

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

1. Menurut Eka Larasati Amalia, 2021, yang berjudul “ Sistem Informasi PT Bintang Sidoraya Dengan Peramalan Penjualan Menggunakan Metode *Statistical Parabolic Projection*”. Tujuan dari penelitian ini yaitu hasil penelitian yang diperoleh dari pengujian yang telah dilakukan sebagai berikut. Pertama, sistem informasi dapat melakukan reporting melakukan pengolahan data penjualan sehari-hari dengan komputerisasi dan secara online, dengan hasil pengujian *usability testing* mengenai tingkat kualitas dari sistem yang dibuat mendapatkan nilai kesesuaian sebesar 82,12%. Kedua, sistem dapat mengimplementasikan metode *statistical prabolic projection* untuk peramalan di periode mendatang.
2. Menurut Abdul Khamid, 2021, yang berjudul “Rancang Bangun Sistem Informasi Peramalan Penjualan pada Songkok Palapa Gresik dengan menggunakan Metode *Time-Series Berbasis Website*” Dalam hasil penelitian, dibuat sebuah rancangan sistem informasi peramalan penjualan berbasis *website* yang sesuai dengan kebutuhan industri, proses perancangan dimulai dari wawancara kepada owner lalu membuat rancangan proses bisnis, *unified modelling language*, dan menyiapkan algoritma permalan, dan hasil yang

diperoleh dari sistem ini terdapat 3 user pengguna utama yaitu, Admin, Bagian penjualan, dan Pemilik. Sistem telah menghitung secara tepat dan akurat data penjualan yang diolah menjadi sebuah informasi baru yang bisa digunakan sebagai penunjang keputusan bagi top level manajemen.

3. Menurut Maftahatul Hakimah, 2017, yang berjudul “Rancang Bangun Aplikasi Peramalan Persediaan Barang Dengan Metode Trend Projection” Dalam hasil penelitian, dibuat Sistem ini dapat membantu CV. Berdikari Surabaya” dalam menentukan jumlah pemesanan barang kepada penyedia barang untuk periode yang akan datang. Hal ini dikarenakan nilai penjualan 2014 yang diramalkan hampir mendekati data sebenarnya meskipun ada beberapa periode menunjukkan perbedaan yang sangat jauh. Hal ini bisa diakibatkan oleh faktor ekonomi. Pengujian sistem terhadap 5 data penjualan menggunakan metode *Trend Projection* diperoleh tingkat akurasinya sebesar 86 %.

Dalam penelitian yang dilakukan oleh penulis dimana perhitungan peramalan dapat dilakukan dengan lebih mudah dengan hasil yang lebih optimal dan menghasilkan grafik peramalan dengan lebih cepat.

## **II.2. Uraian Teoritis**

### **II.2.1 Sistem**

Suatu Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Unsur-unsur suatu sistem terdiri dari subsistem yang

lebih kecil yang terdiri pula dari kelompok- kelompok unsur yang membentuk subsistem tersebut. (Tomi Loveri, 2018).

Terdapat 2 kelompok pendekatan di dalam pendefinisian sistem, yaitu :

1. Pendekatan yang menekankan pada prosedur.

Pendekatan ini mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersamasama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

2. Pendekatan sistem yang lebih menekankan pada elemen atau komponen.

Pendekatan ini mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. (Tomi Loveri, 2018)

#### **2.2.1.1 Karakteristik Sistem**

Suatu sistem mempunyai karakteristik atau sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*prosses*), sasaran (*objectives*), ataupun tujuan (*goal*).

Berikut ini adalah komponen komponen atau sub sistem yang merupakan salah satu unsur dari karakteristik sistem. (Tomi Loveri, 2018)

##### **a. Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses

sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan Supra sistem.

b. Batas Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain

dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah maintenance input yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah signal input yang akan diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

g. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem

tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. (Tomi Loveri, 2018).

### **2.2.2 Informasi**

Di dalam pengolahan sistem pada akhirnya menghasilkan suatu informasi, untuk itu pendefinisian informasi diperlukan untuk menunjang berhasilnya pengembangan sistem yang akan dirancang. Informasi adalah data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya

Jadi informasi adalah data yang diproses kedalam bentuk yang lebih berarti bagi penerima dan berguna dalam pengambilan keputusan. (Tomi Loveri, 2018).

#### **2.2.2.1 Siklus Informasi**

Data merupakan bentuk yang masih mentah yang belum dapat bercerita dengan banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*). Siklus ini

disebut juga dengan siklus pengolahan data (*data processing cycle*). (Tomi Loveri, 2018).

