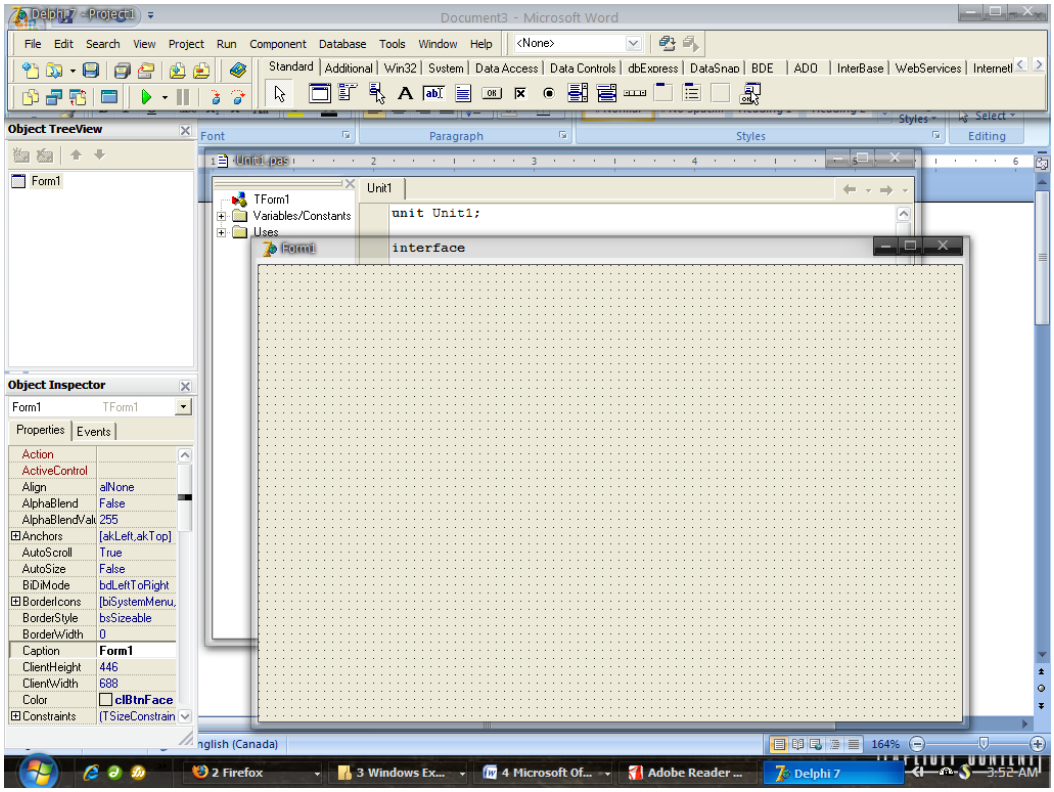
BAB IV
LANDASAN TEORI

4.1 Pengenalan Borland Delphi 7

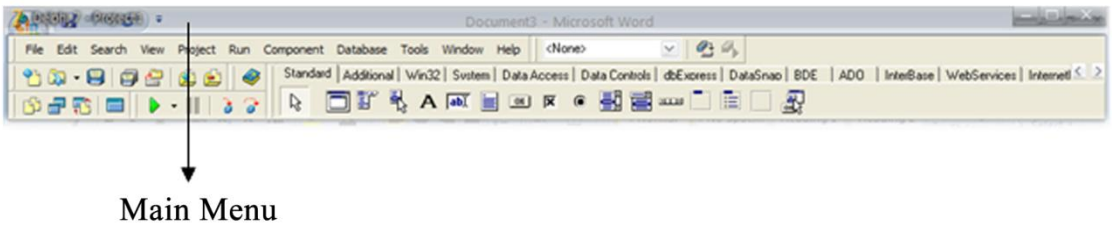
Program computer dapat memecahkan hampir semua masalah yang memerlukan ketelitian yang sangat tinggi serta banyak data yang harus diolah dengan waktu singkat. Dulu, membuat sebuah program sangat sulit dan melelahkan. Karena sekarang ini pembuatan program dituntut harus cepat, tepat dan siap pakai, maka muncullah bahasa pemrograman yang berbentuk visual,antara lain : Visual C ++, Visual Foxpro, Visual Basic, dan lain – lain.

Delphi adalah salah satu dari pemrograman secara visual, bahasa yang digunakan lebih mengarah ke bahasa pascal. Banyak yang mengatakan bahwa Pascal adalah nenek moyangnya Delphi, ini karena Delphi adalah pengembangan dari turbo pascal yang popular saat DOS masih popular. Seperti bahasa lainnya, Delphi mengalami perkembangan yang sangat pesat. Delphi 7, memiliki support yang tinggi terhadap database yang sudah terkenal (seperti MS Access, Paradox, Foxpro, Dbase, Oracle, dan lain sebagainya), dan dilengkapi dengan objek – objek yang memudahkan pembuatan program, baik program database maupun program lainnya (game, utility, dan lain – lain).

