

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Poliklinik Desa Sukaasih**

Sebelum menjadi PKD seperti sekarang ini, dulu masyarakat mengenalnya dengan istilah Polindes. PKD adalah suatu upaya kesehatan yang berfungsi daya masyarakat (UKBM) yang dibentuk oleh, dan untuk masyarakat setempat atas dasar musyawarah desa/kelurahan dan didukung oleh tenaga kesehatan yang professional. Bentuk UKBM artinya PKD dibentuk oleh dan untuk masyarakat yang mempunyai peran serta yang tinggi.

Tujuan PKD yaitu

1. Mendorong kemandirian masyarakat untuk hidup sehat
2. Meningkatkan jangkauan dan mutu pelayanan kesehatan dasar masyarakat
3. Meningkatkan penyuluhan dan konseling
4. Memberikan pelayanan kesehatan lainnya sesuai dengan kewenangan bidan

Unsur PKD yaitu:

1. Adanya partisipasi aktif masyarakat
2. Adanya pengelola (bidan/perawat + masyarakat)
3. Tersedianya sarana, prasarana, dan obat-obatan
4. Keterampilan tenaga kesehatan/kader yang harus dimiliki
5. Peralatan/perlengkapan lainnya (alat-alat penyuluhan)

#### **2.2.1. World Wide Web**

*World Wide Web* atau lebih dikenal dengan nama *web* menurut Berners-Lee (2004), adalah sebuah layanan yang didapat oleh pemakai komputer apabila komputernya tersambung dengan internet. Dengan *web*, pengguna komputer

diseluruh penjuru dunia dapat saling berinteraksi dengan pengguna internet lainnya tanpa harus beranjak dari tempat di mana internet tersebut dapat di akses. Dengan mengakses internet, para pengguna jasa layanan internet dapat dengan mudah memperoleh informasi yang diinginkan, bahkan dapat menyimpan program, atau gambar yang ditampilkan dari media internet.

Pada pengembangannya, *web* merupakan suatu ruangan yang dapat menampung informasi dalam jaringan internet pada sebuah *browser*, dengan menambahkan kemampuan untuk mengolah kode-kode tertentu yang secara umum dinamakan tag-tag dan kemampuan untuk dapat meloncat (*link*) dari halaman satu ke halaman lain. Kemudian kemampuan dari *browser* tersebut ditingkatkan sampai dengan pengelolaan sebuah gambar, suara, animasi, bahkan kemampuan dalam pengelolaan sebuah *database* dari sebuah aplikasi berbasis *web* dengan bahasa pemrograman atau *script* yang dapat di jalankan oeh internet/intranet, seperti ASP, PHP, dan Jscript.

Sedangkan pengertian *Web* Menurut Hidayat (2010), *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

*Web* bekerja didasarkan pada jaringan internet, internet adalah *network of interconnected networks*. Artinya jaringan yang menghubungkan jaringan – jaringan lainnya. Ketika berselancar di internet dan melihat berbagai macam bentuk

dan jenis *web* disitulah anda dapat melihatan begitu hebatnya *web*, sebab *web* dapat menampung berbagai jenis data seperti text, grafis, animasi, foto, suara dan video. Halaman *web* ditulis dengan bahasa pemograman yang disebut *Hypertext Markup Language* atau *HTML*, *file HTML* berisi markup tags, *markup tags* inilah yang memberi tahu *browser* bagaimana halaman *web* ditampilkan, terdapat dua *extension file HTML* yang dapat digunakan yaitu *.htm* dan *.html*.

### 2.2.2. Sejarah Web

Sejarah *web* menurut Berners-Lee (2004), dimulai pada bulan Maret 1989 ketika Tim Berner-Lee yang bekerja di *Laboratorium Fisika Partikel Eropa* atau yang dikenal dengan nama CERN (*Consel European pour Ia Rachere Nuclaire*) yang berada di Genewa, Swiss, mengajukan protokol (suatu tata cara untuk berkomunikasi) *system* distribusi informasi internet yang digunakan untuk berbagai informasi diantara fisikawan. Protokol inilah yang selanjutnya dikenal sebagai protokol *Word Wide Web Consortium* (W3C). W3C ini adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan *web*.

