



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Aplikasi

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

Sugiar (2014:83), “Aplikasi adalah program yang dibuat untuk melaksanakan tugas tertentu yang dibutuhkan oleh pengguna komputer (*user*).”

Dari kedua pengertian tersebut, dapat disimpulkan bahwa aplikasi adalah suatu alat terapan yang dibuat untuk melaksanakan tugas tertentu secara khusus dan terpadu yang dibutuhkan oleh pengguna komputer.

2.1.2. Pengertian Pengolahan

Kristanto (2008:8), Pengolahan berasal dari kata olah yang berarti mengerjakan, mengusahakan supaya menjadi barang lain atau menjadi lebih sempurna. Pengolahan berarti proses, cara, perbuatan mengolah.

Jogiyanto (2008:9), Pengolahan (*processing*) adalah proses data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengolahan adalah mengerjakan atau mengusahakan sesuatu agar menghasilkan tindakan yang lain yang akan membuat sejumlah data kembali.

2.1.3. Pengertian Data

Wahyudi (2012:3) menjelaskan, data adalah suatu *file* ataupun *field* yang berupa karakter atau tulisan dan gambar.

Sutabri (2012:1), data merupakan bentuk jamak dari bentuk tunggal datum, data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.



Dari kedua pengertian tersebut, dapat disimpulkan bahwa data adalah suatu file ataupun field yang merupakan bentuk jamak baik berupa karakter, gambar, maupun tulisan yang menggambarkan suatu kejadian.

2.1.4. Pengertian Pengolahan Data

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

Rajab (2009:66), Pengolahan data adalah suatu kegiatan mengubah data awal menjadi data yang memberikan informasi ke tingkat yang lebih tinggi.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengolahan data adalah untuk menggambarkan bentuk data menjadi informasi yang berguna.

2.1.5. Pengertian Informasi

Pratama (2014:393) menjelaskan, informasi didefinisikan sebagai hasil dari pengolahan data memanfaatkan teknologi yang tersedia, untuk menghasilkan pengetahuan serta memberikan arti dan nilai.

Gaol (2010:8) menjelaskan, Informasi adalah data yang sudah di bentuk ke dalam sebuah formulir bentuk yang bermanfaat dan dapat digunakan untuk manusia.

Amsyah (2005:300) menjelaskan, Informasi merupakan data yang sudah diolah ke dalam bentuk tertentu sesuai dengan keperluan pemakaian informasi tersebut.

Dari ketiga pengertian tersebut, dapat disimpulkan bahwa informasi adalah hasil dari pengolahan data yang memanfaatkan teknologi yang tersedia untuk menghasilkan pengetahuan yang bermanfaat sesuai dengan keperluan pemakaian informasi tersebut.



2.1.6. Pengertian Komoditi

Setiawati dan Ristijono (2010:123) menjelaskan, Komoditi adalah setiap jasa dan barang yang dapat memenuhi kebutuhan dan keinginan konsumennya.

Ferlianto dkk (2006:10) menjelaskan, Komoditi adalah barang dagangan atau ekonomis yang disediakan untuk memenuhi permintaan konsumen.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa komoditi adalah setiap jasa dan barang yang disediakan untuk memenuhi kebutuhan, permintaan dan keinginan konsumen.

2.1.7. Pengertian Lapak

Setyaji dan agus (2011:51) menjelaskan, Lapak merupakan sebuah tempat dimana *seller* bisa menginformasikan, menawarkan, memajang produk kita untuk diperjual belikan.

Yuyun A (2010:13) menjelaskan, Lapak pasar merupakan sarana tradisional yang menjadi pengepul produk makanan atau minuman kemudian dijual melalui agen atau penjual keliling yang “kulak” di tempat tersebut.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa lapak adalah tempat menginformasikan, menawarkan, memajang produk untuk diperjual belikan kemudian dijual melalui agen atau penjual keliling di tempat tersebut.

2.1.8. Pengertian Pasar

Fuad (2000:120) menjelaskan, Pasar adalah tempat bertemunya pembeli dan penjual atau, lebih jelasnya, daerah, tempat, wilayah, area yang mengandung kekuatan permintaan dan penawaran yang saling bertemu dan membentuk harga.

Saraswati dan wadaningsih (2008:87) menjelaskan, Pasar adalah sebagai tempat bertemunya pembeli dan penjual untuk melakukan transaksi jual-beli barang atau jasa.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pasar adalah tempat bertemunya pembeli dan penjual dimana adanya permintaan dan penawaran untuk melakukan transaksi jual-beli barang atau jasa.

2.1.9. Pengertian Aplikasi Pengolahan Data Komoditi dan Lapak pada PT Swarnadwipa Selaras Adiguna Palembang

Aplikasi pengolahan data komoditi dan lapak pada PT Swarnadwipa Selaras Adiguna merupakan aplikasi yang berfungsi mengolah data informasi harga sayur mayur dan informasi lapak secara online pada PT Swarnadwipa Selaras Adiguna meliputi data jenis sayur, informasi harga sayur dan informasi pemilik lapak serta *output* berupa laporan dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

2.2. Teori Khusus

2.2.1. Pengertian DFD (*Data Flow Diagram*)

Rosa dan Shalahuddin (2014:69) menjelaskan, *Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. System yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah system.

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

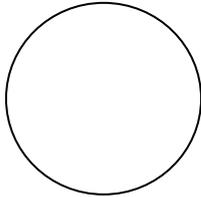
DFD dapat digunakan untuk merepresentasikan sebuah system atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan

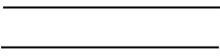
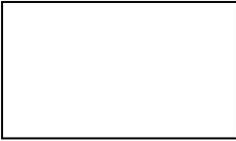
fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan system perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukkan pemrograman terstruktur dan pemrograman berorientasi objek.

Rosa dan Shalahuddin (2014:71) menjelaskan, notasi- notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.1. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

| No | Notasi | Keterangan |
|----|---|---|
| 1. |  | <p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p> |

| | | |
|----|---|---|
| 2. |  | <p><i>File</i> atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM))</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p> |
| 3. |  | <p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p> |
| 4. |  | <p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p> |

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan system yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun system lain. DFD Level 0

digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam system yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah system, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di- *breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah.

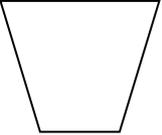
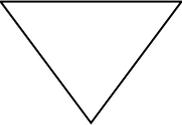
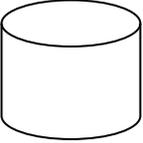
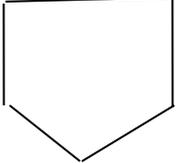
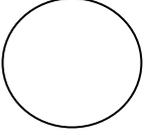
Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

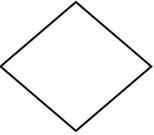
2.2.2. Pengertian *Blockchart*

Kristanto (2008:75) menjelaskan, *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.

Kristanto (2008:75) menjelaskan, simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

Tabel 2.2. Simbol-simbol dalam *Blockchart*

| No | Simbol | Keterangan |
|----|---|--|
| 1. |  | Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan |
| 2. |  | Multi dokumen |
| 3. |  | Proses Manual |
| 4. |  | Proses dilakukan oleh komputer. |
| 5. |  | Menandakan dokumen yang diarsipkan (arsip manual) |
| 6. |  | Data penyimpanan (<i>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. |  | Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama. |

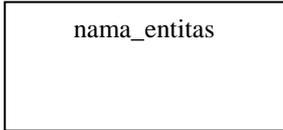
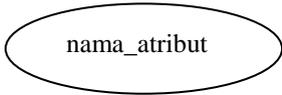
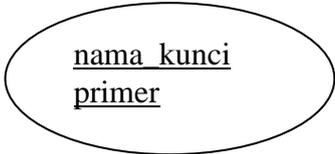
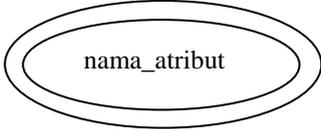
| No | Simbol | Keterangan |
|-----|---|--|
| 10. |  | Terminal yang menandakan awal dan akhir dari suatu aliran. |
| 11. |  | Pengambilan keputusan (<i>Decision</i>). |
| 12. |  | Layar peraga (<i>monitor</i>). |
| 13. |  | Pemasukkan data secara manual. |

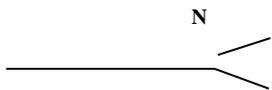
2.2.3. Pengertian ERD (*Entity Relationship Diagram*)

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.

