

BAB II TINJAUAN PUSTAKA

2.1 Perancangan

Menurut John Burch & Gary Grudnitski sebagaimana yang dikutip (Letsoin, 2013), Perancangan sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.

2.2 Pengertian Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2014).

2.3 Informasi

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber yang kemudian diolah, sehingga memberikan nilai, arti dan manfaat (Pratama, 2014).

2.4 Sistem Informasi

Sistem Informasi merupakan suatu sistem yang tujuannya menghasilkan informasi. Informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya. Data yang diolah saja belum cukup menjadi informasi. Untuk menjadi suatu informasi, data yang diolah tersebut harus berguna bagi penggunaannya (Liatmaja and Wardati 2013).

2.5 Data

Data adalah fakta yang sudah ditulis dalam bentuk catatan atau direkam kedalam berbagai bentuk media (Zaki and Supriatna 2015).

2.6 Sistem Informasi Akademik

Pengertian Akademik yaitu kegiatan yang dilakukan didalam lingkungan dunia pendidikan yang berhubungan dengan proses belajar mengajar. Menurut kamus besar Bahasa Indonesia edisi kedua, akademik adalah "lembaga pendidikan tinggi kurang lebih tiga tahun lamanya yang mendidik tenaga profesi". akademik adalah suatu bidang yang mempelajari tentang kurikulum atau pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan/pembelajaran yang dapat dikelola oleh suatu sekolah atau lembaga pendidikan (liatmaja, 2013).

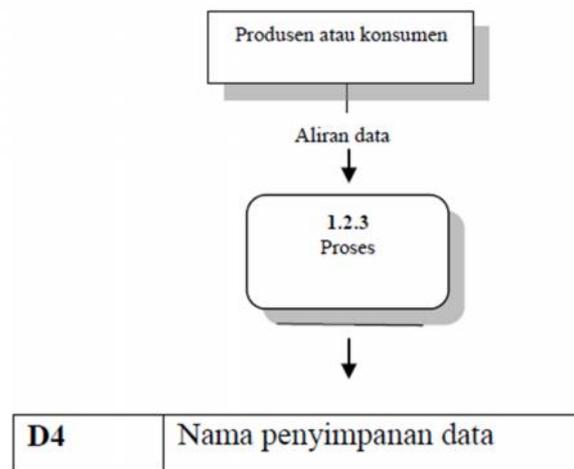
2.7 Kamus Data

Kamus data (*data dictionary*) digunakan untuk menyimpan deskripsi data yang digunakan dalam basis data. Setiap elemen data antara lain memiliki tipe dan ukuran. Kamus data terkadang juga berisi ketentuan yang mengatur nilai yang bisa dimasukkan dalam field.

2.8 Alat Bantu perancangan Sistem

2.8.1 Data Flow Diagram (DFD)

Sukamto dan Shalahuddin (2014:69), *Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem. Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson:

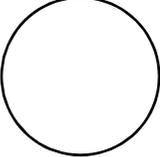
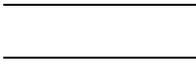


(Sumber: Rosa A.S. dan M. Shalahuddin)

Gambar 2.1 Contoh DFD yang dikembangkan Chris Gane dan Trish Sarson

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur. Sukamto dan Shalahuddin (2014: 71) notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.1 Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

No	Notasi	Keterangan
1.		<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. Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p><i>File</i> atau <i>basisdata</i> 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 ERD, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>.</p>
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan. Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
4.		<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya "data siswa".</p>

Sumber : Sukamto dan Shalahuddin (2014: 71)

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan *DFD*:

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 system lain. *DFD Level 0* digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat *DFD Level 1*

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam system 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* dua. 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 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 persis dengan *DFD Level 1* atau Level 2

Pada satu diagram *DFD* sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit.

2.8.2 Flowchart

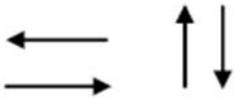
Ladjamudin (2013: 263), *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

2.8.2.1 Simbol-Simbol *Flowchart*

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 kelompok adalah sebagai berikut:

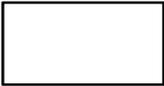
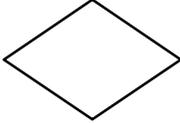
1. *Flow Direction Symbols* (Simbol penghubung/ alur)
2. *Processing Symbols* (Simbol proses)
3. *Input-output Symbols* (Simbol Input-Output)

Tabel 2.2 Simbol-simbol Penghubung/ Alur (*Flow Direction Symbols*)

No	Simbol	Deskripsi
1.		Simbol Arus/ <i>Flow</i> menyatakan jalannya arus suatu proses
2.		Simbol <i>Communication Link</i> Menyatakan bahwa adanya transisi suatu data/ informasi dari satu lokasi ke lokasi lainnya
3.		Simbol <i>Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama
4.		Simbol <i>Offline Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman /lembar yang berbeda

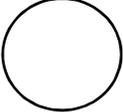
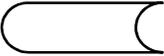
Sumber : Ladjamudin (2013: 263)

Tabel 2.3 Simbol-simbol Proses (*Processing Symbols*)

No	Simbol	Deskripsi
1.		Simbol <i>Offline Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda
2.		Simbol Manual Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual)
3.		Simbol <i>Decision/ Logika</i> Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak
4.		Simbol <i>Predifined Proses</i> Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
5.		Simbol Terminal Meyatakan permulaan atau akhir suatu program
6.		Simbol <i>Keying Operation</i> Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
7.		Simbol <i>Off-line Storage</i> Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
8.		Simbol Manual Input Memasukkan data secara manual dengan menggunakan online keyboard

Sumber : Ladjamudin (2013: 263)

Tabel 2.4 Simbol-simbol Input-Output (*Input-Output Symbols*)

No	Simbol	Deskripsi
1.		Simbol <i>Input-Output</i> Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
2.		Simbol <i>Punched Card</i> Menyatakan input berasal dari kartu atau output ditulis ke kartu
3.		Simbol <i>Magnetic-tape unit</i> Menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic
4.		Simbol <i>Disk Storage</i> Menyatakan input berasal dari disk atau output disimpan ke disk
5.		Simbol <i>Document</i> Untuk mencetak laporan ke printer
6.		Simbol <i>Display</i> Menyatakan peralatan output yang digunakan berupa layar (video, komputer)

Sumber : Ladjamudin (2013: 263)

2.9 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah Metode *Waterfall* (Pendekatan Terstruktur). Metode ini merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan. Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: analisis kebutuhan (requirement), desain sistem (design system), pengkodean (Coding) & pengujian (Testing), Penerapan Program, pemeliharaan.

Tahapan tahapan dari metode waterfall adalah sebagai berikut:

1. *Requirement Analysis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. *Integration Testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. *Operation Maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.10 Bahasa Pemrograman dan Perangkat Lunak Pendukung

2.10.1 Website

Menurut (Wahana, 2013) menyatakan bahwa “*Website* merupakan lokasi yang akan digunakan untuk mengumpulkan berbagai *file* halaman *web* yang terdiri dari gambar, CSS, audio dan sebagainya”.

2.10.2 HTML

Menurut Hidayatullah dan Kawistara (2015) “*Hypertext Markup Language* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman web”. yang bias dilakukan dengan HTML yaitu:

- a. Mengatur tampilan dari halaman *web* dan isinya.
- b. Membuat tabel dalam halaman *web*.
- c. Mempublikasikan halaman *web* secara *online*.
- d. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.

2.10.3 Pengertian PHP

Menurut MADCOMS (2016) “*PHP (Hypertext pre-processor)* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”. PHP dapat digunakan dengan gratis (*free*) dan bersifat *open source*. PHP dirilis dalam lisensi PHP *license*.

2.10.4 Pengertian Notepad++

Menurut MADCOMS (2016) “Notepad++ adalah sebuah text editor yang sangat berguna dalam membuat program. Notepad++ menggunakan komponen scintilla untuk menampilkan text dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi M.Windows”.

2.10.5 XAMPP

XAMPP merupakan paket *server web* PHP dan *database* MySQL yang paling populer di kalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai *databasenya* (Betha Sidik, 2014 : 72).

2.10.6 Pengenalan Java Script

Menurut Hidayatullah dan Kawistara (2015) “*Java script (js)* ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu *web*”.

2.10.7 MySQL

Menurut Kadir (2013:15) Mengemukakan bahwa “MySQL adalah nama database server, database server adalah server yang berfungsi untuk menangani database, MySQL adalah database server yang sangat populer dan banyak digunakan untuk menangani data yang disajikan di halaman Web”. Raharjo (2015:16), menjelaskan *MySQL* merupakan software RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi-user)*, dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

2.10.8 Sublime Text

Bos (2014:12) menjelaskan sublime text merupakan salah satu text editor yang sangat powerful yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi.

2.10.9 Balsamiq Mockup

Balsamiq mockup adalah program aplikasi yang digunakan dalam pembuatan tampilan user *interface* sebuah aplikasi. Software ini sudah menyediakan *tools* yang dapat memudahkan dalam pembuatan desain yang

akan dibuat. *Software* ini berfokus pada konteng yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna (Attaufiq, 2014:73).

