

BAB III

LANDASAN TEORI

3.1. Konsep Dasar Sistem Informasi

Menurut (Jogiyanto 2005) terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

3.2. Definisi Sistem Informasi

Informasi merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan. Informasi diperoleh dari Sistem Informasi. Sistem Informasi didefinisikan oleh Robert A Leitch dan K. Roscoe Davis didalam Jogiyanto (2005) :

“Sistem Informasi adalah suatu Sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dna kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan”.

Suatu Sistem Informasi terdiri dari beberapa komponen yang biasa disebut dengan istilah Blok bangunan yaitu blok masukan, blok keluaran,

blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem keenam blok tersebut masing – masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

3.3. Magang

Menurut DeCenzo dan Robin (1999:227), *Training is a learning experience in that it seeks a relatively permanent change in an individual that will improve the ability to perform on the job.* Ini berarti bahwa pelatihan adalah suatu pengalaman pembelajaran di dalam mencari perubahan permanen secara relatif pada suatu individu yang akan memperbaiki kemampuan dalam melaksanakan pekerjaannya itu.

3.4. Pengelolaan

Menurut Suharsimi Arikunto (1993:31), pengelolaan dapat disamakan dengan manajemen, yang memiliki arti pengaturan atau pengurusan. Secara lebih rinci pengelolaan merupakan terjemahan istilah Inggris dari kata “*Management*”, kemudian di terjemahkan ke dalam bahasa Indonesia yaitu “manajemen” atau “menejemen”. Arti lain dari pengelolaan adalah penyelenggaraan atau pengurusan agar sesuatu yang dikelola dapat berjalan dengan lancar, efektif dan efisien.

3.5. Pendaftaran

Menurut (Dekdikbud 1993:1), Pendaftaran adalah proses, cara, pembuatan mendaftar yaitu pencatatan nama, alamat dsb dalam daftar. Jadi, pendaftaran adalah

proses pencatatan identitas pendaftar kedalam sebuah media penyimpanan yang digunakan dalam proses pendaftaran.

3.6. Aplikasi Web

Menurut Simarmata (2010), Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Fitur-fitur aplikasi web biasanya berupa data *persistence*, mendukung transaksi dan komposisi halaman web dinamis yang dapat dipertimbangkan sebagai hibridisasi antara hipermedia dan sistem informasi.

Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser* web. *Client-side* mempunyai tanggung jawab untuk pengekseskuan proses bisnis.

Interaksi Web dibagi ke dalam tiga langkah yaitu:

1. Permintaan

Pengguna mengirimkan permintaan ke server web, via halaman web yang ditampilkan pada browser web.

2. Pemrosesan

Server web menerima permintaan yang dikirimkan oleh pengguna kemudian memproses permintaan tersebut.

3. Jawaban

Browser menampilkan hasil dari permintaan pada jendela *browser*. Halaman web bisa terdiri dari beberapa jenis informasi grafis (tekstual dari multimedia). Kebanyakan komponen grafis dihasilkan dengan tool khusus, menggunakan manipulasi langsung dan editor *WYSIWYG*.

3.7. Aplikasi Dekstop

Menurut Konixbam (2009) *Desktop Based Application* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi *Internet* di suatu komputer otonom dengan *operating system* atau *platform* tertentu. Aplikasi *Desktop* difokuskan kepada aplikasi yang lebih independen. Hal tersebut bertujuan untuk mempermudah para pengguna aplikasi *desktop* dalam hal memodifikasi pengaturan aplikasi sehingga efektifitas, efesinsi waktu, dana, dan tenaga dapat lebih ditekankan semaksimal mungkin.

Secara garis besar pada pemrograman terutama pada aplikasi yang berbasis *desktop* dapat dibedakan menjadi dua macam yaitu pemrograman konvensional dan pemrograman visual.

1. Pemrograman konvensional merupakan metode mendesain suatu aplikasi, pemrograman dituntut untuk bisa menerapkan baris demi baris kode

program agar bisa menghasilkan sebuah bentuk tampilan aplikasi yang dibuat dan akan memakan waktu lama.

2. Pemrograman visual merupakan metode pembuatan program dimana seorang programmer membuat koneksi antar objek-objek dengan cara menggambar, menunjuk, dan mengklik pada diagram dan ikon dengan berinteraksi dengan diagram jalur.

