ideal Solution* (TOPSIS) di Kota Malang”. Dengan hasil penelitian, sistem yang dibangun dengan menerapkan metode SAW dan TOPSIS dapat mempermudah pengguna atau masyarakat dalam menentukan keputusan pemilihan mitra jasa pengiriman barang terbaik di Kota Malang. Pada penelitian ini juga membahas mitra jasa pengantar barang menggunakan metode SAW dan TOPSIS, sedangkan peneliti menggunakan metode *Weighted Product*.

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. Konsep Dasar Sistem

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*). (Asep Muhidin, 2017:150)

II.2.1.1. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*Abstract System*) dan sistem fisik (*Physical System*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologis, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan.
2. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.

Sistem diklasifikasikan sebagai sistem alamiah (*Natural System*) dan sistem buatan manusia (*Human Made System*) :

1. Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi.
2. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *Human Machine System* atau ada yang menyebut dengan *Man Machine System*. Sistem informasi merupakan contoh *Man Machine System*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

Sistem diklasifikasikan sebagai sistem tertentu (*Deterministic System*) dan sistem tak tentu (*Probabilistic System*) :

1. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan.
2. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem diklasifikasikan sebagai sistem tertutup (*Closed System*) dan sistem terbuka (*Open System*) :

1. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *Relatively Closed System* (secara relatif tertutup, tidak benar-benar tertutup).
2. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja (Asep Muhidin, 2017:150).

II.2.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur.

Tujuan dari SPK adalah untuk membantu pengambil keputusan memilih berbagai alternatif keputusan yang merupakan pengolahan informasi-informasi yang diperoleh atau tersedia dengan menggunakan model pengambilan keputusan. Ciri utama sekaligus keunggulan dari sistem pendukung keputusan tersebut adalah kemampuannya untuk menyelesaikan masalah-masalah yang tidak terstruktur. Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan factor-faktor yang perlu dipertimbangkan dalam pengambilan keputusan. Tahap-tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut (Dyna M Khairina, dkk, 2016:17).

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.2.3. Metode Weighted Product

Weighted Product (WP) adalah suatu metode yang menggunakan perkalian untuk menghubungkan rating atribut, di mana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Proses ini sama halnya dengan proses normalisasi. Metode WP dapat membantu dalam mengambil keputusan pemilihan laptop, akan tetapi perhitungan dengan menggunakan metode WP ini hanya menghasilkan nilai terbesar yang akan terpilih sebagai alternatif yang terbaik. Perhitungan akan sesuai dengan metode ini apabila alternatif yang terpilih memenuhi kriteria yang telah ditentukan. Metode WP ini lebih efisien karena waktu yang dibutuhkan dalam perhitungan lebih singkat. Bobot untuk atribut manfaat berfungsi sebagai pangkat positif dalam proses

perkalian, sementara bobot biaya berfungsi sebagai pangkat negatif (Nur A Syafitri, dkk, 2016:171-172).

Perbaiki bobot untuk $\sum W_j = 1$ menggunakan Persamaan

$$W_j = \frac{w}{\sum w} \dots\dots\dots (1)$$

Variabel W adalah pangkat bernilai positif untuk atribut keuntungan dan bernilai negatif untuk atribut biaya. Preferensi untuk alternatif S_i diberikan oleh Persamaan.

$$S_i = \prod_j^n x_{ij}^{w_j} \dots\dots\dots (2)$$

dengan $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$ sebagai atribut.

Keterangan:

Π : *product*

S_i : skor / nilai dari setiap alternative

X_{ij} : nilai alternatif ke- i terhadap atribut ke- j

W_j : bobot dari setiap atribut atau kriteria

n : Banyaknya kriteria

Untuk mencari alternatif terbaik dilakukan dengan Persamaan.

$$V_i = \frac{S_i}{\prod_{j=1}^n (X_j^*)^{w_j}} \dots\dots\dots (3)$$

dimana :

V : Preferensi alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

$*$: Banyaknya kriteria yang telah dinilai pada vektor S

Nilai V_i yang terbesar menyatakan bahwa alternatif A_i yang terpilih.

Langkah-langkah dalam perhitungan metode WP adalah sebagai berikut :

1. Mengalikan seluruh atribut bagi seluruh alternatif dengan W (bobot) sebagai pangkat positif untuk atribut keuntungan dan bernilai negatif untuk atribut biaya.
2. Hasil perkalian dijumlahkan untuk menghasilkan nilai pada setiap alternative.
3. Membagi nilai V bagi setiap alternatif dengan nilai total dari semua nilai alternatif.
4. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan.

II.2.4. Kemitraan

Dalam prakteknya, strategi kemitraan dalam *Supply Chain management* berpengaruh positif terhadap kualitas hasil produk yang diinginkan. Kemitraan dengan pemasok strategis mengidentifikasi praktik optimal yang dapat memfasilitasi proses perolehan raw material berkualitas yang selaras dan terintegrasi dengan kebutuhan perubahan lingkungan bisnis.