### **2.2.3 Pengertian Sistem Informasi**

Sistem informasi merupakan serangkaian komponen berupa manusia, prosedur data, dan teknologi (seperti komputer) yang digunakan untuk melakukan sebuah proses untuk menghasilkan informasi yang bernilai untuk pengambilan keputusan.”(Lauw Wulandari, Haryanto Tanuwijaya, Julianto Lemantara) Sistem informasi berfungsi menghasilkan informasi untuk memenuhi kebutuhan aktivitas organisasi. Informasi yang bernilai tinggi dihasilkan oleh suatu sistem informasi yang efektif dan efisien. Untuk itu, sistem informasi yang efektif dan efisien menghendaki intervensi manajemen secara tepat. (Tomi Loveri, 2018)

#### **2.2.3.1 Komponen Sistem Informasi**

Sistem informasi terdiri dari komponen-komponen sebagai berikut :

##### **1. Blok Masukan**

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

##### **2. Blok Model**

Blok model ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis

data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

### 3. Blok keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

### 4. Blok Teknologi

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

### 5. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

### 6. Blok kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk

menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi. (Tomi Loveri : 2 : 2018).

#### II.2.4. Peramalan

Peramalan adalah data di masa lalu yang digunakan untuk keperluan estimasi data yang akan datang. Peramalan atau Forecasting merupakan bagian terpenting bagi setiap perusahaan ataupun organisasi bisnis dalam setiap pengambilan keputusan manajemen. Peramalan itu sendiri bisa menjadi dasar bagi perencanaan jangka pendek, menengah maupun jangka panjang suatu perusahaan. Ramalan pada dasarnya merupakan dugaan atau perkiraan mengenai terjadinya suatu kejadian atau peristiwa di waktu yang akan datang. (Ilyas: 2018)

#### II.2.5. Metode *Statistical Parabolic Projection*

Trend parabolic atau dikenal dengan sebutan trend kuadrat adalah trend dengan variabel X berpangkat paling tinggi sehingga akan menghasilkan garis trend melengkung. Untuk mencari nilai peramalan dapat menggunakan rumus pada persamaan berikut :

$$\ddot{y} = a + bX + CX^2 \dots\dots\dots(1)$$

$$a = \frac{(\sum Y)(\sum X^4) - (\sum X^2 Y)(\sum X^2)}{n(\sum X^4) - (\sum X^2)^2} \dots\dots\dots(2)$$

$$b = \frac{(\sum XY)}{\sum X^2} \dots\dots\dots(3)$$

$$c = \frac{n(\sum X^2 Y)(\sum X^2) - (\sum Y)}{n(\sum X^4) - (\sum X^2)^2} \dots\dots\dots(1)$$

Keterangan :

$\hat{y}$  = Hasil peramalan

X = Periode Waktu

a, b, c = Konstanta

n = Jumlah Data

Y = Nilai Trend

(Dwi Bastomi, 2021)

Contoh kasus :

| Tahun akademik | Jumlah Pendaftar (y) | X  | $x^2$ | $x^4$ | x.y   | $x^2.y$ |
|----------------|----------------------|----|-------|-------|-------|---------|
| 2013/2014      | 482                  | -3 | 9     | 81    | -1446 | 4338    |
| 2014/2015      | 178                  | -2 | 4     | 16    | -356  | 712     |
| 2015/2016      | 272                  | -1 | 1     | 1     | -272  | 272     |
| 2016/2017      | 547                  | 0  | 0     | 0     | 0     | 0       |
| 2017/2018      | 508                  | 1  | 1     | 1     | 508   | 508     |
| 2018/2019      | 560                  | 2  | 4     | 16    | 1120  | 2240    |
| 2019/2020      | 561                  | 3  | 9     | 81    | 1683  | 5049    |
| $\Sigma$       | 3108                 |    | 28    | 196   | 1237  | 13119   |

Penyelesaian :

Keterangan :

$\Sigma Y$  = Diperoleh dari total keseluruhan nilai (Y)

$$X = \frac{\text{tahun dasar} - \text{tahun tengah}}{\text{interval}}$$

- Langkah selanjutnya yang harus dilakukan setelah data dihitung dengan menggunakan Metode Jumlah Kuadrat Terkecil (The Least Square's Method) adalah mencari nilai a, b dan c yang merupakan variabel nilai yang akan digunakan untuk melakukan proyeksi nantinya. Untuk mencari nilai a, b dan c harus berdasarkan hasil perhitungan pada Tabel 3 menggunakan rumus berikut:

$$a = \frac{(\sum x^4)(\sum y) - (\sum x^2)(\sum x^2 y)}{n(\sum x^4) - n(\sum x^2)^2}$$

$$a = \frac{(196)(3108) - (28)(13119)}{7(196) - 7(28)^2}$$

$$a = -58,75510204$$

$$b = \frac{\sum xy}{\sum x^2}$$

$$b = \frac{1237}{28}$$

$$b = 44,17857143$$

$$c = \frac{n(\sum x^2 y) - (\sum x^2)(\sum y)}{n(\sum x^4) - (\sum x^2)^2}$$

$$c = \frac{7(13119) - (28)(3108)}{7(196) - (28)^2}$$

$$c = 8,178571429$$

3. Setelah nilai a, b dan c telah diperoleh, selanjutnya adalah tahap melakukan proyeksi data dengan menggunakan rumus :

$$yc = a + bx + cx^2$$

$$yc = -58,75510204 + (44,17857143 \cdot 7) + (8,178571429 \cdot 7^2)$$

$$yc = 651$$

## II.2.6. Database

Menurut Sutabri dalam buku Sistem Informasi Manajemen: ‘Database adalah suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (*controlled redundancy*’. ( Fitri Ayu : 2018)

