

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Aplikasi

Sugiari (2014:83), aplikasi adalah program yang dibuat untuk melaksanakan tugas tertentu yang dibutuhkan oleh pengguna komputer (*user*).

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

Dari kedua definisi diatas, dapat disimpulkan bahwa aplikasi merupakan program yang dibuat sebagai alat terapan untuk melaksanakan tugas tertentu yang dibutuhkan oleh pengguna komputer (*user*).

2.1.2. Pengertian Pendataan

Pendataan merupakan proses, cara, perbuatan mendata atau pengumpulan data. (<http://kbbi.web.id/data>)

Jadi, pendataan adalah suatu cara, perbuatan mendata atau mengumpulkan data melalui suatu poses atau cara tertentu.

2.1.3. Pengertian Pengiriman

Pengiriman merupakan proses, cara, perbuatan mengirimkan suatu barang. (<http://kbbi.web.id/data>)

Jadi, pengiriman adalah suatu cara, perbuatan mengirimkan suatu barang melalui suatu proses atau cara tertentu.

2.1.4. Pengertian Alat

Sugiarto dan Ismawati (2008:175), alat suatu benda yang digunakan dalam melakukan kegiatan praktikum, eksperimen, dan penelitian.

Alat adalah benda yang di gunakan untuk mengerjakan sesuatu yang fungsinya adalah untuk mempermudah pekerjaan. Alat disebut juga sebagai perkakas atau perabotan.



(<http://www.kamusq.com/2013/12/alat-adalah-pengertian-dan-definisi.html>)

Jadi, alat merupakan suatu benda disebut juga sebagai perkakas atau perabotan yang digunakan untuk membantu mempermudah suatu pekerjaan.

2.1.5. Pengertian Material

Anditya (2007:203), material adalah suatu bahan yang diletakan pada objek/elemen konstruksi sehingga kita mengenalnya sebagai suatu benda yang memiliki citra dan kesan karakter tertentu.

Material adalah segala sesuatu yang mempunyai massa dan menempati ruang.

(<https://tajilapak.wordpress.com/2012/12/14/material-teknik/>)

Dari kedua definisi diatas, dapat disimpulkan bahwa material adalah bahan yang digunakan untuk menyusun sebuah benda dan digunakan untuk perancangan dan perancangan.

2.1.6. Pengertian Pengeboran

Pengeboran adalah suatu aktivitas vital baik dalam pengambilan sample maupun pemboran produksi. (<http://rachmatrisejet.blogspot.com/2013/08/drilling-pemboran.html>)

Jadi, pengeboran Minyak adalah suatu proses pengerjaan pemotongan menggunakan mata bor (*twist drill*) untuk menghasilkan lubang yang bulat pada material logam maupun non logam yang masih pejal atau material yang sudah berlubang.

2.1.7. Pengertian Minyak Bumi

Untoro (2010:204), minyak bumi merupakan campuran kompleks yang sebagian besar terdiri atas hidrokarbon.

Yudhistira (2006:66), minyak bumi adalah tersusun dari bermacam-macam hidrokarbon yaitu alkana, sikloalkana, dan aromatik.

Jadi, Minyak bumi merupakan campuran kompleks yang terdiri dari senyawa hidrokarbon yaitu alkana, sikloalkana dan aromatik.

2.1.8. Pengertian PT

Sempal (2013:12) menjelaskan, Perseroan terbatas adalah organisasi bisnis yang memiliki badan hukum resmi yang dimiliki oleh minimal dua orang dengan tanggung jawab yang hanya berlaku pada perusahaan tanpa melibatkan harta pribadi atau perseorangan yang ada di dalamnya. Tiap-tiap persero memiliki satu sero atau lebih yang mempunyai tanggung jawab terbatas hanya pada modal yang diikutsertakan dalam perusahaan. Sehubungan dengan itu, orang yang mempunyai tagihan terhadap PT tidak dapat langsung menagih kepada para pemegang saham, melainkan kepada PT, karena PT merupakan badan hukum.

Winarti dan Syahrizal (2012:1) menjelaskan, PT adalah singkatan dari Perseroan Terbatas. Perseroan Terbatas adalah badan hukum yang merupakan persekutuan modal yang dilakukan oleh minimal dua orang dengan tanggung jawab yang hanya berlaku pada perusahaan saja, tanpa melibatkan harta pribadi atau perseorangan yang ada di dalamnya (para pemegang saham), didirikan berdasarkan perjanjian, melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham dan memenuhi persyaratan yang ditetapkan dalam Undang-Undang Republik Indonesia Nomor 40 Tahun 2007 Tentang Perseroan Terbatas serta peraturan pelaksanaannya.

2.1.9. Pengertian PT. Ginting Jaya Energi

PT. Ginting Jaya Energi merupakan perusahaan yang bergerak di bidang *drilling kontraktor* (pengeboran minyak) dan berfungsi sebagai tempat penyedia alat, material dan mobil unit pengeboran minyak.

2.1.10. Pengertian Aplikasi Pendataan Pengiriman Alat dan Material Pengeboran Minyak pada PT Ginting Jaya Energi

Aplikasi pendataan pengiriman alat dan material pengeboran minyak adalah aplikasi yang dapat membantu Bagian Operasional dalam pendataan pengiriman alat dan material pengeboran minyak.

2.1.11. Metode Pengembangan Sistem

Sukanto dan Shalahudin (2014:28), 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.

2.1.12. Metode Pengujian Sitem

2.1.12.1. 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.

2.1.12.2. 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.

2.1.12.3. 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), *Black Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsionalitas program.

Ciri-ciri *black box testing*, diantaranya sebagai berikut:

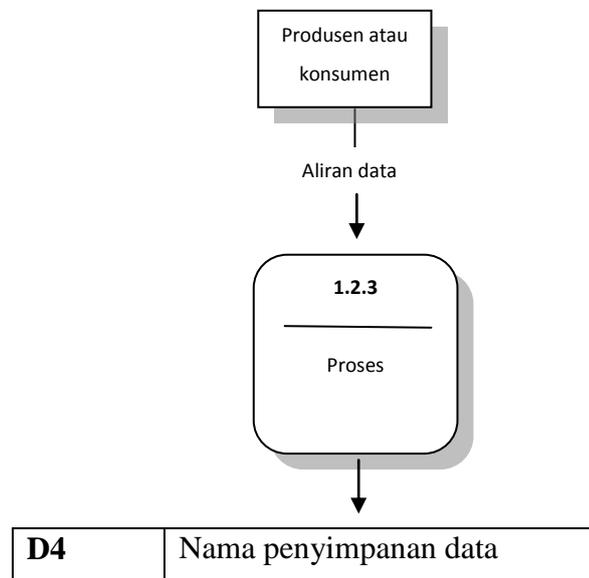
1. *Black box testing* berfokus pada kebutuhan fungsionalitas pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif dari pada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas 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 Shalahudin (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. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem.

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



(*Sumber: Sukamto dan Shalahuddin, 2014:69*)

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

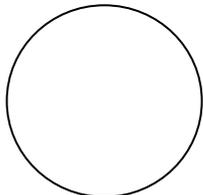
Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk merepresentasikan sebuah sistem 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.

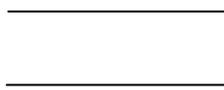
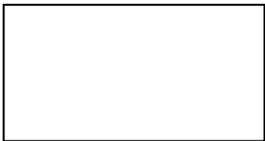
DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampurkan pemrograman terstruktur dan pemrograman berorientasi objek.

Notasi- notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.1. 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. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>

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

Notasi	Keterangan
	<p><i>File</i> atau basis data 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 (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>). Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
	<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. Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>
	<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>). Nama</p>

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

Notasi	Keterangan
	yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”

(*Sumber: Sukanto dan Shalahuddin, 2014:71*)

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.

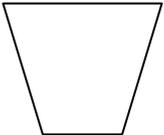
Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

2.2.2. Pengertian *Blockchart*

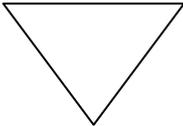
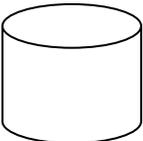
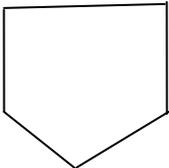
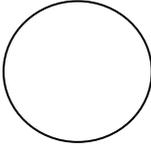
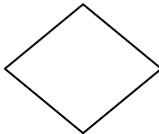
Kristanto (2008:75), *Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

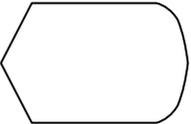
Tabel 2.2. Simbol-simbol dalam *Block Chart*

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.

Lanjutan Tabel 2.2. Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
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.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).

Lanjutan **Tabel 2.2.** Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
12.		Layar peraga (monitor).
13.		Pemasukkan data secara manual.

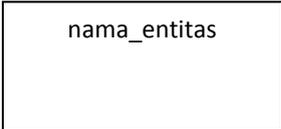
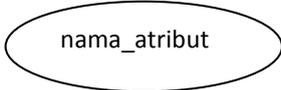
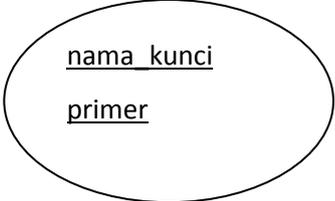
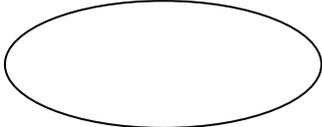
(Sumber: Kristanto, 2008:75)

2.2.3. Pengertian ERD (*Entity Relationship Diagram*)

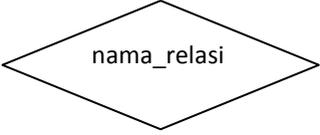
Sukanto dan Shalahudin (2014 : 50), pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis 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. Simbol-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 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> 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 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu

Lanjutan **Tabel 2.3.** Simbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
Relasi 	<i>Relasi</i> 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

(*Sumber: Sukamto dan Shalahuddin, 2014:50*)

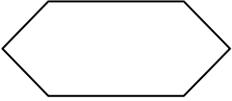
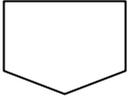
ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak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

2.2.4. Pengertian *Flowchart*

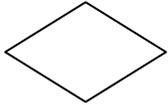
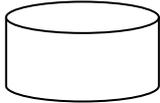
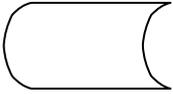
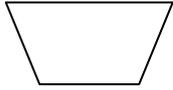
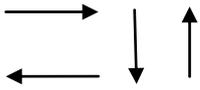
Indrajani (2015:36), *Flow chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.

Simbol-simbol dalam *Flowchart* adalah sebagai berikut:

Tabel 2.4. Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
3.		Proses yang terdefinisi atau sub program.
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Menyatakan masukan dan keluaran (input/output).
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.

Lanjutan **Tabel 2.4.** Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
9.		Menyatakan desicion (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan input/output menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

(Sumber: Indrajani 2015:36)

2.2.5. Pengertian Kamus Data

Sukanto dan Shalahudin (2014:73), kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem 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 simbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.5. Simbol-simbol untuk menjelaskan informasi tambahan dalam Kamus Data

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

(*Sumber: Sukanto dan Shalahuddin, 2013:73*)

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. Tentang 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.

Dari kedua definisi 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. Tentang *HTML*

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

Jubile 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.

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

Saputra et.al (2012:7), MySQL merupakan database yang bekerja menggunakan SQL Language (Structure Query Language).

Dari kedua definisi diatas, dapat disimpulkan bahwa bahwa MySQL adalah sistem manajemen database SQL yang bersifat open source dan paling populer saat ini.

2.3.4. Tentang PHP

Nugroho (2013:153), 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), 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), 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.

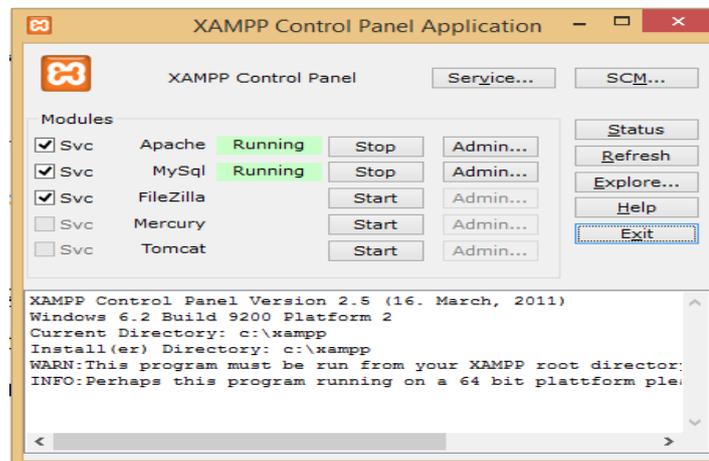
Dreamweaver merupakan software utama yang digunakan oleh web designer maupun web programmer dalam mengembangkan suatu situs web.

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.

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), 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.

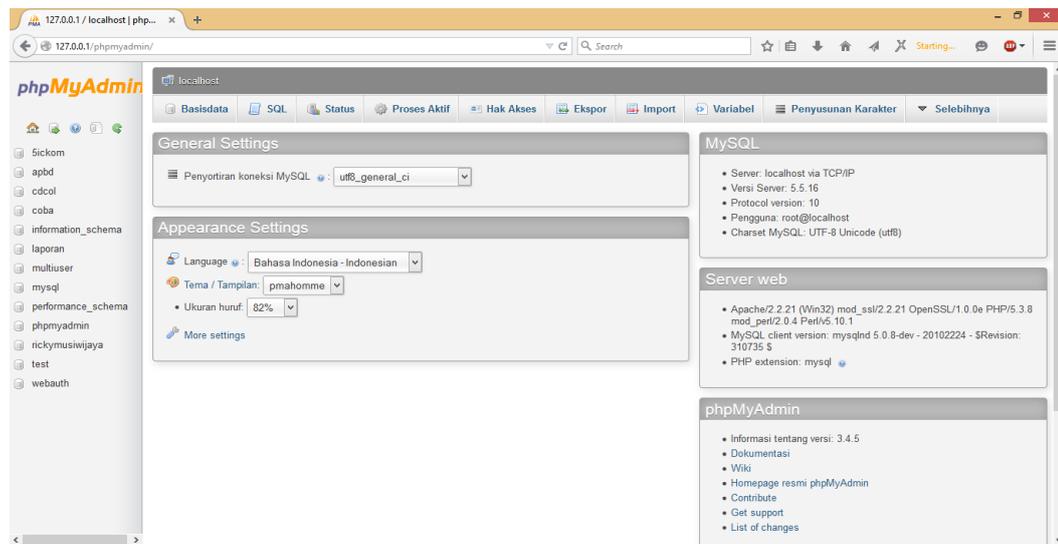


Gambar 2.2 Tampilan XAMPP

(Sumber: Nugroho, 2013:1)

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



Gambar 2.3 Tampilan phpMyAdmin

(Sumber: Nugroho, 2013:71)

2.3.8. Tentang CSS

Saleh et.al (2007:53), 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.

Betha dan Husni (2014 : 132), secara umum disusun oleh tiga bagian yaitu, selector (elemen yang akan didefinisikan), property (atribut yang akan diubah) dan nilai sebagaimana berikut:

Selector {property : value }

Antara property dan nilai disahkan dengan titik-dua (colon) seperti contoh di bawah ini:

Body {color : black }

Jika nilai berupa beberapa kata, gunakan tanda petik ganda:

p{font - family : “sans serif”}

Jika lebih dari satu property pisahkan dengan titik-koma (semi colon):

```
p{text – align : center; color : red}
```

Jika ingin lebih mudah dibaca sebagaimana berikut:

```
p
{
text – align : center ;
color : black ;
font – family : arial
}
```

Jika selector dikelompokkan:

```
H1, H2, H3, H4, H5, H6
{
color : green ;
}
```

Jika menggunakan atribut class (dalam definisi CSS):

```
P.Kanan {text – align : right}
P.Kiri {text – align : center}
```

Berikut ini sintaks yang ditulis dalam dokumen html menggunakan definisi di atas:

```
<P class=”kanan”> This Paragraph will be right aligned <p>
<P class=”kiri”> This Paragraph will be center aligned <p>
```

Jika menggunakan ID atribut:

```
#Kanan {text – align : right}
```

2.3.9. Tentang 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.

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 2.6. 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

(Sumber: Betha dan Husni, 2014:272)