

## B A B III

# M E T O D O L O G I

### 3.1 Pendekatan Terstruktur

Karena banyak terjadi permasalahan-permasalahan di pendekatan klasik, maka kebutuhan akan pendekatan pengembangan sistem yang lebih baik mulai terasa dibutuhkan. Sayangnya sampai sekarang masih banyak orang yang tidak menyadari bahwa hanya dengan mengikuti tahapan di life cycle penulis tidak akan membuat pengembangan sistem informasi menjadi berhasil. Oleh karena itu diperlukan suatu pendekatan pengembangan sistem yang baru yang dilengkapi dengan beberapa alat dan teknik supaya membuatnya berhasil.

Pendekatan ini yang dimulai dari awal tahun 1970 disebut dengan pendekatan terstruktur (*structured approach*). Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang strukturnya didefinisikan dengan baik dan jelas.

Konsep pengembangan sistem terstruktur bukan merupakan konsep yang baru. Teknik perakitan di pabrik-pabrik dan perancangan sirkuit untuk alat-alat elektronik adalah dua contoh dari konsep ini yang banyak digunakan di industri-industri. Konsep ini memang relatif masih baru digunakan dalam mengembangkan sistem informasi untuk dihasilkan produk sistem yang memuaskan pemakainya. Melalui pendekatan terstruktur, permasalahan-

permasalahan yang kompleks di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat pada waktunya, sesuai dengan anggaran biaya pengembangannya, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

### 3.2 Daur Kehidupan Klasik

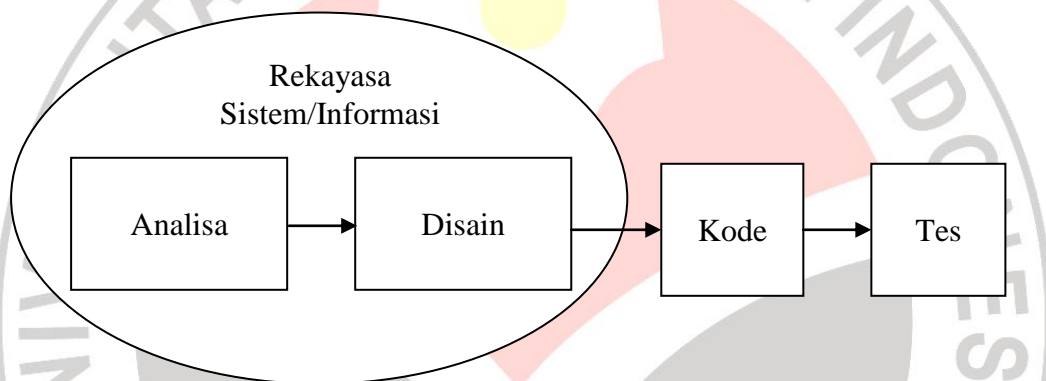
Salah satu dari sekian banyak metode perancangan adalah *Daur Kehidupan Klasik*. Paradigma ini kadang disebut sebagai *waterfall model*. Metode ini membutuhkan pendekatan yang sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem dan kemajuan melalui analisis, disain, *coding*, *testing* dan pemeliharaan ( Roger S. Presman, 1997, h.31-32 ).

Pemodelan ini menyangkut aktifitas berikut :

1. Pemodelan dan Rekayasa Sistem/infomasi. Karena perangkat lunak adalah bagian dari sistem yang lebih besar, pekerjaan ini dimulai dari pembentukan kebutuhan-kebutuhan dari semua elemen sistem dan mengalokasikan suatu subset ke dalam pembentukan perangkat lunak.
2. Analisis kebutuhan Perangkat Lunak. Proses pengumpulan kebutuhan diintensifkan ke perangkat lunak. Harus dapat dibentuk domain informasi, fungsi yang dibutuhkan, performansi dan antarmuka.
3. Disain. Ada empat atribut untuk program : struktur data, arsitektur perangkat lunak, prosedurdetil, dan karakteristik antarmuka. Proses disain mengubah

kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program.

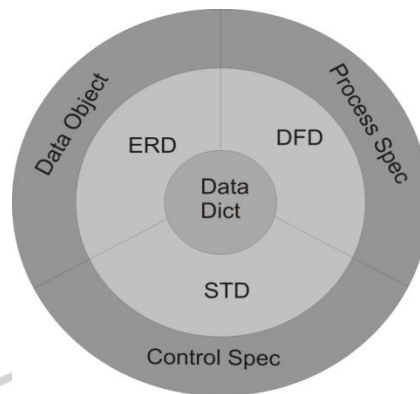
4. *Coding*. Disain harus diubah menjadi bentuk yang dimengerti mesin yaitu penulisan program.
5. *Testing*. Setelah coding selesai, dan program dapat berjalan, testing dapat dimulai. Testing mengfokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan.



Gambar 3.1 Model Linear Sequential

### 3.3 Elements dari Analysis Model

Di dalam subbab ini, model analisa yang umum terdiri dari satu diagram hubungan-entitas (model data), suatu diagram alir data (model fungsional), dan diagram transisi suatu keadaan (model tingkah laku). Uraian-uraian bagaimana caranya membangun masing-masing elemen.



Gambar 3.2 Struktur dari model analisa

### 3.3.1 Pemodelan Data

Sekumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik (makna) data dan batasan data. Model Data bisa disebut juga data logic. Dalam perancangan basis data, model data secara umum dapat dibagi menjadi beberapa kelompok yaitu model data berbasis objek, model data berbasis record, model data fisik, dan model data konseptual.

#### Entity Relationship Diagram (ERD)

Model *Entity-Relationship* berisi komponen-komponen dari suatu himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara *entity-entity* yang ada dengan atribut-atributnya. Selain itu juga bisa menggambarkan hubungan yang ada dalam pengolahan data, seperti hubungan *many to many*, *one to many*, atau *one to one*.

## Notasi ERD

ERD menggunakan beberapa simbol untuk menggambarkan struktur dan hubungan antar entitas. Simbol-simbol tersebut adalah :

### 1. Simbol Entitas

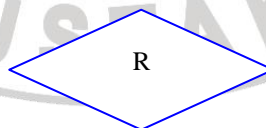
Menurut (Pohan dan Bahri, 1997), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> ) Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.



Gambar 3.3 Simbol Entitas

### 2. Relasi/Hubungan

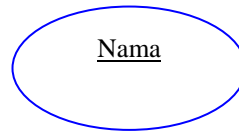
Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda (Fathansyah, 1999). diperoleh (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> )



Gambar 3.4 Simbol Relasi

### 3. Simbol Atribut


Atribut adalah properti yang digunakan untuk menguraikan suatu entitas



Gambar 3.5 Simbol Atribut

#### 4. Penghubung

Notasi simbolik yang menggambarkan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut dalam ERD adalah garis.



Gambar 3.6 Simbol Penghubung

#### 5. Kardinalitas Relasi

Menurut ( Fathansyah, 1999 ), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> ) kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu ke satu, 1 dan N untuk relasi satu ke banyak, N dan N untuk relasi banyak ke banyak). Jadi kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.



Gambar 3.7 Kardinalitas Relasi

## Tahapan Pembuatan ERD

Menurut ( Fathansyah, 1999 ), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> ) langkah-langkah untuk memodelkan data dengan ERD yaitu :

1. Menentukan himpunan entitas yang terlibat di dalam sistem.
2. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
3. Menentukan himpunan relasi di antara himpunan entitas yang ada serta *foreign-key*-nya.
4. Menentukan derajat/ kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).

### 3.3.2 Model Fungsi

Bagian ini menguraikan representasi grafis dari suatu fungsi dalam lingkup yang ditetapkan dan juga menjelaskan fungsi dan proses. Fungsi pemodelan berkonsentrasi pada perspektif yang menggambarkan proses dinamis. Utama dalam konsep ini adalah contoh perspektif proses, hal ini dapat menjadi salah satu fungsi, transformasi, kegiatan, tindakan, tugas lain-lain baik yang dikenal contoh pemodelan bahasa Mempekerjakan perspektif ini adalah data flow diagram. Menggunakan empat perspektif simbol untuk menjelaskan suatu proses yaitu *proses, store, flow, external entity* dan *eksternal entity*. Sekarang, dengan simbol ini, sebuah proses dapat digambarkan sebagai simbol proses DFD ( data flow diagram).

### 3.3.2.1 Diagram Konteks (*Contex Diagram*)

Menurut ( Pohan dan Bahri, 1997), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> ) *Context Diagram* (CD) atau Diagram Konteks adalah kasus khusus DFD (bagian dari DFD yang berfungsi memetakan model lingkaran), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

Berdasarkan definisi di atas Diagram Konteks adalah aliran yang memodelkan hubungan antara sistem dengan entitas luar di luar sistem, yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan siklus. Aliran dalam diagram konteks memodelkan masukan ke sistem dan sistem harus memberikan respon untuk menggambarkan transportasi antar sistem dan entitas luar.

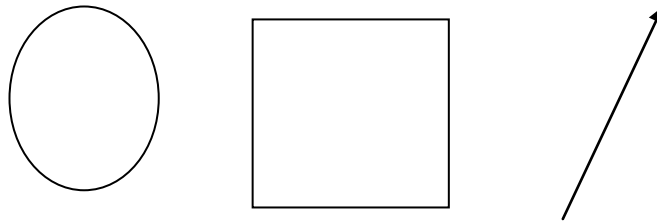
Diagram Konteks menyoroti sejumlah karakteristik penting sistem yaitu:

1. Kelompok pemakai (*User*), organisasi atau sistem lain, dimana sistem kita melakukan komunikasi yang disebut juga sebagai terminator.
2. Data masuk (*input*), data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar (*output*), data yang dihasilkan sistem kita dan diberikan ke dunia luar.
4. Penyimpanan data (*data store*), yang digunakan secara bersama antara sistem kita dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya.



5. Batasan antara sistem dan lingkungan (*rest of the world*).





Gambar 3.8 Simbol Diagram Konteks

### 3.3.2.2 Data Flow Diagram (DFD)

DFD merupakan suatu model yang menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran serta penyimpanan dan aliran data. Sedangkan menurut (McLeod, 1996), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc>) DFD adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan. DFD sering digunakan untuk menggambarkan sistem yang telah ada maupun sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik di mana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan dokumentasi dari sistem yang baik.

#### DFD Level ke-n

Menurut (Burhan.R,2003), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/B>

AB%20II.doc) *Data Flow Diagram* (DFD) adalah salah satu alat dalam perancangan sistem yang menggunakan simbol-simbol untuk menggambarkan aliran data melalui serangkaian proses yang saling berhubungan. *Data Flow Diagram* (DFD) memiliki tingkatan level mulai dari Level 0 sampai tingkatan Level ke-n. *Data Flow Diagram* (DFD) Level n merupakan suatu diagram level rendah yang berfungsi menjabarkan secara detail diagram konteks (diagram level tinggi) pada suatu sistem. Huruf n menggambarkan diagram arus data level tertentu dan proses setiap lingkaran.

Level terendah dalam DFD hanya mempunyai sebuah proses yang memodelkan seluruh sistem. Pemberian nomor pada setiap proses dalam DFD berguna untuk memudahkan penurunan DFD pada level yang lebih rendah. Penurunan mengacu pada aturan tertentu, yaitu :

1. Setiap penurunan ke level yang lebih rendah harus mampu mempresentasikan proses tersebut dalam spesifikasi proses yang jelas. Jika belum jelas harus diturunkan ke level yang lebih rendah.
2. Penurunan dilakukan jika diperlukan, sampai pada tingkatan primitif (tidak dapat diturunkan lagi)
3. Tidak semua bagian sistem harus diturunkan dengan jumlah level yang sama, karena yang kompleks bisa diturunkan dan yang sederhana mungkin tidak perlu diturunkan. Tidak semua proses dalam level yang sama mempunyai derajat kompleksitas yang sama juga.
4. Mengkonfirmasi DFD yang telah dibuat dengan cara *top down*.

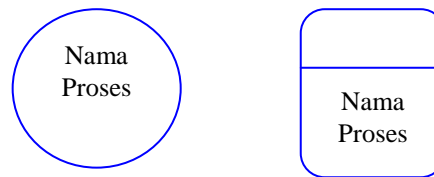
5. Aliran data yang masuk dan keluar pada suatu proses di level-x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1. Level x+1 mendefinisikan sub proses pada level-x.
6. Penyimpanan yang muncul pada level-x harus didefinisikan kembali pada level-x.+1, sedangkan penyimpanan yang muncul pada level-x tidak harus muncul pada level-x.+1 jika penyimpanan tersebut bersifat lokal.
7. Ketika mulai menurunkan DFD level tertinggi, coba untuk mengidentifikasi *external events* dimana sistem harus memberikan respon. *External events* dalam hal ini berarti suatu kejadian yang berkaitan dengan pengolahan data diluar sistem dan menyebabkan sistem membentuk respon ( Pohan dan Bahri, 1997) diperoleh dari diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BAB%20II.doc> )

### **Komponen-komponen DFD**

Komponen-komponen yang digunakan dalam DFD antara lain :

#### **1. Proses**

Proses menunjukkan transformasi dari masukan menjadi keluaran. Sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya. Proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer. Suatu proses disimbolkan dengan lingkaran (*Gane-Sarson*) atau *output* persegi panjang tegak dengan sudut-sudutnya tumpul (*De Marco-Yourdan*).



Gambar 3.9. Proses

Suatu proses terjadi karena adanya arus data yang masuk dan hasil dari proses adalah juga merupakan arus data yang mengalir. Suatu proses harus menerima arus data dan menghasilkan arus data.

Menurut ( Pohan dan Bahri, 1997 ), diperoleh dari (<http://www.mhs.stiki.ac.id/badboy/dion/TUGAS%20MANDIRI/Seminar/Text%20A5/BA5%20II.doc> ) Hal yang harus dihindari dalam menggambarkan suatu proses adalah :

1. Proses yang mempunyai masukan tetapi tidak memiliki keluaran, kesalahan ini disebut *Black Hole*.
2. Proses yang menghasilkan keluaran tetapi tidak mendapatkan masukan/ proses menghasilkan keluaran lengkap dengan data terbatas, kesalahan ini disebut dengan *Miracle*.
3. Proses yang menghasilkan keluaran lengkap tetapi masukan dengan data terbatas/ masukan yang kurang lengkap, kesalahan ini disebut dengan *Gray Hole*.

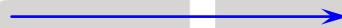
Setiap proses harus diberi penjelasan yang lebih lengkap, meliputi sebagai berikut :

1. Identifikasi Proses

2. Berupa suatu angka yang menunjukkan nomor acuan dari proses dan ditulis pada bagian atas di simbol proses.
3. Nama Proses
4. Menunjukkan apa yang dikerjakan oleh proses, nama harus jelas dan lengkap menggambarkan kegiatan prosesnya dan diletakkan di bawah identifikasi proses.

## 2. Aliran Data (*Data Flow*)

Disimbolkan oleh suatu anak panah yang menunjuk ke data proses. Aliran data ini mengalir diantara proses, simpanan data dan entitas luar. Aliran data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari suatu proses sistem. Aliran data yang digambarkan sebagai anak panah dengan dua ujung menggambarkan terjadinya dialog. Nama *data flow* sebaiknya mengandung kata sifat dan kata keterangan yang tepat untuk menggambarkan status dari *flow*.

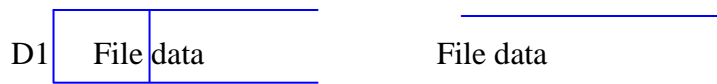


Gambar 3.10. Simbol Aliran Data

## 3. Penyimpanan Data (*Data Store*)

Komponen ini digunakan untuk menyimpan data hasil proses maupun menyediakan data untuk diproses pada waktu yang lain. Dengan kata lain komponen ini digunakan untuk memodelkan kumpulan data atau paket data yang berupa file atau database di sistem komputer, arsip atau catatan manual, kotak tempat data, tabel acuan manual dan agenda atau buku. Data store ini

digambarkan dengan persegi panjang yang sisi kanannya terbuka (*Gane-Sarson*) atau dua garis sejajar yang kedua ujungnya terbuka (*De-Marco Yourdan*). .



Gambar 3.11. Simbol Data Store

#### 4. Kesatuan Luar/ Entitas Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Dalam hal ini, sejumlah input dapat menghasilkan hanya satu output ataupun sebaliknya. Entitas luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang atau kelompok orang. Entitas luar dapat disimbolkan dengan kotak persegi empat.



Gambar 3.12 Simbol Entitas Luar

#### Syarat Pembuatan

Pedoman bagaimana menggambar DFD baik PDFD ataupun LDFD adalah sebagai berikut ini :

1. Identifikasikan terlebih dahulu semua kesatuan luar (external entity) yang terlibat di sistem. Kesatuan luar ini merupakan kesatuan (entity) di luar sistem, karena di luar bagian pengolahan data (sistem informasi). Kesatuan luar ini merupakan sumber arus data ke sistem informasi

serta tujuan penerima arus data hasil dari proses sistem informasi, sehingga merupakan kesatuan di luar sistem informasi.

2. Identifikasikan semua input dan output yang terlibat dengan kesatuan luar.

Tabel 3.1 Identifikasi semua input dan output

Kesatuan Luar	Input	Output
User	Data User	Draft Jurnal
Jurnal	Data Jurnal	-
Editor	Data Editor, Draft Jurnal	Edited Jurnal
Admin	Data Admin	Data Jurnal, User, dan Editor

3. Gambarlah terlebih dahulu suatu diagram konteks (context diagram). DFD merupakan alat untuk structured analysis. Pendekatan terstruktur ini mencoba untuk menggambarkan sistem pertama kali secara garis besar (disebut dengan top level) dan memecah-mecahnya menjadi bagian yang lebih terinci (disebut dengan lower level). DFD yang pertama kali digambar adalah level teratas (top level) dan diagram ini disebut context diagram. Dari context diagram ini kemudian akan digambar dengan lebih terinci lagi yang disebut dengan overview diagram (level 0). Tiap-tiap proses di overview diagram akan digambar secara lebih terinci lagi dan disebut dengan level 1. Tiap-tiap proses di level 1 akan digambar kembali dengan lebih terinci lagi dan disebut dengan level 2 dan seterusnya sampai tiap-tiap proses tidak dapat digambar



67 lebih terinci lagi.

### 3.3.2.3 Kamus Data (KD)

Kamus Data tidak menggunakan notasi grafis sebagaimana halnya DAD, tetapi porsinya dalam memodelkan sistem tidak perlu diragukan lagi (sebuah model tidak lengkap tanpa Kamus Data). Kamus Data juga mempunyai fungsi yang sama dalam pemodelan sistem. Selain itu Kamus Data berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detil, kamus data mereorganisasi semua elemen data yang digunakan dalam sistem dengan presisi yang sedemikian rupa sehingga pemakai dan penganalisis sistem memiliki dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk :

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan
3. Menentukan muatan data yang disimpan dalam file-file
4. Mengembangkan logika untuk proses-proses diagram aliran data

Kamus Data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, Kamus Data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data

yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, Kamus Data digunakan untuk merancang input, merancang laporan-laporan dan database. Kamus data dibuat berdasarkan arus data yang ada di DAD. Kamus Data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos)
3. Mendeskripsikan komposisi penyimpanan data
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam entity-relationship diagram).

Kamus data dibuat dengan memperhatikan dan menggambarkan muatan aliran data, simpanan dataa dan proses-proses seperti pada gambar di atas. Setiap simpanan data dan aliran data bisa ditetapkan dan kemudian diperluas sampai mencakup detail-detail elemen yang dimuatnya. Logika dari setiap proses ini bisa digambarkan dengan menggunakan data yang mengalir menuju dan keluar dari proses tersebut.

### **Elemen-Elemen Data**

Kamus data harus dapat mencerminkan keterangan yang jelas tentang

data yang akan dicatat. Untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut:

1. **Nama arus data**, karena kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di Kamus Data.
2. **Alias**, alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya. Misalnya bagian pembuat faktur dan langganan menyebut bukti penjualan sebagai faktur, sedangkan bagian gudang menyebutnya sebagai tembusan permintaan persediaan. Baik faktur dan tembusan permintaan persediaan ini mempunyai struktur data yang sama, tetapi mempunyai struktur yang berbeda.
3. **Bentuk data**, telah diketahui bahwa arus data dapat mengalir :
  - a. Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.
  - b. Hasil dari suatu proses ke kesatuan luar, data yang mengalir ini biasanya terdapat di media laporan atau query tampilan layar atau dokumen hasil cetakan komputer;
  - c. Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variabel atau parameter yang dibutuhkan oleh proses penerimanya;
  - d. Hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel.

e. Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu field (item data).

Dengan demikian bentuk dari data yang mengalir dapat berupa: dokumen dasar atau formulir, dokumen hasil cetakan komputer, laporan tercetak, tampilan di layar monitor, variabel, parameter, field.

4. **Arus data**, arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan ini perlu dicatat di Kamus Data agar mudah mencari arus data di DAD.
5. **Penjelasan**, Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di Kamus Data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut. Misalnya nama dari arus data adalah Tembusan Permintaan Persediaan, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang.
6. **Periode**, periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di Kamus Data karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.
7. **Volume**, volume yang perlu dicatat di Kamus Data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak. Volume ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan,

kapasitas dan jumlah dari alat input, alat pemroses dan alat output.

**8. Struktur data**, struktur data menunjukkan arus data yang dicatat di Kamus

Data terdiri dari item-item data apa saja.



## Simbol Kamus Data

Pada kebanyakan sistem dalam dunia nyata (dimana kita bekerja), kadang-kadang elemen data terlalu kompleks untuk didefinisikan. Kekompleksan tersebut seharusnya diuraikan melalalui sejumlah elemen data yang lebih sederhana. Kemudian elemen data yang lebih sederhana tersebut didefinisikan kembali hingga nilai dan satuan yang relevan (yang sifatnya elementer). Pendefinisian tersebut menggunakan notasi yang umumnya digunakan dalam menganalisis sistem dengan menggunakan sejumlah symbol yaitu:

Tabel 3.2 Simbol Kamus Data

No	Simbol	Uraian
1	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2	+	Dan
3	{ }	Menunjukkan elemen-elemen repetitive, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaankeadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
4	[ ]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada seara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)

5		Pemisah sejumlah alternatif pilihan antara symbol [ ]
6	@	Identifikasi atribut kunci
7	**	Komentar
8	()	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file.

#### 3.3.2.4 Proses Spesifikasi

Spesifikasi proses, dalam hal ini deskripsi dari apa yang terjadi di dalam setiap level paling dasar, bulatan konvensional pada diagram alir data. Kegunaan proses spesifikasi cukup penting untuk ke depannya, hal ini mendefinisikan apa yang harus dikerjakan untuk merubah input menjadi output. Hal tersebut merupakan gambaran detail kebijakan bisnis user yang dibawa oleh setiap lingkaran.

Spesifikasi Proses menggambarkan kejadian di dalam setiap *bubble* pada level terbawah pada data flow diagram. Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah input menjadi output (*Edward Yourdon, Modern Structured Analysis, 1999, h.203*).

Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD. Model ini berfungsi mendeskripsikan apa yang dilakukan ketika masukan ditransformasi menjadi keluaran.

Ada berbagai macam tools yang dapat kita gunakan untuk menghasilkan suatu spesifikasi proses: tabel keputusan, Bahasa Inggris terstruktur, pre/post condition, flowcharts, diagram Nassishneiderman, dan lain sebagainya.

Sedangkan kebanyakan analisis sistem mengarah ke Bahasa Inggris terstruktur, anda harus ingat bahwa setiap metode dapat digunakan, selama hal tersebut memuaskan dua keadaan penting:

1. Spesifikasi proses harus ditampilkan dalam suatu bentuk/form yang dapat diverifikasi oleh user dan sistem analis. Kondisi ini tepat untuk alasan ini, dimana kita mengurangi paparan Bahasa Inggris sebagai sebuah alat spesifikasi: hal ini dikenal sebagai ambigu, khususnya apabila menggambarkan tindakan (keputusan) alternatif dan tindakan berulang (loops). Secara alami, hal tersebut juga cenderung menyebabkan kebingungan yang sangat ketika mengekspresikan bagian-bagian kondisi Boolean (misalkan kombinasi dari operator Boolean AND, OR dan NOT).
2. Spesifikasi proses harus ditampilkan dalam suatu bentuk/form yang dapat mengkomunikasikan secara efektif berbagai keterlibatan berbagai latar belakang pendengar. Sedangkan hal tersebut akan menjadi tipe dari analisis sistem yang menuliskan spesifikasi proses, hal tersebut biasanya menjadi bermacam-macam pendengar dari para pengguna, manager, auditor, personil quality assurance, dan lainnya yang membaca spesifikasi proses. Suatu proses spesifikasi diharapkan dapat ditampilkan dalam perhitungan yang dapat diprediksi, atau dalam pascal, atau dalam pendekatan format diagram seperti software use-it; tetapi jika komunitas user menolak untuk melihat pada beberapa spesifikasi, mereka adalah tidak berharga.



Spesifikasi proses berfungsi mendeskripsikan tahapan yang dilakukan untuk mentransformasikan input menjadi output. Bentuk penyajian spesifikasi proses adalah sebagai berikut :

Tabel 3.3 penyajian spesifikasi proses

<b>No Proses</b> : Menyatakan nomor proses	
<b>Nama Proses</b> : Menyatakan nama proses	
<b>Deskripsi</b> : Penjelasan tujuan proses	
<b>SOURCE</b>	<b>DATA</b>
( Menyatakan sumber data input menuju proses )	( Menyatakan isi data yang masuk ke proses )
<b>DESTINATION</b>	<b>DATA</b>
( Menyatakan tujuan data output dari proses )	( Menyatakan isi data yang keluar dari proses )
<b>LOGIKA PROSES</b> (menyatakan algoritma dari proses)	

Tujuan membuat spesifikasi proses

1. Mengurangi makna ganda dari proses tersebut
2. Agar memperoleh deskripsi yang tepat mengenai apa yang dicapai
3. Untuk memvalidasi sistem disain

Logika Proses

1. Deskripsi Proses bisa dengan form atau alat bantu CASE
2. *Process descriptions may exist on a form or within a CASE tool repository*

3. Logika proses dapat dipresentasikan dengan

- i) *Structured English*
- ii) *A decision table*
- iii) *A decision tree*
- iv) *A formula*
- v) *Any combination of the above*

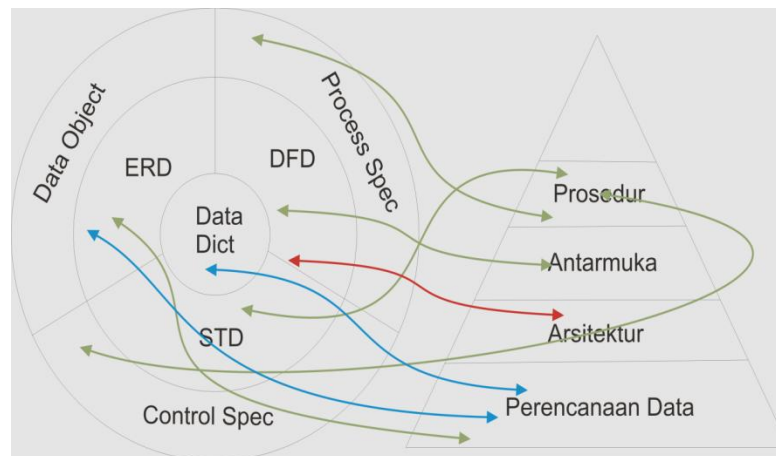
### **Format Spesifikasi Proses**

1. Spesifikasi proses menghubungkan proses ke diagram aliran data dan kamus data
2. Informasi berikut yang perlu dimasukkan
  - i. Jumlah proses, yang harus dicocokkan identitas proyek pada diagram aliran data.
  - ii. Nama proses, harus sama dengan nama pada simbol proses pada diagram aliran data.
  - iii. Deskripsi singkat mengenai apa yang dicapai proses tersebut
  - iv. Daftar data aliran masukan
  - v. Aliran data keluaran
  - vi. Indikasi mengenai jenis proses : *batch*, *online* atau manual
  - vii. Jika proses-proses menggunakan kode yang sudah tertulis sebelumnya, masukkan nama sub program atau fungsi yang memuat kode tersebut

- viii. Deskripsi logika proses yang menyatakan kebijakan dan aturan – aturan bisnis dalam bahasa sehari-hari, bukan bahasa komputer pseudocode.
- ix. Jika tidak ada cukup ruang pada formulir tersebut untuk deskripsi dalam bahasa inggris terstruktur yang lengkap atau jika ada sebuah tabel keputusan atau pohon keputusan yang menggambarkan logika tersebut, masukkan nama pohon atau tabel yang sesuai
- x. Buatlah daftar hal-hal yang belum terselesaikan, bagian logika yang belum lengkap, atau hal-hal lainnya, yang membentuk dasar pertanyaan-pertanyaan yang digunakan untuk wawancara tindak lanjut

### **3.4 Design ( Perencanaan )**

Pada subbab ini menerangkan desain yang terfokus pada empat bidang perhatian utama: data, arsitektur, komponen antarmuka, prosedur. Selama proses perancangan perangkat lunak yang menjadi spesifikasi desain model yang menjelaskan rincian dari struktur data, arsitektur sistem, antarmuka, dan komponen. Setiap desain produk berkualitas untuk dikaji sebelum pindah ke tahap berikutnya dari pengembangan perangkat lunak.



Gambar 3.13 Model Analisis dan Model Desain

### 3.4.1 Perancangan Data

Transformasi model data yang dihasilkan oleh proses analisis menjadi struktur data yang dibutuhkan pada saat implementasi. Petunjuk Teknis perancangan data meliputi menerapkan prinsip-prinsip analisis sistematik ( pada tahap analisis ), mengidentifikasi semua struktur data dan prosedur yang akan digunakan untuk mengakses data tersebut, *me-refine* isi data dictionary, menunda perancangan data yang “low-level” samapi di akhir-akhir proses perancangan, merepresentasikan struktur data sedemikian rupa sehingga hanya modul yang menggunakan data tersebut yang dapat mengaksesnya, merepresentasikan struktur data sedemikian rupa sehingga hanya modul yang menggunakan data tersebut yang dapat mengaksesnya, membangun pustaka untuk struktur data dan prosedur yang sering digunakan, dan mendukung spesifikasi dan realisasi ADT. Sedangkan perancangan data menghasilkan struktur data siap diprogram, dan struktur basis data siap dibuat oleh pemograman, prosedur/operasi untuk

mengakses data, siap untuk diprogram.

### **Physical Data model ( PDM )**

Physical Data model (alias desain database) adalah representasi dari data desain yang memperhitungkan sarana dan kendala suatu sistem manajemen database. *Lifecycle* dalam suatu proyek itu biasanya berasal dari model data logika, meskipun mungkin akan *reverse-engineered* dari suatu database pelaksanaannya. Data model fisik akan memasukkan semua database seni diperlukan untuk membuat hubungan antara tabel atau kinerja mencapai tujuan, seperti indeks, kendala definisi, menghubungkan tabel, tabel partitioned atau kelompok. Fisik data model biasanya dapat digunakan untuk menghitung perkiraan penyimpanan dan mungkin termasuk penyimpanan alokasi khusus untuk rincian yang diberikan sistem basis data.

Saat ini, terdapat empat (mungkin lima) database utama dalam bisnis pasar; Oracle, SQL Server, Sybase, DB2 dan MySQL. Terdapat banyak sekali sistem lainnya RDBMS luar sana, tetapi ini cenderung baik warisan yang akan digunakan dalam database atau akademisi seperti universitas atau kolese pendidikan lanjutan. fisik data model pada setiap pelaksanaan akan berbeda secara nyata, tidak sedikit yang disebabkan oleh OS persyaratan yang duduk di bawah mereka. Contoh akan SQL Server yang hanya berjalan pada sistem operasi Microsoft Windows, sedangkan Oracle dan MySQL dapat berjalan di Solaris, Linux dan UNIX lainnya berbasis sistem operasi Windows serta.

Ini berarti bahwa disk persyaratan, persyaratan keamanan dan banyak aspek lain dari model data fisik akan sepenuhnya dipengaruhi oleh RDBMS yang database administrator (atau organisasi) memilih untuk digunakan.

Sementara ada sekitar perdebatan yang semakin RDBMS adalah lebih baik dalam berbagai domain, secara umum diterima bahwa Oracle dari arsitektur sangat cocok untuk perusahaan besar & implementasi, SQL Server lebih baik untuk UKM dan MySQL yang cukup untuk UKM dan usaha kecil.

### 3.4.2 Perancangan Arsitektur

Definisi keterkaitan antar elemen-elemen utama yang akan membentuk suatu program. Objektif utama perancangan arsitektur adalah membangun struktur program modular dan merepresentasikan keterkaitan kendali antar modul dan memadukan struktur program dan struktur data dan mendefinisikan anatmuka yang memungkinkan data dapat mengalir pada seluruh program. Sedangkan prosesnya mengubah dari aliran informasi ( direpresentasikan dengan DFD ) menjadi struktur PL ( direpresentasikan dengan Structure Chart ).

#### Struktur Chart

Fungsi dari *Structure Chart* digunakan untuk mendefinisikan dan mengilustrasikan dari sistem secara berjenjang dalam bentuk modul dan sub modul. *Structure Chart* juga menunjukkan hubungan elemen data dan elemen kontrol serta hubungan antar modulnya, sehingga *Structure Chart* dapat



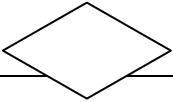
memberikan penjelasan yang lengkap dari sistem dipandang dari elemen data, elemen kontrol, modul dan hubungan antar modulnya.

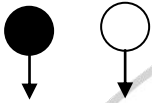
### Simbol – Simbol Dasar

Dalam menggambarkan struktur organisasi sistem secara berjenjang digunakan beberapa macam simbol. Simbol-simbol ini merupakan simbol-simbol standar yang paling banyak digunakan.

Disamping simbol-simbol standar tersebut, pemrogram juga dapat menambah khasanah simbol yang akan digunakan dalam penggambaran tersebut, tetapi pemrogram juga harus memberi penjelasan tentang maksud dari simbol yang dibuat tersebut dalam bentuk kamus simbol. Namun demikian sebaiknya pemrogram menggunakan simbol-simbol standar untuk menggambarkan struktur sistem tersebut agar mudah dipahami oleh pemrogram lain.

Tabel 3.4 Simbol Dasar

Gambar	Nama	Keterangan
	Module	Simbol ini menunjukkan suatu modul
	Connection	Simbol ini digunakan untuk menghubungkan suatu modul dengan modul yang lainnya
	Loop	Simbol ini menunjukkan suatu perulangan didalam modul

	Decision	Simbol ini menunjukkan suatu penyeleksian kondisi didalam modul
	Couple	Simbol ini menunjukkan suatu data/elemen yang dikirimkan dari satu modul ke modul lainnya. Anak panah dengan lingkaran yang kosong menunjukkan data yang dikirimkan, sedangkan anak panah dengan lingkaran padat menunjukkan elemen kontrol yang dikirimkan

*Structure Chart* memiliki dua model penggambaran sistem, yaitu *Transformes-Centered* dan *Transaction-Centered*

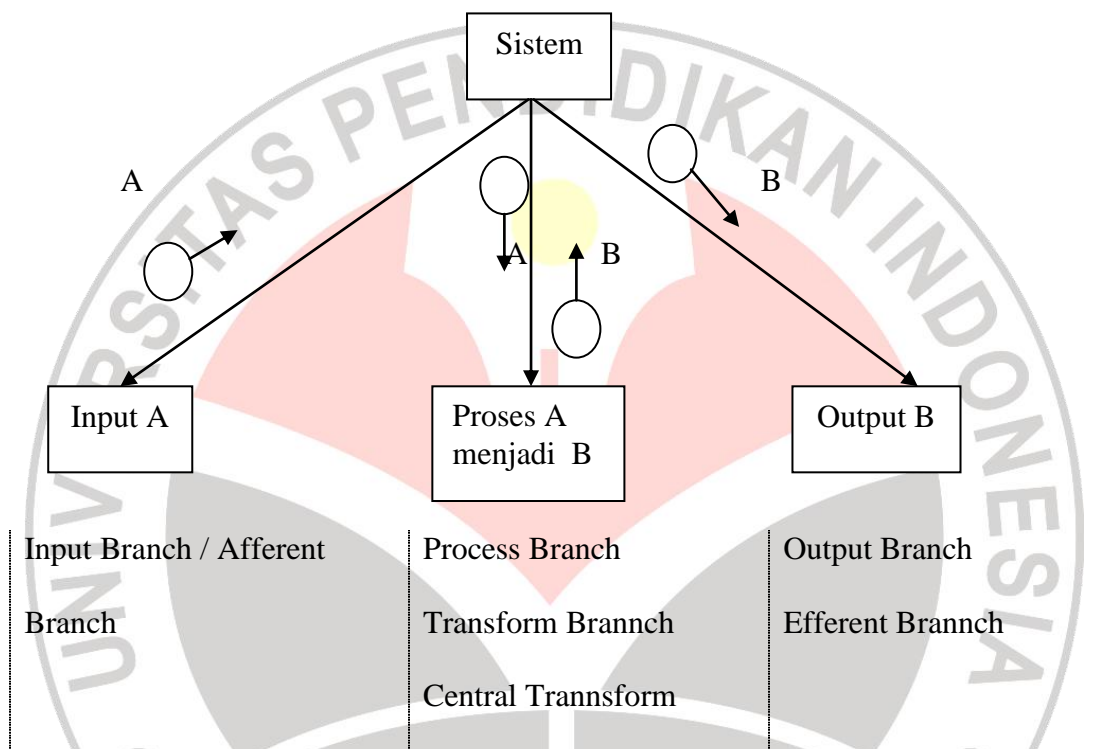
a. *Transformed-Centered*

*Structure Chart* dengan model *Transformed Centered* menggambarkan sistem dalam cabang utama, yaitu :

- i. Cabang Input yang merupakan cabang yang akan menerima input dan menentukan status input untuk siap di proses.

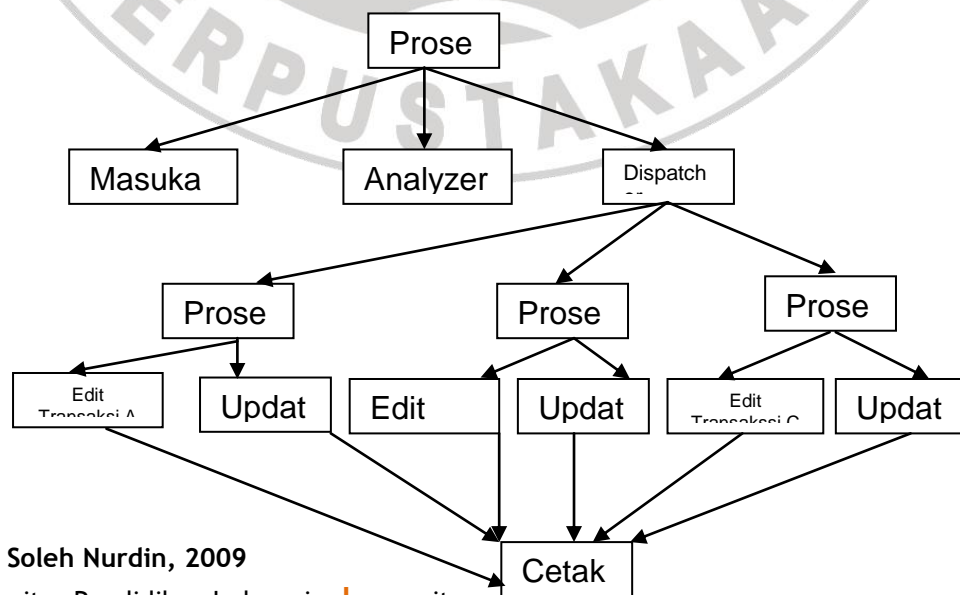


- ii. Cabang Proses yang merupakan cabang yang akan melakukan fungsi utama dari sistem, yaitu memproses input yang dikirim dari cabang input.
- iii. Cabang Output, merupakan cabang yang akan memformat data menjadi output.



Gambar 3.14 *Transformed Centered*

b. *Transaction-Centered*



Gambar 3.15 *Transaction-Centered*



#### 3.4.4. Perancangan Modul/Prosedur

Pada subbab ini menerangkan Transformasi elemen struktural dari arsitektur program menjadi deskripsi prosedur. Perancangan prosedur bisa dikatakan tahapan terakhir pada proses perancangan. Untuk merancang prosedur alat bantu yang dapat digunakan diantaranya flow-chart, algoritma. Pada intinya perancangan prosedural harus memperhatikan dua hal yang paling utama diantaranya coupling (*a measure of the interdependence among software module*), yaitu ukuran kekuatan saling ketergantungan antar modul-modul software dan cohesion (*a measure of the relative functional strength of a module*), yaitu ukuran kekuatan modul-modul perangkat lunak secara fungsional relatif terhadap modul perangkat lunak itu sendiri.

#### Flowchart

*Flowchart* adalah Bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma

Tujuan Utama dari penggunaan Flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Tahap penyelesaian masalah yang disajikan harus jelas, sederhana, efektif dan tepat.

## Langkah-langkah Membuat flowchart

Beberapa petunjuk yang harus diperhatikan, seperti :

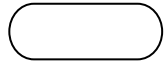
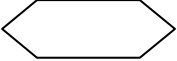





1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakkan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchart yang standar.


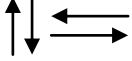
## Simbol- Simbol flowchart

Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart :

Tabel 3.5 Simbol-simbol Flowchart

SIMBOL	KETERANGAN
--------	------------

	<b>Terminal Point Symbol.</b> digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	<b>Preparation Symbol.</b> Simbol <b>Persiapan</b> digunakan untuk memberi nilai awal Suatu besaran atau <i>variabel</i> ( harga awal ).
	<b>Process Symbol.</b> Simbol <b>Proses</b> atau <b>Pengolahan</b> digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	<b>Predefined Process Symbol.</b> Simbol <b>Proses Terdefenisi</b> digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detilnya dijelaskan terpisah, misalnya berbentuk subroutine.
	<b>Decision Symbol.</b> Simbol <b>Keputusan</b> digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	<b>Input/Output Symbol.</b> Simbol <b>Input/Output</b> digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	<b>Connector Symbol.</b> Simbol <b>Penghubung</b> digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.

	<p><b>Off-page Connector.</b> Simbol <b>Penghubung Halaman lain</b> digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.</p>
	<p><b>Flow Lines Symbol.</b> Simbol <b>Garis Alir</b> digunakan untuk menunjukkan aliran atau arus dari proses.</p>

### 3.5 Coding

Proses penulisan script/perintah pada saat pembuatan perangkat lunak (software) kemudian script/perintah tersebut dapat mudah dikompilasi dan dijalankan menjadi sebuah perangkat lunak.

#### 3.5.1 PHP

PHP (Hypertext Preprocessor) dikembangkan pertama kali tahun 1995 oleh Rasmus Lerdorf yang merupakan salah satu anggota group Apache. PHP pertama kali didesain sebagai alat tracking pengunjung web site Lerdorf. Kemudian, fungsinya diperlebar dan dihubungkan dengan Apache. PHP dikembangkan sepenuhnya untuk bahasa skrip side-server programming. PHP bersifat open source dan dapat digabungkan dengan server yang berbeda.



Gambar 3.16 Logo PHP.

PHP adalah bahasa server-side programming yang power full untuk membuat halaman web yang dinamis dan interaktif. Sintak PHP mirip dengan bahasa C. PHP biasanya sering digunakan bersama web server Apache diberagam sistem operasi. PHP merupakan software open source yang dapat digunakan dengan gratis dan dapat dijadikan alternative selain side-server programming lain dan sintak PHP dapat ditulis kan langsung diantara kode HTML.

### **Kelebihan PHP**

Kelebihan-kelebihan PHP antara lain :

- 1 Mudah dibuat dan dijalankan
- 2 Mampu berjalan pada web server dengan sistem operasi yang berbeda-beda: PHP mampu berjalan pada sitem operasi UNIX, Windows dan Macintosh
- 3 Dapat berjalan pada web server yang berbeda: PHP mampu berjalan pada web server yang berbeda seperti Microsoft personal Web Server, Apache, IIS, Xitami dll.
- 4 PHP dapat dijalankan dalam tag HTML.

### **3.5.2 MySQL**

MySQL adalah salah satu database server open source yang populer. Sebuah database mendefinisikan struktur untuk menyimpan informasi. Dalam database ada beberapa tabel dan tabel dalam HTML, sebuah tabel berisi baris, kolom dan cell. Sebuah database biasanya berisi lebih dari satu tabel dan mempunyai nama dan masing-masing tabel berisi baris dan data.



Gambar 3.17 Logo MySQL

MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

#### 1. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

#### 2. Open Source

MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

#### 3. Multiuser



MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

#### 4. Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

#### 5. Column types

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

#### 6. Command dan functions

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

#### 7. Security

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

#### 8. Scalability dan limits

MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

#### 9. Connectivity

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

## 10. Localisation

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

## 11. Interface

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

## 12. Clients dan tools

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

## 13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

### 3.5.3 XAMPP

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang

pada tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server.



Gambar 3.18 Logo XAMPP.

XAMPP yaitu software yang dilengkapi dengan web server yang merupakan tempat melayani *request* ketika mengeksekusi program dan mengirimkan hasilnya ke *client*. Dalam XAMPP terdiri dari Apache (web server), PHP 5.0.(side server programming) dan MySQL ( Database Server).

#### **3.5.4 Photoshop CS**

Photoshop CS disebut juga Photoshop 8. Merupakan software populer yang dibuat oleh Adobe ([www.adobe.com](http://www.adobe.com)). Sangat powerfull dalam pengolahan gambar bitmap, menawarkan berbagai kemudahan dalam menggunakan feature yang beraneka macam di setiap versi terbarunya. Jika anda adalah seorang calon Web Designer, Image Editor, Fotografer dan semua orang yang sering menangani grafis, maka Photoshop sangatlah cocok untuk anda dalam mengembangkan ide dan kreatifitas anda.

#### **3.5.5 Macromedia Dreamweaver 2004**

Macromedia Dreamweaver merupakan sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs web beserta halaman web.Saat

ini terdapat software group Macromedia yang banyak digunakan untuk mendesain situs web.

Pada Dreamweaver MX 2004 terdapat beberapa kemampuan bukan hanya sebagai software untuk mendesain web saja tetapi juga untuk menyunting kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman web base antara lain PHP, JSP, ASP, dan ColdFusion.

Dreamweaver merupakan software populer yang digunakan oleh web designer maupun web programmer guna mengembangkan sebuah situs web. Ruang kerja, fasilitas dan kemampuan Dreamweaver mampu meningkatkan produktivitas dan efektifitas dalam desain dan maintain sebuah web. Dreamweaver juga dilengkapi dengan fasilitas untuk manajemen situs yang cukup lengkap.



Gambar 3.19 Layout Macromedia Dreamweaver MX 2004.

User interface pada Dreamweaver MX 2004 lebih baik dari versi sebelumnya. Karena didalamnya memuat peningkatan kemampuan toolbar, dimana bisa digunakan untuk modifikasi toolbar yang sudah ada atau menambah fungsi baru. Selain user interface baru, Dreamweaver MX 2004 memiliki kemampuan untuk menyunting kode dengan lebih baik, selain itu juga memiliki fasilitas Code Hints yang membantu dalam urusan tag dan Tag Inspector yang sangat berguna dalam menangani Tag HTML. Dreamweaver mx 2004 memiliki panel baru, yaitu Snippets yang memberikan fasilitas untuk menyunting dan menyimpan blok kode yang sering digunakan. Kemampuan baru lainnya adalah fasilitas untuk menangani dokumen-dokumen baru seperti XHTML, Extension Manager ditingkan kemampuannya, kemampuan validasi dengan debug di browser serta fasilitas untuk menangani aplikasi database yang sebelumnya hanya ada di Macromedia MX.

### **3.6 Pengujian Perangkat Lunak**

Pada Subbab ini menerangkan elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas. Meningkatnya visibilitas (kemampuan)

perangkat lunak sebagai suatu elemen sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak, memotivasi dilakukannya perencanaan yang baik melalui pengujian yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak dari pada membangun.

### 3.6.1 Testing Fundamental

Pengujian perangkat lunak adalah sebuah proses *formal* yang dilakukan oleh suatu *tim khusus* yang *menguji* software unit, beberapa perangkat lunak terpadu unites seluruh atau paket perangkat lunak yang diuji dengan menjalankan program pada komputer. Semua pihak terkait yang akan dilakukan tes sesuai dengan prosedur yang telah disetujui pada tes kasus disetujui

### 3.6.2 Metode Pengujian

Pada pengujian ini akan digunakan metode pengujian Black-Box dimana prosedur teknis tersebut diperuntukan untuk menguji kebenaran terhadap software yang dibuat.

#### **Black-Box**

**Black-Box** adalah pengujian berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari

suatu program dengan cara yang memberikan cakupan pengujian yang mendalam. Metode pengujian graph-based mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksa kemampuan program untuk menangani data pada batas yang dapat diterima. Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur client/ server, dokumentasi dan fasilitas help dan sistem real time masing-masing membutuhkan pedoman dan tehnik khusus untuk pengujian perangkat lunak.

#### **Kelebihan dari Black Box Ujian :**

1. lebih efektif pada unit kode yang lebih besar daripada kotak kaca pengujian
2. tester kebutuhan tidak mengetahui pelaksanaan, termasuk spesifik bahasa pemrograman
3. tester dan programmer yang independen satu sama lainnya
4. tes yang dilakukan dari pengguna point of view
5. akan membantu untuk mengungkapkan apapun ambiguities atau ketidakkonsistenan dalam spesifikasi
6. uji kasus dapat dirancang segera setelah spesifikasi yang lengkap

#### **Kekurangan dari Black Box Ujian :**

1. hanya sedikit masukan mungkin dapat benar-benar akan diuji, untuk menguji setiap kemungkinan masukan streaming akan mengambil hampir selamanya
2. jelas dan ringkas tanpa spesifikasi, uji kasus sulit untuk merancang
3. mungkin ada yang tak perlu dari pengulangan ujian masukan jika tester tidak tentang uji kasus programmer telah mencoba
4. tidak dapat diarahkan menuju spesifik segmen kode yang mungkin sangat kompleks (dan karena itu lebih rentan kesalahan)
5. paling pengujian terkait penelitian telah diarahkan ke arah kaca kotak uji.

#### **Strategi Pengujian / Techniques**

1. kotak hitam pengujian harus menggunakan input yang dihasilkan secara acak (hanya tes rentang harus ditentukan oleh tester), untuk menghilangkan semua terkaan oleh tester mengenai metode fungsi
2. data di luar jangkauan ditentukan input harus diuji untuk memeriksa kebenaran program
3. batas kasus harus diuji (atas dan bawah dari rentang yang ditetapkan) untuk memastikan tertinggi dan terendah yang diijinkan masukan benar menghasilkan output
4. angka nol yang harus diuji ketika data numerik menjadi masukan
5. stres pengujian harus dilakukan (coba untuk kelebihan program dengan masukan untuk melihat di mana mencapai kapasitas maksimum), terutama dengan sistem real tim



6. crash pengujian harus dilakukan untuk melihat apa yang diperlukan untuk membawa sistem bawa
7. uji alat monitor harus digunakan untuk melacak bila memungkinkan tes yang telah dilakukan dan output ini untuk menghindari pengulangan ujian dan untuk membantu dalam perawatan perangkat lunak
8. fungsional lainnya pengujian teknik meliputi: pengujian transaksi, pengujian sintaks, domain pengujian, logika pengujian, pengujian.

