

BAB II

TINJAUAN PUSTAKA

II.1. Peneliti Terdahulu

Sebagai bukti penelitian yang akan dibuat, maka penelitian akan dibandingkan terhadap penelitian sejenis yang pernah dilakukan. Adapun Penelitian sebelumnya yang penulis angkat yaitu :

Berdasarkan penelitian Desta Nur Efika Ardini (2019) “Sistem Peramalan Penjualan Menggunakan Metode Trend Moment Pada Toko Mebel Nabila Furniture Paguyangan Brebes Berbasis Desktop” Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah aplikasi untuk peramalan penjualan menggunakan *metode Trend Moment* untuk pengambilan keputusan dalam mengetahui jumlah stok mebel di toko Mebel Nabila Furniture. Dengan adanya sistem peramalan penjualan barang mebel dan furniture menggunakan metode *trend moment* dapat membantu pihak toko dalam pengambilan keputusan tentang pengadaan jumlah stok dan prediksi jumlah penjualan barang pada periode tertentu sehingga meminimalisir kelebihan atau kekurangan stok pada toko Mebel Nabila Furniture.

Menurut Aan Suhatri Wahyono (2018) dengan judul Implementasi Metode *Trend Moment* Untuk Peramalan Penjualan Kubis” Permasalahan yang umum dihadapi penjual kubis adalah bagaimana meramalkan atau memperkirakan penjualan kubis di masa mendatang berdasarkan data yang telah direkam sebelumnya. Peramalan tersebut sangat berpengaruh pada penjual kubis untuk sebagai bahan acuan memperkirakan penjualan kubis yang harus disediakan guna memenuhi permintaan pasar yang ada. Metode peramalan *Trend Moment* digunakan dalam peramalan penjualan kubis. Peramalan menggunakan

data penjualan yang sudah ada selama 3 bulan, dengan data tersebut bisa diperoleh siklus penjualan yang akan dijadikan acuan sistem peralaman ini.

Menurut penelitian Imam Wahyudi (2018) dengan judul “Penerapan Metode Trend Moment Untuk Peramalan Penjualan Sepatu Dan Sandal Pada Toko Batt” Metode *Trend Moment* ini digunakan untuk dapat mengatasi permasalahan yaitu untuk meramalkan penjualan, sehingga pengaruh unsur subyektif dalam menentukan peramalan penjualan nantinya dapat di hindarkan. Semakin banyak data-data historis untuk perhitungan peramalan menggunakan metode *Trend Moment* semakin akurat pula peramalan yang dihasilkan, sehingga hasil peramalan tersebut dapat membantu pelaku pengusaha bisnis dalam mempertimbangkan berapa banyak persediaan stok barang di bulan berikutnya, agar tidak mengalami kekurangan atau kelebihan stok barang.

Menurut penelitian Edy Purnomo (2018) dengan judul “ Penerapan Metode Trend Moment Untuk Forecast Penjualan Barang di Indomaret” Seiring berjalannya waktu dan semakin banyaknya gerai toko Indomaret di Kalimantan Timur, permintaan konsumen terhadap suatu barang sangatlah pesat sehingga membuat perusahaan harus dapat merencanakan penjualan barang untuk memenuhi permintaan konsumen agar dapat meminimalkan pengeluaran dana dalam menyuplai barang dari gudang. Pada penelitian ini menggunakan Metode Trend Moment untuk meramalkan penjualan bulan Desember dari bulan Mei hingga November pada Toko TD9I – CIPTO2 dan Prdcd 10003517. Dari hasil peramalan menggunakan Metode Trend Moment pada bulan Desember mengalami Trend Positif dimana penjualannya adalah 436,17.

Menurut penelitian Ilyas (2018) dengan judul “Implementasi Metode Trend Moment (Peramalan) Mahasiswa Baru Universitas Widyagama Malang” Metode Trend Moment menggunakan cara-cara perhitungan statistika dan matematika. Metode Trend Moment

merupakan metode analisis yang dapat digunakan untuk keperluan peramalan dengan membentuk persamaan $Y = a + bX$ (Purnomo, 2018). Tujuan dari penelitian ini adalah membuat Implementasi Metode Trend Moment (Peramalan) Mahasiswa Baru di Universitas Widyagama Malang. Aplikasi peramalan menggunakan metode trend moment dapat dipergunakan untuk meramalkan jumlah mahasiswa baru di Universitas Widyagama Malang di periode yang akan datang dan Juga aplikasi peramalan ini dapat mempermudah petugas di Universitas Widyagama Malang untuk menentukan dan mengetahui jumlah mahasiswa baru yang akan datang.