Sejarah *web* menurut Kadir (2009), 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 *Appllet* dan 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 pengembangan yang terjadi. Kelemahan ini dapat 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.

### **2.2.3. Aplikasi Web**

Aplikasi *web* menurut Hidayat (2010), Seiring dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokan jenis *web*, lebih diarahkan berdasarkan kepada fungsi, sifat atau style dan bahasa pemrograman yang digunakan. Adapun aplikasi *web* tersebut adalah sebagai berikut :

#### 1) *Web Statis*

Sebuah *website* statis adalah *website* yang kontennya sangat jarang berubah. Bahasa pemrograman yang digunakan adalah HTML dan belum

memanfaatkan *database*. salah satu bentuk *website* yang isinya tidak dimaksudkan untuk di-*update* secara berkala, dan biasanya di *maintain* secara manual oleh beberapa orang yang menggunakan *software editor*. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi *web* dinamis.

## 2) *Web* Dinamis

Sebuah *website* dinamis merupakan sebuah *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan *database MySQL* atau MS SQL. Selain itu, informasi didalamnya berubah atau *website* ini bisa berhubungan dengan *user* dengan berbagai maca cara atau metode (*HTTP cookies* atau *Variabel Database*, sejarah kunjungan, *variabel* sesi dan lain-lain) bisa juga dengan cara interaksi langsung menggunakan *form* dan pergerakan *mouse*.

### **2.2.4. *Web Browser***

*Web browser* menurut Berners-Lee (2004), adalah program yang digunakan untuk menampilkan informasi dari suatu halaman *web* yang tersimpan dalam komputer. Dengan menerapkan *Graptic User Interface* (GUI), maka para pengguna internet dengan mudah mengakses informasi-informasi yang terdapat di

dalam internet. *Browser* yang terkenal saat ini adalah *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*, *Opera*, dan *Safari*.

*Web Browser* menurut Anhar (2010), merupakan *software* yang diinstall di mesin client, berfungsi untuk menerjemahkan *tag HTML* menjadi halaman *web*. *Browser* yang sering digunakan adalah *Internet Explorer*, *Mozilla*, *Opera*, *Netscape*, dan lainnya.

## **2.2. Sistem**

### **2.3.1. Definisi**

Sistem menurut Indrajit (2001), mengemukakan bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya.

Sedangkan Sistem menurut Jugiyanto (2005), mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi.

Sistem menurut Sidharta (1995), adalah himpunan dari bagian-bagian yang saling berhubungan yang secara bersama mencapai tujuan-tujuan yang sama.

Dengan demikian pada penelitian ini, definisi Sistem yang digunakan merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dari sistem tersebut. maksud dari suatu sistem adalah untuk mencapai suatu tujuan dan sasaran dalam ruang lingkup yang sempit.

### 2.3.2. Karakteristik Sistem

Karakteristik sistem menurut Jogiyanto (2005), mengemukakan sistem mempunyai karakteristik atau sifat-sifat tertentu, yakni :

#### 1). Komponen

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu mempengaruhi proses sistem secara keseluruhan.

#### 2). Batasan sistem

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

#### 3). Lingkungan Luar Sistem

Lingkungan luar (*evinronment*) dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan berupa energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

#### 4). Penghubung Sistem

Penghubung (*interfance*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sesistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

#### 5). Masukan Sistem

Masukan (*input*) sistem adalah energi yang masukan kedalam sistem. masukan dapat berupa masukan perawatan (*maintenance input*), dan masukan sinyal (*signal input*). Maintenance input adalah energi yang dimasukan supaya tersebut dapat beroperasi. signal input adalah energi yang diproses untuk didapatkan keluaran. sebagai contoh didalam komputernya dan data adalah signal input untuk diolah menjadi informasi.

#### 6). Keluaran Sistem

Keluaran (*output*) sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

#### 7). Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

#### 8). Sasaran Sistem

Sebuah sistem sudah tentu mempunyai sasaran ataupun tujuan. Dengan adanya sasaran sistem, maka kita dapat menentukan masukan yang dibutuhkan sistem dan keluaran apa yang akan dihasilkan sistem tersebut dapat dikatakan berhasil apabila mencapai/mengenai sasaran atau pun tujuan.

### **2.3. Sistem Informasi**

#### **Definisi**

Sistem Informasi menurut Arbie (2010), sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, membantu dan mendukung kegiatan operasi, bersifat manajerial dari suatu organisasi dan membantu mempermudah penyediaan laporan yang diperlukan.

Sistem Informasi menurut O'Brien (2009), suatu kombinasi terartur apapun dari *people* (orang), *hardware* (perangkat keras), *software* (piranti lunak), *computer networks and data communications* (jaringan komunikasi), dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi.

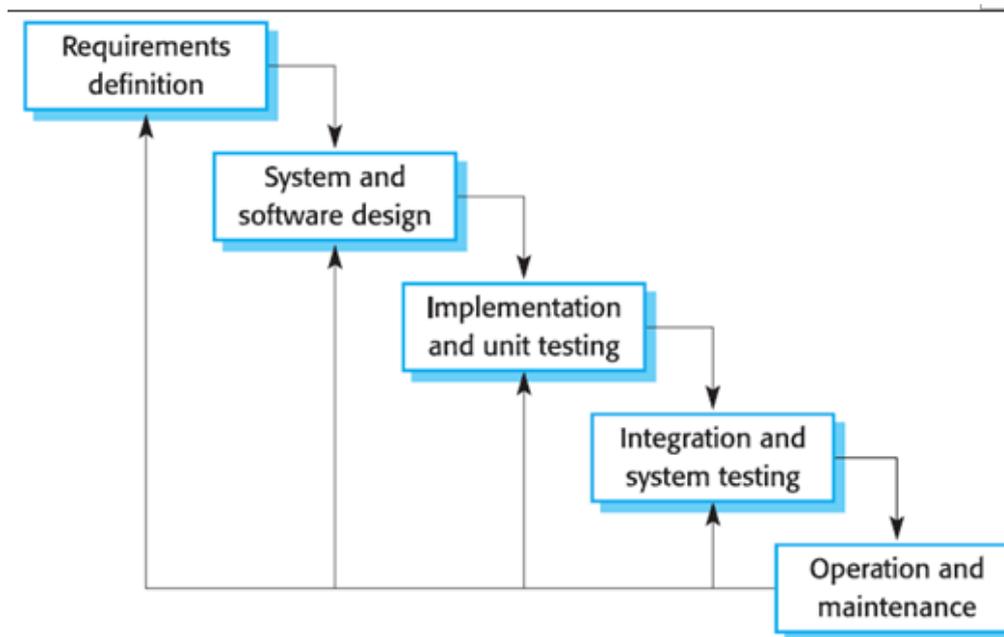
Sistem Informasi menurut Davis (2000), suatu sistem yang menerima masukan data dan instruksi, mengolah data tersebut sesuai dengan instruksi dan mengeluarkan hasilnya.

Dapat disimpulkan Bahwa Sistem Informasi Manajemen adalah suatu sistem yang dirancang untuk menyediakan informasi guna mendukung pengambilan keputusan pada kegiatan manajemen dalam suatu organisasi.

#### **2.4. Metode *Waterfall***

Metode perancangan sistem yang digunakan adalah dengan menggunakan metode Model *Waterfall* yang merupakan metode yang berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak berbasis *web*.

Metode analisis yang digunakan untuk perancangan website yaitu model *waterfall / linear sequential model (life cycle)*. Model ini menuntut suatu pendekatan sistematis – sekuensial dari proses pengembangan website, dimulai pada tingkatan sistem dan dilanjutkan dengan analisis, perancangan, pengkodean, uji coba dan pemeliharaan. Model *waterfall* [Roger S. Pressman, 2010] tersusun atas aktivitas-aktivitas berikut :



Gambar 3.1 Linear Waterfall

Model Waterfall mencakup komponen-komponen berikut :

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, rekayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antar muka (interface) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak di dokumentasikan dan dilihat dengan pelanggan. Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dsb. Tahap ini sering disebut dengan Project Definition.

b. Desain

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer

harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada user. Proses software design untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "blueprint" software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

#### c. Generasi Kode

Desain harus diterjemahkan dalam bentuk mesin yang bisa di baca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.

#### d. Pengujian

Proses Pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah diuji. Pengujian eksternal fungsional untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

#### E. Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja.

## 2.5. Basis Data

Basis data menurut Kristanto (2000), basis data dapat didefinisikan dalam sejumlah sudut pandang tertentu, diantaranya :

- 1) Basis Data dapat diartikan kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- 2) Basis Data dapat diartikan kumpulan *file -file* yang saling berhubungan dan relasi tersebut biasanya ditunjukkan dengan kunci dari tiap *file* yang ada yang disimpan dalam media penyimpanan elektronik.
- 3) Basis Data dapat diartikan sekumpulan data yang terdiri atas satu atau lebih tabel yang saling berhubungan dan *user* mempunyai wewenang untuk mengakses data tersebut, baik menambah, mengubah, atau menghapus data yang ada dalam tabel tersebut. Sedangkan tabel merupakan media untuk menyimpan data yang telah diolah dan mempunyai sesuatu tema tertentu.

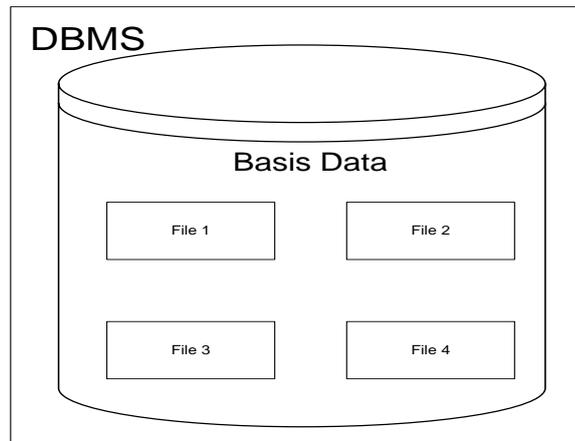
Menurut Kristanto (2000), basis data memiliki beberapa istilah, adapun istilah-istilah dalam basis data diantaranya :

- 1) *Entity* merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain, misalnya orang, tempat, kejadian, atau konsep yang informasinya direkam.
- 2) Atribut merupakan sesuatu hal yang mendeskripsikan karakteristik dari entitas.

- 3) *Data value* adalah data aktual atau informasi yang disimpan pada atribut.
- 4) *Record* yaitu kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap.
- 5) *File* merupakan kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda data *value*-nya.

*Data Base Management Sistem* merupakan kumpulan *file* yang saling berkaitan bersama dengan program untuk mengelolanya. *Database* merupakan kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam satu paket program yang berfungsi untuk membaca data, mengisi data, menghapus data serta melaporkan data dalam *database*.

Basis Data hanyalah sebuah objek yang pasif dan tidak akan berguna jika tidak ada pengelolanya, yang menjadi pengelola secara langsung adalah program atau aplikasi (*software*). Gabungan dari keduanya menghasilkan sistem. Secara umum sebuah sistem basis data merupakan sistem yang terdiri dari kumpulan *file* yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi *file - file* tersebut.



Gambar 2.1. Sistem Basis Data (Kristanto, 2000)

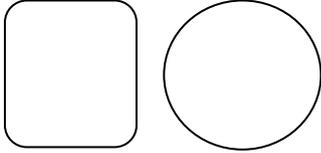
Basis data (*Database*) menurut Connolly (2005), merupakan kumpulan berbagai data logika yang terkait, dan merupakan deskripsi dari data-datanya, yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

*Database* merupakan tempat penyimpanan data yang dapat berukuran besar dan dapat digunakan secara bersama-sama oleh banyak departemen dan *user*. Sebagai pengganti *file -file* yang tidak terhubung dan memiliki banyak duplikasi, data-data pada *database* dibuat terintegrasi dengan jumlah duplikasi yang lebih minimum. *Database* tidak hanya berisi data operasional organisasi, namun juga berisi deskripsi dari data yang disebut sebagai *system catalog* atau *data dictionary* atau *meta data* (data tentang data).

