

BAB II LANDASAN TEORI

2.1 Sanggar Bunga Douglass Florist

Sanggar bunga Douglass Florist merupakan sanggar bunga yang didirikan oleh Maymunah Saad Nabhah pada tahun 1972. Pada awalnya Sanggar Bunga Douglass Florist hanya lah kebun bunga yang dikelola oleh ibu Maymunah karena hobinya dengan merawat bunga, namun seiring berjalannya waktu ibu Maymunah mengubah hobinya tersebut menjadi suatu pekerjaan yang menghasilkan pundi-pundi rupiah lewat hasil dari hobinya merangkai bunga. Hingga pada awal tahun 2000-an munculah ide membuat sewa bunga papan yang usahanya itu berjalan hingga kini.

2.2 Sistem

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Kadir, 2014).

2.3 Informasi

McFadden, dan kawan-kawan mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik melakukan pendekatan secara matematis untuk mendefinisikan informasi (Kroenke). Menurut mereka, informasi adalah jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima. Artinya, dengan adanya informasi, tingkat kepastian menjadi meningkat. Menurut Davis, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Kadir, 2014).

2.4 Sistem Informasi

Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Based Information System* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi berbasis komputer. Ada beragam definisi sistem informasi, yaitu :

- a. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
- b. Bodnar dan Hopwoo, sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
- d. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
- e. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan yang spesifik.
- f. Wilkinson, Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014).

2.5 Metode Pengembangan Sistem

Pada awal pengembangan perangkat lunak, para pembuat program (*programmer*) langsung melakukan pengodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak. Dan ditemuilah kendala-kendala seiring dengan pengembangan skala sistem-sistem perangkat yang semakin besar (Rosa, 2011).

SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem skala usaha besar secara fungsional untuk para konglomerat pada zaman itu. Sistem-sistem yang di bangun mengelola informasi kegiatan dan rutinitas dari perusahaan-perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya.

SDLC atau *Software Development Life Cycle* atau sering 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).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

a. Inisiasi (*Initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

b. Pengembangan Konsep Sistem (*System Concept Development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudaha sistem.

c. Perencanaan (*Planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya.

d. Analisis Kebutuhan (*Requirements Analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan user dan membuat dokumen kebutuhan fungsional.

e. Desain (*Design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

f. Pengembangan (*Development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, mempersiapkan berkas atau file pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program.

g. Integrasi dan Pengujian (*Integration and Test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional dengan diarahkan oleh staf penjamin kualitas dan user sehingga menghasilkan laporan analisis pengujian.

h. Implementasi (*Implementation*)

Implementasi perangkat lunak pada lingkungan produksi (lingkungan pada user) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

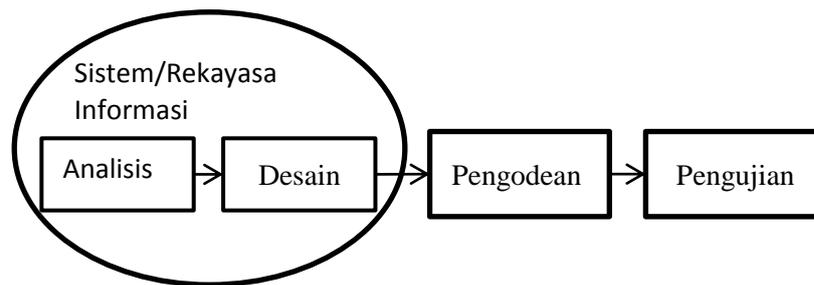
i. Operasi dan Pemeliharaan (*Operations and Maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi, termasuk implementasi akhir dan masuk pada proses peninjauan.

j. Disposisi (*Disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, diantaranya adalah model *waterfall*. Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (Rosa, 2011) seperti pada Gambar 2.1.



Gambar 2.1 Ilustrasi Model *Waterfall* (Sumber : Rosa, 2011)

a. Analisis Kebutuhan Perangkat Lunak

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

b. Desain

Desain perangkat lunak adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Kode Program

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

d. Pengujian

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

e. Pendukung

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab berikut :

- a. Perubahan spesifikasi perangkat lunak terjadi ditengah alur pengembangan.
- b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur. Pelanggan seringkali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
- c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Dengan berbagai kelemahan yang dimiliki model air terjun tapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan model pengembangan perangkat lunak. Model air terjun sangat cocok digunakan kebutuhan pelanggan sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun adalah struktur pengembangan sistem jelas, dokumentasi dihasilkan disetiap tahap dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih pelaksanaan tahap).

2.6 Flowchart

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analisis sistem atau pemograman dapat mengikuti pedoman-pedoman sebagai berikut (Jogiyanto, 2005) :

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
- e. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
- f. Kegiatan yang terpoting dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakanlah simbol-simbol bagan alir yang standar.

Terdapat lima macam bagan alir, yaitu bagan alir sistem (*system flowchart*), bagan alir dokumen (*document flowchart*), bagan alir skematik (*schematic flowchart*), bagan alir program (*program flowchart*) dan bagan alir proses (*process flowchart*). Akan tetapi yang dibahas dalam penelitian ini menggunakan dua bagan alir, yaitu:

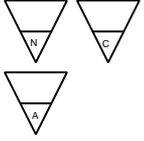
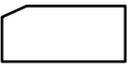
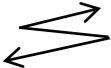
a. Bagan Alir Dokumen (*Document Flowchart*)

Bagan alir dokumen (*document flowchart*) atau yang sering disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

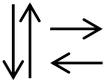
Tabel 2.1 Simbol Bagan Alir Dokumen

| Nama | Simbol | Deskripsi |
|------------------------|---|---|
| Simbol dokumen |  | Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer. |
| Simbol kegiatan manual |  | Menunjukkan pekerjaan manual. |

Tabel 2.1 Simbol Bagan Alir Dokumen (lanjutan)

| | | |
|----------------------------------|---|---|
| Simbol simpanan <i>offline</i> |  | <i>File</i> non-komputer yang diarsipurut angka (<i>numerical</i>). <i>File</i> non-komputer yang diarsipurut huruf (<i>alphabetical</i>). <i>File</i> non-komputer yang diarsipurut tanggal (<i>cronological</i>). |
| Simbol kartu plong |  | Menunjukkan <i>input/output</i> yang menggunakan kartu plong (<i>punched card</i>). |
| Simbol proses |  | Menunjukkan kegiatan proses dari operasi program komputer. |
| Simbol operasi luar |  | Menunjukkan operasi yang dilakukan di luar proses operasi komputer. |
| Simbol pengurutan <i>offline</i> |  | Menunjukkan proses pengurutan data di luar proses komputer. |
| Simbol pita magnetik |  | Menunjukkan <i>input/output</i> menggunakan pita magnetik. |
| Simbol drum magnetik |  | Menunjukkan <i>input/output</i> menggunakan drum magnetik. |
| Simbol pita kertas berlubang |  | Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang. |
| Simbol <i>keyboard</i> |  | Menunjukkan input yang menggunakan <i>on-line keyboard</i> . |
| Simbol display |  | Menunjukkan <i>output</i> yang ditampilkan di monitor. |
| Simbol pita kontrol |  | Menunjukkan penggunaan pita kontrol (<i>control tape</i>) dalam <i>batch control total</i> untuk pencocokan di proses <i>batch processing</i> . |
| Simbol hubungan komunikasi |  | Menunjukkan proses transmisi data melalui channel komunikasi. |

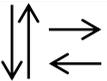
Tabel 2.1 Simbol Bagan Alir Dokumen (lanjutan)

| | | |
|-------------------|---|---|
| Simbol garis air |  | Menunjukkan arus dari proses. |
| Simbol penjelasan |  | Menunjukkan penjelasan dari suatu proses. |
| Simbol penghubung |  | Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain. |

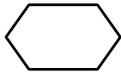
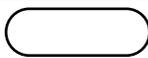
b. Bagan Alir Program (*Program Flowchart*)

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.

Tabel 2.2 Simbol Bagan Alir Program

| Nama | Simbol | Deskripsi |
|----------------------------|---|---|
| Simbol <i>input/output</i> |  | Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> . |
| Simbol proses |  | Simbol proses digunakan untuk mewakili suatu proses. |
| Simbol garis alir |  | Simbol garis alir digunakan untuk menunjukkan arus dari proses. |
| Simbol penghubung |  | Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya. |
| Simbol keputusan |  | Simbol keputusan digunakan untuk suatu penyeleksian kondisi di dalam program. |
| Simbol proses terdefinisi |  | Simbol ini digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain. |

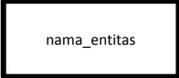
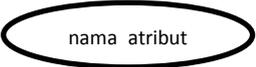
Tabel 2.2 Simbol Bagan Alir Program (lanjutan)

| | | |
|-----------------------|---|---|
| Simbol persiapan |  | Simbol persiapan digunakan untuk memberi nilai awal suatu besaran. |
| Simbol titik terminal |  | Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses. |

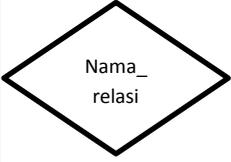
2.7 Entity Relationship Diagram (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan ERD (*Entity Relationship Diagram*). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional, sehingga jika penyimpanan data menggunakan OODBMS, maka perancangan basis data tidak perlu menggunakan ERD. Simbol-simbol ERD, dapat dilihat pada Tabel 2.3 (Rosa, 2011).

Tabel 2.3 Simbol-Simbol ERD

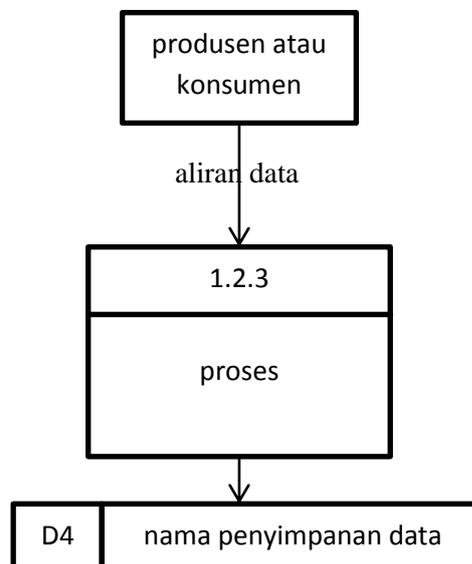
| Simbol | Deskripsi |
|--|---|
| Entitas/Entity  | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data. |
| Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas. |
| Atribut kunci primer  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id. |
| Atribut multi nilai/multi value  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. |

Tabel 2.3 Simbol-Simbol ERD (lanjutan)

| | |
|---|---|
| <p>Relasi</p>  | <p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p> |
| <p>Asosiasi/association</p> <p><u>1</u> 0..*</p> | <p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> |

2.8 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structure System 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 pada Gambar 2.3.



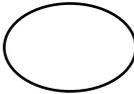
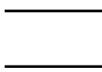
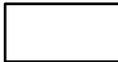
Gambar 2.2 DFD dikembangkan oleh Crish Gane & Trish Sarson (Sumber : Rosa, 2011)

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an dimana mengubah persegi dengan sudut lengkung (pada DFD Crish Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur. Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau 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*).

DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa *level* yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemograman terstruktur, karena pemograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur (Rosa, 2011).

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemograman berorientasi objek. Paradigma pemograman terstruktur dan pemograman berorientasi objek merupakan hal yang berbeda. Simbol-simbol pada DFD Edward Yourdon dan Tom DeMarco adalah seperti pada Tabel 2.4.

Tabel 2.4 Simbol *Data Flow Diagram*

| Nama | Simbol | Deskripsi |
|---|---|---|
| Proses atau fungsi atau prosedur |  | 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. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja. |
| <i>File</i> atau basis data atau penyimpanan (<i>storage</i>) |  | <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>Entity Relationship Diagram, Conceptual Data Model, Physical Data Model</i>). Nama yang diberikan pada sebuah penyimpanan biasanya kata benda. |
| <i>Entitas luar</i> (<i>external entity</i>) |  | 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 sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda |

Tabel 2.4 Simbol *Data Flow Diagram* (lanjutan)

| | | |
|-------------|---|---|
| Aliran data |  | 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>). Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data seperti “siswa”. |
|-------------|---|---|

2.9 Basis Data

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data di maksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda.

Umumnya DBMS menyediakan fitur-fitur sebagai berikut :

a. Independensi data program

Karena basis data ditangani oleh DBMS, program dapat ditulis sehingga tidak tergantung pada stuktur data dalam basis data. Dengan perkataan lain, program tidak akan terpengaruh sekiranya bentuk fisik data diubah.

b. Keamanan

Keamanan dimaksudkan untuk mencegah pengaksesan data oleh orang yang tidak berwenang.

c. Integritas

Hal ini ditujukan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.

d. Konkurensi

Konkurensi memungkinkan data dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.

e. Pemulihan (*recovery*)

DBMS menyediakan mekanisme untuk mengembalikan basis data ke keadaan semula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

f. Katalog sistem

Katalog sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

g. Perangkat produktivitas

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktivitas, DBMS menyediakan sejumlah perangkat produktivitas seperti pembangkit *query* dan pembangkit laporan.

Komponen-komponen yang menyusun lingkungan DBMS terdiri atas:

- a. Perangkat keras. Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa komputer dan periferal pendukungnya. Komputer dapat berupa PC, minikomputer, *mainframe*, dan lain-lain.
- b. Perangkat lunak. Komponen perangkat lunak mencakup DBMS itu sendiri, program aplikasi, serta perangkat lunak pendukung untuk komputer dan jaringan. Program aplikasi dapat dibangun dengan menggunakan bahasa pemrograman seperti C++, Pascal, Delphi, atau Visual BASIC.
- c. Data. Bagi sisi pemakai, komponen terpenting dalam DBMS adalah data karena dari data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

- d. Prosedur. Prosedur adalah petunjuk tertulis yang berisi cara merancang hingga menggunakan basis data. Beberapa hal yang dimasukkan dalam prosedur:
 1. Cara masuk ke DBMS (*login*).
 2. Cara memakai fasilitas-fasilitas tertentu dalam DBMS maupun cara menggunakan aplikasi.
 3. Cara mengaktifkan dan menghentikan DBMS.
 4. Cara membuat cadangan basis data dan cara mengembalikan cadangan ke DBMS.
- e. Orang. Komponen orang dapat dibagi menjadi tiga kelompok, yaitu :
 1. Pemakai akhir (*end-user*).
 2. Pemogram aplikasi.
 3. Administrator basis data.

Terdapat beberapa elemen basis data, yaitu :

a. *Database*

Database atau basis data adalah kumpulan tabel yang mempunyai kaitan antara suatu tabel dengan tabel lainnya sehingga membentuk suatu bangunan data.

b. Tabel

Tabel adalah kumpulan record-record yang mempunyai panjang elemen yang sama dan atribut yang sama namun berbeda data *valuenya*.

c. Entitas

Entitas adalah sekumpulan objek yang terdefiniskan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

d. Atribut

Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas objek atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

e. *Data Value* (Nilai Data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data, elemen atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo dan lain-lain yang merupakan isi data nama pegawai tersebut.

f. *File*

File adalah kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda nilai datanya.

g. *Record/Tuple*

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

2.10 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman web yang dapat disisipkan dalam script HTML. Banyak sintaks di dalamnya yang mirip dengan bahasa C , Java dan Perl. Tujuan dari bahasa ini adalah membantu para pengembang web untuk membuat web dinamis dengan cepat (Ari, 2012).

Ketika seseorang mengunjungi web berbasis PHP, *web server* akan memproses kode-kode PHP. Beberapa perintah atau kode dari PHP tersebut selanjutnya ada yang diterjemahkan ke dalam HTML dan beberapa ada yang disembunyikan (misalnya proses kalkulasi dan operasi). Setelah diterjemahkan ke dalam HTML, web server akan mengirim kembali ke web browser pengunjung tersebut.

2.11 MySQL

Menurut Solichin (2016), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti *Apache* yang merupakan *software* yang

dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Beberapa kelebihan MySQL antara lain : *free* (bebas di *download*), stabil dan tangguh, fleksibel dengan berbagai pemrograman, security yang baik, dukungan dari banyak komunitas, kemudahan management *database*, mendukung transaksi dan perkembangan *software* yang cukup cepat.

2.12 Xampp

XAMPP merupakan paket PHP dan MySQL berbasis *open source* yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket. Beberapa paket yang dibundel adalah *Apache HTTPD*, *mod_autoindex_color module*, *FileZilla FTP Server*, *Mercury Mail Transport Agent*, *OpenSSL*, *SQLite*, *The Webalizer*, *msmtplib (a sendmail compatible SMTP client)*, *MySQL*, *PrimeBase XT Storage Engine for MySQL*, *PHP*, *eAccelerator extension*, *Xdebug extension*, *Ming extension*, *PDFlib Lite extension*, *PEAR*, *phpMyAdmin*, *FPDF Library*, *ADOdb*, *Perl*, *CPAN*, *PPM*, *mod_pearl*, *Apache:ASP* (Riyanto, 2015).

2.13 PHPMyAdmin

PHPMyAdmin merupakan aplikasi web berbasis PHP yang telah banyak digunakan untuk administrasi *database* MySQL. Setelah paket XAMPP terinstal, *PHPMyAdmin* dapat diakses melalui <http://localhost/phpmyadmin>. Karena kehandalannya, fitur seperti *PHPMyAdmin* tidak hanya diikuti untuk paket *web server* maya seperti XAMPP, tetapi juga banyak digunakan para penyedia domain hosting untuk memudahkan pelanggannya melakukan administrasi *database* MySQL (Riyanto, 2015).

2.14 Web Browser

Web browser atau biasa disebut *browser* (perambatan) adalah perangkat lunak yang berguna untuk mengakses informasi web ataupun untuk melakukan transaksi via web. Microsoft Internet Explorer dan Mozilla Firefox merupakan contoh *browser* yang terkenal di lingkungan Windows (Abdul Kadir, 2014).

2.15 SMS Gateway

SMS gateway adalah suatu platform yang menyediakan mekanisme untuk EUA, mengantar dan menerima SMS dari peralatan mobile (HP, PDA *Phone* dan lain-lain) melalui *SMS Gateway's Shortcode*. Sejatinya *SMS gateway* merupakan sebuah sistem aplikasi yang digunakan untuk mengirim dan atau menerima SMS yang biasanya digunakan pada aplikasi bisnis baik untuk *broadcast* promosi, *service* informasi terhadap pengguna, penyebaran konten produk/jasa dan sebagainya (Hilmi, 2015).

2.16 Pengkodean

Pengkodean atau kode adalah suatu pengkodean digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Ada beberapa macam tipe dari kode yang digunakan didalam sistem informasi, diantaranya adalah (Jogiyanto H.M., 2005) :

a. Kode Memonik (*Memonic Code*)

Kode Memonik digunakan untuk tujuan supaya mudah diingat. Kode memonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang diwakili dengan kode ini. Umumnya kode memonik menggunakan huruf, tetapi juga dapat dengan gabungan huruf dan angka.

b. Kode Urut (*Sequential Code*)

Kode urut disebut juga dengan kode seri (*serial code*) merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode Blok (*Block Code*)

Kode blok mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode Group (*Group Code*)

Kode group merupakan kode yang berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

e. Kode Desimal (*Desimal Code*)

Kode desimal mengklasifikasi kode atas dasar 10 unit angka desimal dimulai angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari

1.17 Penelitian Terkait

Penelitian yang terkait dengan penelitian yang akan dilakukan adalah menurut Antonio, dkk (2014) dalam penelitiannya menyimpulkan bahwa hasil pembuatan *website* Toko Amai di kota Dili Timor Leste Berbasis web ini dapat menghemat waktu untuk pencarian, pencatatan, dan pemasukan data; mengurangi pekerjaan yang berulang-ulang atau dapat mengedit data dengan mudah; meningkatkan kinerja dalam rangka melakukan pelayanan dan menyelesaikan tugastugas dengan baik.