Berikut adalah symbol-simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2.3. Symbol-simbol ERD dengan notasi Chen

| No | Simbol | Deskripsi |
|----|---|--|
| 1. | Entitas / <i>entity</i>  | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| 2. | Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas |
| 3. | Atribut kunci primer  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama) |
| 4. | Atribut multivalai / <i>multivalue</i>  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu |

| No | Simbol | Deskripsi |
|----|--|--|
| 5. | Relasi  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja |
| | Asosiasi / <i>association</i>  | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas |

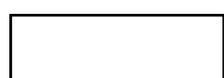
ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

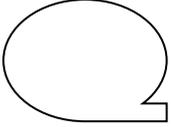
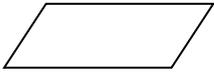
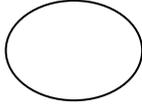
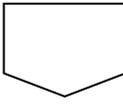
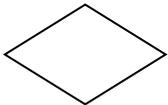
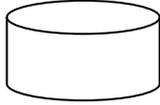
2.2.4. Pengertian *Flowchart*

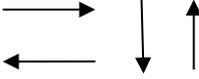
Indrajani (2015:36), “*Flow chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.”

Indrajani (2015:38), “menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:

Tabel 2.4. Simbol-simbol dalam *Flowchart*

| No. | Simbol | Arti |
|-----|---|--|
| 1. |  | Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> . |
| 2. |  | Simbol pemrosesan yang terjadi pada sebuah alur kerja. |

| No. | Simbol | Arti |
|-----|---|---|
| 3. |  | Simbol yang menyatakan bagian dari program (sub program). |
| 4. |  | Simbol masukan atau keluaran dari atau ke sebuah pita magnetic. |
| 5. |  | Simbol Input/Output yang mendefinisikan masukan dan keluaran proses. |
| 6. |  | Simbol konektor untuk menyambung proses pada lembar kerja yang sama. |
| 7. |  | Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda. |
| 8. |  | Simbol masukan atau keluaran dari atau ke sebuah dokumen. |
| 9. |  | Simbol untuk memutuskan proses lanjutan dari kondisi tertentu. |
| 10. |  | Simbol database atau basis data. |
| 11. |  | Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll. |

| | | |
|-----|---|---|
| 12. |  | Simbol yang mendefinisikan proses yang dilakukan secara manual. |
| 13. |  | Simbol untuk menghubungkan antar proses atau antar simbol |

2.2.5. Pengertian Kamus Data

Rosa dan Shalahuddin (2014:73) menjelaskan, Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada system perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) 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:

- a. Nama – nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki beberapa symbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.5. Simbol-simbol dalam Kamus Data

| Simbol | Keterangan |
|------------------|---------------------------------|
| = | disusun atau terdiri dari |
| + | Dan |
| [] | baik ...atau... |
| { } ⁿ | n kali diulang/ bernilai banyak |
| () | data opsional |
| *...* | batas komentar |



Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

2.3. Teori Program

2.3.1. Sekilas Tentang Basis Data (*Database*)

Teguh (2013:1) menjelaskan, secara sederhana database atau basis data dapat diartikan sebagai suatu pengorganisasian data dengan bantuan computer yang memungkinkan data dapat diakses dengan mudah dan cepat. Jadi database adalah sebuah tempat dimana semua data berkumpul saling berinteraksi satu dengan yang lainnya.”

Rosa dan Shalahuddin (2014:43) menjelaskan, sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Whitehorn dan Marklyn (2013:13) menjelaskan, terdapat empat komponen database yang paling penting adalah :

- a. Table: struktur dasar tempat data disimpan di dalam database. Bayangkanlah table sebagai container dimana data berada dan ketiga komponen lainnya sebagai peranti-peranti yang memanipulasi data yang ada dalam table tersebut.
- b. Form: peranti yang memungkinkan Anda untuk melihat dan mengedit data dalam table.
- c. Query: pertanyaan yang dapat Anda lontarkan tentang data dalam suatu table. Jika Anda ingin mencari semua karyawan Anda yang lahir setelah tahun, katakanlah, 1970, Anda biasanya menggunakan query. Query sering digunakan di dalam database karena table biasanya berisi data dalam jumlah

yang sangat banyak padahal yang seringkali ingin kita proses, atau kita lihat, hanyalah sebagian saja dari data tersebut.