Beberapa model *Database* diantaranya :

1. *Object based* data model merupakan himpunan data dan prosedur atau relasi yang menjelaskan hubungan logis antar data dalam suatu *database* berdasarkan objek datanya.
2. *Record Based* data model. Model ini berdasarkan pada *record* untuk menjelaskan kepada *user* tentang hubungan logis antardata dalam *database*. (Fitri Ayu : 2018)

#### **II.2.7. ASP (*Active Server Page*)**

ASP.NET adalah arsitektur yang digunakan untuk mengembangkan *website* dinamis. Untuk mengembangkan *website* berbasis arsitektur ini membutuhkan versi *Microsoft .NET Framework 3.5* atau lebih tinggi. Bahkan *website* yang dikembangkan menggunakan ASP.NET MVC memiliki kelebihan seperti SEO (*Search Engine Optimizer*) *friendly*. Artinya arsitektur ini memiliki kelebihan dasar yaitu salah satunya dapat di-*index* dengan mudah oleh *search engine* seperti Google, Yahoo, dan Bing. ASP.NET dikembangkan oleh Microsoft yang di-*release* pertama kali pada Januari 2002 dan berlisensi *open source*. *Framework* ini dibangun menggunakan CLR (*Common Language Runtime*) dan dapat menulis *code* untuk ASP.NET menggunakan bahasa pemrograman berbasis .NET seperti C# dan Visual Basic. (Firyan Aula Juyuspan : 2017).

### II.2.8. SQL Server

Microsoft SQL Server 2008 merupakan perangkat lunak *Relational Database Management System* (RDBMS) yang handal dan dirancang untuk mendukung proses transaksi yang besar dengan menggunakan arsitektur client/server. *SQL Server* adalah *software* (perangkat lunak) RDBMS kelas *enterprise* yang banyak digunakan dalam dunia korporat. Dengan *SQL Server* dalam penggunaannya, pemakai dapat menyimpan banyak data dan menerapkannya untuk kepentingan bisnis dalam perusahaan. Untuk menjalankan database relasional, maka kita harus mengenal perintah-perintah SQL (*Structured Query Language*). Dalam MS SQL Server dapat mendukung SQL sebagai bahasa dalam memproses *query* ke dalam *database*. Microsoft SQL Server banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi database atau penyimpanan data.

Perintah SQL merupakan perintah untuk mengolah, menampilkan dan memanipulasi data yang tersimpan dalam sebuah RDBMS. *Data Definition Language* (DDL) adalah merupakan perintah untuk membuat atau mengubah struktur objek *database* seperti *create*, *alter*, *drop*, *truncate* dan *rename*. *Data Control Language* (DCL) merupakan perintah untuk mengatur hak akses *user* dalam *database*, seperti *grant* dan *revoke*. Sedangkan untuk *Data Manipulation Language* (DML) merupakan perintah untuk melakukan manipulasi data seperti menambah, menghapus, dan mengubah data dalam *table*, misalkan *insert*, *delete*, *update* dan *merge*. *Data Manipulation Language* merupakan suatu metode *query* yang dapat dipergunakan bila *Data Definition Language* telah terjadi, sehingga

fungsi dari query DML ini untuk melakukan manipulasi data yang telah dibuat dalam *database*. (Didik Setiyadi, 2020).

### II.2.9 UML (*Unified Modelling Language*)

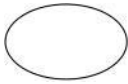
*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

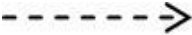
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut (Sinta Maria : 2021) :

#### II.2.9.1. *Use Case Diagram*

Diagram use case merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. Use case digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol diagram use case, seperti yang terlihat pada tabel II.1 :

**Tabel II.1. Simbol *Use Case 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> . |

|                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>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 sistem biasa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> tetapi tidak memiliki <i>control</i> 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 mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem</p>                                                                                                                                                                                                                                                                                          |
|  | <p><i>Include</i>, merupakan didalam <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.</p>                                                                                                                                                                                                                                                    |
|  | <p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat dipenuhi.</p>                                                                                                                                                                                                                                                                                                                                             |

(Sumber : Sinta Maria : 2021)

#### II.2.9.2. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol- simbol class diagram, seperti terlihat pada tabel II.2. berikut ini.

**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 |

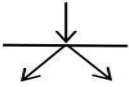
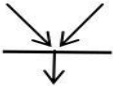
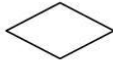
(Sumber : Sinta Maria : 2021)

### II.2.9.2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan actor. Berikut adalah simbol-simbol *activity diagram*, seperti terlihat pada tabel II.3 :

**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.                 |

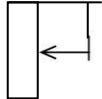
|                                                                                   |                                                                                                                                                            |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <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 : Sinta Maria : 2021)

### II.2.9.3. Sequence Diagram

Diagram sekuen “menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu”. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case (Rosa dan Shalahuddin : 2018). Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel 3 berikut ini

**Tabel II.4. Simbol *Sequence Diagram***

| Gambar                                                                              | Keterangan                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Entity Class</i> , merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system 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 cetak</i> . |
|    | <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 : Sinta Maria : 2021)