BAB II
TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah Instansi

Pada mulanya yang pertama berdiri adalah Madrasah Tsanawiyah (MTs) Al-Mukhtariyah pada tahun 1975. MTs ini berdiri sendiri tanpa ada badan hukum. Setelah ada peraturan pemerintah bahwa instansi pendidikan harus memiliki badan hukum maka didirikanlah Yayasan Al-Mukhtariyah sebagai badan hukumnya. Yayasan ini dibentuk pada tanggal 16 November 1987 atas hasil kerjasama para guru dengan Badan Pembantu Penyelenggaraan Pendidikan (BPPP).

2.1.2 Struktur Organisasi

I. PEMBINA
- Ketua Pembina : DJUHAYA
- Pembina I : AYUB SUPRIATNA, S.Ag.
- Pembina II : Drs. H. ENAN HASAN SYADILI, MBA
- Pembina III : Dr. ACENG SOLAHUDIN AHMAD

II. PENGURUS
- Ketua : Drs. H.A. HANAFIAH, MM
- Wakil Ketua : A.Z. MUTTAQIN, SPd.I.
- Sekretaris : JUHANA, SPd.I.
- Bendahara : ADE SITI JUHRO, S.Ag., MM

III. PENGAWAS
- Pengawas I : ZAINAL ARIFIN
- Pengawas II : KH. E. MA'SUM

2.1.3 Logo Instansi



Gambar 2.1 Logo instansi

2.1.4 Visi dan Misi

2.1.4.1 Visi

Menciptakan Sumberdaya manusia yang tangguh, Cerdas, Kreatif dan Berakhlakul Karimah.

2.1.4.2 Misi

- Menyelenggarakan pendidikan formal dan non formal.
- Meningkatkan kualitas dan kuantitas sarana dan prasarana pendidikan.
- Meningkatkan kualitas layanan pendidikan.
- Meningkatkan Keimanan dan Ketakwaan kepada Allah SWT melalui kegiatan rutin yang bernuansa keagamaan.

2.2 Landasan Teori

2.2.1 Konsep Dasar Sistem

Konsep dasar sistem ini meliputi pengertian dasar dari sistem, bentuk umum sistem, dan karakteristik sistem.

a. Pengertian Sistem

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem menurut JOG[2], yaitu :

- Pendekatan yang menekankan pada prosedurnya.
  Suatu sistem adalah suatu jaringan dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

- Pendekatan sistem yang menekankan pada komponennya.
  Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

b. Bentuk Umum Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses, dan keluaran (*output*). Dalam bentuk umum sistem ini biasa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan rencana yang telah ditentukan sebelumnya,model sistem sederhana dapat dilihat pada gambar 2.1.



Gambar 2.2 Sistem Sederhana [2]

c. Karakteristik Sistem

Adapun karakteristik suatu sistem, yaitu :

a. Komponen-komponen *(Components)*
   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi,yang artinya saling bekerjasama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen system dapat berupa suatu subsistem atau bagian-bagian dari sistem.
b. Batas Sistem*(Boundary)*
   Batas sistem *(Boundary)* merupakan daerah yang membatasi antara satu sistem yang lain atau dengan linkungan luarnya. Batas sistem ini memungkinkan suatu sistemdipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem*(Environment)*

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Perhubungan (*Interface)*

Perhubungan merupakan media penghubung antara satu sub sistem dengansubsistem lain.

e. Masukan *(Input)*

Masukan adalah energi yang dinasukkan kedalam sistem.

f. Keluaran *(Output)*

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadikeluaran yang berguna.

g. Pengolahan

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

h. Sasaran *(Object)*

Suatu sistem pasti mempunyai tujuan atau sasaran-sasaran dari sistem sangat membutuhkan sekali masukan yang dibutuhkan sistemdan keluaran yang akan dihasilkan sistem. [3]

2.2.2 Sistem Informasi Berbasis Web

Sistem Informasi web terdiri dari tiga kata yaitu Sistem, Informasi dan web. Sistem sendiri berarti gabungan dari beberapa sub sistem yang bertujuan untuk mencapai satu tujuan. Informasi berarti sesuatu yang mudah dipahami oleh si penerima dan web adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Sistem Informasi memiliki makna sistem yang bertujuan menampilkan informasi. [4]

2.2.3 Sistem Komputer untuk Web

Sistem komputer biasanya terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

a. Perangkat Keras

Terdiri dari beberapa komponen.

1. *Central processing unit (CPU)*

   CPU menjalankan program komputer dan mengendalikan operasi seluruh komponen.Biasanya digunakan CPU untuk komputer pribadi (*PC/personal computer*), atau *work station* pada sebuah jaringan komputer.