- d. Report: digunakan untuk menghasilkan output cetak dari tabel.

2.3.2. Sekilas Tentang *HTML*

Pratama (2014:218) menjelaskan, *HTML* atau *Hyper Text Markup Language* merupakan salah satu bahasa pemrograman yang digunakan di dalam pembuatan aplikasi dan layanan berbasis web.

Betha dan Husni (2014:9) menjelaskan, dokumen *HTML* adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen *HTML* merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet.

Betha dan Husni (2014:9) menjelaskan, tag-tag dasar *HTML* sebagai berikut:

NN: Netscape, IE: Internet Explorer, W3: Web Standard (W3C)

Tabel 2.6. Tag – tag dasar *HTML*

| Tag Awal | NN | IE | W3 | Kegunaan |
|----------|-----|-----|-----|-------------------------------------|
| <html> | 3.0 | 3.0 | 3.2 | Definisi sebuah dokumen <i>HTML</i> |
| <body> | 3.0 | 3.0 | 3.2 | Definisi body dokumen |
| <h1><h6> | 3.0 | 3.0 | 3.2 | Definisi heading 1 s.d. 6 |
| <p> | 3.0 | 3.0 | 3.2 | Definisi sebuah paragraph |
| | 3.0 | 3.0 | 3.2 | Sisipkan line break |
| <hr> | 3.0 | 3.0 | 3.2 | Definisi sebuah horizontal rule |
| <!.> | 3.0 | 3.0 | 3.2 | Definisi komentar dalam source code |

2.3.3. Pengertian *MySQL*

Kadir (2013:15) menjelaskan, *MySQL* adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan

penyimpanan dan pengaksesan data. Dengan menggunakan MySQL, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.

Pratama (2010:10), MySQL adalah suatu sistem manajemen basis data relasional (*RDBMS-Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah Oracle, Sybase. Bahasa yang digunakan oleh MySQL tentu saja adalah SQL-standar bahasa basis data relasional di seluruh dunia saat ini.

Wahana Komputer (2010:2), MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multiuser. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*.

MySQL memiliki beberapa kelebihan dan keuntungan disbanding database lain, antara lain:

1. Banyak ahli berpendapat bahwa MySQL merupakan server tercepat.
2. MySQL merupakan sistem manajemen database yang open source (kode sumbernya terbuka), yaitu software ini bersifat free atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
3. MySQL mempunyai performa yang tinggi tapi simpel.
4. Database MySQL mengerti bahasa SQL (Structured Query Language).
5. MySQL dapat diakses melalui protocol ODBC (Open Database Connectivity) buatan Microsoft. Ini menyebabkan MySQL dapat diakses oleh banyak software.
6. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses database.
7. Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
8. MySQL merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran Gigabyte.

9. MySQL dapat berjalan di berbagai operating system seperti Linux, Windows, Solaris, dan lain-lain.

2.3.4. Sekilas tentang PHP

2.3.4.1. Pengertian PHP (*Hypertext Preprocessor*)

Nugroho (2013:153) menjelaskan, PHP kepanjangan dari *Hypertext Preprocessor* itu bahasa pemrograman berbasis web. PHP itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web).

Kadir (2013:120) menjelaskan, PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi web. Ditinjau dari pemrosesannya, PHP tergolong berbasis *server side*. Artinya, pemrosesan dilakukan di *server*.

Sidik (2014:4) menjelaskan, PHP merupakan secara umum dikenal sebagai bahasa pemrograman script script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

2.3.4.2. Tipe Data PHP

Sidik (2014:106) menjelaskan tipe data *PHP* yaitu :

1. Tipe Data Integer

Tipe Data Integer adalah tipe data numerik yang digunakan untuk menyatakan bilangan bulat. Tipe data integer dapat dituliskan dengan notasi desimal, oktal atau hexadesimal.

2. Tipe Data *Floating Point*

Tipe Data *Floating Point* adalah tipe data yang digunakan untuk menyatakan data numerik berupa bilangan pecahan.

3. Tipe Data *String*

Tipe Data *String* adalah tipe data yang berjenis *string* atau teks. Teks atau tulisan adalah semua data yang dinyatakan dengan huruf dan tanda baca,

termasuk didalamnya ada data berupa angka. Data angka yang ada didalam suatu teks atau tulisan akan dianggap karakter, karena data tersebut tidak dapat dioperasikan dalam operasi matematika

4. Tipe Data *Boolean*

Tipe Data *Boolean* adalah tipe data yang digunakan untuk menyatakan nilai benar atau salah, atau nilai *Boolean*.

5. Tipe Data *Array*

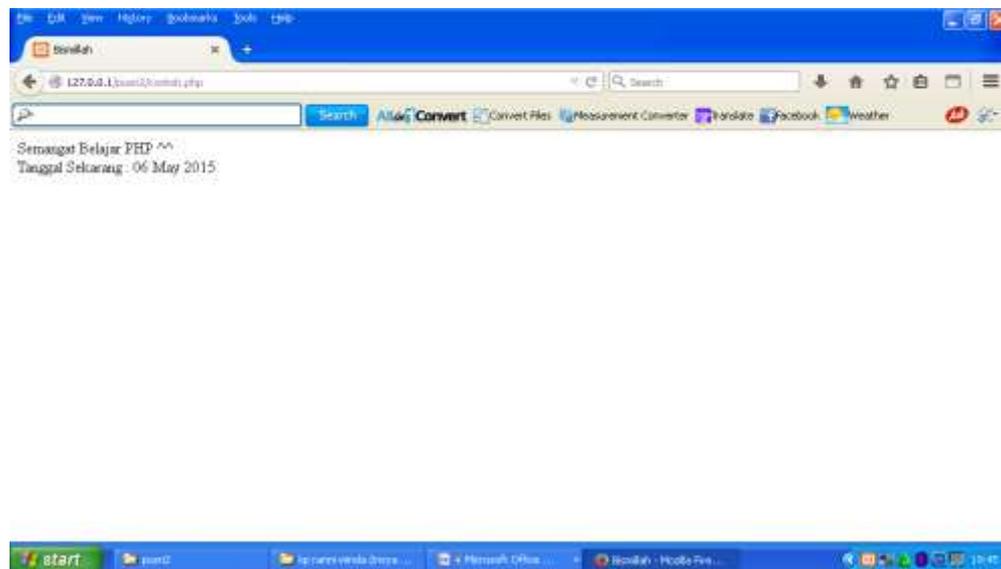
Array dalam *PHP* juga merupakan tipe data, bukan sekedar variabel jamak, variabel yang mempunyai banyak elemen yang diacu dengan satu nama yang sama. Setiap elemen variabel diacu dengan menggunakan nomor variabel.

2.3.4.3. Skrip *PHP*

Badiyanto (2013:32), “skrip *HTML (Hyper Text Markup Language)* adalah bahasa standar untuk membuat halaman *web*. Berikut contoh kode *PHP* yang berada di dalam kode *HTML* (disimpan dengan ekstensi *.php*).

```
<HTML>
<HEAD>
  <TITLE>Bismillah</TITLE>
</HEAD>
<BODY>
  Semangat Belajar PHP ^^<BR>
  <?php
  printf("Tanggal Sekarang : %s", Date("d F Y"));
  ?>
</BODY>
</HTML>
```

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.2 Tampilan *PHP* Sederhana

2.3.5. Pengertian *Adobe Dreamweaver*

Madcoms (2012:2), “Dreamweaver adalah sebuah HTML editor professional untuk mendesain web secara visual dan mengelola situs atau halaman web. Dreamweaver merupakan software utama yang digunakan oleh web designer maupun web programmer dalam mengembangkan suatu situs web.

Puspitosari (2010:9) Dreamweaver adalah salah satu HTML Editor professional yang berfungsi untuk mendesain web secara visual dan mengelola situs atau halaman web.

2.3.5.1. Ruang Kerja *Adobe Dreamweaver CS6*

Madcoms (2012:3) menjelaskan tampilan ruang kerja *Adobe Dreamweaver CS6* adalah sebagai berikut:

a. *Application Bar*

Berada di bagian paling atas jendela aplikasi Dreamweaver CS5. Baris ini berisi tombol *Workspce (Workspace Swithcer)*, *Menu* dan aplikasi-aplikasi dari Dreamweaver.

b. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan jendela *design* atau tampilan jendela *code*.

c. Jendela Dokumen

Adalah lembar kerja tempat anda membuat dan mengedit desain halaman web.

d. *Workspace Swithcer*

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

e. *Panel Groups*

Adalah kumpulan panel yang saling berkaitan, Panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara default, *panel group* berisi *panel insert, CSS styles, Asset, AP Element, dan Files*.

f. *Tag Selector*

Diletakkan di bagian 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.

g. *Panel Property*

Berfungsi untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.

h. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi *code-code* standar.

i. *Panel Insert*

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

j. *Panel group*

Kumpulan *panel* yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul, seperti: *insert, files, css styles*, dan sebagainya.

k. *Panel Files*

digunakan untuk mengatur file-file dan folder-folder yang membentuk

2.3.6 Pengenalan XAMPP

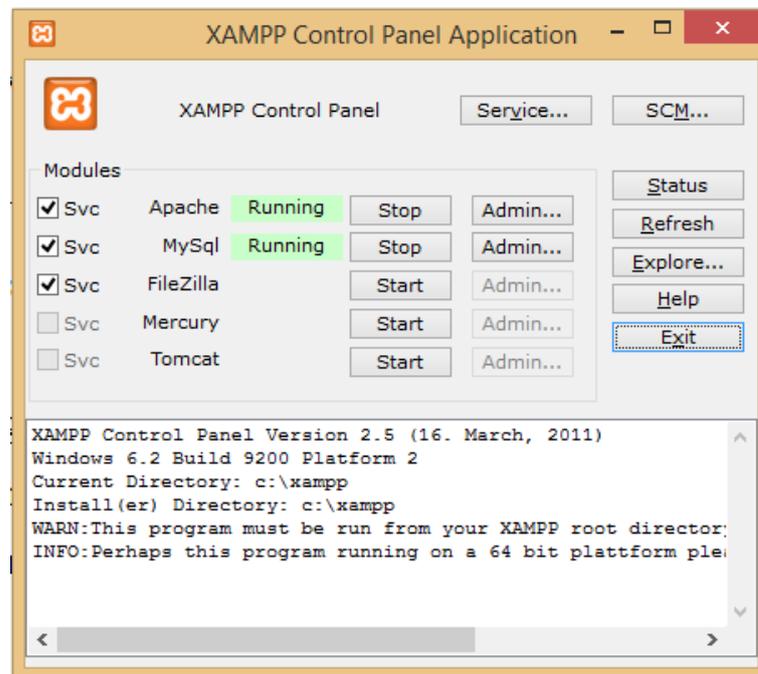
Sidik (2014:72) XAMPP (X/(Windows/Linux) Apache MySQL PHP dan Perl) adalah paket *server web* PHP dan *database* MySQL yang paling populer dikalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai databasenya.

Nugroho (2013:1) menjelaskan, XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legas.”

Dibawah folder utama xampp, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut:

Tabel 2.7 Folder-folder penting dalam Xampp

| No | Nama Folder | Fungsi |
|----|---------------|---|
| 1. | <i>Apache</i> | Folder utama dari <i>Apache Webserver</i> . |
| 2. | <i>Htdocs</i> | Folder utama untuk menyimpan data-data latihan <i>web</i> , baik PHP maupun HTML biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di <i>htdocs</i> bisa diakses dengan mengetikkan alamat http://localhost/ di browser. |
| 3. | Manual | Berisi subfolder yang di dalam terdapat manual program dan <i>database</i> , termasuk manual PHP dan MYSQL. |
| 4. | Mysql | Folder utama untuk <i>database</i> MYSQL server. di dalamnya terdapat subfolder data (lengkapnya: C:\xampp\mysql\data) untuk merekam semua nama <i>database</i> , serta subfolder bin yang berisi <i>tools</i> klien dan server MYSQL. |
| 5. | Php | Folder utama untuk program PHP. |



Gambar 2.3 Tampilan XAMPP

2.3.7 Pengertian phpMyAdmin

Nugroho (2013:71) menjelaskan, “phpMyAdmin adalah tools *MySQL Client* berlisensi *Freeware*, Anda dapat mengunduhnya secara gratis pada situs resminya di www.phpmyadmin.net.