Beberapa keunggulan yang dimiliki oleh aplikasi *desktop* yaitu:

1. Dapat berjalan dengan independen, tidak perlu menggunakan sebuah *web browser*.
2. Tidak memerlukan koneksi internet
3. Prosesnya lebih cepat dibanding aplikasi *web*

Kekurangannya yaitu :

1. Harus menginstal aplikasinya terlebih dahulu jika ingin menjalankannya
2. Bermasalah pada lisensi karena membutuhkan banyak lisensi pada setiap komputer yang berbeda-beda
3. Biasanya memerlukan *hardware* yang mempunyai kualitas yang baik atau cukup tinggi.

3.8. Hypertext Preprocessor (PHP)

Menurut Afriyudi (2008), PHP adalah singkatan dari *hypertext preprocessor*. Merupakan *script* untuk pemrograman berbasis *web server-side*. Dengan menggunakan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan *script* PHP.

Sintaks PHP mirip dengan bahasa C, Perl, Pascal dan basic. PHP dapat dikembangkan sebagai web spesifik yang menyediakan fungsi-fungsi khusus yang membuat pengembangan suatu web dapat dilakukan dengan mudah. PHP juga menyediakan koneksi database, protokol dan modul fungsi lainnya.

3.9. Sistem Basis Data

Menurut Kusrini (2007), Basis data adalah kumpulan data yang memiliki relasi antar entitas, sehingga adanya basis data ini mempunyai tujuan untuk mempermudah perolehan data dalam waktu yang singkat dan ketepatan data yang diperoleh.

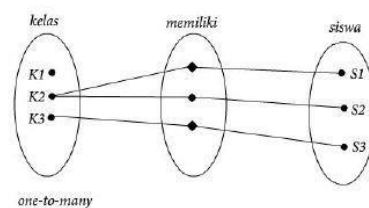
3.10. Entity Relationship Diagram (ERD)

Menurut Simarmata & Paryudi (2006), menyatakan bahwa untuk mengilustrasikan konsep model data, berikut disajikan dua model data yaitu *entity relationship model* dan *relational model*. ERD adalah alat permodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. ER diagram memiliki empat objek, yaitu:

1. Entitas (*entity*) adalah sesuatu yang nyata atau abstrak dimana akan menyimpan data.
2. Relasi (*Relationship*) adalah hubungan yang terjadi antara satu atau lebih entitas. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Pementaan kardinalitas terdiri dari berikut:

a. Relasi *one to many*

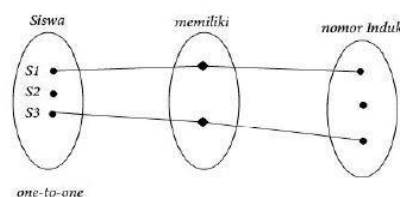
Relasi *one to many* berarti suatu entitas himpunan A dapat berhubungan dengan banyak entitas pada entitas himpunan B, namun tidak sebaliknya. Di bawah ini adalah contoh tipe relasi *one to many*, yaitu satu kelas dapat memiliki banyak siswa.



Gambar 3. 1 Tipe relasi *one to many*

b. Relasi *one to one*

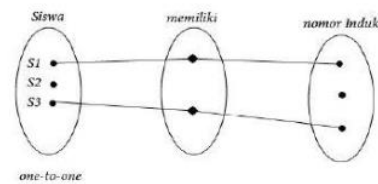
Relasi *one to one* berarti setiap entitas himpunan A hanya berhubungan dengan satu entitas himpunan B, begitu juga sebaliknya. Di bawah ini adalah contoh tipe relasi *one to one*, yaitu satu siswa pasti hanya memiliki satu nomor induk.



Gambar 3. 2 Tipe relasi *one to one*

c. Relasi rekursif *many to one*

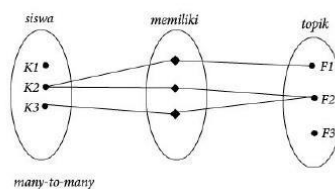
Relasi rekursif *many to one* adalah sebuah tipe relasi yang dimana entitasnya berpartisipasi lebih dari satu peran.



Gambar 3. 3 Tipe relasi *one to one*

d. Relasi *many to many*

Relasi *many to many* berarti setiap entitas himpunan A dapat berhubungan dengan entitas pada himpunan B, begitu juga sebaliknya. Sebagai contoh pada Gambar 6, yaitu satu siswa memiliki banyak topik dan setiap topik dapat dilihat atau dimiliki banyak siswa.



Gambar 3. 4 Tipe relasi *many to many*

3. Atribut (*Attribute*)

Atribut adalah ciri umum atau sebagian besar instansi pada entitas tertentu. Atau disebut properti, elemen data, dan field. Menurut Connolly dan Begg (2002) Atribut dapat dibedakan menjadi 5 jenis, yaitu:

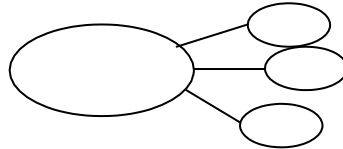
a. *Simple Attribute* adalah atribut yang terdiri dari komponen tunggal.

Simpel atribut tidak dapat dibagi menjadi komponen yang lebih kecil.

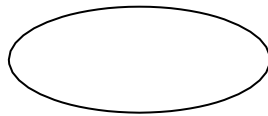


Gambar 3. 5 *Simple Attribute*

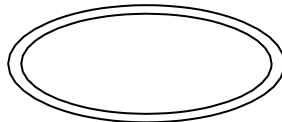
- b. *Composite Attribute* adalah atribut yang terdiri dari beberapa komponen yang bersifat independen.

Gambar 3. 6 *Composite Attribute*

- c. *Single-value Attribute* adalah atribut yang memegang nilai tunggal dari suatu entitas.

Gambar 3. 7 *Single-value Attribute*

- d. *Multi-value Attribute* adalah atribut yang dapat memegang nilai lebih dari suatu entitas.

Gambar 3. 8 *Multi-value Attribute*

- e. *Derived Attribute* adalah atribut yang mewakili turunan nilai sebuah atribut yang saling berkaitan dan belum tentu dalam tipe entitas yang sama.



Gambar 3. 9 *Derived Attribute*

- f. Kunci (*Key*)

Beberapa elemen data memiliki sifat, dengan mengetahui nilai yang telah diberikan oleh sebagian elemen data dari entity tertentu, dapat diidentifikasi nilai-nilai yang terkandung dalam elemen-elemen data lain ada entity yang sama. Elemen penentu tersebut adalah sebagai elemen data kunci (*key*).

Menurut Connolly dan Begg (2002) *keys* terdiri atas beberapa jenis, yaitu:

- a. *Candidate Key*

Candidate key merupakan *set* minimal dari suatu atribut yang secara unik mengidentifikasi setiap *occurrence* dari tipe entitas. *Candidate key* tidak boleh *null* (kosong).

- b. *Primary Key*

Sebuah *candidate key* yang dipilih untuk mengidentifikasi secara unik tiap kejadian pada suatu entitas. *Primary key* harus bernilai *unique* dan tidak boleh *null* (kosong).

c. *Composite Key*

Sebuah *candidate key* yang mempunyai dua atribut atau lebih. Suatu atribut yang membentuk *composite key* bukanlah kunci sederhana karena *composite key* tidak membentuk kunci senyawa.

d. *Alternate Key*

Sebuah *candidate key* yang tidak menjadi *primary key*. *Key* ini biasa disebut dengan *secondary key*.

e. *Foreign Key*

Himpunan atribut dalam suatu relasi yang cocok dengan *candidate key* dari beberapa relasi lainnya. *Foreign key* mengacu pada *primary key* suatu tabel. Nilai *foreign key* harus sesuai dengan nilai *primary key* yang diacunya.

Sebuah ERD memiliki beberapa jenis model yaitu :

3.10.1. Conceptual Data Model (CDM)

Merupakan model yang universal dan dapat menggambarkan semua struktur logic database (DBMS), dan tidak bergantung dari software atau pertimbangan struktur data storage. Sebuah CDM dapat diubah langsung menjadi PDM.

3.10.2. Physical Data Model (PDM)

Merupakan model ERD yang telah mengacu pada pemilihan software DBMS yang spesifik. Hal ini sering kali berbeda dikarenakan oleh struktur

database yang bervariasi, mulai dari model schema, tipe data penyimpanan dan sebagainya.

3.11. Data Flow Diagram (DFD)

Menurut Jogiyanto (1998), DFD banyak digunakan untuk menggambarkan sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa adanya pertimbangan khususnya lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembagkan arus data di dalam sistem dengan terstruktur dan jelas.

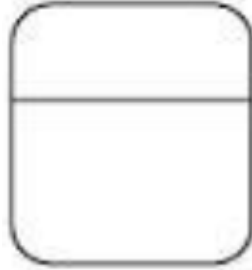
Berikut ini adalah simbol-simbol yang digunakan di DFD:

1. *Eksternal Entity* atau kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luar yang akan memberikan input atau menerima output dari sistem. *Eksternal Entity* disimbolkan dengan notasi kotak.