Karena Delphi berbentuk visual, maka pembuatannya pun sangat mudah, cepat serta menyenangkan. Cukup menaruh obyek – obyek yang dikehendaki. Penulisan bahasa program atau source code-nya pun tidak terlalu banyak.



Gambar 4.1 Delphi 7



Gambar 4.2 *Main Menu*

Di dalam main menu terdapat fasilitas – fasilitas yang disediakan oleh Delphi untuk mengontrol serta menangani masalah yang berhubungan dengan pekerjaan, seperti menu untuk menyimpan, menu untuk meng-compile, dan masih banyak lagi.



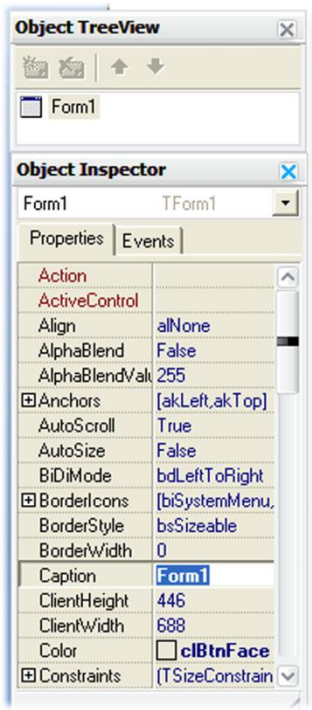
Gambar 4.3 *Speed Bar*

Kegunaan Speed bar sama seperti main menu, tetapi di Speed bar berupa icon sehingga kita cukup meng-klik saja untuk menjalankan fasilitas yang kita kehendaki.



Gambar 4.4 *Palette Bar*

Palette bar adalah tempat object – object yang digunakan dalam penyusunan sebuah program. Disini terdapat 32 palette bar, yaitu : Standart, Additional, Win32, System, Data Access, Data Controls, dbExpress, Danasnap, BDE, ADO, Interbase, Webservices, InternetExpress, Internet, Web Snap, Decision Cube, Dialogs, Win 3.1, Samples, AktiveX, Indy Clients, Indy Servers, Indy Intercepts, Indy I/O Handlers, Indy Misc, COM+, InterBase Admin, IW Standart, IW Data, IW Client Side, IW Control, Servers.

