



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Sujatmiko (2012:70), “komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program. Alat serbaguna ini memegang peranan penting dalam teknologi komunikasi”

2.1.2 Pengertian Basis Data (*Database*)

Kawistara et.al. (2015:147) berpendapat, basis data atau *database* dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Menurut pengertian di atas, basis data adalah kumpulan data yang tersimpan di dalam komputer sehingga dapat diolah menjadi informasi.

2.1.3 Pengertian Perangkat Lunak

Sukamto et.al (2014:2) menyatakan “Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Sujatmiko (2012:256) berpendapat “Perangkat lunak adalah kumpulan beberapa perintah yang di eksekusi oleh mesin komputer dalam menjalankan pekerjaannya. Perangkat lunak ini merupakan catatan bagi bagi mesin komputer untuk menyimpan perintah ataupun dokumen”.

2.1.4 Pengertian Aplikasi

Asropudin (2013:7), “aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Wors, Microsoft Excel*”.

Jadi menurut beberapa pengertian di atas, aplikasi merupakan program komputer yang dibuat dengan tujuan membantu manusia dalam mengerjakan tugas tertentu.



2.1.5 Pengertian Penyedia

Menurut Kamus Besar Bahasa Indonesia, “Penyedia adalah orang atau badan dan sebagainya) yang menyediakan suatu barang atau jasa tertentu”.

2.1.6 Pengertian Jasa

Subarjo (2014:204), “Jasa pada dasarnya merupakan suatu kegiatan yang memiliki beberapa unsur ketakberwujudan yang dapat diberikan oleh suatu pihak kepada pihak lainnya dan memberikan berbagai manfaat bagi pihak-pihak yang terkait”.

Lovelock et.al (2010:16), “Jasa adalah suatu aktivitas ekonomi yang ditawarkan oleh satu pihak kepada pihak yang lain”.

Jadi, dapat disimpulkan bahwa jasa adalah suatu kegiatan atau aktivitas yang ditawarkan oleh suatu pihak kepada pihak lain dan memberikan manfaat bagi pihak-pihak yang terkait.

2.1.7 Pengertian Tukang

Menurut Kamus Besar Bahasa Indonesia , “Tukang adalah orang yang ahli dalam bidangnya; orang yang pekerjaannya memperbaiki atau membuat sesuatu”.

Bertan, dkk (2016:14), “Pekerja (Tukang) adalah orang yang langsung mengerjakan pekerjaan dilapangan dalam bidang tertentu sesuai dengan petunjuk kepala tukang”.

Berdasarkan kedua pengertian di atas, dapat disimpulkan bahwa tukang adalah orang yang ahli dibidangnya dan pekerjaannya membuat ataupun memperbaiki sesuatu.

2.1.8 Pengertian Bangunan

Kamus Besar Bahasa Indonesia menerangkan bahwa bangunan adalah sesuatu yang didirikan atau dibangun seperti rumah, gedung dan menara.

2.1.9 Pengertian PT Sahabat Anugerah Sejati.



Pengertian PT Sahabat anugerah Sejati merupakan perusahaan swasta yang bergerak di bidang jasa konstruksi bangunan. PT Sahabat Anugerah Sejati menyediakan layanan jasa konstruksi bidang pembangunan, pemborong pemborongan sarana prasarana bidang telekomunikasi, pembangunan angkutan vertikal, penyelesaian konstruksi gedung, pemasangan komponen bangunan berat, pembangunan konstruksi gedung, jembatan, jalan, bandar udara dan pembangunan.

2.1.10 Pengertian Aplikasi Penyedia Jasa Tukang Bangunan Pada PT Sahabat Anugerah Sejati.

Pengertian Aplikasi Penyedia Jasa Tukang Bangunan Pada PT Sahabat Anugerah Sejati adalah aplikasi untuk memudahkan masyarakat dalam melakukan pencarian tukang dengan sesuai masalah yang dialami dan tercapainya pelayanan kepada masyarakat kota Palembang yang membutuhkan jasa tukang bangunan tersebut.

2.2 Teori Khusus

2.2.1 Data Flow Diagram (DFD)

2.2.1.1 Pengertian Data Flow Diagram (DFD)

Sukamto et al. (2014:70) menyatakan “*Data Flow Diagram* atau dalam bahasa Indonesia menjadi diagram alir data adalah representatif grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

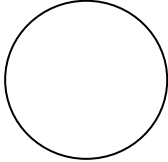
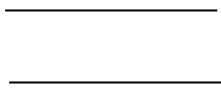

2.2.1.2 Simbol-simbol pada Data Flow Diagram (DFD)

Edward Yourdon dan Tom DeMarco membagi simbol atau notasi pada *Data Flow Diagram* menjadi 4 (empat) buah, yaitu sebagai berikut:

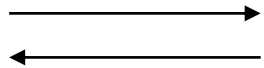
Tabel 2.1 Simbol-simbol pada *Data Flow Diagram*

| NO | Nama dan Simbol | Keterangan |
|----|-----------------|------------|
|----|-----------------|------------|



| | | |
|---|--|---|
| 1 | Proses  | Diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. |
| 2 | Basis data  | Diimplementasikan dengan pemograman terstruktur maka pemodelan inilah yang dibuat menjadi table-tabel basis data yang sesuai dengan table-tabel basis data. |
| 3 | Entitas Luar (<i>External Entity</i>)  | Digunakan untuk menggambarkan asal atau tujuan data. |

Lanjutan Tabel 2.1. Simbol-simbol pada *Data Flow Diagram*

| | | |
|---|--|---|
| 4 | Data Flow  | Digunakan untuk menggambarkan aliran data yang sedang berjalan. |
|---|--|---|

(Sumber: Sukanto et al. 2014:71)

2.2.1.3 Tahapan-tahapan Perancangan *Data Flow Diagram* (DFD)

Dalam merancang suatu diagram alir data, perlu diketahui bahwa diagram alir data memiliki tahapan dalam merancanginya. Adapun tahapan dalam merancang diagram alir data, yaitu:

1. *Data Flow Diagram* (DFD) Level 0 atau *Context Diagram*

Data Flow Diagram Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain ataupun entitas luar.

2. *Data Flow Diagram* (DFD) Level 1

Data Flow Diagram Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. *Data Flow Diagram* Level 1 merupakan hasil *breakdown* dari *Data Flow Diagram* Level 0.

3. *Data Flow Diagram* Level 2



Data Flow Diagram Level 2 merupakan *breakdown* dari *Data Flow Diagram* Level 1. Jumlah *Data Flow Diagram* Level 2 sama dengan modul pada *Data Flow Diagram* Level 1 dalam suatu sistem.

4. *Data Flow Diagram* Level 3 dan seterusnya

Data Flow Diagram Level 3 dan seterusnya merupakan *breakdown* dari *data flow diagram* sebelumnya, namun jarang digunakan karena dapat menyebabkan sistem yang dikembangkan menjadi rumit.

2.2.2 Entity Relationship Diagram (ERD)

2.2.2.1 Pengertian Entity Relationship Diagram (ERD)

Sukamto et al. (2014:50) menyatakan “*Entity Relationship Diagram (ERD)* merupakan bentuk paling awal dalam melakukan perancangan basis data relasional”.

Entity relationship diagram dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* juga memiliki beberapa aliran notasi atau simbol seperti notasi *Chen* (dikembangkan oleh Peter Chen) notasi *Barker* (dikembangkan oleh Richard Barker, Ian Palmer, dan Harry Ellis), notasi *Crow's Foot*, dan juga notasi lainnya.

2.2.2.2 Simbol-simbol pada Entity Relationship Diagram (ERD)

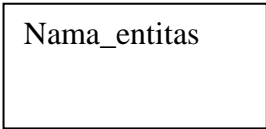
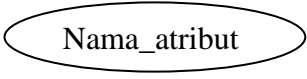
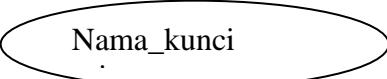
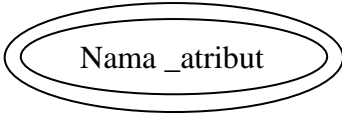
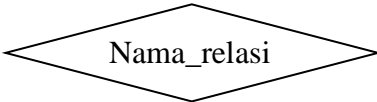
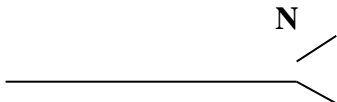
Sukamto et.al (2014:50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*. *ERD* dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *OODBMS* maka perancangan basis data tidak perlu menggunakan *ERD*. *ERD* memiliki beberapa aliran notasi seperti notasi *Chen* (dikembangkan oleh Peter Chen), *Barker* (dikembangkan oleh



Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Sukamto et.al (2014:50) menjelaskan, "Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2.2 Simbol-simbol pada *Entity Relationship Diagram*

| Nama | Simbol | Keterangan |
|--|--|--|
| Entitas/ <i>entity</i> |  | Merupakan data inti yang akan disimpan sehingga dapat diakses oleh aplikasi komputer. |
| Atribut |  | Merupakan <i>field</i> atau kolom data yang akan disimpan dalam suatu entitas. |
| Atribut kunci primer |  | Merupakan <i>field</i> atau kolom data yang disimpan dalam suatu entitas serta menggunakan kunci akses <i>record</i> yang diinginkan ataupun unik. |
| Atribut multinilai/ <i>multivalued</i> |  | Merupakan <i>field</i> atau kolom data yang disimpan dalam suatu entitas dimana memiliki nilai lebih dari satu. |
| Relasi |  | Merupakan relasi yang menghubungkan antar entitas dan biasanya diawali dengan kata kerja. |
| Asosiasi/ <i>association</i> |  | Merupakan penghubung antara relasi dimana kedua ujungnya memiliki <i>multiplicity</i> |



| | | |
|--|--|-------------------------|
| | | dalam jumlah pemakaian. |
|--|--|-------------------------|

(Sumber: Sukanto et al. 2014:50-51)

2.2.3 Block Chart

2.2.3.1 Pengertian Block Chart

Kristanto (2008:75) mengemukakan, “*Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan mengemukakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.



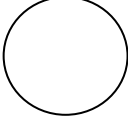
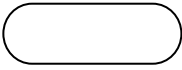
2.2.3.2 Simbol-simbol pada Block Chart

Adapun simbol-simbol dalam *Block Chart* adalah sebagai berikut:

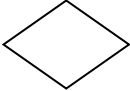


Tabel 2.3 Simbol-Simbol pada *Block Chart*

| No. | Simbol | Keterangan |
|-----|--------|--|
| 1. | | Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan |
| 2. | | Multi dokumen |
| 2. | | Proses manual |
| 4. | | Proses yang dilakukan oleh komputer |
| 5. | | Menandakan dokumen yang diarsipkan (arsip manual) |
| 6. | | Data penyimpanan (<i>data storage</i>) |



| | | |
|-----|---|---|
| 7. |  | Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik |
| 8. |  | Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain |
| 9. |  | Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama |
| 10. |  | Terminasi yang menandakan awal dan akhir dari suatu aliran |

Lanjutan Tabel 2.2. Simbol-Simbol pada *Block Chart*

| | | |
|-----|---|---|
| 11. |  | Pengambilan keputusan (<i>decision</i>) |
| 12. |  | Layar peraga (<i>monitor</i>) |
| 13. |  | Pemasukan data secara manual |

(Sumber: Kristanto, 2008:75-77)

2.2.4. Flowchart

2.2.4.1 Pengertian Flowchart

Ladjamudin (2013:211) mengemukakan bahwa, *flowchart* adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah – langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.



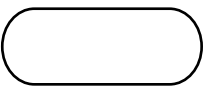
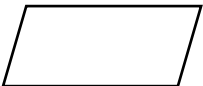
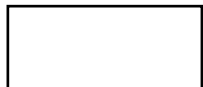
Sitorus (2015:14-15), menjelaskan tentang simbol-simbol yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol-simbol yang bersangkutan dapat dilihat pada tabel berikut ini:

Berdasarkan pengertian di atas, *flowchart* merupakan sebuah alat bantu berbentuk diagram alir yang digunakan untuk menggambarkan urutan prosedur dalam sebuah program.