II.2. Landasan Teori

II.2.1. Sistem

Sistem adalah komponen-komponen yang saling berinteraksi satu sama lain untuk mencapai tujuan yang diterapkan. Dalam sistem terdapat 3 komponen pokok yaitu komponen masukan, komponen proses, dan komponen keluaran. Secara umum sistem adalah kumpulan dari beberapa bagian tertentu yang saling berhubungan secara harmonis untuk mencapai suatu tujuan tertentu. Elemen-elemen yang mewakili suatu sistem secara umum adalah masukan (*input*) pengolahan (*processing*) dan keluaran (*output*). (Herlina Trisnawati : 2018 :27).

II.2.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang berguna dan menjadi berarti bagi penerimanya. Kegunaan informasi adalah untuk mengurangi ketidak pastian di dalam proses pengambilan keputusan tentang suatu keadaan. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut. (Priyo Sutopo, dkk 2018).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

1. Akurat (*Accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Berarti informasi tersebut bermanfaat bagi pemakainya (Deppi Linda, 2018 : 62-63).

II.2.3. Sistem Informasi

Sistem informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Penggunaan sistem informasi telah banyak diterapkan diberbagai bidang termasuk mdalam bisnis. Salah satu tujuan penerapan sistem informasi dalam bidang bisnis agar dapat meningkatkan keuntungan bisnis dengan menggunakan kemampuan yang didapatkan dari sistem informasi. Ada beberapa kemampuan dari sistem informasi yang dapat mendukung dalam bidang bisnis. Kemampuan tersebut seperti pengurangan biaya, mempercepat pekerjaan, dapat meningkatkan

kemudahan dalam pengambilan keputusan, dan peningkatan pelayanan terhadap pelanggan. (Alfian Nurlifa : 2017 :19)

II.2.4. *Trend Moment*

Metode *trend moment* adalah merupakan metode untuk mencari garis *trend* dengan perhitungan statistika dan matematika tertentu guna mengetahui fungsi garis lurus sebagai pengganti garis patah-patah yang dibentuk oleh data historis. Dengan demikian pengaruh unsur subjektif dapat dihindarkan. Persamaan trend dengan metode *trend moment* adalah sebagai berikut (Imam Wahyudi : 2018 : 3) :

$$Y = a + bX \dots \dots \dots (1)$$

Dimana :

Y : nilai trend (Peramalan)

a : bilangan konstant

b : *slope* atau koefisien kecondongan garis *trend*

X : indeks waktu (x = 0, 1, 2, 3, ..., n)

Sedangkan untuk menghitung nilai a dan b digunakan rumus pada persamaan 2 dan persamaan 3.

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{(\sum X^2) - (\sum X)^2} \dots \dots \dots (2)$$

$$a = \frac{\sum Y - b(\sum X)}{n} \dots \dots \dots (3)$$

Dimana :

ΣX : Jumlah kumulatif dari periode waktu

ΣY : Jumlah kumulatif data penjualan

ΣXY : Jumlah kumulatif dari jumlah periode dikalikan jumlah penjualan

n : Banyaknya periode waktu (bulan)

Setelah nilai ramalan yang telah diperoleh dari hasil peramalan dengan metode Trend Moment akan dikoreksi terhadap pengaruh musiman dengan menggunakan indeks musim dengan rumus :

$$\text{Rata – Rata Permintaan Tertentu / rata – rata permintaan perbulan.....(4)}$$

Untuk mendapatkan hasil ramalan akhir setelah dipengaruhi indeks musim maka akan menggunakan perhitungan sebagai berikut

$$Y^* = \text{Indeks Musim} \times Y \text{.....(5)}$$

Dimana :

Y^* = Hasil ramalan dengan menggunakan metode *Trend Moment* yang telah dipengaruhi oleh indeks musim.

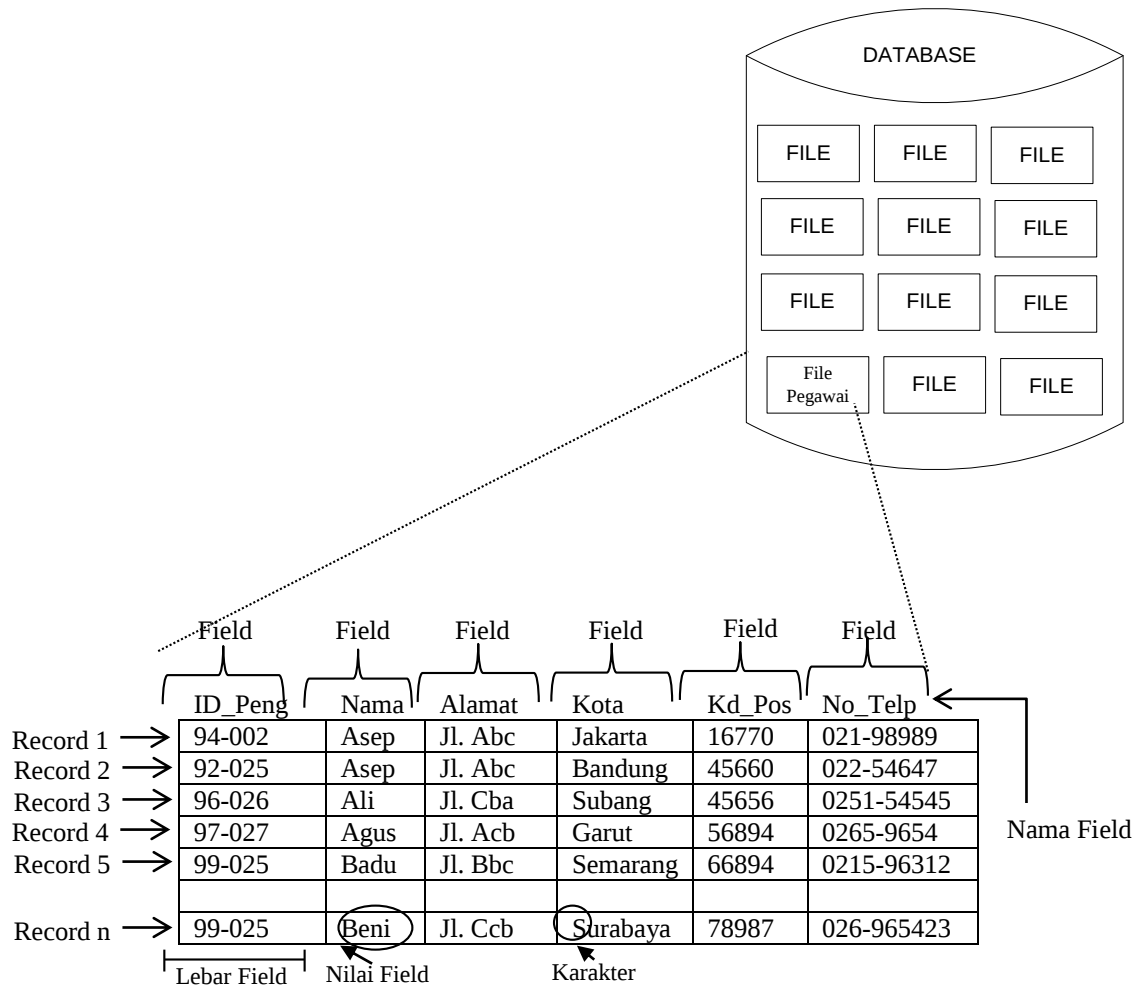
Y = Hasil ramalan dengan menggunakan Trend Moment.

II.2.5. Basis Data (*Database*)

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar

dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati dan Hendra Kurniawan, 2015 : 50).

Database merupakan kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu Database dapat juga diartikan Koleksi data yang terorganisasi untuk melayani beragam aplikasi secara efisien dengan mensentralisasi data dan meminimalisasi data yang berlebih. (Mukhlisulfatih Latief : 2018 : 231).