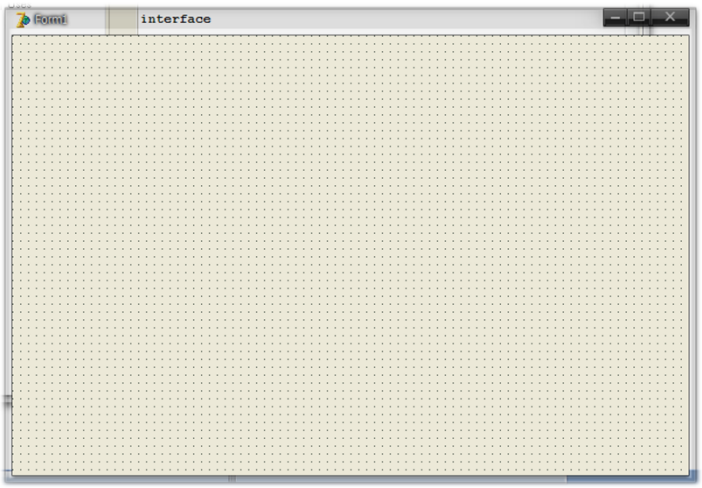


Gambar 4.5 *Object Inspector*

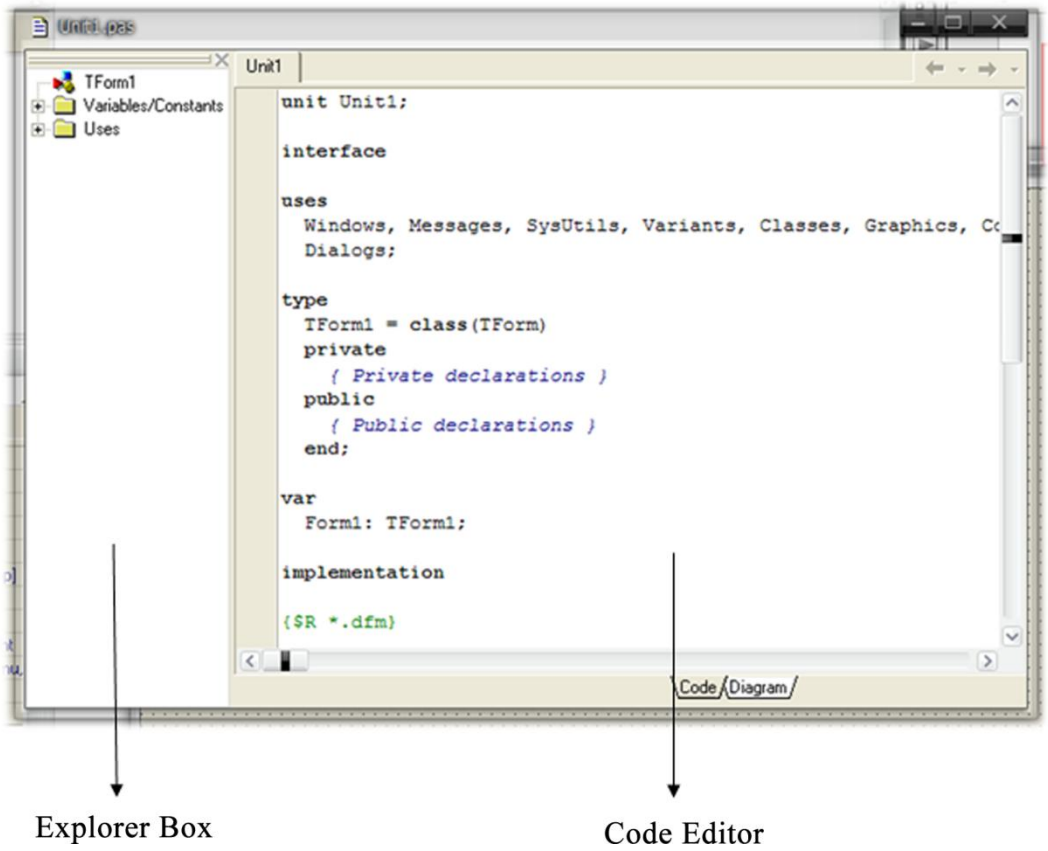
Object Inspector terdapat dua bab, yaitu :

→ *Properties* : berfungsi mengubah sifat-sifat yang dibawa oleh project tersebut.

→ *Events* : sebagai daftar procedure yang dilakukan oleh object tersebut ketika suatu peristiwa terjadi



Gambar 4.6 *form*



Gambar 4.7 *Explorer Box dan Code Editor*

Form adalah tempat untuk memasang objek – objek. Code Editor adalah tempat untuk menuliskan source code / bahasa program. Tombol F12 adalah tombol untuk mengaktifkan code editor apabila from dalam keadaan aktif. Explorer Box yaitu kotak yang menggambarkan macam – macam class yang dimuat dalam unit, procedure atau function yang telah dibuat, nama variable atau kaonstan yang ada di dalam unit, serta unit – unit yang digunakan di dalam unit tersebut.

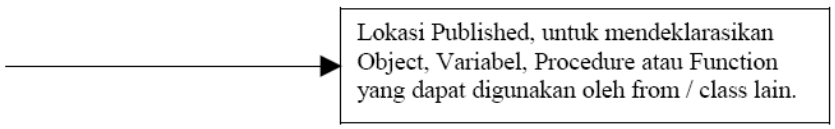
Pada Code Editor diletakkan semua basis pemograman, from, dan unit sehingga menjadi satu kesatuan yang terorganisir. Pada pemograman Delphi, dituntut untuk memahami serta mengetahui cara menulis sebuah procedure atau function. Pada intinya, procedure atau function adalah suatu kelompok yang berisi beberapa

statement perintah yang gunanya untuk menyederhanakan program utama dan biasanya kelompok perintah tersebut sering digunakan.

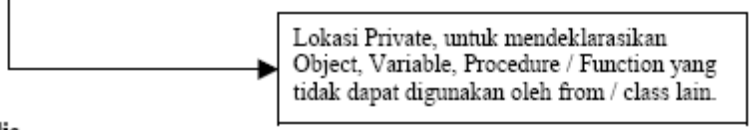
Perbedaan procedure dengan function adalah function dapat menghasilkan suatu hasil dengan tipe tertentu, sedangkan procedure tidak dapat menghasilkan hasil. Procedur atau function dapat diberi beberapa parameter. Parameter procedure atau function dapat dikategorikan didalam dua jenis, yaitu parameter masuk (parameter yang berfungsi untuk memasukkan / menerima nilai), dan parameter keluar (parameter yang berfungsi untuk memasukkan / menerima dan juga menghasilkan / mengeluarkan suatu nilai).

