

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Setiap organisasi atau perusahaan memiliki sistem yang menangani serangkaian kegiatan yang terjadi sebagai alat untuk mencapai tujuan organisasi atau perusahaan tersebut.

Konsep dasar sistem ini menekankan sekumpulan elemen yang terdapat dalam suatu organisasi, perusahaan, aspek dan sudut pandang yang berbeda sesuai dengan keterangan fungsi dalam hal-hal yang berkaitan dengan sistem yang memiliki ciri dan karakteristik tertentu. Adapun uraian dari konsep dasar sistem akan dijelaskan dibawah ini.

2.1.1. Pengertian Sistem

Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir, saling beinteraksi, saling ketergantungan satu sama lain dan terpadu untuk mencapai tujuan tertentu.

Menurut Yakub dan Hisbanarto (2014:1) mengemukakan bahwa, “Sistem adalah kumpulan/grup/komponen-komponen apapun baik phisik yang saling berhubungan satu sama lain san bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu”. Sedangkan menurut Haryanto (2008:27) mendefinisikan sebagai, “Sistem adalah kumpulan elemen yang saling berinteraksi untuk mencapai satu tujuan tertentu”.

Berdasarkan pengertian para ahli di atas, dapat disimpulkan bahwa sistem merupakan serangkaian elemen yang berkaitan dengan jaringan kerja yang saling

berinteraksi, berhubungan, atau ketergantungan antara satu dengan yang lain bertujuan untuk menangani dan menyelesaikan serangkaian kegiatan yang terjadi secara rutin dan berulang-ulang.

A. Karakteristik Sistem

Dalam pembuatan suatu sistem, pembuat sistem harus memahami ciri-ciri atau karakteristik yang terdapat pada sekumpulan elemen yang ada, sebagai dasar pertimbangan dalam pembuatan sistem.

Menurut Ladjamudin (2013:3) mengemukakan bahwa “suatu sistem mempunyai karakteristik atau sifat-sifat tertentu”. Adapun karakteristik yang dimaksud adalah sebagai berikut (Ladjamudin, 2013:4):

1. Komponen Sistem (*Component*)

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran (*Objectives*)

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

9. Tujuan (*Goal*)

Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

B. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, adapun klasifikasi terhadap sistem tersebut adalah sebagai berikut (Ladjamudin, 2013:6):

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisik diartikan sebagai sistem yang nampak secara fisik sehingga setiap mahluk dapat melihatnya, misalnya sistem komputer.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem

robabilistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar.

2.1.2. Basis Data

Dalam pembuatan aplikasi, para pembuat aplikasi atau *programmer* menggunakan basis data yang digunakan untuk pengolahan data atau penataan *file-file* yang ada dan digunakan kembali sesuai dengan kebutuhan aplikasi tersebut.

A. Pengertian Basis Data

Basis data merupakan tempat pengolahan informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang terintegrasi.

Menurut Rosa dan Shalahuddin (2015:43), “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Sedangkan Menurut Yakub dan Hisbanarto (2014:25) menjelaskan, “basis data (*database*) merupakan kumpulan data yang saling berhubungan atau punya relasi”.

Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan data yang diolah menjadi informasi dan dapat digunakan kembali jika dibutuhkan.

B. Aplikasi Basis Data

Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang sering digunakan dalam pengolahan basis data yaitu *MySQL* dan *phpMyAdmin*.

1. *MySQL*

Salah satu aplikasi basis data yang sering digunakan untuk mengolah dan menata *file-file* yaitu *MySQL*. Menurut Manurung (2015:39) “*MySQL* merupakan turunan salah satu konsep utama basis data yang sudah ada sebelumnya *SQL* yaitu pngoperasian basisdata”. Sedangkan menurut Sibero (2013:97) “*MySQL* adalah suatu *RDBMS (Relational Database Manajement System)* yaitu aplikasi sistem yang menjalankan fungsi penglahan data”.

Penulis menyimpulkan bahwa *MySQL* merupakan aplikasi pengolahan *database* yang sering digunakan untuk membuat sebuah aplikasi yang memiliki data-data sebagai sumber pengolahannya.

