

BAB II

LANDASAN TEORI

2.1. Sistem

Keneth Laudon & Jane Laudon (2015) mengungkapkan bahwa sistem adalah serangkaian komponen yang saling berhubungan untuk mencapai suatu tujuan. Menurut Romney dan Steinbart (2015) Sistem adalah : "Serangkaian data atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan.". Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar (Romney dan Steinbart, 2015).

Sistem adalah "suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan".

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan (Mulyadi, 2016)

2.2. Informasi

Laudon, Kenneth C yang diterjemahkan Lukki Sugiato (2015) mengungkapkan bahwa "Informasi dapat diartikan sebagai data yang telah diolah menjadi bentuk yang memiliki arti dan fungsi bagi manusia."

2.3. Sistem Informasi

Sistem informasi adalah cara-cara yang diorganisi untuk mengumpulkan, memasukan, mengolah serta menyimpan data, mengendalikan dan melaporkan informasi yang sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan. (Krismaji.2015).

2.4. Basis Data (*Database*)

Database adalah suatu kumpulan data terhubung (interrelated data) yang disimpan secara bersama – sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu kerangkapan data (*controlled redundancy*) (Sutabri, 2016).

2.5. Aplikasi Yang Digunakan

a. Dreamweaver

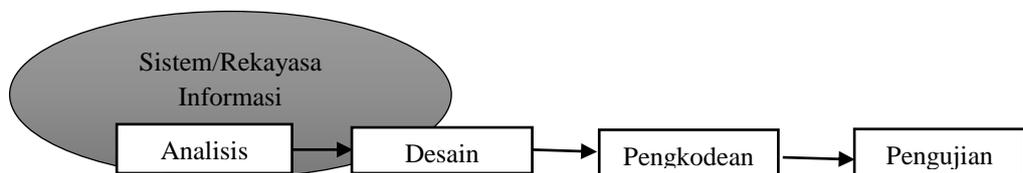
Menurut MADCOMS (2015) Adobe Dreamweaver adalah editor WYSIWYG dan editor kode untuk windows dan macintosh. anda dapat menggunakannya untuk menulis kode HTML, CSS, JSP, XML, PHP, Javascript, dan banyak lagi.

b. MySQL

My Structured Query Language (MySQL) adalah sebuah perangkat lunak sistem manajemen basis data yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL teredia sebagai perangkat lunak gratis dibawah lisensi GNU GPL (*General Public License*) (Sutanto Purba, 2015). “MySQL (baca: mai-se-kyu-el) merupakan *software* yang tergolong *database server* dan bersifat *Open Source*” (Kadir Taufik, 2017).

2.6 Metodologi Pengembangan Sistem

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan *waterfall* (Rosa dan Shalahuddin, 2015) yang terbagi dalam beberapa bagian, yaitu :



Gambar 2.1 Ilustrasi Model *Waterfall*

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user* . Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian focus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan keinginan.

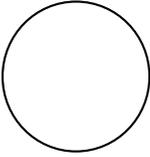
2.7. Alat Bantu Perancangan Sistem

Alat Bantu perancangan sistem yang dapat digunakan adalah :

a. *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) (Rosa A. S dan M. Shalahuddin, 2016) menjelaskan notasi yang digunakan dalam membuat DFD ada empat buah, yaitu sebagai berikut :

Tabel 2.1 Simbol-simbol *Data Flow Diagram*

Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>EntityRelationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>).</p> <p>catatan: nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda</p>
	<p>Aliran data; merupakan data yang dikirim antar catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”. Proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p>
	<p>Entitas luar (<i>externalentity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>catatan: nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>

(Sumber : Rosa A. S, dan M. Shalahuddin, 2016)

Tahapan-tahapan perancangan dengan menggunakan DFD (Rosa A. S dan M. Shalahuddin, 2016):

1. Membuat DFD *Level 0* atau sering disebut juga *Context Diagram DFD Level 0* menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD

Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD *Level 1* DFD *Level 1* digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD *Level 1* merupakan hasil *breakdown* DFD *Level 0* yang sebelumnya sudah dibuat.
3. Membuat DFD *Level 2* Modul-modul pada DFD *Level 1* dapat di *breakdown* menjadi DFD *Level 2*. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di *breakdown* lagi. Untuk sebuah sistem, jumlah DFD *Level 2* sama dengan jumlah modul pada DFD *Level 1* yang di *breakdown*.
4. Membuat DFD *Level 3* dan seterusnya DFD *Level 3, 4, 5,* dan seterusnya merupakan *breakdown* dari modul pada DFD *Level* di atasnya. *Breakdown* pada *level 3, 4, 5,* dan seterusnya aturannya sama dengan DFD *Level 1* atau *Level 2*.

b. Diagram Konteks

Diagram konteks adalah suatu diagram yang menggambarkan suatu proses pengolahan data secara umum dalam satu lingkungan dan hubungan dengan entitas luar.

c. DFD (*Data Flow Diagram*)

DFD adalah penjelasan lebih rinci dari diagram konteks dan proses fungsional yang ada dalam sistem. DFD menjelaskan tentang aliran masuk, aliran keluar, proses serta penyuntingan *file* yang digunakan. Keuntungan menggunakan DFD adalah memudahkan pemakai (*user*) yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan.

Tabel 2.2 Simbol-Simbol DFD

Simbol	Nama Simbol	Keterangan
	Simbol Terminator (<i>Eksternal/Internal Entity</i>)	Suatu kesatuan luar dapat disimbolkan dengan <i>notasi</i> kotak.
	Simbol Proses	Suatu Proses dapat ditunjukkan dengan simbol lingkaran.
	Simbol arus data (<i>Data Flow</i>)	Arus data diberi simbol suatu panah
	Simpanan data (<i>Data Store</i>)	Simpanan data dapat disimbolkan dengan sepasang garis <i>horizontal paralel</i> yang tertutup disalah satu ujungnya.

(Sumber : Rosa A. S, dan M. Shalahuddin, 2016)

Berikut ini keterangan simbol yang digunakan dalam *Data flow Diagram* (DFD) :

1. Kesatuan Luar (*External Entity*)

Setiap sistem mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya (*Eksternal Entity*) merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*Data Flow*)

Arus Data (*Data Flow*) di DFD diberi simbol suatu panah, arus data ini mengalir di antara proses, simpanan data, dan kesatuan luar.

3. Proses (*Process*)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

4. Simpanan Data (*Data Store*)

Simpanan Data (*Data Store*) merupakan simpanan dari data.

A. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan arus data yang ada di diagram aliran data. Dengan menggunakan kamus data, *analisis* sistem dapat mendefinisikan data yang mengalir di sistem secara lengkap. Kamus data digunakan untuk merancang *input*, merancang laporan dan *database*.

Format Kamus Data

Nama *database* :

Nama tabel :

Primary Key :

Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Keterangan: Kondisi berisi (contoh: NULL/NOT NULL)

Gambar 2.2 Format Kamus Data

B. Bagan alir program (*program flowchart*)

Bagan alir program (*program flowchart*) adalah bagian *flowchart* yang menggambarkan arus logika dari data yang akan diproses kedalam suatu program mulai dari awal sampai akhir. Bagan alir merupakan alat yang berguna bagi *programmer* untuk mempersiapkan program yang rumit. Simbol – simbol program yang digunakan antara lain sebagai berikut :

Tabel 2.3 Simbol-Simbol Bagan Alir Program (*Program Flowchart*)

Simbol	Keterangan
 <i>Input / Output</i>	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i>
 Proses	Simbol proses digunakan untuk mewakili suatu proses.
 Garis Alir	Simbol garis alir (<i>flow lines symbol</i>) digunakan untuk menunjukkan arus dari proses
 Penghubung	Simbol penghubung (<i>connector symbol</i>) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang sama / dihalaman yang lain
 Keputusan	Simbol keputusan (<i>decision symbol</i>) digunakan untuk suatu penyelesaian kondisi di dalam program

2.8. Sistem Basis Data

Sistem basis data adalah kumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengolah basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya. Dari pengertian tersebut dapat disimpulkan bahwa sistem basis data mempunyai beberapa elemen penting yaitu :

- a) Basis data sebagai inti dari sistem basis data.
- b) Perangkat lunak (*software*) untuk perancangan dan pengelolaan basis data.
- c) Perangkat keras (*Hardware*) sebagai pendukung operasi pengolahan data.
- d) Manusia (*Brainware*) yang mempunyai peran penting dalam sistem tersebut yaitu sebagai pemakai atau para *spesialis* informasi yang mempunyai fungsi sebagai perancangan atau pengelola.

Istilah Dalam *Database*

Beberapa istilah dalam *database* yang sering dipakai antara lain :

a) *Entity*

Entity adalah konsep informasi yang di rekam, meliputi orang, kejadian dan tempat.

b) *Atribut* atau *Field*

Atribut atau *Field* adalah sesuatu yang mewakili *entity*. *Field* kunci memegang peranan yang sangat penting dalam pembuatan tabel yang berisi *entity* dan relasinya. *Field* kunci merupakan satu *field* atau satu *setfield* yang terdapat dalam satu *file* yang merupakan kunci dan mewakili *record*. Kunci disini akan sangat penting apabila dalam program nanti terdapat fasilitas pencarian, karena *field* yang merupakan kunci akan menjadi penentu dalam pencarian program. *Field* kunci dibagi menjadi beberapa bagian yaitu :

a. Kunci Kandidat

Kunci Kandidat adalah satu *atribut* atau *field* yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu *entity*.

b. Kunci *Primer*

Kunci *Primer* adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu *entity*.

c. Kunci Alternatif

Kunci Alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci *primer*.

d. Kunci Tamu

Kunci Tamu adalah kunci *primer* yang ditempatkan pada *file* lain dan biasanya menunjukkan dan melengkapi suatu hubungan antara *file* satu dengan *file* yang lainnya.

Manfaat dari *database* sebagai berikut :

- a. Media permanen penyimpana pengolahan data.
- b. Petunjuk dan penjelasan bagaimana hasil pengolahan data disimpan.
- c. Data yang tersimpan dapat diubah dan dihapus.
- d. Rujukan pembuatan laporan

Sedangkan tujuan dari konsep *database* sebagai berikut :

- a) Meminimumkan terjadinya pengulangan data yang sama (*redudancy data*).
- b) Mencapai *interpendensi data*.
- c) *Data Value*
Data Value adalah informasi yang tersimpan dalam setiap *atribut*.
- d) *Record*
Record adalah kumpulan *atribut* yang saling berkaitan satu dengan yang lain dan menginformasikan suatu *entity* secara lengkap.
- e) *File*
File adalah kumpulan *record* yang mempunyai panjang atribut yang sama tetapi berbeda data *valuenya*.
- f) *Basis data atau Database*
Database adalah kumpulan *file* satu dengan *file* yang lainnya yang membentuk satu informasi sistem secara keseluruhan.
- g) *Diagram Relasi Entitas (Entity Relationship Diagram)*
 Diagram relasi entitas adalah suatu alat yang dapat digunakan untuk mengorganisasikan data yang dikumpulkan dimana dalam *diagram* ini dapat diperlihatkan *entitas* beserta hubungan antar *entitas* tersebut. *Diagram relasi entitas* memperlihatkan seluruh *entitas* yang ada dalam sistem berikut dengan relasinya.

Untuk membantu memberi gambaran *relasi* secara lengkap terdapat tiga macam *relasi* dalam hubungan antara *entitas* yaitu :

1. *One To One Relationship*

—————▶◀———— Hubungan antara entitas pertama dengan kedua adalah satu berbanding satu, hubungan tersebut dapat diwakilkan dengan tanda panah tunggal.

2. *One To Many Relationship*

—————▶————▶◀———— Hubungan antara *entitas* pertama dengan *entitas* kedua adalah satu berbanding banyak. *Relasi* antara keduanya diwakilkan dengan panah ganda untuk menunjukkan hubungan banyak dan panah tunggal untuk hubungan satu.

3. *Many To Many Relationship*

 Hubungan antara *entitas* pertama dan *entitas* kedua adalah banyak berbanding banyak, keduanya diwakilkan dengan panah ganda.