2. *Memory*

   Memory Utama: adalah bagian paling esensi pada komputer seluruh data dan program berada pada memori utama untuk akses yang lebih cepat. Dibutuhkan setidaknya memori berkapasitas 64 MB untuk SIG berbasis PC.

   Memory Tambahan: digunakan data berukuran besar baik permanen maupun semi-permanen, dengan akses lebih rendah dibanding memori utama. Dikenal juga sebagai media penyimpanan data, seperti *harddisk, disket* (*floppy disk*), pita magnetis atau cakram padat optis (CD-ROM).Untuk *harddisk* dibutuhkan setidaknya yang berkapasitas 1 GB.

b. Perangkat Lunak

   Terdiri atas sistem operasi, compiler dan program aplikasi:

Sistem Operasi (*Operating System*)

Mengendalikan seluruh operasi program, juga menghubungkan perangkat keras dengan program aplikasi. Untuk PC : WINDOWS adalah sistem operasi yang banyak digunakan. Untuk *Workstation*: UNIX dan VMS adalah OS yang dominan.

Compiler

Menerjemahkan program yang ditulis dalam bahasa komputer pada kode mesin sehingga CPU mampu menjalankan program yang harus dieksekusi. Bahasa kompiler yang biasa digunakan adalah PHP, CMS, APACHE, MYSQL, dll.

Program Aplikasi

Program aplikasi merupakan perangkat lunak siap pakai yang nantinya akan digunakan untuk membantu melaksanakan pekerjaan penggunanya.[4]

2.2.4 Konsep Basis Data (*Database)*

Konsep basis data merupakan hal yang sangat penting. Kesulitan utama dalam merancang basis data adalah bagaimana merancang sehingga *database* dapat memuaskan keperluan saat ini dan masa mendatang. Konsep basis data langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Dalam buku konsep dan perancangan database mendefinisikan *database* sebagai berikut :

Menurut KRIS [3] :

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

a. Himpunan kelompok data (arsip) yang saling berhubungan yangdiorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembalidengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersamasedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalammedia penyimpanan elektronik.

2.2.5 Pengertian Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer pribadi saling berhubungan antara satu dengan yang lainnya menggunakan *protokol* komunikasi sehingga dapat saling berbagi informasi, program penggunaan bersama perangkat keras seperti *printer*, *harddisk*. Selain itu jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri dari lebih dari satu komputer yang

saling berhubungan. Dalam sebuah jaringan komputer biasanya terhubung sejumlah komputer ke sebuah/ beberapa *serve*r. *Server* adalah komputer yang difungsikan sebagai "pelayanan" pengiriman data dan penerimaan data diantara komputer-komputer yang tersambung. Tujuan dibangunnya suatu jaringan komputer adalah membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim (*transmisi*) menuju ke sisi penerima (*receiver*) melalui media komunikasi.

TCP/IP (*Trasmission Control Protocol/Internet Protocol*) merupakan protokol standard internet yang digunakan untuk melakukan koneksi ke internet protokol. TCP/IP memiliki beberapa subyek *protocol* yang berbeda yang beroperasi pada lapisan yang berbeda dan mempunyai tugas masing-masing. Berkat adanya protokol ini setiap komputer dapat berhubungan secara fleksibel dengan *host-host* yang terkoneksi.

Jenis - jenis Jaringan Komputer

Jenis- jenis jaringan ada dua, yaitu:

a. Model *Peer To Peer*

Menurut model ini, setiap host dapat menawarkan layanan ke *peer* lain dan juga mengambil layanan dari *peer* lain. Model ini cocok untuk jaringan kecil.

b. Model *Client/Server*

Model ini memisahkan secara jelas, mana yang dapat memberikan layananjaringan (*server*) dan mana yang hanya memberikan layanan (*client*).Beberapa komputer *diset-up* sebagai *server* yang memberikan sumberdaya (*resource*) dari jaringan : printer, modem, dan saluran lainnya kepada komputer lain yang dikoneksi kejaringan yang berfungsi sebagai *client*. [4]