Gambar II.1. Hirarki Database
Sumber : Mukhlisulfatih Latief : 2018 :232

Bahasa SQL tersusun atas 3 kelompok pernyataan berdasarkan fungsi dari pernyataan tersebut yaitu:

1. Data Definition Language (DDL) : Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara relasi data, validasi data dan lainnya.
 Contoh : create, drop, alter table.

2. *Data Manipulation Language* (DML) : Data yang telah dibuat dan didefinisikan tersebut akan dilakukan beberapa pengerjaan, seperti menyaring data, melakukan proses query. Contoh : *select, update, insert*.
3. *Data Control Language* (DCL) : Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*. (Latief : 2018)

II.2.6. Normalisasi

Normalisasi adalah proses penyusunan table-table yang tidak redundan (double) yang dapat menyebabkan anomali pada saat terjadi manipulasi data seperti tambah, ubah dan hapus. Tujuan dari normalisasi adalah:

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi, kompleksitas.
3. Untuk mempermudah pemodifikasian data. (Herlina Trisnawati : 2018 :29)

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redudansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief : 2018)

1. Proses Normalisasi
 - a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.

- 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.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan grup.

Tabel II.2. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2018

- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

Tabel II.3. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2018

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

Tabel II.4. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Sumber : Mukhlisulfatih Latief : 2018

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya. (Mukhlisulfatih Latief : 2018)

Tabel II.5. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlisulfatih Latief : 2018

II.2.7. PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah

web dan bisa digunakan pada dokumen *HTML*. *PHP* dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen *HTML* yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh *PHP* pada umumnya akan memberikan hasil pada web browser, tetapi proses secara keseluruhan dijalankan di server. (Saipul Anwar : 2018)

II.2.8. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (*RDBMS*) yang didistribusikan secara gratis dibawah lisensi *GPL* (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. (Herpendi : 2018)

Kehandalan suatu sistem basisdata (*DBMS*) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, *MySQL* mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat

lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (*wordpress*), *CMS*, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. (Herpendi : 2018)

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang *programmer* komputer asal *Swedia*. *Monty* mengembangkan sebuah sistem *database* sederhana yang dinamakan *UNIREG* yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing*. Pada saat itu *Monty* bekerja pada perusahaan bernama *TcX* di *Swedia*. *TcX* pada tahun 1994 mulai mengembangkan aplikasi berbasis *web*, dan berencana menggunakan *UNIREG* sebagai sistem *database*. Namun sayangnya, *UNIREG* dianggap tidak cocok untuk *database* yang dinamis seperti *web*. (Herpendi : 2018)

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

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami

dibandingkan dengan bahasa pemrograman. *Unified Modelling Language* (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan actor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionalty* dengan *stereo types*.

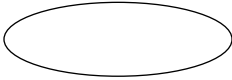
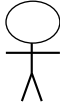
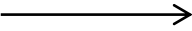
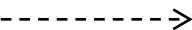
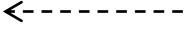
Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Saipul Anwar, et al., 2018 : 75-76).

II.2.9.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2018 : 108).

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.6. Simbol Use Case

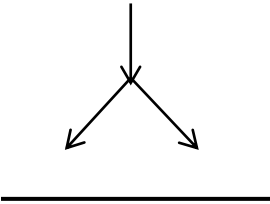
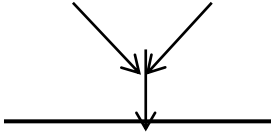
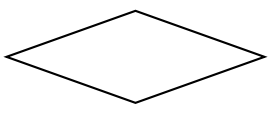
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 control 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 mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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 :: Ade Hendini, 2018)

II.2.9.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.7 dibawah ini:

Tabel II.7. Simbol Activity Diagram

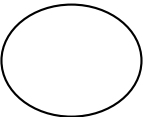
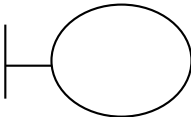
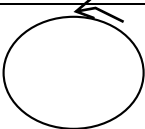
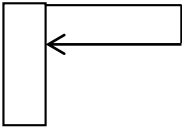
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel 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 diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini, 2018 : 109)

II.2.9.3. 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* dapat dilihat pada tabel II.8 dibawah ini :

Tabel II.8. 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 : Ade Hendini, 2018 : 110)

II.2.9.4. 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 khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2018 : 111).

Tabel II.9. 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 : Ade Hendini, 2018 : 110)