

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Pengertian Katalog**

Katalog berasal dari bahasa Indonesia berasal dari kata *Catalog* dalam bahasa Belanda, serta *Catalogue* dari bahasa Inggris. Istilah katalog itu sendiri berasal dari frase Yunani *Katalogos*. Kata bermakna sarana atau menurut, sedangkan *logos* memiliki berbagai arti seperti kata, susunan, alasan dan nalar.

Jadi katalog dari segi kata bermakna sebuah karya dengan isinya disusun menurut cara yang masuk akal.

Beberapa definisi katalog menurut ilmu perpustakaan dapat disebutkan sebagai berikut :

- a. Katalog berarti daftar berbagai jenis koleksi perpustakaan yang disusun menurut sistem tertentu. (Fathmi, 2004)
- b. Katalog perpustakaan adalah daftar buku atau koleksi pustaka dalam suatu perpustakaan atau dalam suatu koleksi. (Sulistyo-Basuki, 1991)

##### **3.1.1 Tujuan Katalog**

Menurut Sulistyo-Basuki (1991) tujuan dari Katalog adalah sebagai berikut:

1. Memungkinkan seorang menemukan sebuah buku yang diketahui pengarangnya, judulnya atau subjeknya.
2. Menunjukkan buku yang dimiliki perpustakaan oleh pengarang tertentu, berdasarkan subjek tertentu dan dalam jenis literatur tertentu.

3. Membantu dalam pemilihan buku berdasarkan edisinya dan berdasarkan karakternya (sastra atau kah berdasarkan topik).

### 3.1.2 Fungsi Katalog

1. Memudahkan *user* dalam *retrieval* bahan pustaka yang dibutuhkan berdasarkan informasi yang mereka ketahui, bisa berupa judul, pengarang, maupun berdasarkan bidang ilmu yang diperlukan.
2. Untuk menunjukkan apakah perpustakaan memiliki buku yang dikarang oleh pengarang tertentu, mengenai subjek tertentu dan dalam bentuk tertentu.
3. Sebagai wakil ringkas dari bahan pustaka yang dimiliki perpustakaan.

Qalyubi dkk (2007) menyebutkan fungsi katalog adalah sebagai berikut:

1. Mencatat karya seseorang pada tajuk yang sama.
2. Menyusun entri pengarang secara tepat sehingga semua karya seseorang berada pada tajuk yang sama.
3. Mencatat semua judul bahan pustaka yang dimiliki suatu perpustakaan.
4. Menunjukkan rujukan silang (*cross reference*) dari beberapa istilah atau nama-nama yang sama yang digunakan sebagai tajuk.
5. Memberikan petunjuk letak/lokasi bahan pustaka yang disusun pada perpustakaan. memberikan uraian tentang setiap karya yang dimiliki suatu perpustakaan sehingga pengguna perpustakaan (*user*) dapat memperoleh informasi yang lengkap tentang karya itu.

Sedangkan Menurut Kao (2001), fungsi katalog adalah sebagai berikut:

1. Memungkinkan seseorang menemukan sebuah buku yang diketahui pengarangnya, judulnya atau subyeknya.
2. Menunjukkan buku yang dimiliki perpustakaan dari pengarang tertentu, berdasarkan subyek tertentu, atau dalam jenis literature tertentu.
3. Membantu dalam pemilihan buku berdasarkan edisinya atau berdasarkan karakternya.
4. Berfungsi sebagai sarana yang sangat diperlukan oleh staf perpustakaan di bagian pengadaan, pengatalogan, kontrol inventarisasi dan pekerjaan-pekerjaan referensi.

### **3.2 Web Application**

Pada awalnya aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut *HTML (HyperText Markup Language)*. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan *HTML* seperti *PHP* dan *ASP* pada skrip dan *Applet* pada objek. Aplikasi Web dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis.