Gambar 3. 10 Simbol *Eksternal Entity*

2. *Process* adalah kegiatan yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan arus data yang akan keluar proses. Simbol proses berupa lingkaran atau persegi panjang bersudut tumpul.



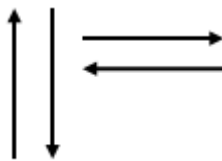
Gambar 3. 11 Simbol *Process*

3. *Data Store* adalah arus data (*data flow*) di DFD diberi simbol panah. Arus Data ini mengalir diantara proses, simpanan data (*data store*) dan kesatuan luar (*eksternal entity*). Arus data ini menunjukkan arus data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.



Gambar 3. 12 Simbol *Data Store*

4. *Data Flow* atau aliran data yang mengalir diantara proses. Aliran data dapat digambarkan dari bawah ke atas, kiri ke kanan, maupun sebaliknya.



Gambar 3. 13 Simbol *Data Flow*

Setiap simbol memiliki aturan tersendiri dalam penggunaannya. Aturan-aturan tersebut antara lain:

1. *External Entity* (Entitas Luar)

Aturan penggunaan untuk *external entity* antara lain:

- a. Data harus bergerak melalui proses, selama data tersebut berhubungan dengan sistem. Jika data tidak berhubungan dengan proses, maka aliran data tidak perlu ditampilkan pada DFD.
- b. Entitas luar diberi label dengan sebuah frase kata benda.

2. *Process*

Aturan penggunaan sebuah proses antara lain:

- a. Sebuah proses tidak hanya memiliki *output*. Jika sebuah objek hanya memiliki *output*, maka objek tersebut adalah *source*.
- b. Sebuah proses tidak hanya memiliki *input*. Jika sebuah objek hanya memiliki sebuah *input*, maka objek tersebut adalah entitas luar.
- c. Sebuah proses diberi label dengan sebuah frase kata kerja.

3. *Data Store*

Aturan-aturan dalam menggunakan *data store* adalah sebagai berikut:

- a. Data harus bergerak melalui proses dimana data diterima melalui suatu *source* untuk disimpan di *data store*.

- b. Data tidak dapat bergerak langsung dari *data source* menuju *external entity*.
- c. *Data store* diberi label dengan frase kata benda.
- 4. *Data Flow* (Aliran Data)

Aturan-aturan dalam menggunakan *data flow* antara lain:

- a. Sebuah aliran data hanya menggunakan satu arah antar simbol.
- b. Sebuah cabang pada aliran data memiliki arti data yang sama dari satu lokasi menuju ke satu atau lebih proses, tempat penyimpanan data, serta entitas luar.
- c. Sebuah aliran data tidak dapat bergerak ke proses asalnya sehingga membutuhkan proses lain untuk menangani, menghasilkan, dan mengembalikan aliran data ke proses asal.
- d. Aliran data atau *data flow* diberi label dengan frase kata benda.

3.11.1. Context Diagram

Context Diagram merupakan langkah pertama dalam pembuatan DFD. Pada *context diagram* dijelaskan sistem apa yang dibuat dan *external entity* apa saja yang terlibat. Dalam *context diagram* harus ada data yang masuk dan arus data yang keluar.

3.11.2. Data Flow Diagram Level 0

DFD *level 0* adalah langkah selanjutnya setelah *context diagram*. Pada langkah ini, digambarkan proses-proses yang terjadi dalam sistem informasi.

3.11.3. Data Flow Diagram Level 1

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*.

3.12. Analisa dan Desain Sistem

Menurut Ladjamudin (2005), Analisa sistem yang ada sangat bergantung pada teori umum sebagai sebuah landasan konseptual. Bertujuan untuk memperbaiki fungsi di dalam sistem yang ada sangat bergantung pada teori umum sebagai sebuah landasan konseptual. Bertujuan untuk memperbaiki fungsi di dalam sistem yang sedang berjalan agar menjadi lebih efisien, mengubah sasaran sistem yang sedang berjalan.

3.13. Document Flow

Menurut Jogiyanto (2005), diagram alir dokumen atau *paperwork flowchart* merupakan diagram alir yang menunjukkan arus laporan dan formulir beserta tembusannya. Berdasarkan pengertian di atas dapat disimpulkan bahwa diagram alir dokumen adalah diagram yang menggambarkan aliran seluruh dokumen. Diagram alir dokumen ini menggunakan simbol-simbol yang sama dengan diagram alir sistem. Diagram alir dokumen digambar dengan menggunakan simbol-simbol yang ada berikut ini:

1. *Terminator* digunakan untuk menunjukkan awal dan akhir proses dokumen.



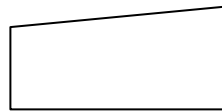
Gambar 3. 14 Simbol *Terminator*

2. *Document* digunakan sebagai input dan output baik secara manual ataupun dengan menggunakan computer.



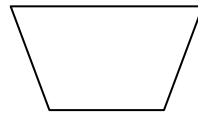
Gambar 3. 15 Simbol *Document*

3. *Manual Input* berfungsi untuk memasukkan data dengan menggunakan online keyboard.



Gambar 3. 16 Simbol *Manual Input*

4. *Manual Process* menunjukkan kegiatan manual.



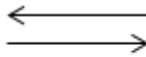
Gambar 3. 17 Simbol *Manual Process*

5. *Offline Storage* merupakan dokumen yang diarsip dan diurutkan berdasarkan N (*numeric*), A (*alphabet*), C (*chronological*).



Gambar 3. 18 Simbol *Offline Storage*

6. *Flow* digunakan sebagai arah aliran dokumen.



Gambar 3. 19 Simbol *Flow*

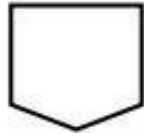
3.14. System Flow

System flow menurut Jogiyanto (1998) adalah bagian alir sistem menunjukkan arus pekerjaan secara keseluruhan dari sistem. System flow menunjukkan sistematika dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dilakukan sistem. Simbol-simbol yang digunakan dalam system flow ada 2 jenis simbol sebagai berikut:

1. *Flow Direction Symbols*

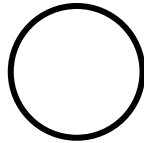
Flow direction symbols digunakan untuk menghubungkan antara satu simbol dengan simbol lainnya (Ladjamudin, 2005). Simbol ini disebut *connecting line*. Simbol-simbol tersebut dijelaskan pada gambar di bawah ini:

1. *Offline Connector* berfungsi untuk menyambungkan antara suatu proses dengan proses lainnya di halaman yang berbeda.



Gambar 3. 20 Simbol *Offline Connector*

2. *Connector* berfungsi untuk menyambungkan antara suatu proses dengan proses lainnya di halaman yang sama.



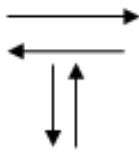
Gambar 3. 21 Simbol *Connector*

3. *Communication Link* berfungsi untuk mentransisi suatu data atau informasi dari setiap lokasi.



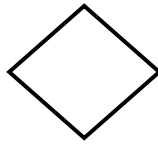
Gambar 3. 22 Simbol *Communication Link*

4. *Flow* berfungsi untuk menyatakan jalannya arus suatu proses.



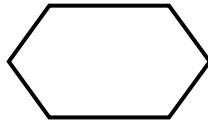
Gambar 3. 23 Simbol *Flow*

5. *Decision* untuk melakukan pengecekan. Biasanya menghasilkan jawaban ya atau tidak.



Gambar 3. 24 Simbol *Decision*

6. *Predefined Process* berfungsi sebagai tempat penyimpanan nilai awal.



Gambar 3. 25 Simbol *Predefined Process*

7. *Terminal* ini berfungsi untuk menyatakan permulaan atau penghentian suatu program.



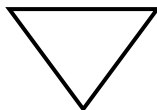
Gambar 3. 26 Simbol *Terminal*

8. *Key Operation* berfungsi untuk menyatakan suatu jenis operasi yang diproses dengan menggunakan mesin yang memiliki *keyboard*.



Gambar 3. 27 Simbol *Key Operation*

9. *Offline Storage* digunakan untuk menyimpan data ke suatu media tertentu.



Gambar 3. 28 Simbol *Offline Storage*

10. *Manual Input* berfungsi untuk memasukkan data dengan menggunakan *online keyboard*.



Gambar 3. 29 Simbol *Manual Input*

3.15. MySQL

Menurut (Arief, 2011d:152), MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya.

3.16. Xampp

Menurut (Widijanuarto, 2010), Xampp adalah sebuah aplikasi yang berisi MySQL dan Apache. Aplikasi ini dapat membantu untuk membangun sebuah aplikasi berbasis web tanpa harus menginstal MySQL dan Apache secara terpisah/sendiri-sendiri.

3.17. Notepad++

Menurut (Kurniawan, 2010) Notepad++ adalah sebuah aplikasi yang ditulis menggunakan jenis bahasa C++ yang berfungsi sebagai editor pengganti notepad default dari windows. Notepad++ sering digunakan untuk meng-*edit* web yang berformat html.