



---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

Adapun dalam penyusunan laporan akhir ini penulis akan menuliskan landasan teori, pengertian-pengertian yang terdapat pada penulisan laporan ini adalah sebagai berikut:

##### 2.1.1 Pengertian Komputer

Robert H.Blissmer (Dikutip Sutarman, 2012:3), Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas sebagai berikut:

1. Menerima *input*,
2. Memproses *input* tadi sesuai dengan programnya,
3. Menyimpan perintah-perintah dan hasil dari pengolahan,
4. Menyediakan *output* dalam bentuk informasi.

Donald H.Sanders (Dikutip Sutarman, 2012:3), Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan agar secara otomatis menerima dan menyimpan data *input*, memprosesnya dan menghasilkan *output* dibawah pengawasan suatu langkah-langkah instruksi program yang tersimpan pada memori (*stored program*).

Wahyudi (2012:3), Komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*).

Hartono (2013:27), Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan perhitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis .



### 2.1.2 Pengertian Sistem

Kristanto (2008:1), Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Sutarman (2012:5), Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

Al-Bahra (2013:6), Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya.

### 2.1.3 Pengertian Data

Sutarman (2012:3), Data adalah fakta dari sesuatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengujian/pengamatan.

Hartono (2013:15), Data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan atau kejadian.

Al-Bahra (2013:8), Data adalah deskripsi dari sesuatu dan kejadian yang kita hadapi (*the description of things and events that we face*).

Pipin (2013:22), Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.

### 2.1.4 Pengertian Aplikasi

Hendrayudi (2009:143), Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).

Sutabri (2012:147), Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.



---

Sutarman (2012:145), Aplikasi merupakan *software* dioperasikan untuk keperluan tertentu, seperti bahasa pemrograman, permainan dan aplikasi olah kata, angka, dan sebagainya.

Pipin (2013:6), Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*.

### **2.1.5 Pengertian Pengolahan Data**

Kristanto (2008:8), Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Sutarman (2012:4), Pengolahan data adalah proses perhitungan/transformasi data input menjadi informasi yang mudah dimengerti ataupun sesuai yang diinginkan.

Al-Bahra (2013:9), Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

### **2.1.6 Pengertian Ekspedisi**

Kamus Besar Bahasa Indonesia (2009:382), Ekspedisi adalah pengiriman sesuatu (surat, barang, dan sebagainya) ke suatu tempat.

Ekspedisi adalah Perjalanan yang dilakukan untuk tujuan tertentu, biasanya penjelajahan dan/atau penelitian.

(<http://id.wikipedia.org/wiki/Ekspedisi>, diakses pada tanggal 13 Mei 2014)

Ekspedisi adalah pengiriman surat, barang atau perusahaan pengangkutan barang.

(<http://artikata.com/arti-326100-ekspedisi.html>, diakses pada tanggal 13 Mei 2014)



---

### **2.1.7 Barang**

Kamus Besar Bahasa Indonesia (2009:139), barang adalah benda umum (segala sesuatu yang berwujud atau berjasad), cair, keras.

Barang atau komoditas dalam pengertian ekonomi adalah suatu objek atau jasa yang memiliki nilai.

(<http://id.wikipedia.org/wiki/Barang>, diakses pada tanggal 13 Mei 2014)

Barang adalah benda-benda yang berwujud, yang digunakan masyarakat untuk memenuhi kebutuhannya atau untuk menghasilkan benda lain yang akan memenuhi kebutuhan masyarakat.

(<http://axellelessons.blogspot.com/2012/05/pengertian-barang-dan-jasa.html>, diakses pada tanggal 13 Mei 2014)

### **2.1.8 Pengertian Aplikasi Pengolahan Data Ekspedisi Barang pada CV.Mulya Karya Palembang**

Merupakan sebuah aplikasi yang menghimpun data-data ekspedisi barang yang dapat memudahkan bagian administrasi dalam mengelola data dan mengecek data ekspedisi barang pada CV.Mulya Karya Palembang.

## **2.2 Teori Khusus**

Teori ini akan membahas mengenai desain sistem yang merupakan suatu fase dimana diperlukan suatu keahlian perencanaan untuk elemen-elemen komputer yang akan menggunakan sistem baru. Ada beberapa alat bantu yang digunakan dalam desain sistem yaitu:

### **2.2.1 Pengertian Diagram Aliran Data/*Data Flow Diagram* (DFD)**

Kristanto (2008:61), DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.