2. *phpMyAdmin*

selain *MySQL*, aplikasi yang dapat mengolah basis data yaitu *phpMyAdmin*. *phpMyAdmin* memiliki fungsi yang sama dengan *MySQL*, namun untuk pengaksesan aplikasi menggunakan *browser*. Menurut Rahman (2013:21) “*phpMyAdmin* adalah sebuah *software* berbasis pemograman yang dipergunakan sebagai administrator *MySQL* melalui *browser (web)* yang digunakan untuk managemen *database*”.

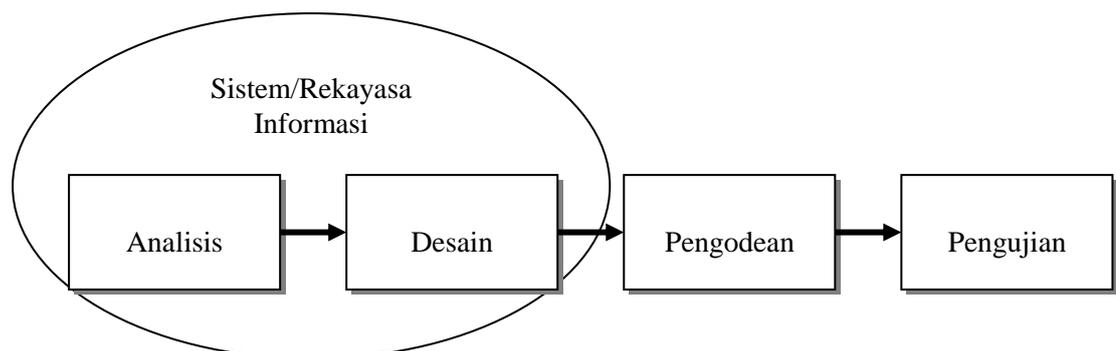
Sedangkan menurut Sibero (2013:376) mengemukakan bahwa :
phpMyAdmin digunakan untuk administrasi *database* pada *phpMyAdmin* seperti fitur pembuatan *database*, mengubah *database*, pembuatan tabel, mengubah tabel, menghapus tabel, menambah data, menampilkan data, mengubah data, menghapus data, membuat *view*, menghapus *view*, membuat *index* kolom dan menghapus *index* kolom.

Maka dari itu, penulis menyimpulkan bahwa *phpMyAdmin* merupakan aplikasi yang digunakan untuk melakukan pengolahan basis data dengan *browser* sebagai medianya.

2.1.3. Model Pengembangan Perangkat Lunak

Disiplin ilmu yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun (*waterfall*).

Menurut Rosa dan Shalahuddin (2015:28), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (*support*)”.



Sumber : Rosa dan Shalahuddin (2015:29)

Gambar II.1. Ilustrasi Model *Waterfall*

Adapun metode air terjun menurut Rosa dan Shalahuddin (2015:29) yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan 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 fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, 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 ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

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

5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan 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.

2.2. Teori Pendukung

Selain teori-teori para ahli yang berkaitan dengan konsep dasar sistem, penulis menggunakan teori dari para ahli lainnya untuk mendukung penulisan Tugas Akhir ini. Adapun teori yang mendukung penulisan Tugas Akhir yaitu:

2.2.1. Sistem Informasi

Sistem memiliki tujuan untuk menghasilkan informasi yang nantinya informasi tersebut akan digunakan sebagai dasar pertimbangan pengambilan keputusan suatu organisasi dalam perencanaan di masa mendatang.

Menurut Yakub dan Hisbanarto (2014:32), “Sistem informasi merupakan komponen dalam sebuah informasi yang berhubungan dengan proses penciptaan dan aliran informasi”. Sedangkan, menurut Ladjamudin (2013:33), Sistem informasi adalah “Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi”.

Dapat disimpulkan bahwa sistem informasi merupakan kompilasi yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang saling memiliki kaitan, hubungan atau interaksi untuk melakukan pengolahan data menjadi informasi dalam suatu organisasi sebagai dasar dalam pengambilan keputusan untuk menentukan langkah di masa mendatang.

2.2.2. Jasa Pengiriman Barang

Menurut PER-178/PJ/2006 mendefinisikan jasa pengiriman barang (*Freight Forwarding Service*) yaitu yang mengacu pada keputusan menteri Perhubungan No.KM/10 Tahun 1998 tentang jasa pengurusan transportasi. Berdasarkan SK Menhub tersebut, yang dimaksud dengan jasa pengiriman barang

(*Freight Forwarding Service*) adalah” usaha yang ditunjukan untuk mewakili kepentingan pemilik barang, untuk mengurus semua kegiatan yang diperlukan bagi terlaksananya pengiriman dan pengiriman barang melalui transportasi darat, laut dan udara yang dapat mencakup kegiatan penerimaan, penyimpanan, sortasi, pengepakan, penandaan pengukuran, penimbangan, pengurusan penyelesaian dokumen, penerbit dokumen angkutan, klaim asuransi, atas pengiriman barang serta, penyelesaian tagihan dan biaya-biaya lainnya berkenan dengan pengiriman barang-barang tersebut sampai dengan diterimanya barang oleh yang berhak menerimanya.

Dari definisi tersebut dapat disimpulkan bahwa jasa pengiriman barang adalah seluruh kegiatan yang perlu dilakukan untuk mengirim barang sampai pada orang yang berhak menerimanya.

2.2.3. Diagram Alir Data (DAD)

Diagram alir data sering digunakan untuk menggambarkan prosedur sistem yang sedang berjalan di suatu organisasi atau perusahaan. Diagram alir data juga sering disebut sebagai *Data Flow Diagram* (DFD). Menurut Rosa dan Shalahuddin (2015:70) “DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek”. Sedangkan menurut Jogiyanto (2009:701), “DFD adalah diagram yang menggunakan notasi simbol untuk menggambarkan arus data sistem”.

Berdasarkan teori yang dilanturkan oleh para ahli di atas, penulis menarik sebuah kesimpulan yang berisikan tentang pengertian dari diagram alir data (DAD) yaitu gambaran akan arus data sistem yang saling berkaitan dengan menggunakan notasi simbol sebagai perwakilan dari sistem yang terjadi.

Simbol atau lambang yang digunakan dalam membuat diagram alir data yang lazim digunakan, menurut Rosa dan Shalahuddin (2015:71) DAD terdiri dari empat buah simbol yaitu:

1. Entitas/Lingkungan Luar (*External Entity*)

Simbol ini digunakan untuk menggambarkan asal atau tujuan data, menunjukkan entitas atau kesatuan yang berhubungan dengan sistem, dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan input atau menerima input dari sistem atau keduanya digunakan dengan simbol empat persegi panjang.

2. Proses (*Process*)

Simbol ini digunakan untuk proses pengolahan atau transformasi data, menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dan hasil suatu data yang masuk kedalam proses untuk menghasilkan arus data yang akan keluar dari proses, digambarkan dengan simbol lingkaran.

3. Arus Data (*Data Flow*)

Simbol ini digunakan untuk menggambarkan aliran data yang berjalan, menunjukkan arus data yang berupa masukan untuk sistem atau hasil dari proses sistem yang mengalir diantara proses (*process*), simpanan data (*data store*) dan entitas (*external entity*) digambarkan dengan arah panah.

4. Simpanan Data (*Data Store*)

Simbol ini digunakan untuk menggambarkan *data flow* yang sudah disimpan, menunjukkan suatu tempat penyimpanan data yang dapat berupa suatu file di

sistem komputer, arsip atau catatan manual, tabel acuan dan lain-lain digambarkan dengan sepasang garis horizontal.

Tahap pembuatan Diagram Alir Data (DAD) (Rosa dan Shalahuddin, 2015:72) dibagi menjadi tiga tingkatan konstruksi Diagram Alir Data yaitu:

1. Diagram Konteks

Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut untuk menggambarkan sistem secara global dari keseluruhan sistem yang ada.

2. Diagram Nol

Diagram ini dibuat untuk menggambarkan tahap-tahap proses yang akan ada didalam konteks atau penjabaran secara rinci.