2.2.6 Web

2.2.6.1 Sejarah Lahirnya Web

Sejarah Web dimulai pada bulan Maret 1989 ketika Tim Berner-Lee yangbekerja di Laboratorium Fisika Partikel Eropa atau yang dikenal

dengan nama *Consei European pour la Recherce Nuclaire (CERN)* yang berada di Genewa, Swiss, mengajukan protokol atau suatu tatacara untuk berkomunikasi systemdistribusi informasi internet yang digunakan untuk berbagai informasi diantara para fisikawan. Protokol inilah yang yang selanjutnya dikenal sebagai *protokol World Wide Web (WWW)* dan dikembangkan oleh *World Wide Web Consortium (W3C)*.Dimana W3C adalah konsorsium dari sejumlah organisasi yang berkepentingan dalampengembangan berbagai standar yang berkaitan denganWeb.W3C meletakan gabungan spesifikasi dalam standar web, berikut adalah hasil dari W3C:

a. Standar web yang paling mendasar adalah HTML, *Cascade Style Sheet* (CSS), dan *Extended Markup Language* (XML)
b. Standar HTML yang terakhir adalah *Extended Hypertext markupLanguage* 1.0(XHTML 1.0) [4].

2.2.6.2 Aplikasi Web

Aplikasi web pada awalnya dibangun hanya dengan menggunakan bahasa yang disebut dengan HTML dan protokol yang digunakan yang dinamakan *HyperText Transfer Protocol* (HTTP). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML, antara lain yaitu *PHP Hypertext Preprocessor* (PHP) dan *Active Server Pages* (ASP), sedangkan contoh yang berupa objek antara lain adalah *applet* (Java). Aplikasi Web sendiri dapat dibagi menjadi dua, yaitu:

a. Web Statis
b. Web Dinamis

Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi, karena suatu web dapat cepat sekali populer dikalangan pengguna internet apabila terdapat kemudahan yang disediakan untuk pengguna web dalam melakukan penelusuran, penjelajahan dan pencarian

informasi (*surfing*) dan juga suatu web akan dikenal dengan cepat apabila informasi yang disajikan selalu up to date dan atau lengkap. Kelemahan ini dapat diatasi dengan model aplikasi Web Dinamis yang tidak memerlukan pemeliharaan program secara terus menerus untuk mengikuti perubahan yang terjadi, salah satu bahasa yang sering digunakan untuk aplikasi web dinamis adalah PHP, ASP dan banyak lagi. [4]

2.2.6.3 Aplikasi Pendukung Perancangan Web

Adapun beberapa program yang dianggap mampu membantu dalam proses design sebuah web adalah :

a. Macromedia dreamweaver V.8 : Macromedia Dreamweaver V.8 merupakan perangkat lunak yang dikenal sebagai *web authoring tool*, yaitu perangkat lunak yang digunakan untuk desain tampilan halaman web. Macromedia Dreamweaver V.8 mempunyai kemampuan untuk *editing* kode pada aplikasi web dengan bahasa pemrograman yang berbeda-beda, seperti PHP, ASP, *Java Server Pages* (JSP) dan lainnya.
b. Wampserver V.2.0 : adalah sebuah aplikasi yang di gunakan untung meng-*compile script* PHP dari Macromedia dreamweaver ke *browser* sehingga tampil menjadi sebuah web secara offline, wamp server juga di bekali dengan beberapa aplikasi seperti PHPmyadmin, Apache.
c. Mozilla firefox V.7.0.1: sebuah aplikasi *browser* yang digunakan untuk menampilkan hasil *compiler* dari wampserver sehingga web yang kita buat sudah seperti web yang sudah ter-*upload* ke *file hosting*.

2.2.7 HTML

HTML atau Hyper Text Markup Language, adalah bahasa yang digunakan untuk mendesain dan memformat halaman web. Kita mungkin sering mendengar tentang bahasa program seperti C, C++, Java, dan Visual basic. Masing-masing bahasa ini terdiri dari perintah sintak dan programming

.Sintak ini yang sering digunakan programmer untuk memanggil kode. Sangatlah penting mengetahui bagaimana cara menulis kode menggunakan bahasa yang relevan. Lebih dari itu, harus konvensional dengan aturan menyangkut bahasa tertentu. Didalam HTML, sintak ini disebut tag. Tag ditulis dengan tanda-kurung bersudut <sintak>. Ada kelompok tag yang sudah dikenal didalam HTML, yang mana digunakan untuk berbagai tujuan. Sebagai contoh, dalam rangka memodifikasi satu baris teks ke dalam bold, kita menerapkan tag bold dengan suatu tag , kemudian tulis beberapa teks atau suatu paragraph yang berisi beberapa teks,dan tutup tag menggunakan tag seperti dalam contoh listing 1. Semua tag didalam HTML harus ditutup menggunakan sintak </sintak>. Tetapi ada beberapa perkecualian pada aturan ini. Kita memakai tag ini sebab HTML bukanlah bahasa yang sensitif seperti C++ dan Java.[4]

