



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 definisi diatas, dapat disimpulkan bahwa aplikasi adalah alat terapan yang melaksanakan tugas tertetu yang dibutuhkan oleh *user* secara khusus dan terpadu yang sesuai dengan kemampuannya.

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.”

2.1.3. Pengertian Data

Ladjamuddin (2013:8), “Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata.”

Sutabri (2012:2), “Data adalah bahan mentah untuk diolah yang hasilnya kemudian menjadi informasi.”

Dari kedua definisi diatas, dapat disimpulkan bahwa data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan.

Whitehorn dan Marklyn (2003:20) menjelaskan, tipe data yang paling lazim digunakan ialah:

Table 2.1. Tipe-tipe data

No	Tipe	Keterangan
1.	Text	sebagian besar karakter yang ada pada keyboard, termasuk angka; biasanya dibatasi hingga 255 karakter.
2.	Memo	blok-blok besar berisi teks
3.	Number/ numeric	hanya bilangan, tanpa karakter teks
4.	Currency	pada dasarnya bilangan numeric dengan empat tempat decimal dan sebuah lambang mata uang.
5.	AutoNumber	nomor yang secara otomatis ditambahkan untuk setiap record baru.
6.	Yes/No	untuk informasi terpisah yang bisa dibagi ke dalam dua kategori; seperti True/False, Yes/No, Up/Down.
7.	OLE Object	suatu objek seperti lembar-kerja Microsoft Excel, dokumen Microsoft Word, atau informasi grafis
8.	Hyperlink	untuk menyimpan URL (Uniform Resource Locator)

2.1.4. Pengertian Pengolahan Data

Ladjamuddin (2013:9), “Pengolahan data adalah massa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

Ashkaf (2014), “Pengolahan data adalah pengubahan atau transformasi simbol-simbol seperti nomor dan huruf untuk tujuan peningkatan kegunaannya.”

Dari kedua definisi diatas, dapat disimpulkan bahwa pengolahan data adalah manipulasi data kedalam bentuk yang lebih berarti berupa informasi.

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.”

Pratama (2014:393) menjelaskan, “Informasi didefinisikan sebagai hasil dari pengolahan, manipulasi, dan pengorganisasian data, sehingga menambah nilai dan pengetahuan di dalamnya.”

Dari kedua definisi diatas, informasi merupakan hasil dari pengolahan data yang memanfaatkan teknologi yang tersedia sehingga menambah nilai dan pengetahuan .

2.1.6. Kinerja

Jelantik (2015:37) menjelaskan, “Kinerja atau *performance* adalah usaha yang dilakukan dari hasil kerja yang dapat dicapai seseorang atau sekelompok orang dalam suatu organisasi sesuai dengan tugas dan tanggung jawab.”

Nurlaila (2010:71) menjelaskan, “*Perfomance* atau kinerja merupakan hasil atau keluaran dari suatu proses.”

Dari kedua definisi diatas, kinerja merupakan hasil usaha yang dilakukan dari hasil kerja yang merupakan hasil atau keluaran dari suatu proses.

2.1.7. Pengertian Pegawai

Prastowo dkk (2011:132) menjelaskan, “Pegawai adalah pribadi yang bekerja pada pemberi kerja baik sebagai pegawai tetap atau pegawai tidak tetap atau tenaga kerja lepas berdasarkan perjanjian atau kesepakatan baik secara tertulis maupun tidak tertulis.”

Muljono (2010:113) menjelaskan, “Pegawai adalah setiap orang pribadi yang melakukan pekerjaan berdasarkan perjanjian atau kesepakatan kerja, baik tertulis maupun tidak tertulis, termasuk yang melakukan pekerjaan dalam jabatan negeri atau badan usaha milik negara atau badan usaha milik daerah.”

Dari kedua definisi diatas, pegawai adalah setiap orang pribadi yang melakukan pekerjaan berdasarkan perjanjian di dalam jabatan negeri atau badan usaha milik negara atau badan usaha milik negara.

2.1.8. Pengertian Badan Pusat Statistik

Badan Pusat Statistik adalah Lembaga Pemerintah Non Kementerian yang bertanggung jawab langsung kepada Presiden. Sebelumnya, BPS merupakan Biro Pusat Statistik, yang dibentuk berdasarkan UU Nomor 6 Tahun 1960. Sebagai pengganti kedua UU ditetapkan UU Nomor 16 Tahun 1997 tentang Statistik. Berdasarkan UU ini yang ditindaklanjuti dengan peraturan perundangan dibawahnya, secara formal nama Biro Pusat Statistik diganti menjadi Badan Pusat Statistik.

2.1.9. Metode Pengembangan Sistem

Sukanto dan Shalahudin (2014:28), menjelaskan tentang metode pengembangan sistem model *waterfall*, sebagai berikut:

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 dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentransaksi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi dan fungsional dan memastikan bahwa bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.10. Metode Pengujian Sistem

2.1.10.1. Pengertian Pengujian Sistem

Al Fatta (2007:169), Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian bug, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.

2.1.10.2. Metode Pengujian

Shihab (2011), “Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menentukan kesalahan.”

Al Fatta (2007:170), Dalam melakukan pengujian, beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, transaksi, query atau jalur navigasi yang mewakili penggunaan sistem yang tipikal, kritis atau abnormal. Pengujian harus mencakup unit testing, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul, dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

2.1.10.3. Metode *Black Box Testing*

Al Fatta (2007:172), Pengujian *blackbox* adalah metode pengujian yang berfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dengan spesifikasi. Pada *blackbox testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit

atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.

Iskandaria (2012), Pengujian *blackbox* (*blackbox testing*) adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada *input* dan *output* aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak.

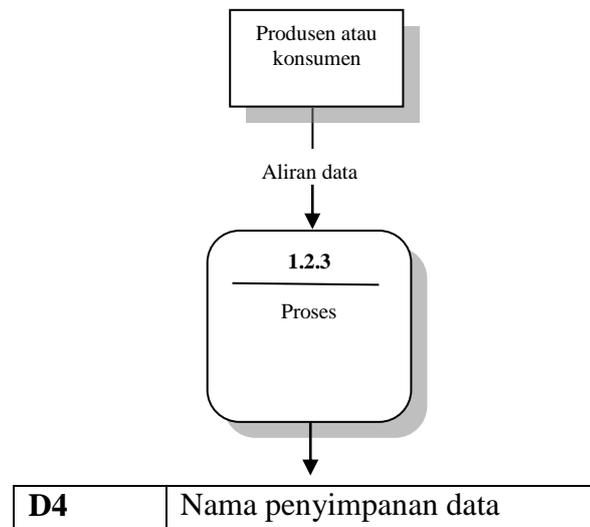
Shihab (2011), mengemukakan cirri-ciri *black box testing*, diantaranya sebagai berikut:

1. *Black box testing* berfokus pada kebutuhan fungsionalitas pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif dari pada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

2.2. Teori Khusus

2.2.1. Pengertian DFD (*Data Flow Diagram*)

Sukamto 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.” Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson:



Gambar 2.1. Contoh DFD yang dikembangkan Chris Gane dan Trish Sarson

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.

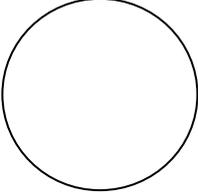
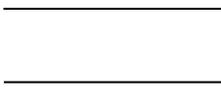
Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai 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 mencampurkan pemrograman terstruktur dan pemrograman berorientasi objek.

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

Tabel 2.2. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<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>
	<p>File atau basisdata atau penyimpanan (storage); 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 (Entity Relationship Diagram (ERD), Conceptual Data Model (CDM), Physical Data Model (PDM))</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
	<p>Entitas luar (external entity) atau masukan (input) atau keluaran (output) 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 (input) atau keluaran (output) biasanya berupa kata benda</p>

Lanjutan Tabel 2.2. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (input) atau keluaran (output) 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”

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.

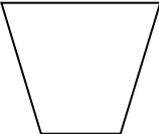
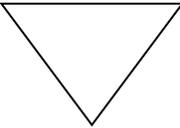
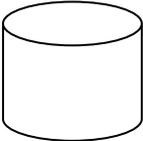
Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

2.2.2. Pengertian *Blockchart*

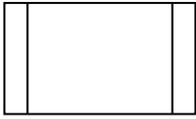
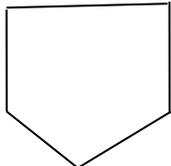
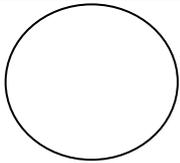
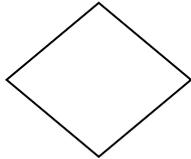
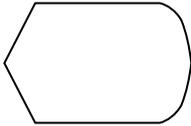
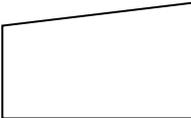
Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

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

Tabel 2.3. 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 (Storage)

Lanjutan Tabel 2.3. Simbol-simbol dalam *Blockchart*

No	Simbol	Keterangan
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.
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.

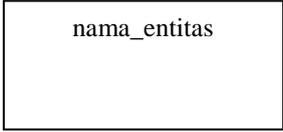
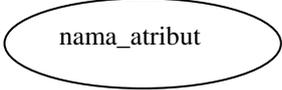
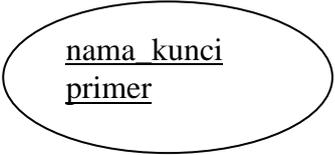
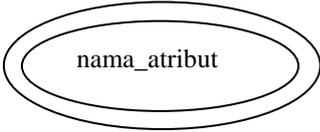
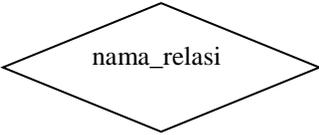
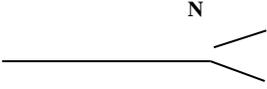
2.2.3. Pengertian ERD (*Entity Relationship Diagram*)

Sukamto dan Shalahuddin (2014 : 50) menjelaskan, “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 DBMS 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 2.4. Symbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
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.
Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record 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).
Atribut multivalai / multivalue 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
Asosiasi / association 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity 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*.

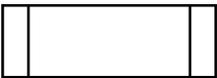
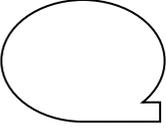
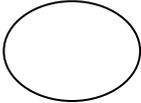
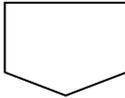
2.2.4. Pengertian *Flowchart*

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

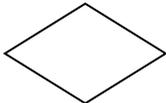
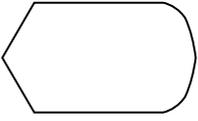
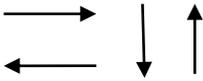
eWolf Community (2011:16), “*flowchart* adalah simbl-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung.”

eWolf Community (2011:17), menjelaskan simbol-simbol dalam *flowchart* adalah sebagai berikut:

Tabel 2.5. 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.
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.

Lanjutan Tabel 2.5. Simbol-simbol dalam *Flowchart*

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, printer, 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

Sukanto 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 2.6. 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. Pengertian Basis Data (*Database*)

Indrajani (2015:70), “Basis data adalah sekumpulan elemen data terintegrasi yang secara logika saling berhubungan.”

Sukanto dan Shalahudin (2014:43), “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.”

Sutanta (2011:35), “Basis data adalah sekumpulan dari berbagai macam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.”

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa basis data adalah kumpulan informasi yang disimpan didalam computer secara sistematis untuk memperoleh informasi dari basis data tersebut.

2.3.2. HTML (*Hyper Text Markup Language*)

Winarno (2014:1), “*Hypertext Markup Language* (HTML) adalah sebuah bahasa untuk menampilkan konten di web.”

Enterprise (2015:135), “*Hypertext Markup Language* (HTML) adalah *script* pemrograman yang mengatur bagaimana menyajikan informasi di dunia internet dan bagaimana informasi itu membawa kita dari satu tempat ke tempat lainnya.”

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa *Hypertext Markup Language* (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web.

2.3.3. Pengertian *MySQL*

Winarno (2014:101), “*MySQL* adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.”

Kadir (2013:15), “*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.”

Saputra dkk (2012:7), “*MySQL* merupakan *database* yang bekerja menggunakan *SQL Language* (*Structure Query Language*).”

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa *MySQL* adalah sistem manajemen database SQL yang bersifat *open source* dan paling populer saat ini.

2.3.4. Tentang *PHP*

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.”

2.3.5. 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.”

Sadeli (2014:12), “*Dreamweaver* adalah suatu perangkat lunak web editor keluaran adobe sistem yang digunakan untuk mengembangkan dan mendesain suatu *website* dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya.”

Bekti (2015:2), “*Adobe Dreamweaver* adalah aplikasi design dan pengembangan web yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai *Design view*) dan kode editor dengan fitur standar seperti *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur lebih canggih seperti *real-time syntax checking* dan *code introspection*.”

2.3.6. Pengenalan XAMPP

Sidik (2014:72) menjelaskan, “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.”

2.3.7. Pengertian phpMyAdmin

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

2.3.8. Tentang CSS

Saleh dkk (2007:53) menjelaskan, “CSS (*Cassading Style Sheet*) adalah template yang mengontrol pemformatan tag HTML pada halaman web anda. Bila anda telah terbiasa menggunakan Microsoft Word, anda dapat melihat bahwa konsep Style Sheet mirip dengan template pada Microsoft Word: anda dapat mengubah penampilan pada dokumen Word dengan mengubah format pada style dokumen. Begitu pula anda dapat mengubah penampilan halaman web dengan mengubah format pada tag HTML tertentu melalui Style Sheet, untuk selanjutnya menggantikan spesifikasi default dari browser untuk tag-tag tersebut.”

2.3.9. JavaScript

Betha dan Husni (2014:267) menjelaskan, “JavaScript merupakan modifikasi dari bahasa c++ dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan asp ataupun internet explorer.” Secara khusus beberapa hal yang penting dalam JavaScript adalah:

- a. Menggunakan blok awal “{“ dan blok akhir “}”
- b. Automatic conversion dalam pengoperasian tipe data yang berbeda
- c. Sensitive case, sehingga programmer java harus ekstra hati-hati dalam menggunakan nama variable, fungsi dan lain-lain
- d. Extension umumnya menggunakan “*.js”
- e. Setiap statement dapat diakhiri dengan “;” sebagaimana halnya c++ tetapi dapat juga tidak
- f. Jika tidak didukung oleh browser versi lama, scriptnya dapat disembunyikan di antara tag “<!--” dan “-->”
- g. Jika program dalam satu baris terlalu panjang dapat disambung ke baris

berikut dengan karakter

Betha dan Husni (2014 : 272) menjelaskan, “Tipe data pada JavaScript umumnya sama pada setiap mesin, hal ini diakibatkan desain awal java memang dikhususkan untuk bisa berjalan pada semua mesin.”

Tabel 2.7. Tipe data JavaScript

Primitive Type	Size	Minimum	Maximum	Version Type
Boolean	1 bit	-	-	
Char	16 bit	Unicode 0	Unicode 2E16-1	
Byte	8 bit	-128	+127	Versi 1.1 keatas
Short	16 bit	-2E-15	+2E15-1	Versi 1.1 keatas
Int	32 bit	-2E-31	+2E31-1	
Long	64 bit	-2E-63	+2E63-1	
Float	32 bit	IEEE754	IEEE754	
Double	64 bit	IEEE754	IEEE754	
Void	-	-	-	Versi 1.1 keatas