

## BAB III

### LANDASAN TEORI

Pada bab ini akan membahas beberapa landasan teori penulis yang meliputi hal-hal dari permasalahan yang ada dan teori yang membahas tentang ilmu yang terkait dalam permasalahan pada kerja praktik ini.

#### 3.1 Jasa

Jasa menurut Phillip Kotler jasa adalah setiap tindakan atau unjuk kerja yang ditawarkan oleh salah satu pihak ke pihak lain yang secara prinsip *intangibel* dan tidak menyebabkan perpindahan kepemilikan apapun. Produksinya bisa terkait dan bisa juga tidak terikat pada suatu produk fisik.

Jasa adalah aktifitas ekonomi yang mempunyai sejumlah elemen (nilai atau manfaat) *intangibel* yang berkaitan dengannya, yang melibatkan sejumlah interaksi dengan konsumen atau dengan barang-barang milik, tetapi tidak menghasilkan transfer kepemilikan. Perubahan dalam kondisi bisa saja muncul dan produksi suatu jasa bisa memiliki atau bisa juga tidak mempunyai kaitan dengan produk fisik. (Adrian Payne, 2001:9)

Jasa adalah proses yang terdiri atas serangkaian aktivitas intangible yang biasanya (namun tidak harus selalu) terjadi pada interaksi antara pelanggan dan karyawan jasa dan atau sumber daya fisik atau barang dan atau sistem penyedia jasa, yang disediakan sebagai solusi atas masalah pelanggan. Interaksi antara penyedia jasa dan pelanggan kerap kali terjadi dalam jasa, sekalipun pihak-pihak yang terlibat mungkin tidak menyadarinya. (Christian Gronross, C.1992. *Service Management and Marketing*. Lexington Books. Massachusetts, Toronto)

Seringkali dikatakan bahwa jasa memiliki karakteristik unik yang membedakannya dari barang atau produk-produk manufaktur. Empat karakteristik yang paling sering dijumpai dalam jasa dan pembeda dari barang pada umumnya adalah :

A. Tidak berwujud

Jasa bersifat abstrak dan tidak berwujud, berarti jasa tidak dapat dilihat, dirasakan, dicicipi atau disentuh seperti yang dapat dirasakan dari suatu barang.

B. *Heterogenitas*

Jasa merupakan variabel non – standar dan sangat bervariasi. Artinya, karena jasa itu berupa suatu unjuk kerja, maka tidak ada hasil jasa yang sama walaupun dikerjakan oleh satu orang. Hal ini dikarenakan oleh interaksi manusia (karyawan dan konsumen) dengan segala perbedaan harapan dan persepsi yang menyertai interaksi tersebut.

C. Tidak dapat dipisahkan

Jasa umumnya dihasilkan dan dikonsumsi pada saat yang bersamaan, dengan partisipasi konsumen dalam proses tersebut. Berarti, konsumen harus berada di tempat jasa yang dimintanya, sehingga konsumen melihat dan bahkan ikut ambil bagian dalam proses produksi tersebut.

#### D. Tidak tahan lama

Jasa tidak mungkin disimpan dalam persediaan. Artinya, jasa tidak bisa disimpan, dijual kembali kepada orang lain, atau dikembalikan kepada produsen jasa dimana ia membeli jasa.

### 3.1.1 Pengiriman

Menurut Kamus Umum Bahasa Indonesia (1999, p512), pengiriman adalah kiriman, hal (perbuatan dan sebagainya) mengirimkan.

Pengiriman adalah proses pengangkutan barang-barang. Kebanyakan barang-barang diantarkan lewat jaringan transportasi. Muatan (barang-barang fisik) terutama diantarkan melalui darat dengan memakai kereta api, melalui jalur laut dengan menggunakan kapal laut dan melalui udara dengan menggunakan perusahaan penerbangan. ([www.wikipedia.org](http://www.wikipedia.org))

### 3.1.2 Pihak-pihak yang terlibat dalam Jasa Pengiriman

Pihak-pihak yang terlibat dalam Jasa Pengiriman yaitu :

#### A. Penyedia Jasa

Penyedia Jasa adalah pemilik usaha jasa pengiriman yang menyalurkan jasa pesanan dari pelanggan.

#### B. Pelanggan

Pelanggan adalah pihak pemakai jasa pengiriman yang disediakan oleh pihak penyedia jasa.

## 3.2 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau

elemennya. Pendekatan sistem lebih menekankan pada prosedur sistem adalah sebagai berikut:

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

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Prosedur (*procedure*) didefinisikan oleh (Riyanto, 2009) sebagai berikut :

“Prosedur adalah sebuah perintah yang dapat digunakan untuk membagi beberapa kejadian dalam suatu kumpulan perintah yang lebih kecil dengan berbagai kelengkapan di dalamnya baik itu pengecekan kondisi, fungsi matematika maupun fungsi *string*.” (Riyanto, 2009)

### **3.2.1 Pengertian Sistem**

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu, (Hartono, 1999). Sedangkan definisi menurut Kusri (2007:5), Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

### **3.2.2 Karakteristik Sistem**

Menurut Kusri (2007:7) Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain :

1. Komponen Sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling bekerja sama untuk membentuk suatu komponen sistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*Boundary*)

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem

Bagian-bagian dari sistem yang beraktifitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. Lingkungan Luar Sistem (*Environment*)

Suatu sistem yang ada di luar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media penghubung antara suatu subsitem dengan subsitem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsitem ke subsistem lainnya.

6. Masukan Sistem (*Input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

### 7. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

### 8. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.

### 9. Sasaran Sistem (*Object*)

Tujuan yang ingin di capai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

## 3.3 Konsep Dasar Aplikasi

Aplikasi adalah “penerapan, pengimplementasian suatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru, tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan.” (Kristanto, 2003).

## 3.4 Data

Menurut Herlambang (2005:121) data berasal dari bahasa Latin yaitu *datum* yang berarti fakta, kenyataan, kejadian atau peristiwa. Jadi data atau fakta adalah kenyataan dari sesuatu kejadian atau peristiwa. Data dapat didefinisikan yaitu kumpulan fakta-fakta yang berupa fisik dan bukan fisik, kejadian-kejadian dan prosedur yang belum diolah manusia atau peralatan yang digunakan oleh manusia. Kegunaan dari data adalah sebagai bahan dasar yang objektif di dalam proses penyusunan kebijaksanaan dan keputusan oleh pimpinan organisasi. Data merupakan

keterangan yang masih mentah (belum diolah). Agar dapat dipergunakan, maka data tersebut harus diolah terlebih dahulu ke dalam bentuk informasi yang sesuai dengan keperluan yang dibutuhkan.

### 3.5 *System Development Life Cycle (SDLC)*

Menurut McLeod (2008 : 199) Siklus hidup sistem (*system life cycle*) disingkat SLC adalah proses *evolusioner* dalam menetapkan sistem dan sub sistem informasi berbasis komputer. SLC yang juga dikenal sebagai pendekatan air terjun (*waterfall approach*) terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem, karena proses tersebut mengikuti sebuah pola yang teratur dan dilakukan secara *top-down*.

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni: siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan *prototyping* (*life cycle using prototyping*), dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*).

Sedangkan *System Development Life Cycle* atau yang disingkat SDLC adalah metoda tradisional yang digunakan untuk membangun, memelihara dan mengganti suatu sistem informasi. *System Development Life Cycle (SDLC)* terdiri dari tujuh fase, diantaranya adalah :

a. *Project Identification and Selection*

Fase dimana kebutuhan sistem informasi secara keseluruhan diidentifikasi dan analisa.

b. *Project Initiation and Planning*

Fase dimana suatu proyek sistem informasi yang potensial dilakukan dan direncanakan terinci dikembangkan untuk pengembangan sistem.

c. *Analisis*

Suatu fase dimana sistem yang sedang berjalan dipelajari dan alternatif sistem baru diusulkan.

d. *Logical Design*

Suatu fase dimana semua kegiatan fungsional dari sistem yang diusulkan untuk dikembangkan dan digambarkan secara independent.

e. *Physical Design*

Fase rancangan logis dari sebelumnya diubah dalam bentuk teknis yang terinci dimana pemrograman dan bentuk sistem dapat dibuat.

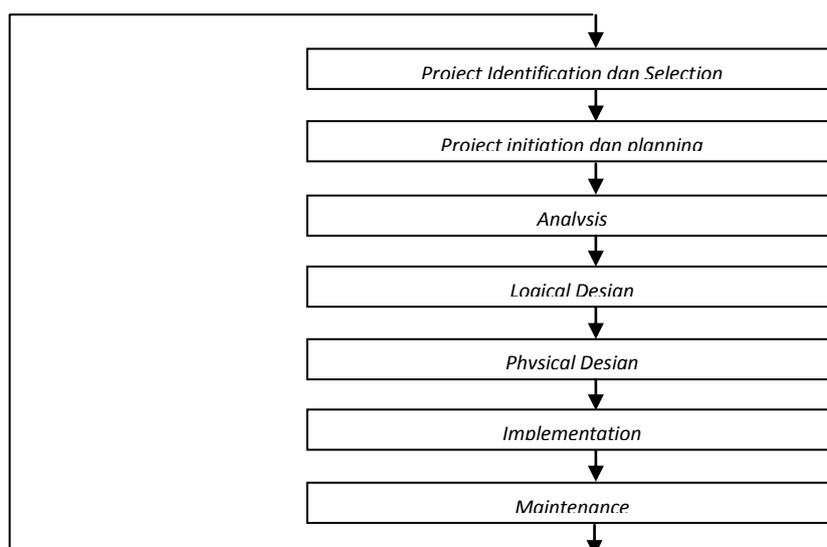
f. *Implementation*

Suatu fase dimana sistem informasi diuji dan digunakan untuk mendukung suatu organisasi.

g. *Maintenance*

Dimana sistem informasi secara sistematis diperbaiki dan dikembangkan.

Komponen dari *System Development Life Cycle* (SDLC) ini seperti yang terlihat pada gambar 3.1:



**Gambar 3.1:** *System Development Life Cycle*

Siklus hidup pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan-tahapan utama, dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya. Tiap-tiap pengembangan sistem itu dibagi menjadi beberapa tahapan kerja. Tiap tahapan ini mempunyai karakteristik tersendiri. Sebagai awal dari pelaksanaan pengembangan sistem adalah proses kebijaksanaan dan perencanaan sistem. Dimana kebijaksanaan sistem merupakan landasan dan dukungan dari manajemen puncak untuk membuat perencanaan sistem. Sedangkan perencanaan sistem merupakan pedoman untuk melakukan pengembangan dari sistem tersebut.

### **3.6 Data Flow Diagram (DFD)**

Menurut (Kristanto, 2003), *Data Flow Diagram* (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan

atau menjelaskan DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

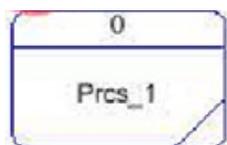
### 3.6.1 Simbol-Simbol yang digunakan dalam DFD

#### 1. *Process*

Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan *level* tertinggi dari DFD yang menggambarkan hubungan sistem dengan terlebih dahulu menentukan *terminator* yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

Nomor 1,2,3 dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses *level-0* (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses *level-0* membahas sistem secara lebih *detail*. Baik dipandang dari segi kegiatan melalui sebuah bagian, alur data yang ada, maupun *database* yang digunakan didalamnya. Pembuatanya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem. Menentukan alur data yang diterima dan diberikan masing-masing proses daripada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar

dari suatu *level* harus sama dengan alur data yang masuk dan keluar pada *level* berikutnya), memunculkan *data store* sebagai sumber maupun tujuan data (*optional*), menggambarkan diagram *level-0*, menghindari perpotongan arus data, dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses). Simbol dari *process* dapat dilihat pada gambar 3.2



**Gambar 3.2:** *Process*

## 2. *External Entity*

Disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang organisasi maupun instansi. *External Entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. Simbol dari *External Entity* dapat dilihat pada gambar 3.3



**Gambar 3.3:** *External Entity*

### 3. *Data Store*

Media	Nama Data Store
-------	-----------------

**Gambar 3.4:** *Data Store*

Digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file* dan tabel-tabel dalam *database*. Penanaman *data store* harus sesuai dengan bentuk data yang tersimpan pada *data store* tersebut, misalnya tabel pelamar, tabel pendidikan, tabel lulus seleksi dan lain-lain. Simbol dari *Data Store* dapat dilihat pada gambar 3.4

### 4. *Data Flow*

Merupakan penghubung antara *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran dari satu titik ketitik lainnya dengan tanda anak panah mengarah ke tujuan data. Penanaman *data flow* harus menggunakan kata benda, karena di dalam *data flow* mengandung sekumpulan data. Simbol dari *Data Flow* dapat dilihat pada gambar 3.5



**Gambar 3.5:** *Data Flow*

### **3.6.2 Context Diagram**

Menurut (Kristanto, 2003), *Context Diagram* merupakan langkah pertama dalam pembuatan DFD. Pada *context diagram* dijelaskan sistem apa yang dibuat dan eksternal entity apa saja yang terlibat. Dalam *context diagram* harus ada arus data yang masuk dan arus data yang keluar.

### **3.6.3 Data Flow Diagram Level 0**

Menurut (Kristanto, 2003), DFD *level 0* adalah langkah selanjutnya setelah context diagram. Pada langkah ini, digambarkan proses-proses yang terjadi dalam Aplikasi.

### **3.6.4 Data Flow Diagram Level 1**

Menurut (Kristanto, 2003), DFD *Level 1* merupakan penjelasan dari DFD *level 0*. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD *level 0*.

### **3.6.5 Entity Relational Diagram**

Menurut (Sutanta, 2004), *Entity Relational Diagram* (ERD) adalah sebuah konsep yang mendeskripsikan hubungan antara penyimpanan dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek, disebut entiti & relasi diantar objek-objek tersebut. jadi singkatnya ERD memodelkan data seperti pada dunia nyata, jadi ada Entitas Mobil, Entitas Orang, dan sebagainya, lalu Entitas itu di relasikan satu sama lain.

Simbol-simbol ERD :

## 1. Entitas

Jenis entitas dapat berupa suatu elemen lingkungan, sumber daya atau transaksi yang field-fieldnya dipergunakan dalam aplikasi program. Simbol dari Entitas dapat dilihat pada gambar 3.6



**Gambar 3.6:** Entitas

## 2. Hubungan atau *Relasi*

Penghubung antara suatu entitas dengan entitas yang lain dan merupakan bagian yang sangat penting di dalam mendesain *database*. Ada empat (4) tipe *relasi* yang dikenal yaitu :

### a. *One-to-One Relationship*

Jenis hubungan antara tabel yang menggunakan secara bersama sebuah kolom *primary key*. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk alasan keamanan atau kecepatan akses data. Seperti *departemen* hanya mengrejakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu *departemen* saja.

### b. *One-to-Many Relationship*

Jenis hubungan antara tabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel ini. Jenis hubungan ini yang paling sering digunakan. Misalnya satu pekerjaan hanya dikerjakan oleh satu *departemen* saja. Namun satu *departemen* dapat mengerjakan beberapa macam pekerjaan sekaligus.

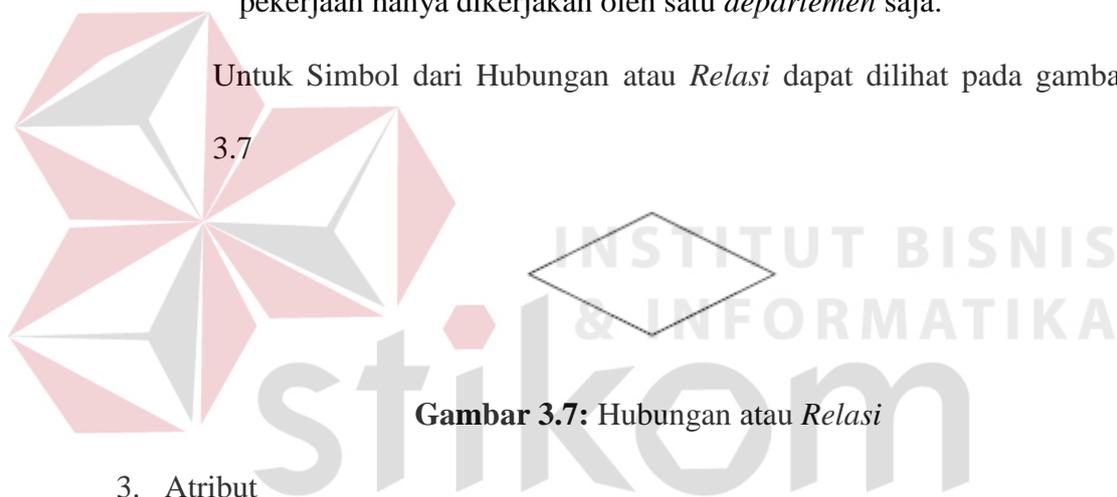
### c. *Many-to-Many Relationship*

Jenis hubungan antara tabel dimana beberapa *record* satu tabel terhubung dengan beberapa *record* pada tabel lain. Misalnya satu *departemen* mampu mengerjakan banyak pekerjaan, sedangkan satu pekerjaan dapat ditangani oleh banyak *departemen*.

d. *Many-to-One Relationship*

Jenis hubungan antara tabel dimana beberapa *record* pada satu tabel terhubung dengan satu *record* pada tabel ini. Misalnya satu *departemen* mampu mengerjakan banyak pekerjaan, namun satu pekerjaan hanya dikerjakan oleh satu *departemen* saja.

Untuk Simbol dari Hubungan atau *Relasi* dapat dilihat pada gambar



3. Atribut

Atribut adalah karakteristik dari sebuah entitas yang merupakan penjelasan-penjelasan dari entitas yang membedakan entitas satu dengan yang lain. Simbol dari Atribut dapat dilihat pada gambar 3.8



**Gambar 3.8:** Atribut

#### 4. Garis Relasi

Menunjukkan hubungan (keterkaitan) antar entitas. Simbol dari Atribut dapat dilihat pada gambar 3.9



**Gambar 3.9:** Garis Relasi

#### 5. Entitas Lemah

Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat. Simbol dari Atribut dapat dilihat pada gambar 3.10



**Gambar 3.10:** Entitas Lemah

### 3.7 Database Management System

Menurut Marlinda (2004:6), *Database Management System* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

## A. Bahasa-bahasa yang terdapat dalam DBMS

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

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut *data dictionary / directory*.

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

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

### 3. *Query*

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi.

## B. Fungsi DBMS

### 1. *Data Definition*

DBMS harus dapat mengolah *data definition* atau pendefinisian data.

### 2. *Data Manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

### 3. *Data Security dan Integrity*

DBMS dapat memeriksa *security* dan *integrity* data yang didefinisikan oleh DBA.

#### 4. *Data Recovery dan Concurrency*

- a. DBMS harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan *disk*, dan sebagainya.
- b. DBMS harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.

#### 5. *Data Dictionary*

DBMS harus menyediakan *data dictionary* atau kamus data.

### 3.8 *Tools Pemrograman*

Dalam pengembangan suatu Aplikasi, tentunya membutuhkan suatu *tool* atau alat berupa bahasa pemrograman. Salah satu *tool* dalam bahasa pemrograman yang sekarang dipakai adalah notepad++ dan XAMPP.

#### A. Definisi Notepad++

Notepad++ adalah sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi Windows. Notepad++ menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman. ([www.wikipedia.org](http://www.wikipedia.org))

Bahasa Pemrograman yang didukung oleh Notepad++ adalah :

1. ActionScript, Ada, ASP, Assembler, autoIt
2. Batch
3. C, C++, C#, Caml, Cmake, COBOL, CSS

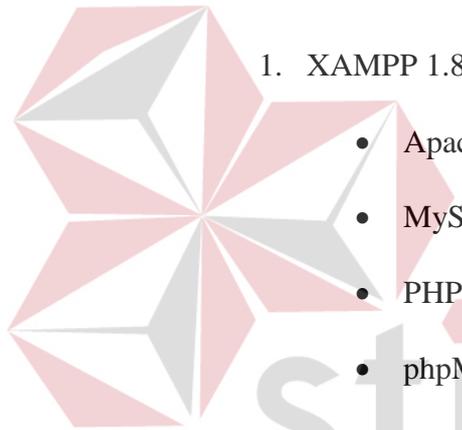
4. D, Diff
5. Flash ActionScript, Fortran
6. Gui4CLI
7. Haskell, HTML
8. Berkas INI, InnoSetup
9. Java, Javascript, JSP
10. KiXtart
11. LISP, Lua
12. Makefile, Matlab, MS-DOS
13. NSIS
14. Objective-C
15. Pascal, Perl, PHP, Postscript, PowerShell, Properties file, Python
16. R, Resource file, Ruby
17. Shell, Scheme, Smalltalk, SQL
18. TCL, TeX
19. Pascal, Perl, PHP, Postscript, PowerShell, Properties file, Python
20. XML
21. YAML

## **B. Definisi XAMPP**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan

singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web *server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya. ([www.wikipedia.org](http://www.wikipedia.org))

#### Komponen XAMPP :



##### 1. XAMPP 1.8.3 untuk windows :

- Apache 2.4.4
- MySQL 5.6.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.41 (with mod\_proxy\_ajp as connector)
- Strawberry Perl 5.16.3.1 Portable
- XAMPP Control Panel 3.2.1 (from hackattack142)

##### 2. XAMPP 1.8.3 untuk linux :

- Apache 2.4.4
- MySQL 5.6.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- OpenSSL 1.0.1e

3. XAMPP untuk Solaris
4. XAMPP untuk Mac OS X

### **3.9 Interaksi Manusia dan Komputer**

Suatu Aplikasi yang baik tentunya harus mempertimbangkan interaksi antara pengguna dan program yang dibuat. Di sinilah pentingnya penerapan ilmu Interaksi Manusia dan Komputer.

Menurut Santoso (2004:4), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya.

Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang terdapat dalam sistem.

Pada implementasinya, interaksi manusia dan komputer dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan.