BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

A. Sistem

Menurut Rosa dan Shalahuddin (2016:291) “sistem adalah kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai”. Sedangkan menurut Al Fatta (2007:3) “secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain”

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. Batasan *(boundary)*: Penggambaran dari suatu elemen atau unsur mana yang termasuk didalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*): Segala sesuatu diluar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*): Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*): Sumber daya atau produk (informasi, laporan, dokumen, tampilan *layer computer*, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*): Kegiatan-kegiatan atau proses dalam suatu sistem yang mentranformasikan input menjadi bentuk setengah jadi (*output*).

6. Penyimpanan (*storage*): Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya.

B. Perancangan Sistem *(system design)*

> Menurut Rosa dan Shalahuddin (2016:17) menyimpulkan bahwa:
> Desain atau perancangan dalam perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segibiaya, waktu, dan perangkat.
>
> Sedangkan menurut Pressman (2010:515) mengemukakan bahwa:
> "*The system design depicts the overall product architecture, the subsystems that compose the product, the manner in which subsystems are allocated to processors, the allocation of classes to subsystems, and the design of the user interface*. Perancangan sistem menggambarkan keseluruhan arsitektur produk, subsistem yang menyusun produk, cara subsistem dialokasikan ke prosesor, alokasi kelas ke subsistem, dan perancangan antarmuka pengguna".
>
> Jika ditinjau dari segi tujuan menurut Hall (2011:615) yaitu:
> "*the purpose of the design phase is to produce a detailed description of proposed system that both satisfies the system requirements identified during system analysis and is in accordance with the conceptual design.* Tujuan dari tahap perancangan ini adalah untuk menghasilkan uraian rinci tentang sistem yang diusulkan yang memenuhi persyaratan sistem yang diidentifikasi selama analisis sistem dan sesuai dengan desain konseptual".

C. Sistem Informasi

Menurut Hall (2011:7) "*the information system is the set of formal procedures by which data are collected, processed into information, and distributed to users.* Sistem informasi adalah seperangkat prosedur formal dimana data dikumpulkan, diproses menjadi informasi, dan didistribusikan kepada pengguna".

Secara umum sistem informasi merupakan suatu sistem yang di dalamnya memuat tentang berbagaii informasi yang terkait dengan operasional suatu

organisasi yang berguna untuk mengambil keputusan dalam mencapai tujuan organisasi.

D. Basis Data

Menurut Rosa dan Shalahuddin (2016:43) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah diolah atau diinformasi dan membuat informasi tersedia saat dibutuhkan”. Sedangkan menurut Mulyarto (2008:261) “Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan dalam perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya

Untuk membuat basis data penulis menggunakan MySQL dan Xampp sebagai aplikasi server. Menurut MADCOMS (2016:2) “MySQL adalah sistem manajemen database SQL yang bersifat *Open Source* dan paling polpuler saat ini”. MADCOMS (2016:186) juga mengemukakan “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain-lain”.

E. Model Pengembangan Perangkat Lunak

> Menurut Rosa dan Shalahuddin (2016:26) menyimpulkan bahwa:
> “SDLC atau *Software Development Life Cycle* atau disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik)”.

Salah satu model yang cukup dikenal dalam dunia rekayasa perangkat lunak adalah Model Air Terjun (*Waterfall*). Ada 5 tahapan utama dalam Model Air Terjun (*Waterfall*) seperti analisis, desain, pengkodean, pengujian, dan tahap

pendukung (*support*). Disebut *waterfall* (berarti air terjun) karena memang diagram tahapan prosesnya mirip dengan air terjun yang bertingkat.

2.2. Teori Pendukung

A. *Data Flow Diagram*

Menurut Kendall dan Kendall (2011:558) *Data Flow Diagram* (DFD) adalah "*Graphical depiction of data processes, data flows, and data stores in a business system*. Penggambaran grafik dari proses data, arus data, dan penyimpanan data dalam sistem bisnis*"*.

1. Simbol-simbol yang digunakan

   Umumnya ada empat simbol yang sering digunakan dalam DFD menurut Kendall dan Kendall (2011:194), adalah:

   a. *External Entity*

      The double square is used to depict an external entity (another department, a business, a person, or a machine) that can send data to or receive data from the system. Kotak ganda digunakan untuk menggambarkan *external entity* (departemen lain, bisnis, seseorang, atau mesin) yang dapat mengirim data ke atau menerima data dari sistem.

   b. *Data Flow*

      Data flows occurring simultaneously can be depicted doing just that through the use of parallel arrows. Because an arrow represents data about a person, place, or thing, it too should be described with a noun. Data Flow yang terjadi secara bersamaan dapat digambarkan melakukan hal itu melalui

penggunaan panah sejajar. Karena panah mewakili data tentang seseorang, tempat, atau benda, hal itu juga harus dijelaskan dengan kata benda.

c. *Process*

Arectangle with rounded corners is used to show the occurrence of a transforming process. Sebuah persegi panjang dengan sudut membulat digunakan untuk menunjukkan terjadinya proses transformasi.

d. *Data Store*

Data stores represent a person, place, or thing, they are named with a noun. Data Store mewakili seseorang, tempat, atau benda, mereka diberi nama dengan kata benda.

2. Aturan-aturan dalam DFD menurut Kendall dan Kendall (2011:195), yaitu:

a. *The data flow diagram must have at least one process, and must not have any freestanding objects or objects connected to themselves. Data Flow Diagram* harus memiliki setidaknya satu *process*, dan tidak boleh memiliki objek yang berdiri bebas atau obyek bebas yang terhubung dengan dirinya sendiri.

b. A process must receive at least one data flow coming into the process and create at least one data flow leaving from the process. Suatu *process* harus menerima setidaknya satu *data flow* yang masuk ke dalam *process* dan membuat setidaknya satu *data flow* yang keluar dari *process*.

c. A data store should be connected to at least one process. Sebuah *data store* harus terhubung ke setidaknya satu *process*.

d. *External entities should not be connected to each other. External entity* seharusnya tidak terhubung satu sama lain.

3. Langkah-langkah Mengembangkan DFD menurut Kendall dan Kendall (2011:196), adalah:
   a. *Make a list of business activities and use it to determine various*. Buat daftar kegiatan usaha dan gunakanlah untuk menentukan berbagai macam:
      1. *External entity*
      2. *Data flow*
      3. *Process*
      4. *Data strore*
   b. *Create a context diagram that shows external entities and data flows to and from the system. Do not show any detailed processes or data stores.* Menciptakan diagram konteks yang menunjukkan *external entity* dan *data flow* mengalir menuju dan dari sistem.
   c. *Draw Diagram 0, the next level. Show processes, but keep them general. Show data stores at this level.* Menggambar Diagram 0, level selanjutnya. Tampilkan *process*, tapi jagalah agar tetap umum. Tampilkan *data store* di level ini.
   d. *Create a child diagram for each of the processes in Diagram 0*.
      Menciptakan *child diagram* untuk setiap proses dalam diagram 0.
   e. *Check for errors and make sure the labels you assign to each process and data flow are meaningful.* Mengecek kesalahan dan memastikan label-label yang anda tetapkan untuk setiap *process* dan *data flow* sangat berarti.
   f. *Develop a physical data flow diagram from the logical data flow diagram. Distinguish between manual and automated processes, describe actual files and reports by name, and add controls to indicate when processes are*

complete or errors occur. Kembangkan *physical data flow diagram* dari *logical data flow diagram*. Bedakan antara proses manual dan otomatis, jelaskan file dan laporan aktual berdasarkan nama, dan tambahkan kontrol untuk menunjukkan kapan proses selesai atau terjadi kesalahan.

g. *Partition the physical data flow diagram by separating or grouping parts of the diagram in order to facilitate programming and implementation.* Partisi *physical data flow diagram* dengan memisahkan atau mengelompokkan bagian diagram agar memudahkan pemrograman dan implementasinya.

4. Tahapan Pembuatan DFD menurut Kendall dan Kendall (2011:195), sebagai berikut :

   a. Buat *Context Diagram*

      The context diagram is the highest level in a data flow diagram and contains only one process, representing the entire system. The process is given the number zero. All external entities are shown. Context diagram adalah tingkat tertinggi dalam diagram alir data dan hanya berisi satu proses, mewakili keseluruhan sistem. Prosesnya diberi angka nol. Semua *external entity* ditampilkan.

   b. Menggambar Diagram 0

      Inputs and outputs specified in the first diagram remain constant in all subsequent diagrams. Masukan dan keluaran yang ditentukan pada diagram pertama tetap konstan pada semua diagram selanjutnya.

c. Buat *Child Diagram*

All data flow into or out of the parent process must be shown flowing into or out of the child diagram. Semua *data flow* masuk atau keluar dari proses induk harus ditunjukkan mengalir masuk atau keluar dari *child diagram*.

B. Kamus Data *(Data Dictionary)*

Menurut Rosa dan Shalahuddin (2016:73) “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunaksehingga masukan *(input)* dan keluaran *(output)* dapat dipahami secara umum (memiliki standar cara penulisan)”. Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur.

1. Hal yang harus dimuat dalam Kamus Data menurut Rosa dan Shalahuddin (2016:74) Kamus data biasanya berisi:
   a. Nama – nama dari data
   b. Digunakan pada – merupakan proses-proses yang terkait data
   c. Deskripsi – merupakan
   d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data
2. Notasi Kamus Data

   Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

   a. Notasi Tipe Data

      Untuk membuat spesifikasi masukkan dan keluaran suatu data.

Tabel II.1

Notasi Tipe Data

Notasi	Keterangan
X	*May enter or display/print any character* Dapat memasukkan atau menampilkan / mencetak karakter apapun
9	*Enter or display only numbers* Masukkan atau tampilkan hanya angka
Z	*Display leading zeros as spaces* Menampilkan angka nol sebagai spasi
,	*Insert commas into a numeric display* Masukkan koma ke tampilan numerik
.	*Insert a period into a numeric display* Masukkan periode menjadi tampilan numerik
/	*Insert slashes into a numeric display* Masukkan garis miring ke tampilan numerik
–	*Insert a hyphen into a numeric display* Masukkan tanda hubung ke tampilan numerik
V	*Indicate a decimal position (when the decimal point is not included)* Tunjukkan posisi desimal (bila titik desimal tidak disertakan)

Sumber : Kendall dan Kendall (2011:236)

b. Notasi Struktur Data

Berfungsi untuk membuat spesifikasi elemen data.

Tabel II.2

Notasi Struktur Data

Notasi	Keterangan
=	Disusun atau terdiri dari
+	Dan
[|]	Baik... atau...
$\{\}^n$	n kali diulang/ berninlai banyak
( )	Data opsional
...	Batas komentar

Sumber : Rosa dan Shalahuddin (2016:74)

C. *Entity Relationship Diagram* (ERD)

Menurut Rosa dan Shalahuddin (2016:53) “ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”.

1. Komponen ERD

Dalam memodelkan suatu desain menjadi ERD diperlukan beragam teknik, salah satunya pembuat ERD harus membuat gambar simbol dan beberapa notasi yang menunjukkan makna desain dalam rupa ERD. Adapun komponen dalam pembuatan ERD, yaitu sebagai berikut :

a. Entitas / *Entity*

Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses

oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

b. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

c. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan;biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)

d. Atribut kunci nilai / *multivalue*

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

e. Relasi

Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

f. Asosiasi / *association*

Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian

2. Derajat *Relationship*

Menurut Rosa dan Shalahuddin (2016:51) mengemukakan bahwa:
"ERD biasanya memiliki hubungan binary (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi ternary (satu relasi menghubungkan tiga buah relasi) atau N-ary (satu relasi menghubungkan banyak entitas), Tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary"*.

D. *Logical Record Structure* (LRS)

Menurut Fathansyah (2007:77) “menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”. Kardinalitas relasi yang terjadi diantara dua himpunan entitas, yaitu:

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

   Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B.

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

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

3. Banyak ke satu (*Many to one*)

   Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B.

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

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

E. Pengkodean

Menurut Kendall dan Kendall (2011:485) “*The process of putting ambiguous or cumbersome data into short, easily entered digits or letters is called coding.* Proses menempatkan data ambigu atau tidak praktis menjadi pendek, mudah masuk digit atau huruf disebut pengkodean”.

Ada beberapa macam tipe kode antara lain Kode Mnemonik (*Mnemonic Code*), Kode Urut (*Sequential Code*), Kode Blok Urut (*Block Code*), Kode Derivasi Alfabetik (*Alphabetic Derivasion Code*), Kode Chiper (*Chiper Code*),

Kode Subset Digit-Signifikan (*Significant-Digit Subset Code*), Urutan Kode Sederhana (*Simple Sequential Code).*

E. *Hierarchy Input Output Chart* (HIPO)

Menurut Al Fatta (2007:147) "HIPO merupakan teknik untuk mendokumentasikan pengembangan suatu sistem yang dikembangkan oleh IBM". HIPO dapat digunakan untuk memenuhi kebutuhan beberapa pengguna untuk kepentingan berbeda-beda, antara lain:

1. Seorang manajer dapat menggunakan dokumentasi HIPO untuk memperoleh gambaran umum sistem.
2. Seorang programer menggunakan HIPO untuk menentukan fungsi-fungsi dalam program yang dibuatnya.
3. Programer juga dapat menggunakan HIPO untuk mencari fungsi-fungsi yang dimodifikasi dengan cepat