Al-Bahra (2013:64), Diagram Aliran Data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

### 2.2.1.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem.

### 2.2.1.2 Diagram Nol/ Zero

Diagram nol adalah diagram yang menggambarkan proses dari dataflow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data dan eksternal entity.

### 2.2.1.3 Diagram Rinci (Level Diagram)

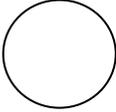
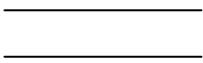
Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram level di atasnya.

Kristanto (2008:62), simbol DFD yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

**Tabel 2.1** Simbol-simbol *data flow diagram*

| No. | Nama        | Simbol De Marco and Jourdan   | Arti  |
|-----|-------------|---|---|
| 1.  | Entiti luar |  | Simbol ini digunakan untuk menggambarkan sumber atau tujuan dari aliran data dari atau ke sistem. |



|    |                            |   |  |
|----|----------------------------|---|--|
| 2. | Aliran data                |  | Simbol ini digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya. |
| 3. | Proses                     |  | Simbol ini digunakan untuk mentransformasikan data secara umum.                          |
| 4. | Berkas/ tempat penyimpanan |  | Simbol ini digunakan untuk menyimpan data tau file.                                      |

Sumber: Kristanto (2008:62)

Dalam penggambaran simbol DFD, ada beberapa peraturan yang harus diperhatikan sehingga dalam penggambarannya tidak terjadi kesalahan.

1. Antar entity luar tidak diijinkan terjadi hubungan relasi.
2. Tidak boleh ada aliran data antara entiti luar dengan *data store*.
3. Untuk alasan kerapian, entiti luas atau *data store* boleh digambar beberapa kali dengan tanda khusus, misalnya diberi nomor.
4. Satu aliran data boleh mengalirkan beberapa struktur data.
5. Bentuk anak panah aliran data boleh bervariasi.
6. Semua *obyek* harus mempunyai nama.
7. Aliran data selalu diawali dan diakhiri dengan proses.
8. Semua aliran data harus mempunyai tanda arah.

### 2.2.2 Pengertian *Flowchart*

Al-Bahra (2013:263), *Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.

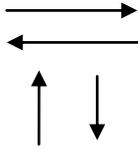
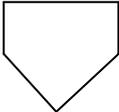
*Flowchart* disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses didalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut:



### 1. *Flow Direction Symbols* (Simbol penghubung/alur)

Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan yang lain. Simbol ini disebut juga *connecting line*. Simbol-simbol tersebut sebagai berikut:

**Tabel 2.2** *Flow Direction Symbols*

| No. | Nama symbol                         | Simbol  | Arti   |
|-----|-------------------------------------|---|--|
| 1.  | Simbol arus/<br><i>flow</i>         |    | Untuk menyatakan jalannya arus suatu proses.   |
| 2.  | Simbol<br><i>communication link</i> |    | Untuk menyatakan bahwa adanya transisi suatu data/informasi dari suatu lokasi ke lokasi lainnya. |
| 3.  | Simbol<br><i>connector</i>          |  | Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/ lembar yang sama.   |
| 4.  | Simbol <i>offline connector</i>     |  | Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/ lembar yang berbeda |

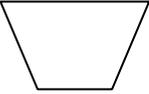
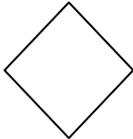
Sumber: Al-Bahra (2013:266)

### 2. *Processing Symbols* (Simbol proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur. Simbol-simbol tersebut adalah sebagai berikut:



Tabel 2.3 Processing Symbols

| No | Nama symbol                     | Simbol  | Arti  |
|----|---------------------------------|---|---|
| 1. | Simbol <i>offline connector</i> |    | Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/ lembar yang berbeda.                   |
| 2. | Simbol manual                   |    | Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).                               |
| 3. | Simbol <i>decision/ logika</i>  |    | Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.                  |
| 4. | Simbol <i>predefined proses</i> |  | Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk member harga awal.                            |
| 5. | Simbol terminal                 |  | Untuk menyatakan permulaan atau akhir suatu program.  |
| 6. | Simbol <i>keying operation</i>  |  | Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> . |
| 7. | Simbol <i>off-line storage</i>  |  | Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.                                |
| 8. | Simbol manual <i>input</i>      |  | Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .                                     |

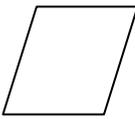
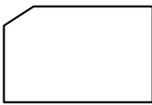
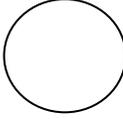
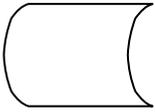
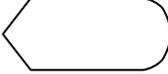
Sumber: Al-Bahra (2013:267)



### 3. *Input-output symbols* (Simbol *input-output*)

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Simbol-simbol tersebut adalah sebagai berikut:

**Tabel 2.4** *Input-output symbols*

| No | Nama symbol                      | Simbol  | Arti  |
|----|----------------------------------|---|---|
| 1. | Simbol <i>input-output</i>       |    | Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.                    |
| 2. | Simbol <i>punched card</i>       |    | Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.                                 |
| 3. | Simbol <i>magnetic-tape unit</i> |  | Untuk menyatakan <i>input</i> berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> . |
| 4. | Simbol <i>disk storage</i>       |  | Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .                   |
| 5. | Simbol <i>document</i>           |  | Untuk mencetak laporan ke printer.  |
| 6. | Simbol <i>display</i>            |  | Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).                               |

Sumber: Al-Bahra (2013:268)



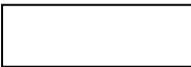
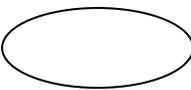
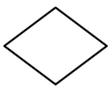
### 2.2.3 Pengertian *Entity Relationship Diagram*/ *Diagram Hubungan Antara Entitas (ERD)*

Kristanto (2008:91), ERD adalah gambaran dunia nyata diistilahkan dalam objek dan relasinya. Teknik ini biasa digunakan untuk mengembangkan inisial dan desain basis data.

Al-Bahra (2013:142), ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak.

Adapun simbol-simbol dari *Entity Relationship Diagram* adalah sebagai berikut:

**Tabel 2.5** Simbol-simbol *Entity Relationship Diagram* (ERD)

| No. | Nama symbol            | Simbol  | Arti   |
|-----|------------------------|---|--|
| 1.  | <i>Entitas</i>         |  | Menyatakan <i>himpunan entitas/ entitas</i> .  |
| 2.  | <i>Atribut</i>         |  | Menyatakan <i>atribut (atribut yang berfungsi sebagai key digarisbawahi)</i> .   |
| 3.  | <i>Himpunan relasi</i> |  | Menyatakan <i>himpunan relasi/ relasi</i> .  |
| 4.  | <i>Link</i>            |  | Penghubung antara <i>himpunan relasi</i> dengan <i>himpunan entitas</i> dan <i>himpunan entitas</i> dengan <i>atributnya</i> . |

Sumber: Al-Bahra (2013:147-148)

*Entity Relationship Diagram* memiliki elemen-elemen diagram hubungan entitas yaitu:

#### 1. *Entity*

*Entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata abstrak dimana data tersimpan atau dimana terdapat data.



## 2. *Relationship*

*Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya.

## 3. *Relationship Degree*

*Relationship Degree* adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*. Derajat *relationship* yang sering dipakai di dalam *ERD* yaitu:

- a. *Unary Relationship*
- b. *Binary Relationship*
- c. *Tertiary Relationship*

## 4. *Atribut*

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas manapun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa yang sebenarnya yang dimaksud dengan entitas maupun *relationship*, sehingga sering dikatakan atribut adalah element adalah element dari setiap entitas dan *relationship*.

## 5. *Kardinalitas (Cardinality)*

*Kardinalitas* relasi menunjukkan jumlah maksimum tupel yang dapat belasi dengan entitas yang lain. *Kardinalitas* relasi menunjukkan kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu:

### a. *One To One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya, yang berarti setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B dan

---



begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

b. *One To Many atau Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian untuk entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. *Many To Many*

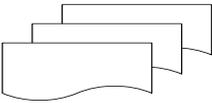
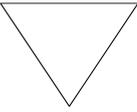
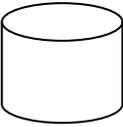
Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas yang lainnya. Baik dilihat dari sisi entitas yang pertama, maupun yang dilihat dari sisi kedua, yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas B.

#### 2.2.4 Pengertian *Blockchart*

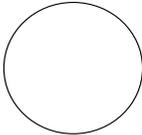
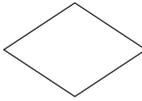
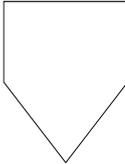
Kristanto (2008:75) menyatakan *blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol digunakan dalam *blockchart* adalah sebagai berikut:

**Tabel 2.6** Simbol-simbol *blockchart*

| No. | Nama Simbol             | Simbol  | Arti  |
|-----|-------------------------|---|---|
| 1.  | <i>Document</i>         |    | Simbol yang digunakan dalam menandakan suatu dokumen, bisa dalam bentuk surat, formulir, berkas atau cetakan.             |
| 2.  | <i>Multi Document</i>   |    | Suatu simbol yang digunakan dalam menandakan suatu multi dokumen, bisa dalam bentuk surat, formulir, berkas atau cetakan. |
| 3.  | <i>Manual Operation</i> |  | Suatu simbol yang digunakan dalam suatu proses yang dilakukan secara manual.  |
| 4.  | <i>Process</i>          |  | Suatu proses yang dilakukan dengan menggunakan komputer.  |
| 5.  | <i>Merge</i>            |  | Simbol yang digunakan dalam dokumen yang menandakan dokumen diarsipkan (arsip manual).                                    |
| 6.  | <i>Magnetik Disk</i>    |  | Suatu simbol yang digunakan dalam data penyimpanan ( <i>data storage</i> ).   |
| 7.  | <i>Tampilan</i>         |  | Layar peraga (monitor)  |



|     |                           |   |   |
|-----|---------------------------|---|---|
| 8.  | <i>Connector</i>          |    | Suatu simbol yang digunakan dalam terminasi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama. |
| 9.  | <i>Alternate Process</i>  |    | Simbol yang digunakan dalam terminasi yang menandakan awal; atau akhir dari suatu aliran.   |
| 10. | <i>Decision</i>           |    | Suatu simbol yang digunakan dalam pengambilan keputusan akhir.  |
| 11. | <i>Off-Page Connector</i> |  | Suatu simbol yang digunakan dalam terminasi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain. |

Sumber: Kristanto (2008:75-77)

### 2.2.5 Kamus Data (*Data Dictionary*)

Kristanto (2008:72), Kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file di dalam sistem. Simbol-simbol yang ada dalam kamus data adalah sebagai berikut:

**Tabel 2.7** Simbol-simbol dalam Kamus Data

| No. | Simbol | Arti         |
|-----|--------|--------------|
| 1.  | =      | Terdiri atas |
| 2.  | +      | Dan          |
| 3.  | ()     | Opsional     |



|    |     |                               |
|----|-----|-------------------------------|
| 4. | [ ] | Memilih salah satu alternatif |
| 5. | **  | Komentar                      |
| 6. | @   | Identifikasi atribut kunci    |
| 7. |     | Pemisah alternatif simbol [ ] |

Sumber: Kristanto (2008:72)

## 2.3 Teori Program

### 2.3.1 Mengetahui *PHP (Hypertext Preprocessor)*

Andi (2011:216), PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah webserver dan berfungsi sebagai pengolah data pada sebuah server.

Kode-kode bahasa PHP dalam penulisannya menyatu dengan tag-tag HTML dalam satu file. Kode PHP diletakkan antara tanda `<? atau <?php` dan diakhiri dengan tanda `?>`, sebagai identitas bahasa pemrograman PHP.

Abdul Kadir (2013:17), PHP adalah bahasa pemrograman yang ditujukan untuk kepentingan pembuatan aplikasi *web*.

### 2.3.2 Mengetahui *Adobe Dreamweaver CS5*

#### 2.3.2.1 Pengertian *Adobe Dreamweaver CS5*

Andi (2011:2), *Dreamweaver* merupakan software utama yang digunakan adalah sebuah *HTML* editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web. Saat ini terdapat *software* dari kelompok *Adobe* yang belakangan banyak digunakan untuk mendesain suatu situs web.

Salah satu versi dari *Adobe Dreamweaver* adalah *Adobe Dreamweaver CS5*. Pada *Adobe Dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk desain web saja tetapi juga untuk menyunting kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman web, antara lain : *JPS, PHP, ASP* dan *Coldfusion*.