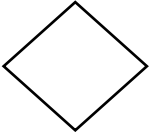
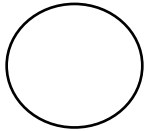
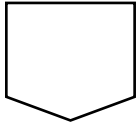

2.2.4.2 Simbol-simbol pada *Flowchart*

Sitorus (2015:14-15), menjelaskan tentang simbol-simbol yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol-simbol yang bersangkutan dapat dilihat pada tabel berikut ini:


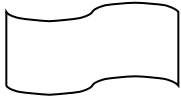

Tabel 2.4 Simbol-simbol pada *Flowchart*

| No | Simbol | Nama | Keterangan |
|----|---|-----------------------|---|
| 1. |  | <i>Terminal</i> | Menyatakan permulaan atau akhir suatu program |
| 2. |  | <i>Input / Output</i> | Menyatakan proses input atau output tanpa tergantung jenis peralatannya |
| 2. |  | <i>Proses</i> | Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer |



| | | | |
|----|--|---------------------------|--|
| 4. |  | <i>Decision</i> | Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya / tidak. |
| 5. |  | <i>Connector</i> | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama |
| 6. |  | <i>Offline Connector</i> | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |
| 7. |  | <i>Predefined Process</i> | Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal. |

Lanjutan Tabel 2.4 Simbol-simbol pada *Flowchart*

| | | | |
|-----|---|---------------------|--|
| 8. |  | <i>Punched Card</i> | Menyatakan input berasal dari kartu atau output ditulis ke kartu |
| 9. |  | <i>Punch Tape</i> | Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang. |
| 10. |  | <i>Document</i> | Mencetak keluaran dalam bentuk dokumen (melalui printer) |



| | | | |
|-----|--|-------------|---------------------------------------|
| 11. | | <i>Flow</i> | Menyatakan jalannya arus suatu proses |
|-----|--|-------------|---------------------------------------|

(Sumber: Siturous, (2015:14-15))

2.2.5 Kamus Data (*Data Dictionary*)

Sukamto et al. (2014:73) menyatakan “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*ouput*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

1. Nama, merupakan nama dari data.
2. Digunakan pada, merupakan proses-proses yang terkait data.
3. Deskripsi, merupakan deskripsi data.
4. Informasi tambahan, merupakan tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol yang bertujuan untuk menjelaskan informasi tambahan. Adapun simbol dari kamus data, yaitu:

Tabel 2.5 Simbol-simbol pada Kamus Data

| Simbol | Keterangan |
|------------------|--------------------------------|
| = | Disusun atau terdiri dari |
| + | Dan |
| [] | Baik...atau... |
| { } ⁿ | N kali diulang/bernilai banyak |
| () | Data opsional |
| *...* | Batas komentar |

(Sumber: Sukamto et al. 2014:74)



2.2.6 Pengertian Waterfall

Sukanto et al. (2014:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

2.3 Teori Program

2.3.1 Pengertian Android

Kadir, (2013:2), “Android Adalah sistem operasi yang bersifat open source untuk digunakan pada ponsel, netbook, e-reader, dan tablet.”

Eva et al, (2016:2), “Android adalah sebuah sistem operasi untuk smartphone dan tablet, sistem operasi sebagai ‘jembatan’ antara peranti (device) dan penggunanya, sehingga pengguna dapat berinteraksi dengan device-nya dan menjalankan aplikasi-aplikasi yang tersedia pada device.”

Kesimpulannya, Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet.

2.3.2 Pengertian Java

Hariyanto (2014:1), “Java merupakan bahasa pemrograman objek untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet*, aplikasi untuk perangkat cerdas yang dapat berkomunikasi *internet*/jaringan komunikasi.”



Naughton (2002:1), “Java adalah bahasa untuk menciptakan program yang aman, portabel, kokoh, berorientasi-objek, multi-jalanan, dan interaktif”

Berdasarkan kedua pengertian di atas, Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

2.3.3 Pengertian *PHP*

Winarno, dkk. (2014:49) menjelaskan, *PHP* atau *Hypertext Preprocessor*, adalah sebuah bahasa pemrograman *web* berbasis server (server-side) yang mampu memarsing kode *PHP* dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan website yang dinamis di sisi client (browser).

Badiyanto (2013:32) “*PHP* : *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML/PHP* banyak dipakai untuk membuat situs web dinamis”.

Jadi menurut beberapa pendapat di atas, *PHP (Hypertext Preprocessor)* adalah sebuah bahasa pemrograman berbasis *web* yang digunakan untuk membuat halaman *HTML*.

2.3.3.1 *Script PHP*

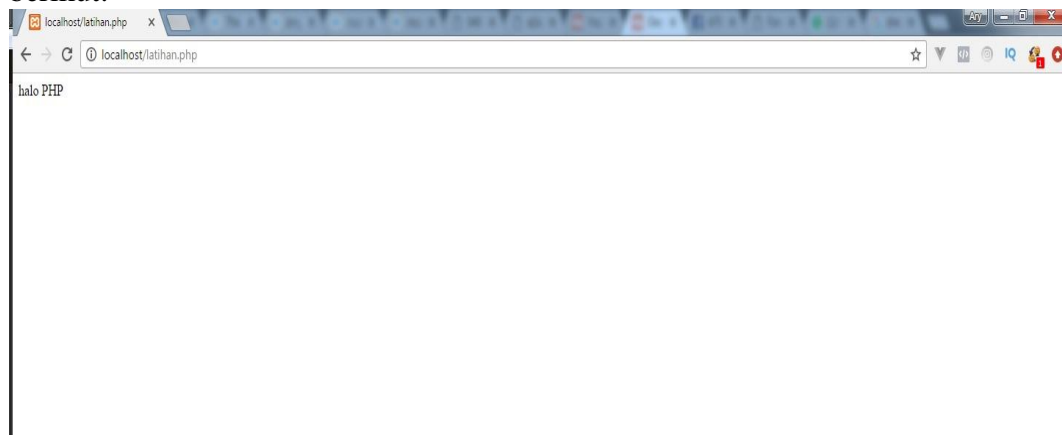
PHP yang merupakan sebuah bahasa scripting yang terpasang pada *HTML*, dan skrip *HTML (Hyper Text Markup Language)* adalah bahasa standar untuk membuat halaman *web*. Biasa *file* dituliskan dengan ekstensi *.htm* atau *.html*.

```
<html>
  <head>
    <title>Latihan PHP</title>

  </head>
  <body>
    halo PHP
  </body>
</html>
```



Bila dijalankan melalui *browser*, kode diatas akan menampilkan hasil seperti berikut:



Gambar 2.1 Tampilan program HTML pada *browser*

2.3.3.2 Teknik Penulisan Script *PHP*

Zaki et al. (2014:24), block kode *PHP* dimulai dengan sintak tag `<?PHP` dan di akhiri dengan `?>`, berikut adalah contoh blok kode *PHP*:

```
<?PHP
//kode tertentu diisikan di sini
?>
```

Selain itu ada beberapa tag lain yang bisa di pakai, seperti:

Tabel 2.6 Script Dasar *PHP*

| Jenis Tag | Tag Pembuka | Tag Penutup |
|-------------|--|------------------------------|
| Tag standar | <code><?PHP</code> | <code>?></code> |
| Tag pendek | <code><?</code> | <code>?></code> |
| Tag ASP | <code><%</code> | <code>%></code> |
| Tag script | <code><script language="PHP"></code> | <code></script></code> |

(Sumber: Zaki et al. 2014:24-25)

Echo sering digunakan di dalam *PHP*. Fungsinya untuk meng-ouput data. Berikut ini contoh penggunaan echo:

```
<?PHP
Echo '<p> selamat datang </p>';
?>
```



Di dalam *script PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah *script*. Beberapa cara untuk memberi keterangan di dalam *script PHP* adalah sebagai berikut:

- 1) Gunakan *tag /** dan akhiri *tag */* apabila jumlah keterangan lebih dari 1 baris.
- 2) Gunakan *tag //*, *tag* ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
- 3) Gunakan *tag #*, *tag* ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah *script PHP* selalu diakhiri dengan menuliskan tanda titik koma (;). Satu baris *script PHP* tidak harus berada dalam satu baris, melainkan dapat menuliskan perintah *script PHP* lebih dari satu baris.

2.3.4 Pengertian MySQL

Sukamto et al. (2014:46), “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus”.

Winarno (2014:102), “MySQL adalah sebuah software database. MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”.

Berdasarkan , MySQL merupakan sebuah bahasa yang digunakan untuk mengelolah data dalam bentuk tabel-tabel yang saling berhubungan.

2.3.5.1 Fungsi-Fungsi MySQL

Sejumlah fungsi yang berawalan **mysql_** yang digunakan untuk mengakses *database server MySQL* adalah sebagai berikut:

Tabel 2.7 Fungsi-fungsi MySQL

| No. | Fungsi MySQL | Penggunaan |
|-----|--------------------------|---|
| 1 | <i>Mysql_connect()</i> | Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu host |
| 2 | <i>Mysql_close()</i> | Menutup hubungan ke <i>database MySQL</i> |
| 3 | <i>Mysql_select_db()</i> | Memilih <i>database</i> |
| 4 | <i>Mysql_query()</i> | Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel |



| | | |
|----|------------------------------|--|
| 5 | <i>Mysql_db_query()</i> | Menjalankan suatu permintaan terhadap suatu <i>database</i> |
| 6 | <i>Mysql_num_rows()</i> | Memperoleh jumlah baris dari suatu hasil permintaan yang menggunakan <i>SELECT</i> |
| 7 | <i>Mysql_affected_rows()</i> | Memperoleh jumlah baris yang dikenai operasi <i>INSERT, DELETE, UPDATE</i> |
| 8 | <i>Mysql_num_fields()</i> | Memperoleh jumlah kolom pada suatu hasil permintaan |
| 9 | <i>Mysql_fetch_row()</i> | Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil |
| 10 | <i>Mysql_fetch_array()</i> | Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasilnya akan disimpan dua kali pada <i>array</i> hasil |
| 11 | <i>Mysql_fetch_field()</i> | Menghasilkan informasi suatu kolom |

Lanjutan Tabel 2.7. Fungsi-fungsi *MySQL*

| | | |
|----|----------------------------|---|
| 12 | <i>Mysql_data_seek()</i> | Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu |
| 13 | <i>Mysql_field_seek()</i> | Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu |
| 14 | <i>Mysql_create_db()</i> | Membuat database <i>MySQL</i> |
| 15 | <i>Mysql_drop_db()</i> | Menghapus database <i>MySQL</i> |
| 16 | <i>Mysql_list_dbs()</i> | Menghasilkan daftar database <i>MySQL</i> |
| 17 | <i>Mysql_list_tables()</i> | Memperoleh daftar nama tabel dalam suatu database |
| 18 | <i>Mysql_list_fields()</i> | Memperoleh daftar nama kolom dalam suatu database |

(Sumber: Abdul Kadir (2008:360-379))

2.3.5.2 Alasan Menggunakan *MySQL*

Berikut ini beberapa alasan mengapa mereka memilih *MySQL* sebagai *server database*

1. Fleksibel. *MySQL* dapat digunakan untuk mengembangkan aplikasi *desktop* maupun aplikasi *web* dengan menggunakan teknologi yang bervariasi. Ini



berarti bahwa *MySQL* memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi, apakah itu *PHP*, *JSP*, *Java*, *Delphi*, *C++*, maupun yang lainnya dengan cara menyediakan *plug-in* dan *driver* yang spesifik untuk masing-masing teknologi tersebut.

2. Performa Tinggi. *MySQL* memiliki *query* performa tinggi dengan demikian proses transaksional dapat dilakukan dengan sangat cepat.
3. Lintas Platform. *MySQL* dapat digunakan pada *platform* atau lingkungan sistem operasi yang beragam.
4. Gratis. *MySQL* dapat digunakan secara gratis.
5. Proteksi Data yang Handal. Perlindungan terhadap keamanan data merupakan hal nomor satu yang dilakukan oleh para profesional di bidang *database*. *MySQL* menyediakan mekanisme yang powerful untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.