Kemitraan merupakan sebuah upaya untuk beradaptasi dengan perubahan lingkungan bisnis melalui kapabilitas teknologi aplikatif dan inovasi yang dapat dicapai. Hal ini dapat dicapai jika perusahaan mampu mengembangkan sebuah mekanisme untuk mengelola pengetahuan/knowledge management yang diperoleh

serta melakukan komunikasi dan menggunakan teknik konflik resolusi yang tepat dengan pemasok mitra. (Agung Sofani, dkk, 2017:116).

II.2.5. PHP (*Hypertext Preprocessor*)

PHP merupakan bahasa *Scripting Server Side*, dimana pemrosesan datanya dilakukan pada sisi *Server*. Sederhananya, *Server*lah yang akan menerjemahkan *Script* program, baru kemudian hasilnya akan dikirim kepada *Client* yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*Script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *Web Browser* menjadi kode HTML.

Pada prinsipnya *Server* akan bekerja apabila ada permintaan dari *Client*. Dalam hal ini *Client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *Server*. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *Website* oleh *Browser*. Berdasarkan URL atau alamat *Website* dalam jaringan internet, *Browser* akan menemukan sebuah alamat dari *Websserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *Web Server*.

Selanjutnya *Web Server* akan mencari berkas yang diminta dan menampilkan isinya di *Browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh *User* adalah halaman yang mengandung *Script* PHP? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan

dikirim ke *Web Server* selanjutnya akan memeriksa tipe file yang diminta *User*. Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi *Script* dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung *Script* PHP, permintaan user akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan script-script PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user (Astria dkk, 2016:30).

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, 2018:25-26).

Basis data sendiri dapat didefinisikan dari beberapa sudut pandang yang ada seperti :

1. Himpunan kelompok data arsip yang saling berhubungan dengan terorganisir sedemikian rupa agar kelak dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan/*Redudancy* yang tidak perlu untuk memenuhi berbagai kebutuhan.

3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Tujuan utama dari pemanfaatan basis data dalam pengolahan data adalah :

1. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan atau manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika kita menyimpan data secara manual / non elektronik atau secara elektronik tetapi tidak dalam bentuk penerapan basis data misalnya dalam bentuk *Spread Sheet* atau dokumen teks biasa.

2. Efisiensi Ruang Penyimpanan (*Space*)

Karena keterkaitan yang erat data dalam sebuah basis data , maka redundansi data pasti terjadi. Banyaknya redundansi ini tentu akan memperbesar ruang penyimpanan yang disediakan. Dengan basis data, efisiensi atau optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik atau dengan menerapkan sejumlah aturan dalam pengkodean atau dengan membuat relasi-relasi antar kelompok antar data dan saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (constraint) tipe data, domain data, keunikan data dan sebagainya sangat berguna untuk menekan ketidakakuratan pemasukan atau penyimpanan data.

4. Ketersediaan (*Availability*)

Data akan selalu siap dibutuhkan jika sewaktu - waktu akan di butuhkan atau digunakan.

5. Kelengkapan (*Completeness*)

Kelengkapan data menyimpan struktur yang mendefinisi detail dari tiap objek sehingga kita dapat melakukan perubahan struktur dalam basis data baik dalam penambahan objek baru / table atau dengan penambahan kolom pada suatu tabel.

6. Keamanan (*Security*)

Keamanan untuk menentukan user yang boleh menggunakan basis data beserta objek-objek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan user.

7. Kebersamaan Pemakai (*Shareability*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja di satu lokasi atau satu sistem aplikasi saja. Data pegawai dalam basis data kepegawaian, misalkan dapat digunakan oleh banyak sistem (sistem penggajian, sistem akuntansi, sistem inventori, dan lain-lain). Basis data yang dikelola oleh sistem aplikasi yang mendukung lingkungan *Multiuser*, akan dapat memenuhi kebutuhan ini, tapi tetap dengan menjaga atau menghindari munculnya persoalan baru seperti inkonsistensi data karena data yang sama diubah oleh banyak pemakai disaat yang bersamaan atau kondisi *Deadlock* karena ada banyak pemakai yang saling menunggu untuk menggunakan data.

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 yang penulis gunakan dalam membuat tugas akhir ini adalah Appserv-win32-8.5.0. exe. (Rita dkk, 2018:27)

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 *Free Software* 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, 2017:176)

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

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

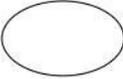
UML merupakan metodologi dalam mengembangkan system berorientasi Objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Dalam membangun perancangan system dengan alat bantu perancangan *UML* ada beberapa tahapan yang akan dilakukan yaitu *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram* (Gellysa Urva, dkk (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 actor 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 *Usecase Diagram*

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukaran pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>Use Case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mendefinisikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau system biasa muncul dalam beberapaperan. Perlu dicatat bahwa actor berinteraksi dengan <i>use case</i> tetapi tidak memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara actor 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 actor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila actor berinteraksi secara pasif dengan system
	<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 dipenuhi.

(Sumber : Gellysa Urva, dkk : 2015)

2. Class Diagram

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan 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 khusus 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. Simbol *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

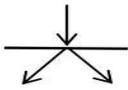
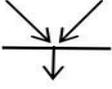
(Sumber : Gellysa Urva, dkk : 2015)

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.3. *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

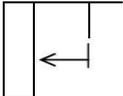
	<i>End Point</i> , akhir aktifitas
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (Percabangan) digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision points</i> menggambarkan pilihan untuk pengambilan keputusan, true, false
	<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 (Gellysa Urva, dkk (2015):

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambar 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 actor dengan sistem. Seperti tampilan <i>formentry</i> 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 kelas.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

(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:A-341)

Tujuan dari normalisasi adalah :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *Storage* yang dipakai oleh *Base Relations* (file).
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data

Gambaran proses normalisasi adalah :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian
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.

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.

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*). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu *3NF*.

Berdasarkan tahapan normalisasi, terdapat lima bentuk normal yaitu :

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

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

1. Tidak adanya atribut multi-value, atribut komposit atau kombinasinya.
2. Mendefinisikan atribut kunci.
3. Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi)

Tabel II.5. Contoh Bentuk Tidak Normal

id_alter natif	nama_alter natif	alamat	biaya	no_telp
M01	Aceh Tamiang	Aceh Tamiang, Kec. Manyak Pahit, No.14	5000/kg	085270670105
M02	APM	Jl. Letda Sudjono	5000/kg	085265209801
M03	Bagus Jaya	Jl. Bersama No.29, Bandar Selamat.	6000/kg	081375286634
M04	HGM	Jl. Kapten Sumarsono, No. 73. Helvetia.	4000/kg	085275565164
M05	IPL Logistic	Jl. Merci Raya, Suka Makmur.	8000/kg	082320067680
M06	Makaro	Jl. Sumarsono No. 77A	8000/kg	082369686334

M07	Mefa	Jl. Akasia Blok. V No.1, Kisaran.	15000/kg	081360775000
M08	Rahman Logistic	Jl. Kataso, Indah Taksi.	10000/kg	082369900536
M09	Tapanuli Logistic	Ruko Metrolink, Pangkalan Masyhur.	5000/kg	085362022141
M10	Union	Jl. Pertahanan No. 102, Sigara-gara.	15000/kg	081360727537

Tabel II.6. Contoh Bentuk *1NF*

id_alter natif	nama_alternatif	alamat	biaya
M01	Aceh Tamiang	Aceh Tamiang, Kec. Manyak Pahit, No.14	5000/kg
M02	APM	Jl. Letda Sudjono	5000/kg
M03	Bagus Jaya	Jl. Bersama No.29, Bandar Selamat.	6000/kg
M04	HGM	Jl. Kapten Sumarsono, No. 73. Helvetia.	4000/kg
M05	IPL Logistic	Jl. Merci Raya, Suka Makmur.	8000/kg
M06	Makaro	Jl. Sumarsono No. 77A	8000/kg
M07	Mefa	Jl. Akasia Blok. V No.1, Kisaran.	15000/kg
M08	Rahman Logistic	Jl. Kataso, Indah Taksi.	10000/kg
M09	Tapanuli Logistic	Ruko Metrolink, Pangkalan Masyhur.	5000/kg
M10	Union	Jl. Pertahanan No. 102, Sigara-gara.	15000/kg

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

Bentuk Normal Kedua / 2NF, memiliki aturan :

1. Sudah memenuhi dalam bentuk normal kesatu (1NF)

2. Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci
3. Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain
4. Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi

Tabel II.7. Contoh Bentuk 2NF

id_alter_natif	nama_alternatif	alamat
M01	Aceh Tamiang	Aceh Tamiang, Kec. Manyak Pahit, No.14
M02	APM	Jl. Letda Sudjono
M03	Bagus Jaya	Jl. Bersama No.29, Bandar Selamat.
M04	HGM	Jl. Kapten Sumarsono, No. 73. Helvetia.
M05	IPL Logistic	Jl. Merci Raya, Suka Makmur.
M06	Makaro	Jl. Sumarsono No. 77A
M07	Mefa	Jl. Akasia Blok. V No.1, Kisaran.
M08	Rahman Logistic	Jl. Kataso, Indah Taksi.
M09	Tapanuli Logistic	Ruko Metrolink, Pangkalan Masyhur.
M10	Union	Jl. Pertahanan No. 102, Sigaragara.

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

Bentuk Normal Ketiga / 3NF, memiliki aturan :

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

Tabel II.8. Contoh Bentuk 3NF

id_alternatif	nama_alternatif
M01	Aceh Tamiang
M02	APM
M03	Bagus Jaya
M04	HGM
M05	IPL Logistik
M06	Makaro
M07	Mefa
M08	Rahman Logistic
M09	Tapanuli Logistic
M10	Union

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.