3. Diagram Detail

Diagram ini dibuat untuk menggambarkan arus data secara lebih detail dan terperinci dari tahapan proses yang ada dalam diagram.

2.2.4. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut.

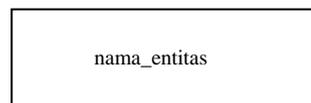
Menurut Rosa dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”. Sedangkan menurut Ladjamudin (2013:142) “*Entity Relationship Diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

Dapat disimpulkan bahwa *Entity Relationship Diagram* (ERD) merupakan teknik penggambaran model jaringan suatu basis data dengan susunan data yang disimpan dalam sistem secara abstrak dengan menggunakan notasi dan simbol.

Simbol-simbol atau komponen-komponen yang digunakan dalam penggambaran *Entity Relationship Diagram* (ERD) yaitu:

1. Entitas (*entity*)

Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

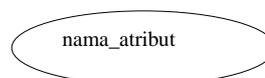


Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.2. Entitas

2. Atribut (*Attribute*)

Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah *entitas* yang perlu disimpan sebagai basis data.

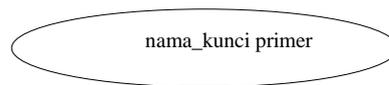


Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.3.Atribut

3. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id.

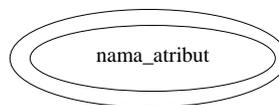


Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.4.Atribut Kunci Primer

4. Atribut multivalai (*multivalue*)

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

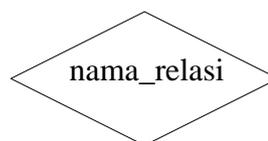


Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.5.Atribut Multinilai

5. Relasi (*relation*)

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua entity yang keterangannya perlu disimpan dalam basis data.



Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.6.Relasi

6. Asosiasi/*association*

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiciplicity*.



Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.7.Asosiasi / Association

Relasi yang terbentuk antar entitas dalam *Entity Relationship Diagram* (ERD) dinamakan derajat *relationship*. Menurut Ladjamudin (2013:144) mengemukakan bahwa “Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *Relationship*”.

Derajat *relationship* yang sering dipakai didalam ERD Ladjamudin (2013:145) adalah sebagai berikut:

1. *Unary Relationship*

Unary relationship adalah model yang terjadi diantara entity yang berasal dari *entity* set yang sama. Sering juga disebut sebagai *recursive relationship* atau *reflective relationship*.

2. *Binary Relationship*

Adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

3. *Ternary Relationship*

Ternary Relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas sepihak.

2.2.5. Struktur Navigasi

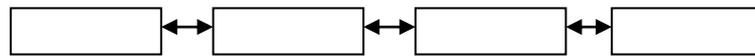
Struktur navigasi berfungsi untuk menggambarkan hubungan rantai kerja yang terjadi dalam aplikasi. Struktur navigasi memberikan gambaran terhadap sistem kerja aplikasi tersebut.

Struktur navigasi menurut Vaughan (2007:367) “merupakan peta navigasi yang menggambarkan koneksi atau hubungan di antara bermacam area ini dan membantu mengorganisasi isi dan pesan”.

Struktur navigasi dikelompokkan menjadi 4 struktur yang berbeda, Vaughan (2007:367) yaitu:

1. Linier

Pengguna melakukan navigasi secara berurutan, dari *frame* atau *bite* informasi satu ke yang lainnya

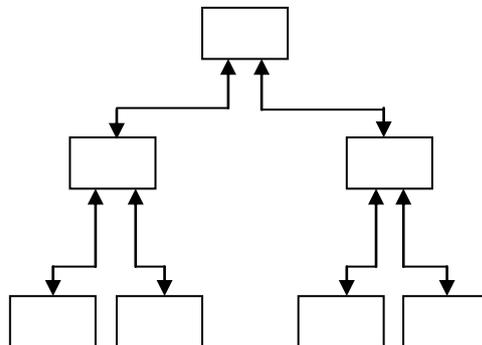


Sumber : Vaughan (2007:367)

Gambar II.8. Contoh Struktur Linier

2. Hierarkis

Disebut juga linier dengan percabangan, karena pengguna melakukan navigasi sepanjang cabang pohon struktur yang terbentuk oleh *natural logic* dari isi.

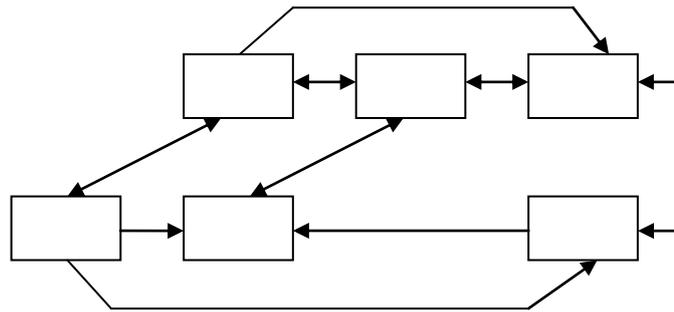


Sumber : Vaughan (2007:368)

Gambar II.9. Struktur Hirarki

3. Non Linier

Pengguna melakukan navigasi dengan bebas melalui isi proyek, tidak terikat dengan rute yang telah ditetapkan sebelumnya.

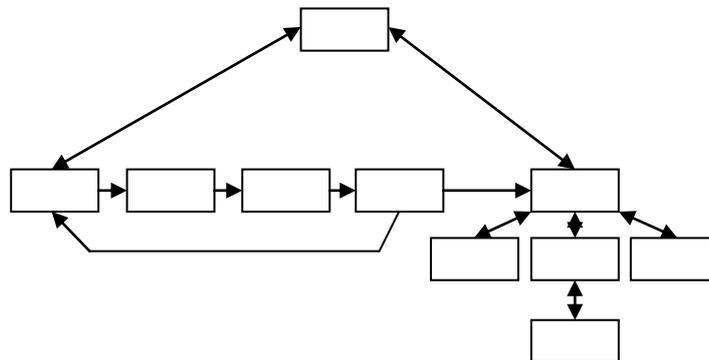


Sumber : Vaughan (2007:368)

Gambar II.10. Struktur Non Linier

4. Komposit

Pengguna melakukan navigasi dengan bebas secara nonlinier, tetapi terkadang dibatasi oleh presentasi linier film atau informasi kritis dan atau pada data yang paling terorganisasi secara logis dalam suatu hierarki.



Sumber : Vaughan (2007:368)

Gambar II.11. Struktur Komposit

2.2.6. Pengkodean

Pengkodean merupakan teknik untuk merancang kode suatu program dimana kode yang dibuat tersusun dari aturan-aturan yang dirancang berdasarkan elemen-elemen tertentu yang digunakan oleh perancang.

Menurut Jogiyanto (2009:384), “struktur kode adalah suatu bentuk struktur yang berfungsi untuk mengklasifikasikan data, memasukkan data ke

dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya”.

Beberapa kemungkinan susunan digit (angka), huruf dan karakter-karakter khusus dapat dirancang ke dalam bentuk kode. dalam merancang suatu kode harus diperhatikan beberapa hal (Jogiyanto, 2009:384) yaitu:

1. Mudah diingat

Supaya kode mudah diingat maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

2. Unik

Kode harus unik untuk masing-masing item yang diwakilinya.

3. Fleksibel

Kode harus fleksibel sehingga memungkinkan perubahan-perubahan atau penambahan *item* baru dapat tetap diwakili oleh kodenya masing-masing.

4. Efisien

Kode harus sesingkat mungkin, selain mudah diingat juga akan efisien apabila direkam dan disimpan di luar komputer.

5. Konsisten

Kode harus konsisten dengan kode yang telah digunakan.

6. Harus distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Hindari penggunaan spasi

Spasi dalam kode sebaiknya dihindarkan karena dapat menyebabkan kesalahan dalam menggunakannya.

8. Hindari karakter yang mirip

Karakter-karakter yang hampir mirip bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode.

9. Panjang kode harus sama

Masing-masing kode yang sejenis harus memiliki panjang kode yang sama.