---

*Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* maupun *Web Programmer* dalam mengembangkan suatu situs *web*. Hal ini disebabkan ruang kerja, fasilitas dan kemampuan *Dreamweaver* yang mampu meningkatkan produktifitas dan efektivitas dalam desain maupun membangun suatu situs *web*.

### 2.3.2.2 Elemen Kerja *Dreamweaver CS5*

Andi (2011:3-11), elemen-elemen kerja pada *Dreamweaver CS5* adalah sebagai berikut:

a. *Application Bar*

Berada dibagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace* (*workspace switcher*), menu dan aplikasi lainnya.

b. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *Design* atau tampilan *Code*. Selain itu juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.

c. Jendela Dokumen

Lembar kerja tempat membuat dan mengedit desain halaman *web*.

d. *Workspace Switcher*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Dreamweaver CS6*. Sebagai contoh mengubah tampilan menjadi tampilan *classic*, yaitu tampilan ruang kerja *Dreamwaver* versi sebelumnya.

e. *Panel Groups*

Kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara default, panel *groups* berisi panel *Insert*, *CSS Styles*, *Asset*, *AP Element* dan *Files*.



f. *Tag Selector*

Terletak dibagian bawah jendela dokumen, satu baris dengan status bar. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja *Design*.

g. *Panel Property*

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*. Properti untuk satu objek dengan objek lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela kode.

h. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi code-code standar. Jendela ini hanya tampil pada jendela *Code*.

i. *Panel Insert*

Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.

j. *Panel Groups*

*Panel Groups* adalah kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul, seperti: *Insert*, *Files*, *CSS Style* dan sebagainya.

k. *Panel Files*

Digunakan untuk mengatur file-file dan folder-folder yang membentuk situs *web* yang dibuat. Sebagai contoh mengcopy, memindahkan, atau mengganti nama file.



---

### 2.3.3 Mengenal MySQL

#### 2.3.3.1 Pengertian MySQL

Kadir (2008:22), *SQL* adalah kependekan dari *Structured Query Language*. Bahasa ini merupakan standar yang digunakan untuk mengakses database relasional.

Kadir (2008:2-3), *MySQL* merupakan software yang tergolong sebagai *DBMS (Database Management System)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di internet secara gratis.

Sebagai software *DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan dibawah ini:

1. *Multiplatform*

*MySQL* tersedia pada beberapa platform (*Windows, Linux, Unix*, dan lain-lain).

2. Andal, cepat, dan mudah digunakan

*MySQL* tergolong sebagai *database sever* (server yang melayani permintaan terhadap database) yang andal, dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan. Berbagai pendukung *tool* juga tersedia. Perlu diketahui, *MySQL* dapat menangani sebuah tabel yang berukuran dalam terabyte (1 *terabyte* = 1024 *gigabyte*). Namun, ukuran yang sesungguhnya dapat bergantung pada batasan sistem operasi. Sebagai contoh pada sistem *Solaris 9/10*, batasan ukuran file sebesar 16 *terabyte*.

3. Jaminan keamanan akses

*MySQL* mendukung pengamanan database dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia, sedangkan *user* lain tidak



boleh. *MySQL* juga mendukung konektivitas ke berbagai *software*. Sebagai contoh, *MySQL* bisa diakses melalui aplikasi berbasis *web*; misalnya dengan menggunakan *PHP*.

#### 4. Dukungan *SQL*

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL* (*Structured Query Language*). Sebagaimana diketahui, *SQL* merupakan standar dalam pengaksesan *database* relasional. Pengetahuan akan *SQL* akan memudahkan siapapun untuk menggunakan *MySQL*.

Secara umum, *Database* merupakan koleksi data yang saling terkait, sehingga basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat.

Sesungguhnya ada beberapa macam *database*, antara lain yaitu *database* hierarkis, *database* jaringan, dan *database* relasional. *Database* relasional pada saat ini telah diterapkan pada berbagai *platform*, dari *PC* hingga minikomputer.

#### 2.3.3.2 Istilah Kunci Primer dan Kunci Asing

Kadir (2008:6-7), menurut konsep *database relasioal*, setiap tabel memiliki sebuah kunci primer (*primary key*), walaupun dalam praktiknya bisa saja tidak memilikinya. Kunci primer adalah suatu nilai yang bersifat unik (tidak ada yang kembar) sehingga dapat digunakan untuk membedakan suatu baris dengan baris yang lain dalam sebuah tabel.

Pada *database* relasional, dikenal pula dengan istilah kunci asing (*foreign key*) atau sering juga disebut kunci tamu. Kunci tamu adalah sembarang kolom yang menunjuk ke kunci primer milik tabel lain.

#### 2.3.3.3 Hubungan

Kadir (2008:7), pada model *database* relasional, kaitan atau asosiasi antara dua buah tabel disebut hubungan (*relationship*). Hubungan dapat berupa:



- a. 1 - 1, yakni satu data pada suatu tabel berpasangan dengan hanya satu data pada tabel lain.
- b. 1 - M, yakni satu data pada suatu tabel berpasangan dengan banyak data pada tabel lain.

### 2.3.3.4 Berbagai Tipe Data Pada *MySQL*

Kadir (2008:48-53), setiap kolom dalam tabel didefinisikan dengan melibatkan tipe kolom. Tipe kolom merupakan suatu bentuk kekangan data yang akan menentukan jenis data yang akan dimasukkan dalam suatu kolom. Secara garis besar *MySQL* menyediakan beberapa kategori tipe kolom, yaitu sebagai berikut:

- a. Tipe Numeris

Tipe numeris adalah tipe yang menyatakan bilangan. Berikut adalah tabel tipe numeris:

**Tabel 2.8** Tipe Numeris

| No. | Tipe Data    | Keterangan   | Kebutuhan Memori         |
|-----|--------------|--|--------------------------|
| 1.  | BIT          | Menyatakan tipe bit. Satu bit dapat berupa bilangan 0 atau 1.<br>a. BIT identik dengan BIT(1) yang berarti 1 buah bit<br>b. BIT(2) berarti mengandung 2 bit<br><br>Angka maksimal di dalam kurung yaitu 64, yang berarti 64 bit. | $(M/7)/8$<br><i>byte</i> |
| 2.  | TINYINT[(M)] | Menyatakan data bilangan bulat yang berkisar antara -128 sampai 127. Bila ditambahkan kata UNSIGNED (berarti bilangan tidak bertanda),   | 1 <i>byte</i>            |



|    |  |   |        |
|----|--|---|--------|
|    |  | jangkauan bilangan berkisar dari 0 sampai dengan 127.   |        |
| 3. | BOOL atau<br>BOOLEAN, atau<br>TINYINT(1) | Untuk menyatakan nilai benar dan salah. Nilai nol dianggap salah dan nilai selain nol dianggap benar.   | 1 byte |
| 4. | SMALLINT[(M)]                            | Menyatakan data bilangan bulat yang berkisar antara -32768 sampai 32767. Bila ditambahkan kata UNSIGNED (berarti bilangan tidak bertanda), jangkauan bilangan berkisar dari 0 sampai dengan 65535.                | 2 byte |
| 5. | MEDIUMINT[(M)]                           | Menyatakan data bilangan bulat yang berkisar antara -8388608 sampai 8388607. Bila ditambahkan kata UNSIGNED (berarti bilangan tidak bertanda), jangkauan bilangan berkisar dari 0 sampai dengan 16777215          | 3 byte |
| 6. | INT[(M)]atau<br>INTEGER[(M)]             | Menyatakan data bilangan bulat yang berkisar antara -2147483648 sampai 2147483647. Bila ditambahkan kata UNSIGNED (berarti bilangan tidak bertanda), jangkauan bilangan berkisar dari 0 sampai dengan 4294967295. | 4 byte |
| 7. | BEGINT[(M)]                              | Menyatakan data bilangan bulat yang berkisar antara -9223372036854775808 sampai   | 6 byte |



|    |   |   |        |
|----|---|---|--------|
|    |   | 9223372036854775807. Bila ditambahkan kata UNSIGNED (berarti bilangan tidak bertanda), jangkauan bilangan berkisar dari 0 sampai dengan 18446744073709551615  |        |
| 8. | FLOAT[(M,D)]                                    | <p>Tipe bilangan pecahan yang dapat menampung bilangan antara -3.402823466E+38 sampai dengan 1.175494351E-38, 0, dan 1.175494351E+38 sampai 3.402823466E-38.</p> <p>a. <i>M</i> menyatakan jumlah digit desimal.</p> <p>b. <i>D</i> menyatakan jumlah digit di belakang titik desimal.</p> <p>c. Jika kata UNSIGNED disertakan di belakangnya, nilai negatif tidak diperkenankan.</p> | 4 byte |
| 9. | DOUBLE[(M,D)]<br>atau OUBLE<br>PRECISION[(M,D)] | <p>Tipe bilangan pecahan yang dapat menampung bilangan antara 1.79769348623157E+308 sampai dengan -2.2250738585072014E-308, 0, dan 2.2250738585072014E-308 sampai 1.79769348623157E+308.</p> <p>a. <i>M</i> menyatakan jumlah digit desimal.</p> <p>b. <i>D</i> menyatakan jumlah digit di</p>  | 8 byte |



|     |                                   |   |  |
|-----|-----------------------------------|---|--|
|     |                                   | <p>belakang titik desimal.</p> <p>Jika kata UNSIGNED disertakan di belakangnya, nilai negatif tidak diperkenankan.</p>  |  |
| 10. | FLOAT( $p$ )                      | <p>Tipe bilangan pecahan. Dalam hali ini, <math>p</math> menyatakan jumlah presisi dalam bit.</p> <p>a. Jika <math>p</math> berkisar antara 0 sampai dengan 24, tipe data identik FLOAT tanpa nilai <math>M</math> ataupun <math>D</math>.</p> <p>b. Jika <math>p</math> berkisar antara 25 sampai dengan 53, tipe data identik DOUBLE tanpa nilai <math>M</math> ataupun <math>D</math>.</p>   | <p>4 byte jika <math>0 \leq p \leq 24</math>, 8 byte jika <math>25 \leq p \leq 53</math></p> |
| 11. | DEC( $M$ ) atau DECIMAL( $M, D$ ) | <p>Tipe bilangan dengan sejumlah titik desimal.</p> <p>a. <math>M</math> menyatakan jumlah digit desimal.</p> <p>b. <math>D</math> menyatakan jumlah digit di belakang titik desimal.</p> <p>c. Nilai terbesar untuk <math>M</math> adalah 65.</p> <p>d. Jika <math>D</math> tidak disertakan, dianggap <math>D</math> bernilai 0 (yang berarti tidak ada bagian pecahan).</p> <p>e. Cocok untuk menyimpan data yang memerlukan keakurasian tinggi misalnya uang.</p> |  |

Sumber: Kadir (2008:49-50)



### b. Tipe Tanggal dan Waktu

Tipe ini adalah tipe menyatakan tanggal dan waktu. Berikut adalah tabel tipe tanggal dan waktu:

**Tabel 2.9** Tipe Tanggal dan Waktu

| No. | Tipe Data   | Keterangan  | Kebutuhan Memori |
|-----|-------------|---|------------------|
| 1.  | DATE        | Menyatakan tipe untuk data tanggal. Tanggal yang didukung berkisar antara '1000-01-01' sampai dengan '999-12-31'. Perlu diketahui, notasi tanggal biasa ditulis dengan format tahun-bulan-tanggal.  | 3 byte           |
| 2.  | TIME        | Menyatakan tipe untuk data waktu yang berformat: 'jam:menit:detik'.   | 3 byte           |
| 3.  | DATETIME    | Menyatakan tipe data yang menggabungkan tanggal dan waktu. Format: 'tahun-bulan-tanggal jam:detik:menit.  | 8 byte           |
| 4.  | TIMESTAMP   | Tipe kolom yang nilainya akan diisi secara otomatis dengan tanggal dan jam saat ada operasi INSERT atau UPDATE.   | 4 byte           |
| 5.  | YEAR[(2 4)] | Tipe untuk menyatakan tahun. Bila angka tidak disebutkanm tahun dianggap berformat 4 digit. Angka 2 berarti tahun menggunakan 2 digit.<br>a. Untuk format 4 digit, tahun dapat berkisar 1901 sampai dengan 2155 bisa juga 0000.<br>b. Untuk format 2 digit, tahun dapat | 1 byte           |



|  |  |  |  |
|--|--|--|--|
|  |  | berkisar 70 sampai dengan 69. 70 berarti 1970 dan 69 berarti 2069.<br>c. Nilai untuk YEAR(4) dapat diisi berupa string ataupun bilangan. |  |
|--|--|--|--|

Sumber: Kadir (2008:51)

### c. Tipe String

String berarti deretan karakter. Berikut adalah tabel tipe string:

**Tabel 2.10** Tipe String

| No. | Tipe Data      | Keterangan   | Kebutuhan Memori                                 |
|-----|----------------|--|--|
| 1.  | CHAR( $M$ )    | String dengan panjang tetap yaitu $M$ karakter. Bila string yang dimasukkan memiliki panjang kurang dari $M$ maka sisanya diisi dengan spasi. Nilai terbesar untuk $M$ adalah 255. | $M$ byte, $0 \leq M \leq 255$                    |
| 2.  | CHAR           | Sinonim dengan CHAR(1).  | 1 byte   |
| 3.  | VARCHAR( $M$ ) | String dengan panjang variabel. $M$ menyatakan jumlah maksimum karakter yang bisa disimpan. Nilai terbesar untuk $M$ yaitu 65535.  | $L+1$ byte, $L \leq M$ dan $0 \leq M \leq 65535$ |
| 4.  | BINARY( $M$ )  | Serupa dengan CHAR( $M$ ), tetapi disimpan dalam bentuk biner.   | $M$ byte, $0 \leq M \leq 255$                    |



|     |                       |  |  |
|-----|-----------------------|--|--|
| 5.  | VARBINARY( <i>M</i> ) | Serupa dengan VARCHAR( <i>M</i> ), tetapi disimpan dalam bentuk biner. | <i>L</i> +1 byte, <i>L</i> ≤ <i>M</i> and 0 ≤ <i>M</i> ≤ 255 |
| 6.  | TINYBLOB              | BLOB dengan ukuran paling kecil.                                       | <i>L</i> +1 byte, <i>L</i> < 2 <sup>8</sup>                  |
| 7.  | BLOB[( <i>M</i> )]    | BLOB dengan ukuran lebih besar daripada TINYBLOB.                      | <i>L</i> +2 byte, <i>L</i> < 2 <sup>16</sup>                 |
| 8.  | MEDIUMBLOB            | BLOB berukuran medium.   | <i>L</i> +3 byte, <i>L</i> < 2 <sup>24</sup>                 |
| 9.  | LOB                   | BLOB berukuran paling besar.   | <i>L</i> +4 byte, <i>L</i> < 2 <sup>32</sup>                 |
| 10. | TINYTEXT              | String teks dengan ukuran paling kecil.                                | <i>L</i> +1 byte, <i>L</i> < 2 <sup>8</sup>                  |
| 11. | TEXT[( <i>M</i> )]    | String teks dengan ukuran lebih besar                                  | <i>L</i> +2 byte, <i>L</i> < 2 <sup>16</sup>                 |
| 12. | MEDIUMTEXT            | String teks berurutan medium.  | <i>L</i> +3 byte, <i>L</i> < 2 <sup>24</sup>                 |
| 13. | LONGTEXT              | String teks berurutan paling besar.                                    | <i>L</i> +4 byte, <i>L</i> < 2 <sup>32</sup>                 |



|     |                                 |   |  |
|-----|---------------------------------|---|--|
| 14. | ENUM('nilai1',<br>'nilai2',...) | <p>Tipe kolom yang memungkinkan nilai kolom berupa salah satu string yang disebutkan pada ENUM. Jumlah elemen ENUM dapat mencapai 65535.</p> <p>Nilai untuk kolom dapat juga diisi dengan nomor indeks. Daftar berikutnya menunjukkan nilai dan indeksnya:</p> <ol style="list-style-type: none"> <li>NULL (indeks NULL).</li> <li>' ' (indeks 0).</li> <li>'nilai 1' (indeks 1).</li> <li>'nilai 2' (indeks 2), dan seterusnya.</li> </ol> | 1 atau 2<br><i>byte</i> ,<br>bergantung<br>pada<br>jumlah nilai<br><i>enumerasi</i>        |
| 15. | SET('nilai1',<br>'nilai2',...)  | Merupakan suatu objek string yang bisa berisi nol, satu, atau beberapa nilai. Sebuah kolom bertipe SET maksimum berisi 64 elemen yang berbeda. Nilai antar-elemen ditulis dengan pisah koma.  | 1, 2, 3, 4,<br>atau 8 <i>byte</i> ,<br>bergantung<br>pada<br>jumlah<br>anggota<br>himpunan |

Sumber: Kadir (2008:52-53)