2.2.8 CSS

CSS memungkinkan web developer untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah website .CSS (*Cascading StyleSheet*) digunakan untuk melengkapi file HTML, dan tugas utamanya adalah menetapkan aturan tampilan/*style* yang akan digunakan pada sebuah *website*. CSS diperkenalkan untuk pengembangan website pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan *parent-child* pada setiap *style*,Setelah CSS distandarisasikan, *Internet Explorer* dan Netscape melepas *browser* terbaru yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS. CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar mejadi rujukan oleh HTML dalam pendefinisian *style*.CSS menggunakan kode yang tersusun untuk menetapkan *style* pda elemen HTML atau dapat juga digunakan membuat *style* baru yang biasa disebut *class*.CSS dapat mengubah besar kecilnya *text*, mengganti warna *background* pada sebuah halaman, atau dapat pula mengubah warna *border* pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS.Singkatnya, CSS digunakan

untuk mengatur susunan tampilan pada halaman HTML. CSS dapat digunakan untuk menggantikan <font>, <b>, <u> dan <u>, dikarenakan hal berikut:sebuah*file* css dapat menjadi rujukan banyak halaman HTML. Hanya dibutuhkan 1 baris kode untuk melakukan hal tersebut. Ini berarti akan meminimalkan *file* HTML yang akan dibuat.Jika ingin mengubah tampilan *website* yang telah dibuat, maka yang perlu dilakukan hanya mengganti barisbaris kode pada css nya saja, tanpa perlu mengutak-atik *file-file* HTML nya, CSS dapat mengatur banyak atribut pada sebuah halaman secara mudah. Misalnya: warna *background, border, shadow,* yang berbeda pada masing-masing *tag* yang digunakan. [4]

2.2.9 PHP (*PHP Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

2.2.9.1 Sejarah PHP

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessin'/FormInterpreter*. Dengan perilisan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0.Pada rilis ini *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat.Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP

3.0.Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0.PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21.Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web *kompleks* tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0.Dalam versi ini, inti dari interpreterPHP mengalami perubahan besar.Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. [6]

2.2.9.2 Kelebihan PHP

Kelebihan PHP diantaranya :

a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
b. *Web Server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai IIS sampai dengan apache, dengan configurasi yang relatif mudah.
c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan *developer* yang siap membantu dalam pengembangan.
d. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
e. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui console serta juga dapat menjalankan perintah-perintah *system*. [6]

2.2.10 ERD (*Entity Relationship Diagram*)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi**.** Dalam rekayasa perangkat lunak, sebuah *Entity-Relationship Model* (ERM) merupakan abstrak dan konseptual representasi data. *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. Dimana sistem seringkali memiliki basis data relasional, dan ketentuannya bersifat *top-down.* Diagram untuk menggambarkan model *Entitiy-Relationship* ini disebut *Entitiy-Relationship diagram*, *ER diagram,* atau *ERD.*

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

a. *Entiti* (Entitas)

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah (weak entity) adalah suatu Entity dimana keberadaan dari entity tersebut tergantung dari keberadaan entity lain. Entity yang merupakan induknya disebut *Identifying Owner* dan relationshipnya disebut *Identifying Relationship.*

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain.

Jenis-jenis atribut :

- Key

Atribut yang digunakan untuk menentukan suatu entity secara unik.

- Atribut Simple
  Atribut yang bernilai tunggal.

- Atribut Multivalue
  Atribut yang memiliki sekelompok nilai untuk setiap instan entity.

- Atribut Composite
  Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.

- Atribut Derivatif
  Suatu atribut yang dihasilkan dari atribut yang lain.

c. Hubungan / Relasi

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

1. Satu ke satu (*One to one*)

   Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B. atau Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

2. Satu ke banyak (*One to many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B hanya dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. Banyak ke banyak (*Many to many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan demikian pula sebaliknya.[7]

2.2.11 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut.

Kita dapat menggunakan DFD untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru.

DFD dapat digambarkan dalam Diagram Context dan Level n. Huruf n dapat menggambarkan level dan proses di setiap lingkaran.

- Diagram Context
- Diagram Level n
  - DFD Logis
  - DFD Fisik

1. Context Diagram (CD)

Jenis pertama Context Diagram, adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD;

- Terminologi sistem:
  - *Batas Sistem* adalah batas antara "daerah kepentingan sistem".

- *Lingkungan Sistem* adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.

- *Interface* adalah aliran yang menghubungkan sebuah sistem dengan linkungan sistem tersebut.

- Menggunakan satu simbol proses,

- Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut,

- Antara Entitas Eksternal/Terminator tidak diperbolehkan komunikasi langsung

- Jika terdapat termintor yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda asterik (*) atau garis silang (#).

- Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipermainkan personil tersebut.

- Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

2. Diagram Level *n* / Data Flow Diagram Levelled

Dalam diagram n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram diagram logis. Dimana Diagram Level *n* merupakan hasil pengembangan dari *Context Diagram* ke dalam komponen yang lebih detail tersebut disebut dengan top-down partitioning. Jika kita melakukan pengembangan dengan benar, kita akan mendapatkan DFD-DFD yang seimbang.

Beberapa hal yang harus diperhatikan dalam membuat DFD ialah:

- Pemberian Nomor pada diagram level *n* dengan ketentuan sebagai berikut:

  - Setiap penurunan ke level yang lebih rendah harus mampu merepresentasikan proses tersebut dalam sepesifikasi proses yang jelas. Sehingga seandainya belum cukup jelas maka seharusnya diturunkan ke level yang lebih rendah.

  - Setiap penurunan harus dilakukan hanya jika perlu.

  - Tidak semua bagian dari sistem harus diturunkan dengan jumlah level yang sama karena yang kompleks bisa saja diturunkan, dan yang sederhana mungkin tidak perlu diturunkan. Selain itu, karena tidak semua proses dalam level yang sama punya derajat kompleksitas yang sama juga.

  - Konfirmasikan DFD yang telah dibuat pada pemakai dengan cara top-down.

  - Aliran data yang masuk dan keluar pada suatu proses di level n harus berhubungan dengan aliran data yang masuk dan keluar pada level n+1. Dimana level n+1 tersebut mendefinisikan sub-proses pada level n tersebut.

  - Penyimpanan yang muncul pada level n harus didefinisikan kembali pada level n+1, sedangkan penyimpanan yang muncul pada level n tidak harus muncul pada level n-1 karena penyimpanan tersebut bersifat lokal.

  - Ketika mulai menurunkan DFD dari level tertinggi, cobalah untuk mengidentifikasi external events dimana sistem harus memberikan respon. External events dalam hal ini berarti suatu kejadian yang berkaitan dengan pengolahan data di luar sistem, dan menyebabkan sistem kita memberikan respon.

- Jangan menghubungkan langsung antara satu penyimpanan dengan penyimpanan lainnya (harus melalui proses).

- Jangan menghubungkan langsung dengan tempat penyimpanan data dengan entitas eksternal / terminator (harus melalui proses), atau sebaliknya.

- Jangan membuat suatu proses menerima input tetapi tidak pernah mengeluarkan output yang disebut dengan istilah “black hole”.

- Jangan membuat suatu tempat penyimpanan menerima input tetapi tidak pernah digunakan untuk proses.

- Jangan membuat suatu hasil proses yang lengkap dengan data yang terbatas yang disebut dengan istilah “magic process”.

- Jika terdapat terminator yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda asterik (*) atau garis silang (#), begitu dengan bentuk penyimpanan.

- Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

3. DFD Fisik

Adalah representasi grafik dari sebuah sistem yang menunjukan entitas-entitas internal dan eksternal dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari entitas-entitas tersebut. Entitas-entitas internal adalah personel, tempat (sebuah bagian), atau mesin (misalnya, sebuah komputer) dalam sistem tersebut yang mentransformasikan data. Maka DFD fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan. (Tidak Bahas).

Perlu diperhatikan didalam memberikan keterangan di lingkaran-lingkaran (simbol proses) dan aliran-aliran data (simbol aliran data) dalam DFD fisik menggunakan label/keterangan dari kata benda untuk menunjukan bagaimana sistem mentransmisikan data antara lingkaran-lingkaran tersebut.

4. DFD Logis

Adalah representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan DFD logis untuk membuat dokumentasi sebuah sistem informasi karena DFD logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan.

Keuntungan dari DFD logis dibandingkan dengan DFD fisik adalah dapat memusatkan perhatian pada fungsi-funsi yang dilakukan sistem.[5]

2.2.12 *Flowmap*

Flowmap adalah penggambaran secara grafik dari langkah – langkah dan urutan prosedur dari suatu program. *Flowmap* berguna untuk membantu analis dan *programer* untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian. Biasanya *flowmap* mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Fungsi *Flowmap* yaitu mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan aliran data (dalam bentuk dokumen keluaran dan masukan).

Aturan membuat *Flowmap* untuk membuat sebuah analisis menggunakan *flowmap* seorang analis dan *programer* memerlukan beberapa tahapan, diantarnya:

1. *Flowmap* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol – simbol *flowmap* yang standar.[5]