Web statis dibentuk dengan menggunakan *HTML*. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi web dinamis. Pada aplikasi web dinamis, perubahan informasi dalam halaman web dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi web dapat dikoneksikan ke basis

data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari *webmaster*.

Arsitektur aplikasi web meliputi klien, *web server*, *middleware* dan basis data. Klien berinteraksi dengan *web server*. Secara internal, *web server* berkomunikasi dengan *middleware* dan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah *PHP* dan *ASP*. Pada mekanisme aplikasi web dinamis, terjadi tambahan proses yaitu server menerjemahkan kode *PHP* menjadi kode *HTML*. Kode *PHP* yang diterjemahkan oleh mesin *PHP* yang akan diterima oleh klien. (Kadir, 2010)

### 3.3 *PHP (Personal Home Page)*

*PHP* adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML (HyperText Markup Language)*. (Oktavian, 2010)

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (Situs personal). Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP/FI* pada sekitar tahun 1995, dan diperkenalkan kepada beberapa programmer pemula dengan alasan bahasa yang digunakan oleh *PHP (Personal Home Page)* cukup sederhana dan mudah dipahami. Selanjutnya Rasmus menulis ulang *PHP (Personal Home Page)* dengan bahasa *C* untuk meningkatkan kecepatan aksesnya.

*Script PHP (Personal Home Page)* adalah bahasa program yang berjalan pada sebuah *web server*, atau sering disebut *server-side*. Oleh karena itu, *PHP*

(*Personal Home Page*) dapat melakukan apa saja yang bisa dilakukan program *CGI* lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan *cookies*, dan bahkan *PHP* bisa melakukan lebih dari itu.

Arti *script server-side* adalah, agar dapat menjalankan *script* ini dibutuhkan tiga program utama, yaitu *web-server* (dapat berupa *IIS* dari *windows* atau *apache*), modul *PHP* dan juga *web browser*.

*PHP* dapat berjalan pada semua jenis sistem operasi, antara lain pada *Linux* dan varian *Unix* (*HP-UX*, *Solaris* dan *OpenBSD*), pada *Ms Windows*, *Mac* dan masih banyak lagi, selain itu *PHP* juga dapat berjalan pada beberapa jenis *web-server* antara lain *Apache*, *Microsoft IIS*, *personal web server*, *Netscape* dan *Iplanet Server*, *Caudium*, *Xitami*, *Omnihttpd* dan masih banyak lagi.

Apabila Anda menggunakan *PHP*, maka Anda banyak memiliki alternatif Sistem Operasi atau *web server* untuk menjalankannya, selain itu Anda juga dapat menjalankan atau menggunakan program berorientasi object atau sering disebut *Object Oriented Programming (OOP)*.

*Output* yang dihasilkan *PHP* bukan hanya *HTML*, namun juga dalam bentuk gambar, *file PDF*, serta gambar animasi menggunakan *Libswf* dan *Ming*. *Output* yang lain dengan jenis teks dapat berupa *file XHTML* dan *XML*.

*Script PHP* Anda akan disembunyikan dalam *server* dan *output* teks yang ditampilkan di *browser* adalah format teks *HTML*, *XHTML* atau *XML*.

*PHP* juga banyak mendukung *database* populer yang sering digunakan dalam beberapa *web server*, antara lain *Adabas D*, *Empress*, *FrontBase*,

*PostgreSQL* , *FilePro(read only)*, *mSQL*, *Solid*, *Hyperwave*, *Direct MS-SQL*, *Sybase*, *IBM DB2*, *MySQL*, *Velocis*, *Informix*, *ODBC* dan *Unix dbm*.

*PHP* juga mendukung koneksi dengan layanan jasa yang menggunakan protokol seperti *LDAP*, *IMAP*, *SNMP*, *NNTP*, *POP3*, *HTTP* dan *COM* (pada *windows*). Selain itu, juga dapat melakukan koneksi dengan jaringan yang menggunakan protokol lain, *PHP* mempunyai dukungan *WDDX*, sehingga dapat bertukar data antar bahasa pemrograman web.

### **3.4 Pengertian *Bootstrap***

*Bootstrap* adalah sebuah *Framework* atau *Tools* untuk membuat desain *template website* yang *responsive* secara cepat, mudah dan gratis. Desain ini memiliki kemampuan untuk mengelola aset media dengan efektif. Website dengan desain responsif mampu memberikan respon secara otomatis terhadap ukuran layar, sehingga situs dapat beradaptasi dengan optimal dalam perangkat apapun yang digunakan pengguna untuk mengaksesnya. (Clatworthy, 2011)

*Twitter Bootstrap* dibangun dengan teknologi *HTML* dan *CSS* yang dapat membuat *layout* halaman *website*, tabel, tombol, *form*, navigasi, dan komponen lainnya dalam sebuah *website* hanya dengan memanggil fungsi *CSS (class)* dalam berkas *HTML* yang telah didefinisikan. Selain itu juga terdapat komponen-komponen lainnya yang dibangun menggunakan *JavaScript*.

### **3.5 Oracle XE**

*Oracle XE (Express Edition)* adalah sebuah produk *database server* yang berlisensi *freeware* dari *Oracle Corporation*. Dengan produk ini, para pemakai

*Oracle XE* dapat menggunakannya tidak hanya untuk percobaan, tetapi juga dapat digunakan untuk pengembangan dan *deployment* sistem. Sampai saat ini, terdapat dua versi *Oracle*, yaitu *Oracle 10g XE* dan *Oracle 11g R2 XE*. Keduanya memberikan konsep antarmuka pemakai untuk pemakaian *database* yang berbeda serta sama-sama menggunakan teknologi *Oracle Application Express* untuk menyediakan layanan antarmuka pemakai berbasis web untuk mengatur dan memanfaatkan *database Oracle XE*. (Susanto, 2012)

Dalam *Oracle 10g XE* terdapat beberapa batasan yang perlu diperhatikan terkait dengan penggunaan sumber komputer. Pembatasan pertama adalah pemakaian ruang memori pada *database Oracle 10g XE* hanya dapat mengalami sampai 1 GB RAM. Pembatasan ini akan mempengaruhi bagaimana pemakai dapat mengakses *database* secara bersamaan dan bagaimana kinerja yang diberikan oleh *database Oracle 10g XE*. Dengan kondisi *Oracle 10g XE* yang terpasang pada komputer *Pentium IV* dan 1 GB RAM, maka jumlah maksimum pemakai yang dapat membentuk sesi koneksi adalah kurang lebih 16 koneksi.

Pembatasan kedua, *Oracle 10g XE* hanya akan menggunakan satu CPU. *Oracle 10g XE* akan berjalan pada sebuah komputer *multi-CPU*, tapi hanya akan menggunakan satu CPU saja. Jika ingin memfungsikan *multi-CPU* dalam satu komputer, maka dibutuhkan *Oracle Database Standard Edition* atau *Enterprise Edition*. Batasan ketiga, *database Oracle 10g XE* hanya dapat berjalan secara tunggal di sembarang komputer. Pemakai dapat menggunakan konsep skema untuk memisahkan satu aplikasi dengan aplikasi lain sehingga tidak memerlukan *database* untuk setiap aplikasi.

Pembatasan yang terakhir adalah pembatasan ruang penyimpanannya hanya sampai 4 GB tidak boleh lebih, tapi pemakai dapat menghubungkan *Application Express* ke *database* lain untuk menambah ruang penyimpanan sesuai kebutuhan. (Susanto, 2012)

Beberapa kebutuhan yang harus diperhatikan saat akan menginstalasi *Oracle 10g XE* pada lingkungan *Microsoft Windows*, sebagai berikut:

1. *Random Access Memory (RAM)* minimum adalah 256 MB (direkomendasikan 512 MB).
2. Ruang penyimpanan minimal komponen *server* adalah 1,6 GB dan komponen *client* 75 MB (disarankan agar sistem file diformat menggunakan *NTFS* yang memiliki *Access Control*).
3. Prosesor minimum adalah *Intel (x86) Pentium 4*.
4. Sistem operasi minimum adalah *Windows SP1* atau *Windows Server 2003* atau *Windows 2000 SP4*.