Berikut adalah contoh source code dan letak – letak untuk mendeklarasikan serta menulis isi dari procedure atau function :

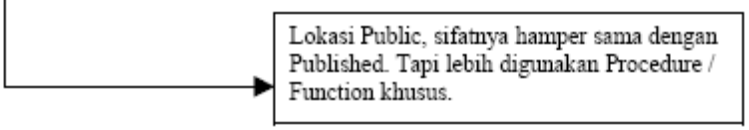
```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Froms, Dialogs;
type
TFrom1 = class(TFrom)
```

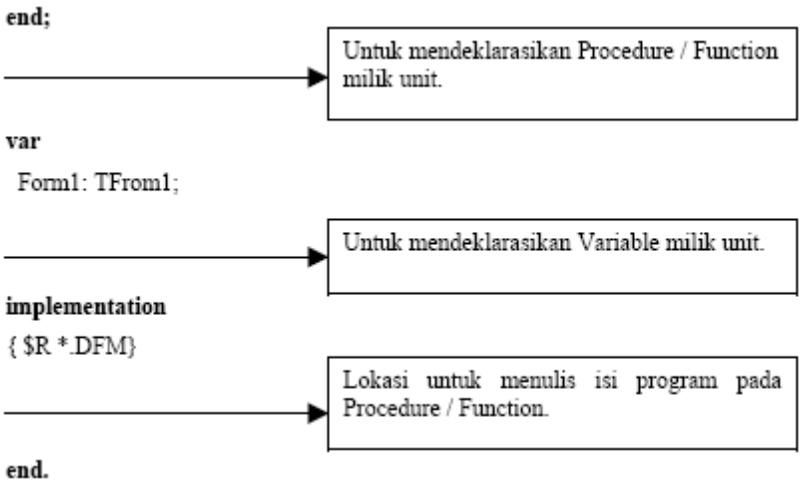



```
private
  { Private declarations }
```




```
public
  {Public declarations }
```





Gambar 4.8 Sistem Penulisan Program Pada Delphi

4.2 Definisi Database

Database adalah susunan record data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam computer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna.

Borland Delphi 7 mendefinisikan database sebagai keterangan mengenai kumpulan sejumlah table, prosedur tersimpan dan hubungan relasi antar table yang saling berhubungan daam bentuk suatu program aplikasi atau dapat disebut juga dengan database relational. Jadi, file database dalam Borland Delphi 7 hanya menampung nama file, hubungan relasi dan keterangan dari file – file table lainnya.

4.3 Konsep Dasar Database

Banyak sekali kegiatan manusia yang menggunakan computer sebagai sarana pengolahan data, sehingga diperlukan suatu perangkat lunak database. Jika dikaji

lebih mendasar tentang batasan suatu database, maka dapat disebutkan bahwa segala bentuk koleksi data adalah suatu database. Mulai dari kelompok data pegawai, sampai dengan kelompok file, merupakan database.

Kelompok data tersebut lalu diorganisir terhadap jenisnya dan diwujudkan dalam bentuk table, sehingga terbentuklah berbagai macam table sesuai dengan jenisnya. Namun demikian meskipun table – table telah dipisahkan terhadap jenis yang berlainan, suatu table biasanya masih memiliki kaitan informasi dengan table lainnya.

Dalam perancangan system database ada dua hal utama yang perlu diperhatikan dan tidak boleh diabaikan, yaitu keamanan dan otoritas. Keamanan menyangkut segala potensi database dalam menjaga data dari pihak – pihak yang tidak diinginkan. Sedangkan otoritas terkait dengan pemberian wewenang atau hak istimewa untuk mengakses, mengedit dan menggunakan program pada para pengguna.

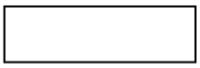
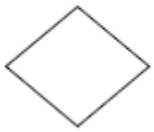
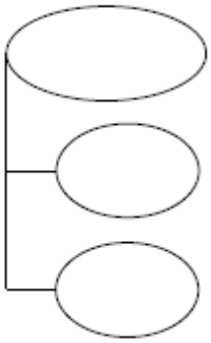
4.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam system secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur – struktur dan relation data.

4.5 Simbol – symbol Entity Relationship Diagram

ERD merupakan alat untuk pembuatan model data secara grafik, maka ERD memiliki symbol – symbol yang digunakan untuk menggambarkan model data seperti terdapat pada table berikut :

Tabel 4.1 *Simbol – symbol Entity Relationship Diagram*

No	Simbol	Arti	Definisi
1.		Entitas	Suatu kumpulan objek atau sesuatu yang dapat dibedakanatau didefinisikan secara unik.
2.		Relationship	Hubungan yang terjadi antara satu atau lebih entity.
3.		Atribut	Karakteristik dari entity atau relationship yang menyediakan penjelasan detail tentang entity atau relationship tersebut.
4.		Atribut Kunci	Atribut yang digunakan untuk menentukan suatu entity secara unik
5.		Atribut komposisi	Atribut yang memiliki sekelompok nilai untu setiap instansi entity.

4.6 Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasikan data kedalam table – table untuk memenuhi kebutuhan pemakai didalam suatu organisasi. Tingkatan dalam normalisasi :

1. First normal form (1NF)

   Suatu relasi dikatakan berada pada 1NF bila setiap data bersifat atomic yaitu setiap irisan baris dan kolom mempunyai satu nilai data.

2. Second normal form (2NF)

   Suatu relasi dikatakan berada pada 2NF bila relasi tersebut berada pada 1NF dan atribut yang bukan key bergantung penuh terhadap keynya.

3. Third normal form (3NF)

   Suatu relasi sudah berada pada 3NF bila sudah berada dalam 2NF dan atribut yang bukan key tidak tergantung transitif terhadap keynya.

4.7 Flowchart

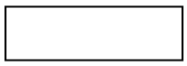
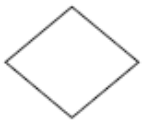
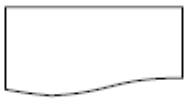
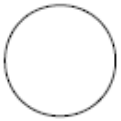
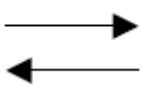
Flowchart adalah penggambaran secara grafik dari langkah – langkah dan urutan prosedur dari suatu program. Flowchart menolong analisis dan programmer untuk memecahkan masalah kedalam segmen – segmen yang lebih kecil dan menolong dalam menganalisis alternative – alternative lain dalam pengoperasian.

Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

4.8 Simbol – simbol Flowchart

Flowchart merupakan penggambaran urutan prosedur suatu program secara grafik, maka flowchart memiliki symbol – symbol yang digunakan untuk menggambarkan urutan program seperti terdapat pada table berikut :

Tabel 4.2 *Simbol Flowchart*

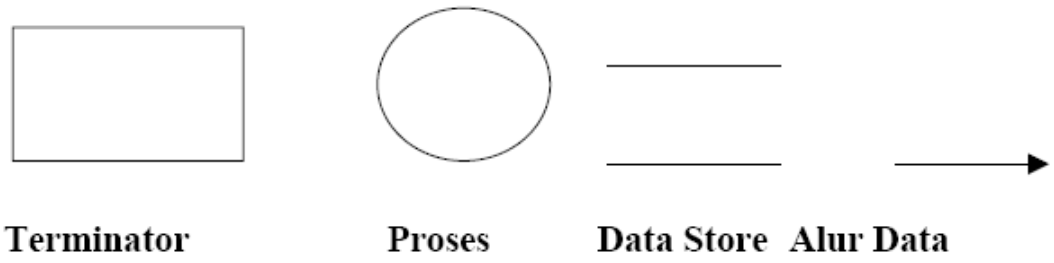
No	Simbol	Arti	Definisi
1.		Proses	Mempresentasikan operasi.
2.		Input / Output	Mempresentasikan Input data atau Output data yang diproses atau informasi.
3.		Keputusan	Keputusan dalam program.
4.		Dokument	I / O dalam format cetak.
5.		Terminal points	Awal / akhir flowchart.
6.		Preparation	Pemberian harga awal.
7.		Manual input	Input yang dimasukkan secara manual dari keyboard.
8.		Penghubung	Keluar atau masuk dari bagian lain flowchart khususnya halaman yang sama.
9.		Penghubung	Keluar atau masuknya dari bagian lain flowchart khususnya halaman lain.
10.		Display	Output yang ditampilkan pada terminal
11.		Anak panah	Mempresentasikan alur kerja.

4.9 Data Flow Diagram

DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional system kepada pemakai maupun pembuat program.

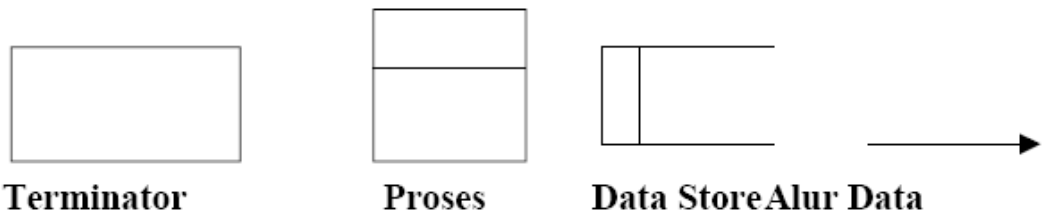
Komponen DFD :

1. Menurut Yourdan dan DeMarco



Gambar 4.9 Komponen DFD Menurut Yourdan dan DeMarco

2. Menurut Gene dan Serson



Gambar 4.10 Komponen DFD Menurut Gene dan Serson

Terminator Proses Data Store Alur Data

Adalah Entitas diluar sistem yang berkomunikasi / berhubungan langsung dengan sistem.

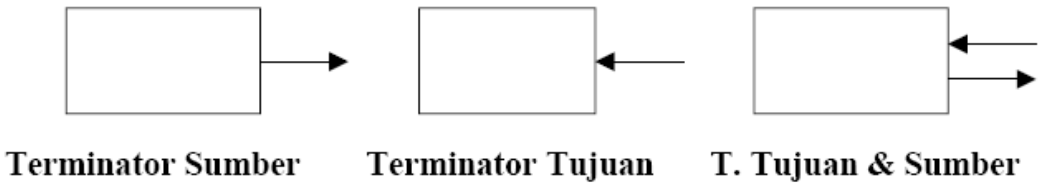
Terdapat 2 jenis Terminator :

1. Terminator Sumber

   Merupakan Terminator yang menjadi sumber

2. Terminator Tujuan

   Merupakan Terminator yang menjadi tujuan data / informasi sistem.



Gambar 4.11 Terminator Tujuan

Terminator dapat berupa orang, sekelompok orang, organisasi, perusahaan/departemen yang berada diluar sistem yang akan dibuat, diberi nama yang berhubungan dengan sistem tsb dan biasanya menggunakan kata benda.

Contoh : Dosen, Mahasiswa.

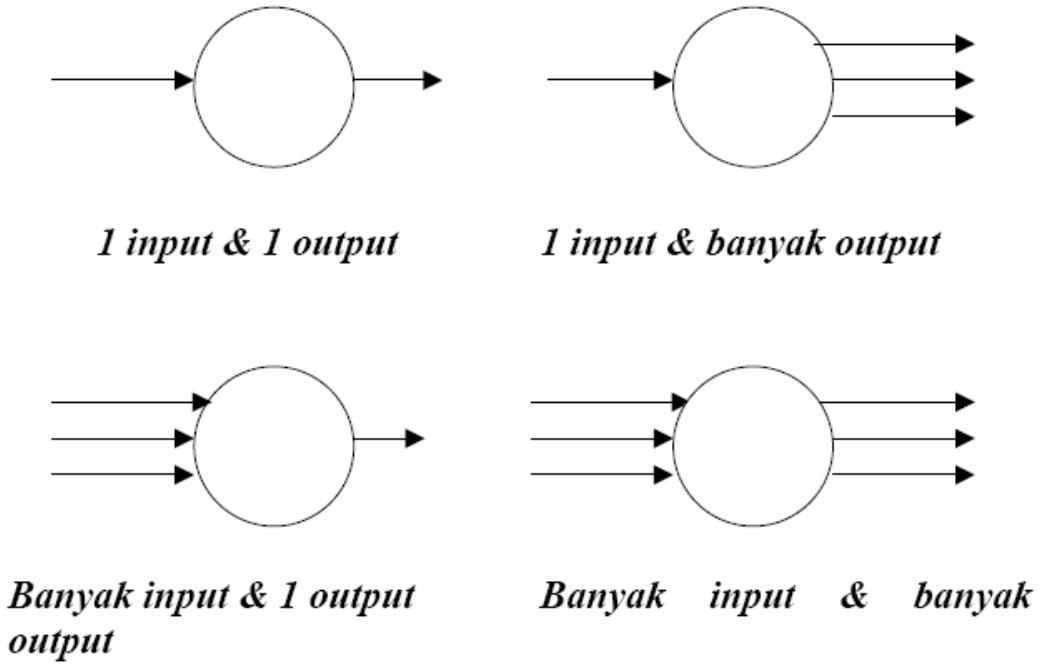
Hal yang perlu diperhatikan tentang terminator :

1. Alur data yang menghubungkan terminator dengan sistem, menunjukkan hubungan sistem dengan dunia luar.
2. Profesional sistem tidak dapat mengubah isi/cara kerja, prosedur yang berkaitan dengan Terminator.
3. Hubungan yang ada antar terminator tidak digambarkan dalam DFD.

Komponen Proses

Komponen proses menggambarkan transformasi input menjadi output. Penamaan proses disesuaikan dengan proses/kegiatan yang sedang dilakukan. Ada 4 kemungkinan yang dapat terjadi dalam proses sehubungan dengan

input dan output :



Gambar 4.12 Komponen Proses

Ada beberap hal yang perlu diperhatikan tentang proses :

1. Proses harus memiliki input dan output.
2. proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
3. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh professional sistem digambarkan dengan komponen proses.

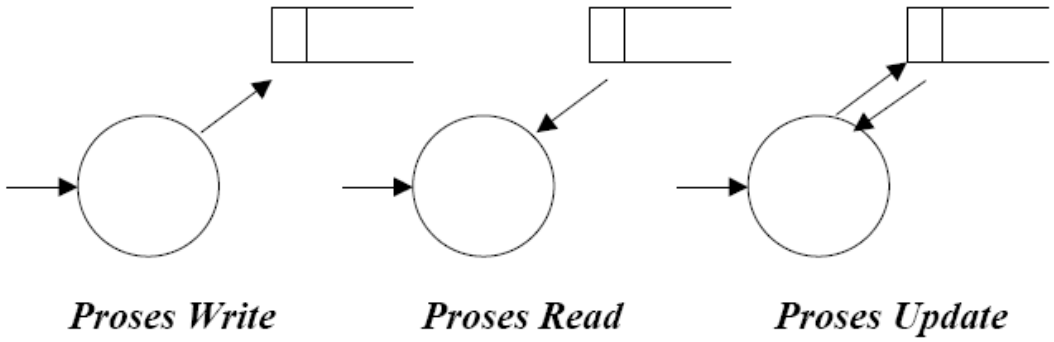
Komponen Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda bersifat jamak. Data store dapat berupa file/database yang tersimpan dalam disket, harddisk atau bersifat manual seperti buku alamat, file folder.

Yang perlu diperhatikan tentang data store :

1. Alur data dari proses menuju data store, hal ini berarti data store berfungsi sebagai tujuan/tempat penyimpanan fari suatu proses (proses write).
2. Alur data dari data store ke proses, hal ini berarti data store berfungsi sebagai sumber/ proses memerlukan data (proses read).
3. Alur data dari proses menuju data store dan sebaliknya berarti berfungsi sebagai sumber dan tujuan.

Lihat gambar berikut :



Gambar 4.13 Komponen Data Store

Komponen Alur Data

Alur data digunakan untuk menerangkan perpindahan data / paket datadari satu bagian ke bagian lainnya. Alur data dapat berupa kata, pesan, formulir / informasi.

Ada 4 konsep tentang alur data :

1. Packets of data

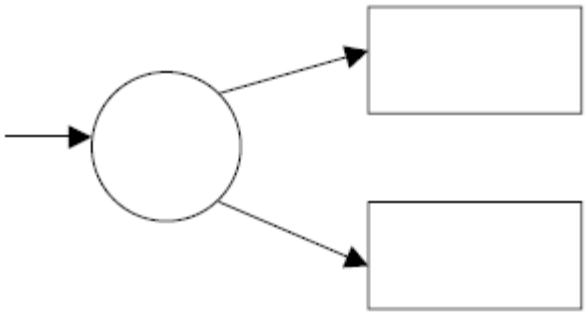
Apabila ada 2 data / lebih yang mengalir dari 1 sumber yang sama menuju pada tujuan yang sama & mempunyai hubungan digambarkan dengan 1 alur data.



Gambar 4.14 Packets of data

2. Diverging data flow

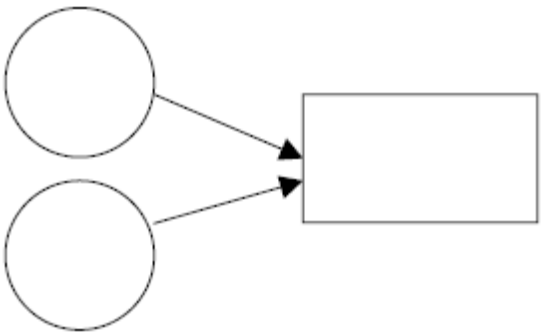
Apabila ada sejumlah paket data yang berasal dari sumber yang sama menuju pada tujuan yang berbeda atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda.



Gambar 4.15 Diverging data flow

3. Converging data flow

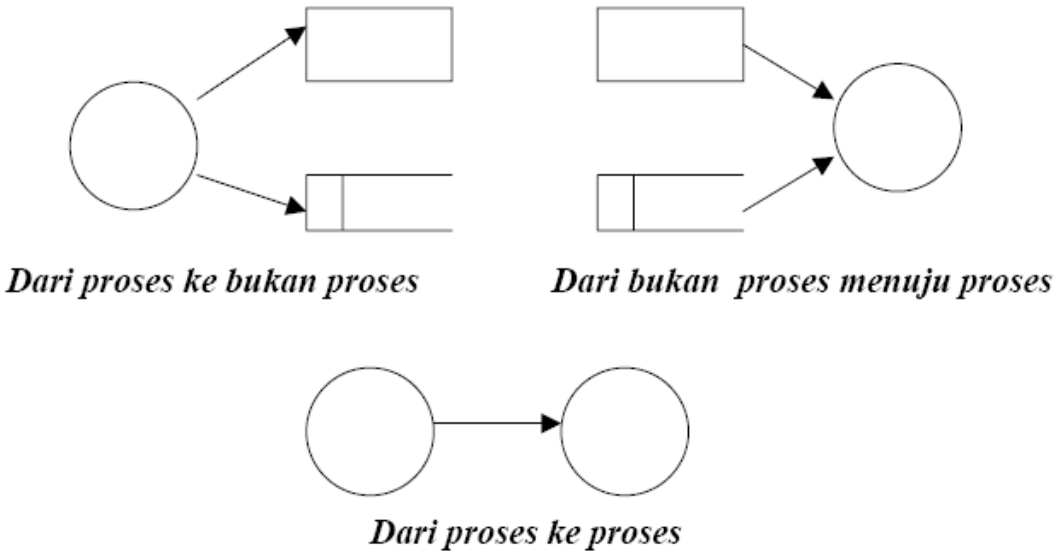
Apabila ada beberapa alur data yang berbeda sumber menuju ke tujuan yang sama.



Gambar 4.16 Converging data flow

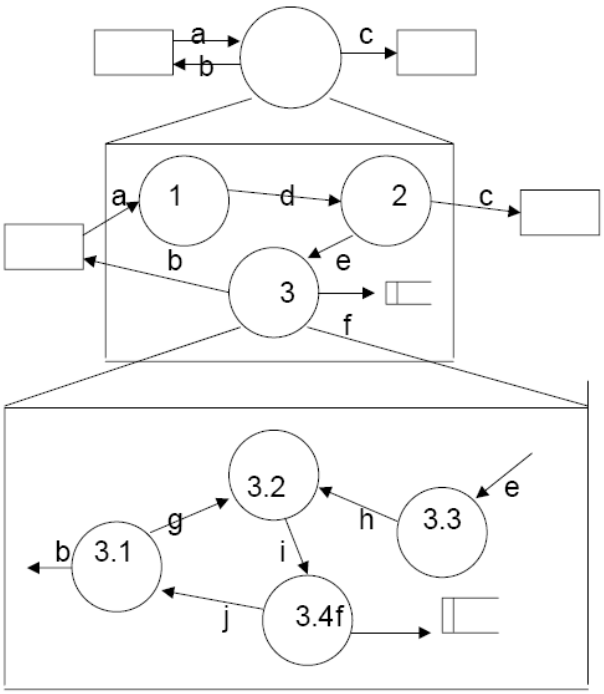
4. Sumber dan Tujuan

Arus data harus dihubungkan pada proses, baik dari maupun yang menuju proses.



Gambar 4.17 Sumber dan Tujuan

Levelisasi DFD



Gambar 4.18 Levelisasi DFD

Penggambaran DFD

Tidak ada aturan baku untuk menggambarkan DFD, tapi dari berbagai referensi yang ada, secara garis besar:

1. Buat diagram context

Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya.

Cara :

- Tentukan nama sistemnya.
- Tentukan batasan sistemnya.
- Tentukan terminator apa saja yang ada dalam sistem.
- Tentukan apa yang diterima/diberikan terminator dari/pada sistem.
- Gambarkan diagram context.

2. Buat diagram level Zero

Diagram ini adalah dekomposisi dari diagram Context.

Cara :

- Tentukan proses utama yang ada pada sistem.
- Tentukan apa yang diberikan/diterima masing-masing proses pada/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya)
- Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
- Gambarkan diagram level zero.

- Hindari perpotongan arus data
- Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

3. Buat diagram level Satu

Diagram ini merupakan dekomposisi dari diagram level zero.

Cara :

- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
- Tentukan apa yang diberikan/diterima masing-masing sub-proses pada/dari sistem dan perhatikan konsep keseimbangan.
- Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
- Gambarkan DFD level Satu
- Hindari perpotongan arus data.
- Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya. Contoh : 1.1, 1.2, 2.1

4. DFD level dua, tiga, ..

Diagram ini merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan level satu.