



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Sistem

Menurut Romney dan Steinbart dalam Dianty (2016:1), sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Menurut Mulyadi dalam Dianty (2016:1), sistem adalah “suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”.

Dari pengertian diatas dapat disimpulkan bahwa sistem merupakan rangkaian dari komponen-komponen yang saling berhubungan untuk mencapai tujuan suatu organisasi.

##### 2.1.2 Pengertian Web

Menurut Asropudin dalam Utama (2014:9), Web adalah sebuah kumpulan halaman yang diawali dengan halaman muka yang berisikan informasi, iklan, serta program aplikasi.

Menurut Ardhana dalam Utama (2014:9), Web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet).

Dari pengertian diatas penulis menyimpulkan bahwa web adalah suatu layanan atau kumpulan halaman yang berisi informasi, iklan, serta program aplikasi yang dapat digunakan oleh *surfer*.



### 2.1.3 Pengertian Database

Menurut Sutarman dalam Utama (2014:8), Database adalah sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan record-record yang menyimpan data dan hubungan diantaranya.

Menurut Ladjamudin dalam Utama (2014:8), Database adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya.

Dari pengertian diatas penulis menyimpulkan Database adalah sekumpulan file yang saling berhubungan yang menyimpan data dan tersimpan dalam sebuah media penyimpanan.

## 2.2 Teori Judul

### 2.2.1 Sistem Pendukung Keputusan

#### 2.2.1.1 Pengertian Sistem Pendukung Keputusan

Menurut Dewanto dalam Sukatmi (2017:30) Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision System. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditunjukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur.

#### 2.2.1.2 Karakteristik Sistem Pendukung Keputusan

Menurut Dudung (2015), karakteristik dari sistem pendukung keputusan yaitu:

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
- b. Adanya *interface* manusngia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi erstruktur serta mendukung keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan keputusan.



- e. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
- f. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen.

### 2.2.1.3 Tahap Pengambilan Keputusan

Menurut Herbert A. Simon dalam Farida (2016) ada 4 tahap yang harus dilalui dalam proses pengambilan keputusan yaitu :

1. Tahap Pemahaman ( *Intelligence Phace* )

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasikan masalah.

2. Tahap Perancangan ( *Design Phace* )

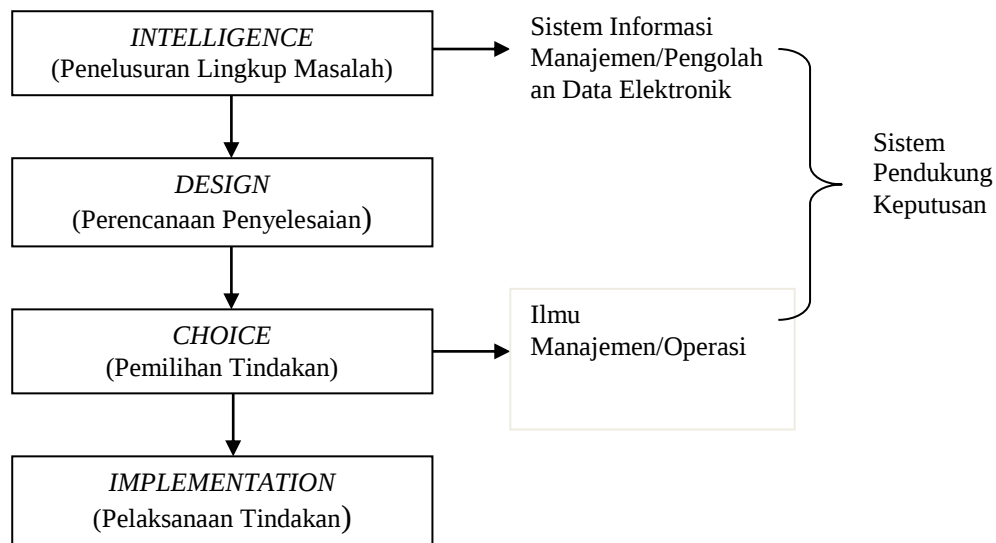
Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan vertifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan ( *Choice Phace* )

Tahap ini dilakukan pemilihan terhadap diantaraberbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan / dengan memperhatikan kriteria – kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi ( *Implementation Phace* )

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.



**Gambar 2.1** Fase Proses Pengambilan Keputusan

Sumber: Kadarsah dan Ali (1998:16)

### 2.2.2 Pengertian Pertanian

Menurut Lubis dalam Suyatmo (2017:2) menyampaikan bahwa pertanian adalah suatu kegiatan yang mengusahakan tanaman pada tanah atau media tumbuh lainnya, untuk dibudidayakan pada lahan. Sebagai negara agraris, Indonesia memiliki banyak pulau dan gunung, sehingga tanaman yang dihasilkan beragam. karena dipengaruhi beberapa faktor di antaranya ketinggian tempat, suhu, kondisi tanah, ph tanah, dan sumber air. sehingga dalam menentukan lokasi tanam disesuaikan berdasarkan kriteria tanaman.

Sedangkan menurut Bukhori dalam Rahman (2016:13) pengertian pertanian yang dalam arti luas tidak hanya mencakup pembudidayaan tanaman saja melainkan membudidayakan serta mengelola dibidang perternakan seperti merawat dan membudidayakan hewan ternak yang bermanfaat bagi pemenuhan kebutuhan masyarakat banyak seperti: ayam, bebek, angsa. Serta pemanfaatan hewan yang dapat membantu tugas para petani kegiatan ini merupakan suatu cakupan dalam bidang pertanian.



Dari pengertian diatas dapat disimpulkan bahwa, pertanian merupakan kegiatan manusia untuk memperoleh hasil yang berasal dari tumbuh-tumbuhan atau hewan guna memenuhi kebutuhan manusia.

## **2.3. Teori Khusus**

### **2.3.1 *Unified Modelling Language (UML)***

#### **2.3.1.1 *Pengertian Unified Modelling Language (UML)***

Menurut Sukamto dan Shalahuddin dalam Paramanindo (2014:3), UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian gambar dan diagram yang sangat baik.

Beberapa diagram memfokuskan diri pada ketangguhan teori objectoriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna.



**Gambar 2.2 Logo UML**

*Sumber : Google*



Menurut Widodo dan Herlawati dalam Paramanindo (2014:4), menjelaskan tentang kegunaan UML sebagai berikut :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

**Tabel 2.1** Tipe Diagram UML

| No. | Diagram              | Tujuan                                                                                                                                  |
|-----|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 1   | <i>Class</i>         | Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.                                   |
| 2   | <i>Package</i>       | Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen.                                                                   |
| 3   | <i>Use case</i>      | Diagram ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (suatu jenis khusus dari kelas)                                     |
| 4   | <i>Sequence</i>      | Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.                                                     |
| 5   | <i>Communication</i> | Sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan. |
| 6   | <i>Statechart</i>    | Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status ( <i>state</i> ), transisi kejadian serta aktivitas.           |
| 7   | <i>Activity</i>      | Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.                |



| No. | Diagram           | Tujuan                                                                                                                  |
|-----|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| 8   | <i>Component</i>  | Memperlihatkan organisasi serta kebergantungan sistem /perangkat lunak pada komponen-komponen yang telah ada sebelumnya |
| 9   | <i>Deployment</i> | Memperlihatkan konfigurasi saat aplikasi di jalankan ( <i>run-time</i> ).                                               |

Sumber: Widodo dan Herlawati dalam Paramanindo (2014:4-5)

### 2.3.1.2 Jenis-Jenis Diagram UML

#### 2.3.1.2.1 Diagram Use case

Menurut Sukanto dan Shalahuddin dalam Paramanindo (2014:5), *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem .

Komponen pembentuk diagram *use case* adalah:

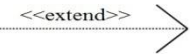
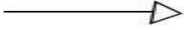
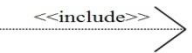
1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

**Tabel 2.2.** Simbol-simbol *Use case* Diagram

| No. | Gambar                                                                              | Nama            | Keterangan                                                                                                |
|-----|-------------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------------------------------------|
| 1   |  | <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |



| No. | Gambar                                                                              | Nama           | Keterangan                                                                                                                                                           |
|-----|-------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2   |    | Actor          | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.                    |
| 3   |    | Association    | Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.                                                |
| 4   |  | Extend         | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.                                |
| 5   |  | Generalization | Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.             |
| 6   |  | Include        | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya sebagai syarat dijalankan use case. |

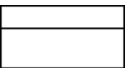
Sumber: Widodo dan Herlawati dalam Paramanindo, F (2014:5-6)



### 2.3.1.2.2 Diagram Class

Menurut Sukanto dan Shalahuddin dalam Paramanindo (2014:7), *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