Sedangkan pada lingkungan *Linux*, kebutuhan komputer yang harus dipersiapkan adalah, sebagai berikut:

1. *Random Access Memory (RAM)* minimum adalah 256 MB (direkomendasikan 512 MB).
2. Ruang penyimpanan minimal komponen *server* adalah 1,5 GB dan komponen *client* 100 MB.
3. Kebutuhan *Swapfile* (dalam MB): (a) *RAM* antara 0 – 256, ukuran *swap* 3 x *RAM* (b) *RAM* antara 256 – 512, ukuran *swap* 2 x *RAM* (c) *RAM* antara 512 – 1024, ukuran *swap* 1024 M.

4. Prosesor minimum adalah *Intel (x86) Pentium 4*.

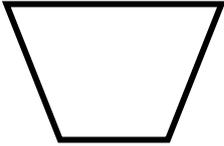
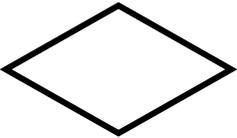
### 3.6 Pengertian *Document Flowchart*

Menurut Jogiyanto (2005), *flowchart* (bagan alir) adalah bagan (*charts*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem.

*Document Flowcharts* adalah Bagan alir dokumen atau bisa disebut juga sebagai bagan alir formulir yang merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Dalam pembuatannya, *document flowcharts* memiliki ketentuan-ketentuan yang harus diperhatikan. Salah satunya adalah notasi-notasi yang ada di dalamnya. Notasi yang umumnya ada pada *document flowcharts* antara lain:

Tabel 1. Simbol *Document Flowcharts*

No	Simbol	Nama Simbol	Fungsi
1.		<i>Terminator</i>	Merupakan bentuk dari <i>terminator</i> simbol yang digunakan pada awal pembuatan <i>document flowcharts</i> sebagai mengawali ( <i>Start</i> ) dan mengakhiri ( <i>End</i> ) <i>flowcharts</i> .

2.		<p><i>Manual Process</i> (Proses Manual)</p>	<p>Merupakan notasi dari proses manual yang pada <i>document flowcharts</i>. Dinyatakan sebagai proses manual karena dalam notasi <i>document flowcharts</i> segala bentuk proses masih belum dilakukan oleh komputer.</p>
3.		<p>Dokumen</p>	<p>Merupakan notasi dari dokumen pada <i>document flowcharts</i>. Notasi document ini umumnya digambarkan sebagai bentuk lain dari arsip, laporan atau dokumen lainnya yang berbentuk kertas</p>
4.		<p><i>Decision</i> (Keputusan)</p>	<p>Merupakan notasi dari suatu keputusan dalam pengerjaan <i>document flowcharts</i>. Dalam penggambaran notasi <i>decision</i> ini selalu menghasilkan dengan keputusan ya atau tidak.</p>

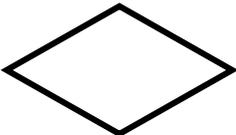
### 3.7 Pengertian System Flowchart

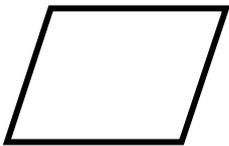
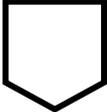
Menurut Jogiyanto (2005), bagan alir program (*system flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Pembuatan *System Flowcharts* memiliki aturan dan ketentuan yang harus diikuti. Seperti halnya dalam pembuatan *Document Flowcharts* sebelumnya, *System Flowcharts* memiliki notasi-notasi sebagai representasi dari proses kerja suatu sistem.

Sebagian notasi dalam *System Flowchart* memiliki kesamaan dengan notasi yang ada pada *document flow* seperti, *terminator (start/end)*, dan notasi laporan. Selain kedua notasi tersebut terdapat perbedaan secara bentuk dan fungsinya. Adapun simbol-simbol lain yang ada pada *System Flowchart* akan di jelaskan pada tabel 2.

Tabel 2. Simbol *System Flowcharts*

No	Simbol	Nama Simbol	Fungsi
1.		Proses Komputerisasi	Simbol di samping merupakan representasi dari proses sistem. Menggambarkan sistem yang dikerjakan oleh komputer (otomatis).
2.		<i>Database</i>	Gambar disamping adalah representasi dari <i>Database</i>

No	Simbol	Nama Simbol	Fungsi
			yang mana fungsinya untuk menyimpan data dari proses sebelumnya.
3.		<i>Connector</i>	<i>Connector</i> difungsikan sebagai penunjuk arah aliran dari satu proses ke proses yang lainnya yang saling berkaitan.
4.		<i>Sub-Process</i>	Simbol <i>Sub-Process</i> difungsikan untuk menunjukkan adanya proses yang lebih rinci dari suatu proses utama.
5.		<i>Document</i>	Simbol <i>Document</i> menunjukkan tentang dokumen yang dihasilkan
6.		<i>Decision</i> (Keputusan)	Simbol di samping difungsikan sebagai langkah pengambil keputusan . Keputusan yang ada terkait “ya” atau “tidak” keputusan diambil.

No	Simbol	Nama Simbol	Fungsi
7.		<i>Input/Output</i>	Simbol di samping difungsikan untuk menunjukkan masukan data ( <i>input</i> ) dan data yang dihasilkan ( <i>output</i> ).
8.		<i>Connector (On Page Reference)</i>	Untuk menunjukkan hubungan simbol yang saling terkait dalam <i>System Flowchart</i> . Selain itu juga sebagai pengganti garis <i>connector</i> untuk menghubungkan simbol yang saling berjauhan.
9.		<i>Connector (Off- Page Reference)</i>	Fungsinya hampir sama dengan <i>connector (on page reference)</i> perbedaannya adalah untuk menghubungkan simbol yang berada pada halaman yang berbeda.

### 3.8 Pengertian *Data Flow Diagram (DFD)*

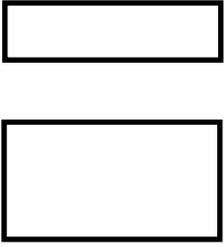
*Data Flow Diagram (DFD)* menurut Jogiyanto (2005), *DFD* adalah diagram yang menggunakan notasi simbol untuk menggambarkan arus *data system*. *Data Flow Diagram (DFD)* menggambarkan komponen-komponen yang

ada pada suatu sistem serta aliran-aliran data. Fungsi utama dari *DFD* yaitu untuk memudahkan analisis sistem untuk menggambarkan suatu sistem sebagai satu fungsi yang saling berhubungan antara satu dengan yang lain.

Pada perancangannya, *Data Flow Diagram* berorientasi pada alur data dengan konsep dekomposisi yang digunakan untuk penggambaran analisa maupun rancangan sistem yang memudahkan seorang profesional TI dalam mengkomunikasikan rancangan sistem yang akan dibuat kepada pengguna atau pembuat program. Dalam pembuatannya, *DFD* memiliki simbol-simbol di dalamnya. Adapun simbol tersebut akan dijelaskan pada tabel 3.

Tabel 3. Komponen *Data Flow Diagram*

No	Simbol	Nama Simbol	Fungsi
1.		<i>Terminator</i>	Untuk menggambarkan asal atau tujuan data di luar sistem. Sebagai penggambaran dari entitas eksternal (orang, sekelompok orang, organisasi, departmen organisasi, dsb) yang sama tetapi di luar kendali sistem yang akan dibuat.

No	Simbol	Nama Simbol	Fungsi
2.		Proses	<p>Untuk menggambarkan bagian dari sistem yang memproses inputan menjadi <i>output</i>. Setiap proses diberi nama dengan menggunakan kata kerja transitif, seperti ; mencatat data <i>master</i>, dsb.</p>
3.		<i>Data Store</i>	<p>Untuk menggambarkan media penyimpanan data, seperti file atau <i>database</i>. Media penyimpanan berkaitan dengan penyimpanan secara komputerisasi.</p>
4.		Alur Data	<p>Untuk menggambarkan arah keluar masuknya dari suatu proses. Alur data digunakan untuk menjelaskan proses perpindahan data/informasi dari satu proses ke proses lain.</p>

Terdapat beberapa jenis *DFD* yang mana dibuat dan digunakan oleh seorang analis dalam merancang suatu sistem. Dalam perancangan *Data Flow Diagram* (*DFD*), terdapat beberapa jenis *DFD* tersebut memiliki fungsi dan aturan penulisan yang berbeda, yaitu :

1. *Context Diagram*

*Context Diagram* merupakan jenis pertama dari *Data Flow Diagram* (*DFD*). *Context Diagram* ini merupakan diagram tingkat atas di dalam penulisan *DFD*. Fungsi dari *Context Diagram* sendiri adalah untuk menggambarkan sebuah sistem informasi secara global (umum) termasuk masuk dan keluarnya data. Di dalam *context diagram* terdapat suatu lingkaran yang mana lingkaran tersebutlah yang menjadi simbol dari sistem secara keseluruhan.

2. *Data Flow Diagram Fisik*

Merupakan representasi grafik dari sebuah sistem untuk menunjukkan entitas-entitas internal dan eksternal dari sistem itu sendiri. Entitas internal dapat berupa tempat atau bagian dari sistem untuk mentransformasikan data. *DFD* fisik tidak menunjukkan unsur proses dari suatu sistem melainkan lebih kepada bagaimana, dimana dan oleh siapa proses dilakukan.

3. *Data Flow Diagram Logis*

Merupakan representasi grafik dari sebuah sistem yang menjelaskan tentang proses-proses di dalamnya dan aliran-aliran data ke dalam maupun ke luar dari proses tersebut. *DFD* Logis dibuat untuk mendokumentasikan sebuah sistem

informasi karena *DFD* tersebut mewakili logika dari suatu sistem. Jadi, *DFD* Logis ini lebih fokus pada fungsi-fungsi yang dilakukan sistem.

#### 4. *Data Flow Diagram Levelled*

*DFD Levelled* dapat digunakan untuk menggambarkan diagram fisik maupun diagram logis. *DFD* ini merupakan penjabaran dari *Context Diagram* ke diagram yang lebih terperinci dengan cara pendekomposisian. *DFD Levelled* dapat dijabarkan ke dalam level atau tingkat tertentu tergantung seberapa rinci penjabarannya. Ketentuan dalam pembuatan *DFD Levelled* ini harus mampu menggambarkan proses dari suatu sistem sedetail mungkin. Apabila penggambaran masih dianggap luas atau belum cukup jelas maka seharusnya diturunkan ke *level* yang lebih rendah lagi.

### 3.9 Pengertian *Database Management System (DBMS)*

Menurut Connolly (2005), Sistem Manajemen Basis data (*DBMS*) merupakan suatu sistem perangkat lunak (*software*) yang membantu pemakai dalam mendefinisikan, menciptakan, mengatur dan mengontrol akses pada suatu basis data.

*DBMS* menyediakan beberapa fasilitas sebagai berikut :

#### 1. *Data Definition Language (DDL)*

Memungkinkan *user* untuk membuat spesifikasi tipe data, mendefinisikan basis data, struktur data dan *data constraint*.

#### 2. *Data Manipulation Language (DML)*

Memungkinkan pemakai untuk memasukkan, memperbaharui, menghapus, dan mengirimkan atau mengambil data dari basis data.

Menurut Connolly (2002), ada lima komponen Sistem Manajemen Basis data (*DBMS*), yaitu :

1. *Hardware* (Perangkat Keras)

Perangkat keras yang dibutuhkan untuk menjalankan *DBMS* dan aplikasi-aplikasi. Contoh: *single personal computer*, *single mainframe*, atau komputer yang menggunakan jaringan.

2. *Software* (Perangkat Lunak)

Komponen perangkat lunak terdiri dari perangkat lunak *DBMS* itu sendiri dan program-program aplikasi, bersama dengan sistem operasi, termasuk perangkat lunak jaringan.

3. Data

Data merupakan komponen yang paling penting dari *DBMS*, khususnya dari sudut pandang pemakai akhir mengenai data.

4. Prosedur

Cara untuk menjalankan sistem, seperti bagaimana masuk ke dalam *DBMS* memulai dan menghentikan *DBMS*, bagaimana membuat *data backup* dari basis data.

5. Manusia

Komponen terakhir adalah manusia yang terlibat dengan sistem, termasuk didalamnya adalah *Database Administrator (DBA)*, perancang basis data, pengembang aplikasi, dan pemakai akhir.

### 3.10 Pengertian Perancangan Basis Data

*Database Design* (Perancangan Basis Data) merupakan salah satu aplikasi yang paling umum dan dikenal oleh banyak orang. Untuk merancang suatu *database* dimulai dengan melakukan analisis terhadap kebutuhan perusahaan. di dalam perancangan *database* terdapat metodologi sebagai acuan dalam pembuatan suatu basis data.

Menurut Connolly (2005) Metodologi perancangan basis data terdiri dari tahap- tahap yang membantu perancang dengan teknik yang tepat dalam setiap merancang basis data. Metodologi perancangan basis data juga membantu perancangan untuk merencanakan, mengatur dan mengevaluasi pengembangan dari proyek pembuatan basis data tersebut.

Dalam metodologi perancangan basis data menurut Connolly, terdapat 3 proses, yaitu *conceptual database design*, *logical design*, dan *physical database design*.

1. *Conceptual Database Design (CDM)*

Proses membangun sebuah model dari informasi yang digunakan dalam perusahaan, terbebas dari semua pertimbangan fisik. *Conceptual database design* meliputi pembuatan sebuah konseptual data model sebagai bagian dari perusahaan. *Data model* dibangun menggunakan informasi yang didokumentasikan dari *user requirement*. *Conceptual database design* secara keseluruhan tidak mencakup detail implementasinya, seperti *DBMS software* yang dipakai, aplikasi program, *programming, language, hardware platform* atau pertimbangan fisik lainnya.

## 2. *Logical Database Design*

Proses membangun sebuah model informasi yang digunakan dalam sebuah perusahaan berdasarkan pada sebuah data model tertentu tetapi terbebas dari penggunaan *DBMS* tertentu dan pertimbangan fisik lainnya. *Conceptual data model* yang dibuat pada tahap sebelumnya disempumakan dan dipetakan menjadi sebuah *logical data model*.

## 3. *Physical Database Design (PDM)*

*Physical database design* dilakukan untuk memutuskan struktur *logic* secara fisik diimplementasikan ke dalam tujuan Sistem Manajemen Basis data (*DBMS*), para perancang juga harus membuat keputusan mengenai bagaimana basis data tersebut dapat diimplementasikan dan diterapkan. Oleh karena itu, *physical database design* harus disesuaikan dengan *DBMS* yang spesifik. Terdapat hubungan antara *physical* dan *logical database design*, karena keputusan yang diambil pada *physical database design* untuk meningkatkan kinerja dari basis data tersebut dapat mempengaruhi *logical data model*.