## 2.6. Data Flow Diagram (DFD)

DFD menurut McLeod (1995), Diagram Arus Data (*Data Flow Diagram*) atau DFD adalah Suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk simbol-simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan. Walau nama diagram ini menekankan pada data, situasinya justru sebaliknya penekanannya ada pada proses.

Tabel 2.1. Simbol-simbol *Data Flow Diagram* (DFD)

Simbol	Fungsi
Entitas 	Digunakan untuk menggambarkan suatu entitas eksternal, yang dapat mengirim data atau menerima data dari sistem.
Aliran Data 	Digunakan untuk menunjukkan aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.
Proses 	Digunakan untuk menunjukkan kegiatan atau kerja yang dilakukan orang, mesin atau komputer. Proses harus diberi nama yang jelas untuk memudahkan menangani proses.
Simpanan Data 	Digunakan untuk menunjukkan simpanan dari data.

DFD Menurut Whitten (2004), “*Data Flow Diagram is a process model used to depict the flow of data through a system and the work or processing performed by the system*” , yang jika diartikan dalam bahasa Indonesia adalah DFD merupakan model proses yang digunakan untuk menggambarkan aliran data yang melalui sebuah sistem dan proses-proses yang dilakukan oleh sistem tersebut.

Terdapat 3 simbol yang dihubungkan oleh satu koneksi pada DFD, yaitu :

- Kotak persegi dengan ujungnya yang bulat, menggambarkan proses atau sebuah pekerjaan yang telah diselesaikan.
- Kotak, menggambarkan entitas luar atau *external agents*. Simbol ini dapat digambarkan sebagai terminator yang menjelaskan batasan dari sebuah sistem.
- Kotak tanpa garis penutup di ujungnya (*the open-ended boxes*) menggambarkan penyimpanan data, berupa *database* atau *file* .
- Garis anak panah menggambarkan *data flow* atau aliran atau *input* dan *output*. Garis anak panah ini terhubung dari atau menuju proses.

### **2.7. Entity Relationship Diagram (ER-D)**

*Entitas Relationship Diagram* (ERD) menurut Connolly (2002), adalah pendekatan *top-down* untuk mendesain basis data yang dimulai dengan mengidentifikasi data yang penting, yang disebut sebagai entitas dan hubungan antara data harus digambarkan.

Batasan utama dalam relasi disebut *multiplicity*. *Multiplicity* adalah jumlah kejadian yang mungkin muncul dari entitas satu ke entitas lainnya yang mempunyai hubungan khusus.

Hubungan yang paling umum adalah berpasangan seperti :

1. *one-to-one*(1:1)

Sebuah entitas di A hanya dapat diasosiasikan dengan paling banyak satu entitas di B.

2. *One-to-many* (1:\*)

Sebuah entitas di A dapat diasosiasikan dengan satu atau lebih entitas di B, namun entitas di B hanya dapat diasosiasikan dengan paling banyak satu entitas di A.

3. *many-to-many* (\*:\*)

Sebuah entitas di A dapat diasosiasikan dengan nol atau lebih entitas di B dan sebuah entitas di B dapat diasosiasikan dengan nol atau lebih entitas di A.

ERD menurut Yakub (2008), merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi.

Adapun simbol-simbol ERD sebagai berikut :

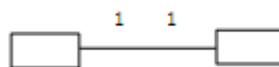
- I. Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
- II. Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu, satu ke banyak, dan banyak ke banyak.
- III. Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
- IV. Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasinya.

Dalam ERD juga dikenal dengan derajat relasi/ kardinalitas. Menurut Yakub (2008), kardinalitas yaitu kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas. Dapat berupa satu ke satu (*one-to-one*), satu ke banyak (*one-to-many*), dan banyak ke banyak (*many-to-many*).

Penjelasan-penjelasan ketiga kardinalitas tersebut, akan dijelaskan dibawah ini :

i. *One-to-one*

Hubungan antara lain *file* pertama dan *file* kedua adalah satu berbanding satu.



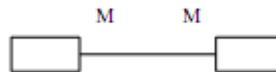
Gambar 2.3 Hubungan *one-to-one*

ii. *One-to-many*

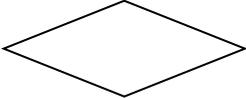
Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu.

Gambar 2.4 *One-to-many*iii. *Many-to-many*

Hubungan antara *file* pertama dan kedua adalah banyak berbanding banyak.

Gambar 2.5 *Many-to-many*

Tabel 2.2. Notasi Simbolik ER-D

Simbol	Fungsi
Persegi Panjang 	Menyatakan himpunan entitas
Lingkar atau Elips 	Menyatakan atribut yang berfungsi sebagai <i>key</i> diberi garis bawah.
Simbol	Fungsi
Belah Ketupat 	Menyatakan himpunan relasi

Simbol	Fungsi
Garis _____	Sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya
Kardinalitas Relasi 1 dan 1 ( Satu ke Satu ) 1 dan N ( Satu ke Banyak ) N dan N ( Banyak ke Banyak )	Menyatakan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain

## 2.8. *Hypert Text Markup Language (HTML)*

*Hypert Text Markup Language* menurut Turban (2006), adalah bahasa pemrograman yang digunakan di *web*, yang memformat dokumen dan memadukan link hiperteks dinamis ke dokumen-dokumen lainnya yang disimpan didalam komputer. HTML berasal dari *Standard Generalized Markup Language (SGML)* yang lebih rumit, bahasa berbasis teks untuk mendeskripsikan isi dan struktur dari dokument digital. HTML adalah subset yang lebih sederhana dari SGML dan mencakup tabel, *applet*, aliran teks, dalam gambar, *superskrip*, *subskrip*.

Menurut Nugroho (2004), HTML adalah *Hyper Text Markup Language* yang merupakan sebuah bahasa *scripting* berguna untuk menuliskan halaman *web*. Pada *web*, HTML dijadikan sebagai Bahasa *Script* dasar yang berjalan bersama berbagai bahasa *scripting* pemrograman lainnya. Semua *tag-tag* HTML bersifat dinamis artinya kode HTML tidak dapat dijadikan sebagai *file executable* program. Hal tersebut disebabkan, HTML hanyalah sebuah bahasa *scripting* yang dapat berjalan apabila dijalankan di dalam browser (pengakses *web*).

*Browser-browser* yang mendukung HTML antara lain *Internet Explorer*, *Netscape Navigator*, *Opera*, *Mozilla*, dan lain-lain.

## 2.9. MySQL

*MySQL* menurut Sutarman (2003), adalah program pembuat *database* yang bersifat *open source*. Sebagai sebuah program penghasil *database*, *MySQL* tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (interface). *MySQL* dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti PHP dan sebagainya.

*MySQL* menurut Prasetyo (2004), merupakan salah satu *database server* yang berkembang di lingkungan *open source* dan didistribusikan secara *free* (gratis) dibawah lisensi GPL. *MySQL* merupakan RDBMS (*Relational Database Management System*) *server*. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya.

*MySQL* menurut Kadir (2008), merupakan *software* yang tergolong sebagai DBMS (*database Management System*) yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (code yang dipakai untuk membuat *MySQL* ). Selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi. Dan bisa diperoleh secara gratis dengan *download* di internet.

*MySQL* awalnya dibuat oleh perusahaan konsultan bernama TeX yang berlokasi di Swedia. Saat ini pengembangan *MySQL* berada dibawah naungan

*MySQL* AB. Sebagai *software* DBMS, *MySQL* memiliki sejumlah fitur seperti yang akan dijelaskan dibawah ini.

a. *Mutliplatform*

*MySQL* tersedia pada beberapa platform (*windows, linux, unix, dan lain-lain*)

b. Andal, cepat dan mudah digunakan.

*MySQL* tergolong sebagai *database server* (*server* yang melayani permintaan terhadap *database*) yang andal, dapat menangani *database database* yang besar dengan kecepatan tinggi. Mendukung banyak sekali fungsi untuk mengakses *database* dan sekaligus mudah untuk digunakan.

c. Jaminan keamanan akses

*MySQL* mendukung pengamanan *database* dengan berbagai criteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misal gaji pegawai), sedangkan *user* lain tidak boleh sesuai dengan hak aksesnya.

d. Dukungan SQL

Seperti tersirat namanya, *SQL* mendukung perintah *SQL* (*Structured Query Language*). Sebagaimana diketahui *SQL* merupakan bahasa standar dalam pengaksesan *database* rasional. Pengetahuan akan *SQL* akan memudahkan siapapun untuk menggunakan *MySQL* .

## **2.10. Personal Home Page (PHP)**

PHP menurut Rafizah (2006), adalah sebuah bahasa pemrograman yang berjalan dalam sebuah *web server* . PHP diciptakan oleh seorang *programmer*

*Unix dan Perl* yang bernama Rasmus Lerdorf pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan *script* dalam *website* pribadinya dengan tujuan untuk memonitor siapa saja yang pernah berkunjung di *website*-nya. Namun kemudian dikembangkan lagi sehingga menjadi sebuah *Personal Home Page*.

Menurut Anhar (2010), yaitu bahasa pemrograman *web server -side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML scripting)* PHP adalah *script* yang digunakan untuk membuat halaman *website* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang di terima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan.

PHP (akronim dari *PHP Hypertext Preprocessor*) yang merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

Ketika menggunakan PHP sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut :

- 1) Membaca permintaan dari *client/browser*.
- 2) Mencari halaman/page di *server*.

- 3) Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*.

### **2.11. XAMPP**

XAMPP menurut Nugroho (2008), XAMPP adalah suatu bundel *web server* yang populer digunakan untuk coba-coba di Windows karena kemudahan instalasinya. Bundel program open source tersebut berisi antara lain *server web* Apache, interpreter PHP, dan basis data *MySQL*. Setelah menginstall XAMPP, 17 kita bisa memulai pemrograman PHP di komputer sendiri maupun mencoba menginstall aplikasi-aplikasi *web*.

### **2.12. Metode Pengujian *Black-Box***

*Black box* Menurut Myers (1979), Proses menjalankan program dengan maksud menemukan kesalahan. Sedangkan menurut IEEE (1990), Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada *output* yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi. Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu.

*Black-Box testing* menurut Perry (2006), adalah metode pengujian perangkat lunak yang melakukan pengetesan fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (pengujian *white-box*). Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, apa yang seharusnya aplikasi lakukan. Menggunakan deskripsi

*eksternal* perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang *valid* dan tidak *valid* dan menentukan *output* yang benar.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan masukan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit *testing* juga.

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan pendekatan yang melengkapinya untuk menemukan kesalahan lainnya, selain menggunakan *metode whitebox*.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- 1) Fungsi-fungsi yang salah atau hilang
- 2) Kesalahan *interface*
- 3) Kesalahan dalam struktur data atau akses *database* eksternal
- 4) Kesalahan performa
- 5) Kesalahan inisialisasi dan terminasi

### 2.13. Integrasi

Integrasi menurut Markus (2001), Proses integrasi dan fungsinya pada perusahaan sangat berkaitan dengan batasan cangkupannya yang hanya terjadi pada beberapa unit atau proses pada satu organisasi. Dengan seiringnya perkembangan teknologi dan ketatnya persaingan antar perusahaan, saat ini perusahaan sudah berusaha untuk beralih ke integrasi dengan skala yang lebih besar.

Sedangkan Integrasi menurut Porter (2001), integrasi mengatur *value chain* dalam seluruh koresponden perusahaan ke tahap keempat dalam evolusi teknologi informasi dalam bisnis, dengan menggunakan integrasi proses pengembangan produk akan menjadi pada tahap yang lebih lanjut. Integrasi ditempatkan pada posisi penting didalam beberapa bagian, termasuk *management, strategy, organization theory, production/operation management, dan information systems*.