**Tabel 2.3** Simbol-simbol *Class* Diagram

| No | Gambar                                                                              | Nama                        | Deskripsi                                                                                          |
|----|-------------------------------------------------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------|
| 1  |    | <i>Class</i>                | Kelas pada stuktur sistem.                                                                         |
| 2  |  | <i>Interface</i>            | Sama dengan konsep <i>interfac</i> dalam pemrograman berorientas objek.                            |
| 3  |  | <i>Association</i>          | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 4  |  | <i>Directed association</i> | Relasi antar kelas dengan makna kelas yang satu digunakan ole kelas yang lain.                     |
| 5  |  | <i>Generalization</i>       | Relasi antar kelas dengan makna generalisasi-spesialisasi (umu khusus).                            |
| 6  |  | <i>Dependency</i>           | Relasi antar kelas dengan makna kebergantungan antar kelas.                                        |
| 7  |  | <i>Aggregation</i>          | Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).                                |

Sumber: Widodo dan Herlawati dalam Paramanindo (2014:7)



### 2.3.1.2.3 Diagram Activity

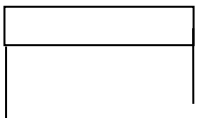
Menurut Sukamto dan Shalahuddin dalam Paramanindo (2014:7), *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

**Tabel 2.4.** Simbol-simbol *Activity Diagram*

| No | Gambar                                                                              | Nama            | Keterangan                                                                                 |
|----|-------------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------------|
| 1  |  | Status Awal     | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.        |
| 2  |  | Aktivitas       | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.             |
| 3  |  | <i>Decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.                    |
| 4  |  | <i>Join</i>     | Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.              |
| 5  |  | Status Akhir    | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |



| No | Gambar                                                                            | Nama     | Keterangan                                                                           |
|----|-----------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------|
| 6  |  | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber: Widodo dan Herlawati dalam Paramanindo (2014:8)

#### 2.3.1.2.4 Diagram Sequence

Menurut Sukanto dan Shalahuddin dalam Paramanindo (2014:9), *sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence* diagram menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal.

Adapun simbol-simbol yang digunakan dalam *sequence diagram* adalah sebagai berikut:

**Tabel 2.5.** Simbol-simbol *Sequence* Diagram

| No | Simbol                                                                              | Nama         | Keterangan                                                  |
|----|-------------------------------------------------------------------------------------|--------------|-------------------------------------------------------------|
| 1  |  | An Actor     | Menggambarkan orang yang sedang berinteraksi dengan sistem. |
| 2  |  | Entity Class | Menggambarkan hubungan kegiatan yang akan dilakukan.        |



| No | Simbol | Nama                      | Keterangan                                                                                              |
|----|--------|---------------------------|---------------------------------------------------------------------------------------------------------|
| 3  |        | <i>Boundary Class</i>     | Menggambarkan sebuah menggambaran dari <i>form</i>                                                      |
| 4  |        | <i>Control Class</i>      | Menggambarkan penghubung antara boundary dengan tabel.                                                  |
| 5  |        | <i>A focus of control</i> | Menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> (pesan).                               |
| 6  |        | <i>A line of life</i>     | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |

Sumber: Widodo dan Herlawati dalam Paramanindo (2014:9)

### 2.3.2 *Perl Hypertext Preprocessor (PHP)*

Menurut Hidayatullah dan Kawistara dalam Mahdiati dan Fridayanthie (2016:130) PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang berintegrasi dengan HTML dan berada pada server (*server side HTML*)



*embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman web dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/up to date. Semua *script* PHP dieksekusi pada server dimana *script* tersebut dijalankan.

Pengertian PHP menurut Kristanto dalam Mahdiati dan Fridayanthie (2016:130), “PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML”. PHP merupakan singkatan dari “PHP : *Perl Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML sekaligus bekerja di sisi server (*server-server HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada halaman HTML biasa, sehingga *script*nya tak tampak disisi client. 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 dibangun oleh PHP pada umumnya akan memberikan hasil kepada web browser, tetapi proses keseluruhannya dijalankan di server.

Adapun kelebihan-kelebihan dari PHP yaitu:

- a. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
- b. PHP dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat open source.
- c. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya.
- d. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.



- e. PHP dapat berjalan pada 3 (tiga) operating system, yaitu: *Linux, unix, dan windows*, dan juga dapat dijalankan secara runtime pada suatu *console* f. PHP bersifat bebas dipakai (*free*).

### 2.3.3 MySQL

#### 2.3.3.1 Pengertian MySQL

Menurut Rosul (2016 : 14), MySQL merupakan database yang dikembangkan dari bahasa SQL (*Structure Query Language*). SQL sendiri merupakan bahasa yang terstruktur yang digunakan untuk interaksi antara script program dengan database server dalam hal pengolahan data. Dengan SQL, kita dapat membuat tabel yang nantinya akan diisi dengan data, memanipulasi data ( misalnya menambah data, menghapus data dan memperbaharui data ), serta membuat suatu perhitungan dengan berdasarkan data yang ditemukan. MySQL merupakan software resmi yng dikembangkan oleh perusahaan Swedia bernama MySQL AB, yang waktu itu bernama TcX Data Konsult AB. Pada awalnya MySQL memakai nama mSQL atau “mini SQL” sebagai antarmuka yang digunakan, ternyata dengan menggunakan mSQL itu mengalami banyak hambatan, yaitu sangat lambat dan tidak *fleksibel*. Oleh karena itu, Michael Widenius berusaha mengembangkan *interface* yang tersebut hingga ditemukan MySQL. Kala itu, MySQL didistribusikan secara khusus, yakni untuk keperluan nonkomersial bersifat gratis.

#### 2.3.3.2 Fungsi – Fungsi MySQL

Menurut Kadir dalam Paramanindo (2014:19), menjelaskan tentang fungsi-fungsi *mysql* sebagai berikut:

**Tabel 2.6** Fungsi-fungsi MySQL

| No | Nama Fungsi              | Kegunaan                                                                                              | Bentuk Pemanggilan                                              |
|----|--------------------------|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| 1  | <i>MySQL_connect()</i>   | Membuat hubungan ke <i>database</i> <i>MySQL</i> yang terdapat pada suatu <i>host</i> .               | <i>mysql_connect (host, nama_pemakai, password)</i>             |
| 2  | <i>MySQL_close()</i>     | Menutup hubungan ke <i>database</i> <i>MySQL</i> .                                                    | <i>mysql_close (pengenal_hubungan)</i>                          |
| 3  | <i>MySQL_select_db()</i> | Memilih <i>database</i>                                                                               | <i>mysql_select_db (database, pengenal_hubungan)</i>            |
| 4  | <i>MySQL_query()</i>     | Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel.                                    | <i>mysql_query (permintaan, pengenal_hubungan)</i>              |
| 5  | <i>MySQL_db_query()</i>  | Menjalankan suatu permintaan terhadap suatu <i>database</i> .                                         | <i>mysql_db_query (database, permintaan, pengenal_hubungan)</i> |
| 6  | <i>MySQL_num_rows()</i>  | Memperoleh jumlah baris dari suatu hasil permintaan ( <i>query</i> ) yang menggunakan <b>SELECT</b> . | <i>mysql_num_rows (pengenal_hasil)</i>                          |



| No | Nama Fungsi                | Kegunaan                                                                                                                      | Bentuk Pemanggilan                                      |
|----|----------------------------|-------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| 8  | <i>MySQL_num_fields()</i>  | Memperoleh jumlah kolom pada suatu hasil permintaan.                                                                          | <i>mysql_num_fields</i> (penenal_hasil).                |
| 9  | <i>MySQL_fetch_row()</i>   | Menghasilkan suatu array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil.                               | <i>mysql_fetch_row</i> (pengenal_hasil)                 |
| 10 | <i>MySQL_fetch_array()</i> | Mempunyai kegunaan serupa dengan <i>mysql_fetch_row()</i> . Hanya saja, setiap kolom akan disimpan dua kali pada array hasil. | <i>mysql_fetch_array</i> (pengenal_hasil)               |
| 11 | <i>MySQL_fetch_field()</i> | Memperoleh informasi suatu kolom.                                                                                             | <i>mysql_fetch_field</i> (pengenal_hasil, nomor_kolom). |
| 12 | <i>MySQL_data_seek()</i>   | Memindah pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu.                                                 | <i>mysql_data_seek</i> (pengenal_hasil, nomor_baris)    |
| 13 | <i>MySQL_field_seek()</i>  | Memindah pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu.                                                 | <i>mysql_data_seek</i> (pengenal_hasil, nomor_kolom)    |
| 14 | <i>MySQL_create_db()</i>   | Menciptakan database MySQL.                                                                                                   | <i>mysql_create_db</i> (database [pengenal_hubungan])   |



| No | Nama Fungsi                | Kegunaan                                           | Bentuk Pemanggilan                                                |
|----|----------------------------|----------------------------------------------------|-------------------------------------------------------------------|
| 16 | <i>MySQL_list_dbs()</i>    | Menghasilkan daftar database MySQL.                | <code>mysql_list_db</code><br>([pengenal_hubungan] )              |
| 17 | <i>MySQL_list_tables()</i> | Memperoleh daftar nama tabel dalam suatu database. | <code>mysql_list_tables</code><br>(database [pengenal_hubungan] ) |
| 18 | <i>MySQL_list_fields()</i> | Memperoleh daftar nama kolom dalam suatu database. | <code>mysql_list_fields</code><br>(database [pengenal_hubungan] ) |

Sumber :Kadir dalam Paramanindo (2014:19-20)

## 2.4 Teori Jurnal

Menurut penelitian yang dilakukan oleh Elvina Lubis pada tahun 2013 dalam jurnal yang berjudul “Sistem Pendukung Keputusan Penentuan Daerah Pertanian Menggunakan Metode SAW”, tujuan penelitian ini adalah membangun sistem pendukung keputusan dengan metode SAW. Penerapan sistem pendukung keputusan dengan metode *simple additive weighting* ini dilakukan berdasarkan hasil dari perangkingan mulai dari alternatif terbesar sampai terkecil. Hasil dari proses sistem pendukung keputusan penentuan kelayakan daerah ini dapat digunakan sebagai bahan pertimbangan untuk menentukan daerah yang dapat dijadikan sebagai daerah pertanian di Kabupaten Asahan.

Menurut penelitian yang dilakukan oleh Dwi Puspitasari dkk pada tahun 2016 dalam jurnal yang berjudul “Sistem Pendukung Keputusan untuk Menentukan Lokasi Pembangunan Minimarket Baru di Kota Bojonegoro dengan Metode TOPSIS Berbasis GIS”, dalam penelitian ini penulis mengimplementasikan metode TOPSIS dalam sistem pendukung keputusan penentuan lokasi



pembangunan minimarket baru berdasarkan kriteria-kriteria yang ada antara lain, jarak antara pasar tradisional dengan lokasi pembangunan *minimarket*, jumlah kompetitor, luas lahan, arus lalu lintas wilayah, dan kepadatan penduduk. Hasil dari penelitian ini dapat membantu proses pengambilan keputusan untuk menentukan lokasi pembangunan *minimarket* baru di Kota Bojonegoro dengan mudah karena pengguna tidak perlu menghitung secara manual dan hasil dari sistem ini menampilkan titik lokasi dalam peta wilayah Kota Bojonegoro, dan urutan nilai dari masing – masing penilaian.

Menurut penelitian yang dilakukan oleh Irfan Abdul Rosul pada tahun 2016 dalam jurnal yang berjudul “Sistem Pendukung Keputusan Penentuan Daerah Pertanian Menggunakan Metode SAW Studi Kasus : Kabupaten Banyuwangi”, penelitian ini bertujuan untuk membangun sistem pendukung keputusan yang dapat mempermudah menentukan daerah pertanian serta mengembangkan sistem penyimpanan data analisis. Hasil dari implementasi sistem ini dengan menggunakan metode SAW didapatkan tingkat akurasi 77,7% dari 24 total data uji yang diterima oleh sistem.

Menurut penelitian yang dilakukan oleh Rossalie Dahniar pada tahun 2016 dalam jurnal yang berjudul “Sistem Pendukung Keputusan Pemilihan Tempat Kuliner dengan Metode TOPSIS Beserta Informasi Geografis Kota Malang”. Sistem berbasis web yang dibuat ini memberikan rekomendasi alternatif sebagai tempat kuliner berdasarkan kriteria menggunakan metode TOPSIS serta pengguna dapat mengetahui lokasi tempat kuliner dan mengetahui rating dari tempat makan dan dapat secara langsung memberikan rating & komentar terhadap tempat makan melalui website.

Menurut penelitian yang dilakukan oleh Mohammad Adiwisanghagni pada tahun 2015 dalam jurnal yang berjudul “Penggunaan Metode TOPSIS Dalam Rancangan Sistem Penunjang Keputusan Untuk Menentukan Lokasi Usaha Baru (Studi Kasus : Arena Disc Yogyakarta)”, metode topsis yang digunakan dalam kasus ini menghasilkan jarak terpendek dari solusi ideal positif serta jarak terpanjang dari solusi ideal negative yang memberikan rekomendasi pemilihan lokasi usaha baru yang ideal dengan menentukan nilai bobot untuk setiap atribut,



---

kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif.