



## Modul : 11

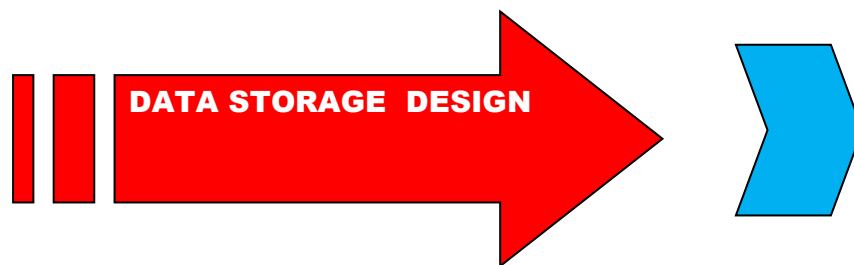
CPL-230-PENGEMBANGAN PERANGKAT LUNAK

Oleh :

5165 –Kundang K Juman

Prodi : Teknik Informatika

[www.esaunggul.ac.id](http://www.esaunggul.ac.id)



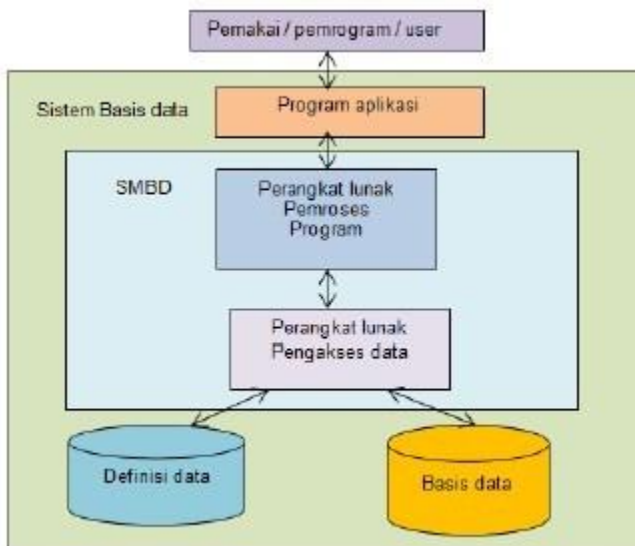
Pengertian basis data, yaitu :

Sebuah kumpulan data yang berhubungan secara logis dan merupakan penjelasan dari data tersebut yang dirancang dengan tujuan untuk menemukan data yang dibutuhkan oleh suatu perusahaan atau organisasi. Basis data juga dapat dikatakan sebagai kumpulan data yang saling terintegrasi karena basis data dirancang untuk dapat digunakan oleh banyak pemakai, memegang data operasional dan juga penjelasan mengenai data tersebut, dan menghindari duplikasi data. Sebuah kumpulan elemen data yang terintegrasi dan berhubungan secara logika. Basis data menggabungkan berbagai catatan yang sebelumnya disimpan dalam file terpisah ke dalam suatu elemen data. Menurut Connolly & Begg (2010, p65), basis data adalah sebuah kumpulan data yang saling berelasi secara logika dan dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi

## Konsep database

Konseptual Menggambarkan data apa yang sebenarnya ( secara fungsional ) disimpan dalam Basis Data dan hubungannya dengan data lain. Fisik bagaimana sesungguhnya suatu data disimpan Bahasa Basis Data : merupakan cara berinteraksi pemakai dengan basis data yang sesuai dengan aturan yang ditetapkan oleh pembuat DBMS.

Bagian Basis Data : – Data Definition Language (DDL) bahasa yang digunakan untuk menggambarkan basis data secara keseluruhan. Operasi yang dapat dijalankan adalah membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur. Hasil Kompilasi dari DDL ini adalah Kamus Data ( Data Dictionary ). – Data Manipulation Language (DML) Bahasa untuk melakukan manipulasi dan pengambilan data pada suatu basis data, seperti : penyisipan, penghapusan dan update



Keterangan File Manager : mengelola alokasi ruang dalam disk dan struktur data dalam disk. Database Manager : menyediakan interface antara low level dengan program Aplikasi dan query. Query Processor : menterjemahkan perintah dalam query language ke perintah low level yang dimengerti oleh Database Manager. DML Precompiler : mengkonversi perintah DML yang ada di program Aplikasi ke pemanggilan prosedur normal dalam bahasa induk.

DDL Compiler : mengkonversi perintah-perintah DDL kedalam sekumpulan tabel yang mengandung metadata.

Diagram Alir Data (DAD)/ Data Flow Diagram(DFD) dengan Bahasan:

- Pengertian DFD
- Latar Belakang DFD
- Manfaat DFD
- Tujuan DFD
- Notasi/Symbol DFD
- Langkah membuat/menggambar DFD
- Kesalahan dalam menggambar DFD
- Tips dalam membuat DFD
- Contoh kasus DFD (Perpustakaan)

### **Pengertian DFD**

**Diagram Alir Data (DAD)** atau **Data Flow Diagram (DFD)** adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

### **Latar Belakang DFD**

Suatu yang lazim bahwa ketika menggambarkan sebuah sistem kontekstual data flow diagram yang akan pertama kali muncul adalah interaksi antara sistem dan entitas luar. DFD didisain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil adan untuk menggarisbawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu “dikembangkan” untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya.

### **Tujuan DFD**

**Tujuan DFD** adalah :

1. Memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem
2. Menggambarkan fungsi-fungsi(dan sub fungsi) yang mentransformasi aliran data

### **Manfaat DFD**

**Manfaat DFD** adalah :

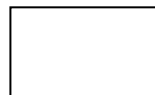
- Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
- DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.
- DFD ini 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 sistem kepada pemakai maupun pembuat program.

**Simbol DFD**

**TERMINATOR/KESATUAN LUAR (*EXTERNAL ENTITY*)**

Setiap sistem pasti mempunyai batas sistem (boundary) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (external entity) merupakan kesatuan (entity) di lingkungan luar sistem yang berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem (Jogiyanto, 1989).

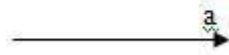
Suatu kesatuan luar dapat disimbolkan dengan suatu notasi kotak.



Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

## ARUS DATA (DATA FLOW)

Arus data (data flow) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (Process), simpanan data (data store) dan kesatuan luar (external entity). Arus data ini menunjukkan arus data yang dapat berupa masukan untuk sistem atau hasil



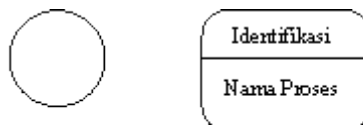
Notasi Arus Data di DFD

Arus data data dapat berbentuk berbentuk sebagai sebagai berikut berikut : :

- Formulir atau atau dokumen dokumen yang yang digunakan digunakan perusahaan perusahaan
- Laporan tercetak tercetak yang yang dihasilkan dihasilkan sistem sistem
- Output dilayar komputer
- Masukan untuk komputer komputer
- Komunikasi ucapan
- Surat atau memo
- Data yang dibaca atau atau direkam di file
- Suatu isian yang yang dicatat pada buku agenda
- Transmisi data dari suatu komputer ke komputer lain

## PROSES (PROCESS)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dan hasil suatu arus data yang masuk ke dalam proses untuk dilakukan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul.



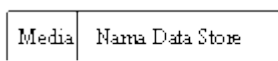
Notasi Proses di DFD

Ada beberapa hal yang perlu diperhatikan tentang proses :

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

### **SIMPANAN DATA (DATA STORE)**

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa file atau database di sistem komputer, arsip atau catatan manual, kotak tempat data di meja seseorang, tabel acuan manual, agenda atau buku. Simpanan data di DFD dapat disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.



Simbol dari Simpanan Data di DFD

Syarat-syarat pembuatan DFD ini adalah :

1. Pemberian nama untuk tiap komponen DFD
2. Pemberian nomor pada komponen proses
3. Penggambaran DFD sesering mungkin agar enak dilihat
4. Penghindaran penggambaran DFD yang rumit
5. Pemastian DFD yang dibentuk itu konsisten secara logika

Berikut ini tips-tips dalam membuat DFD :

1. Pilih notasi sehingga proses yang didekomposisi atau tidak didekomposisi dapat dibaca dengan mudah
2. Nama proses harus terdiri dari kata kerja dan kata benda
3. Nama yang dipakai untuk proses, data store, dataflow harus konsisten (identitas perlu)
4. Setiap level harus konsisten aliran datanya dengan level sebelumnya
5. Usahakan agar external entity pada setiap level konsisten peletakkannya
6. Banyaknya proses yang disarankan pada setiap level tidak melebihi
7. 7 proses

8. Dekomposisi berdasarkan kelompok data lebih disarankan (memudahkan aliran data ke storage yang sama)
9. Nama Proses yang umum hanya untuk proses yang masih akan didekomposisi
10. Pada Proses yang sudah tidak didekomposisi, nama Proses dan nama Data harus sudah spesifik
11. Aliran ke storage harus melalui proses, tidak boleh langsung dari external entity
12. Aliran data untuk Proses Report .. : harus ada aliran keluar. Akan ada aliran masuk jika perlu parameter untuk mengaktifkan report
13. Aliran data yang tidak ada datastorenya harus diteliti, apakah memang tidak mencerminkan persisten entity (perlu disimpan dalam file/tabel), yaitu kelak hanya akan menjadi variabel dalam program.

### **Langkah membuat/menggambar DFD**

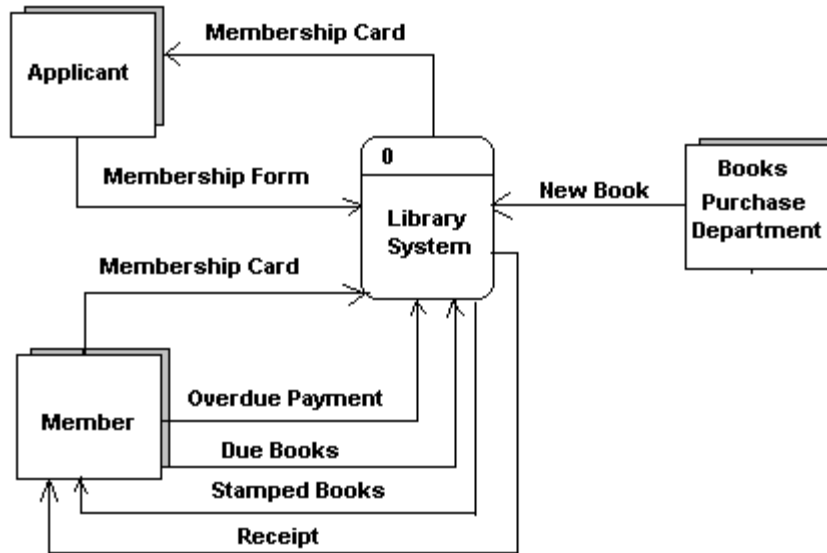
Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

#### **BUAT DIAGRAM KONTEKS (DIAGRAM CONTEXT)**

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

Caranya :

- Tentukan nama sistemnya.
- Tentukan batasan sistemnya.
- Tentukan terminator apa saja yang ada dalam sistem.
- Tentukan apa yang diterima/diberikan external entity dari/ke sistem.
- Gambarkan diagram konteks.



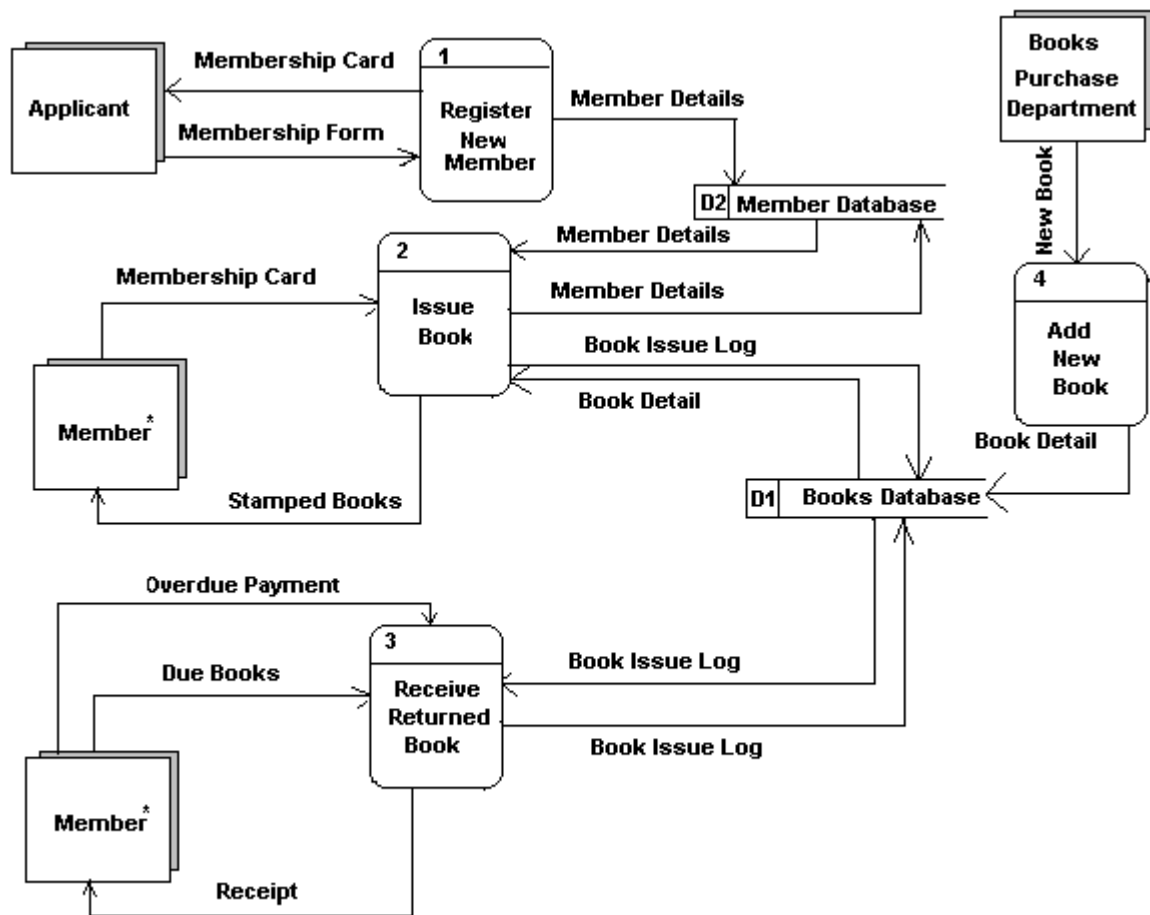
### BUAT DIAGRAM LEVEL ZERO (OVERVIEW DIAGRAM)

Diagram ini adalah dekomposisi dari diagram konteks.

Caranya :

- Tentukan proses utama yang ada pada sistem.
- Tentukan apa yang diberikan/diterima masing-masing proses ke/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.
- Hindari perpotongan arus data
- Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).



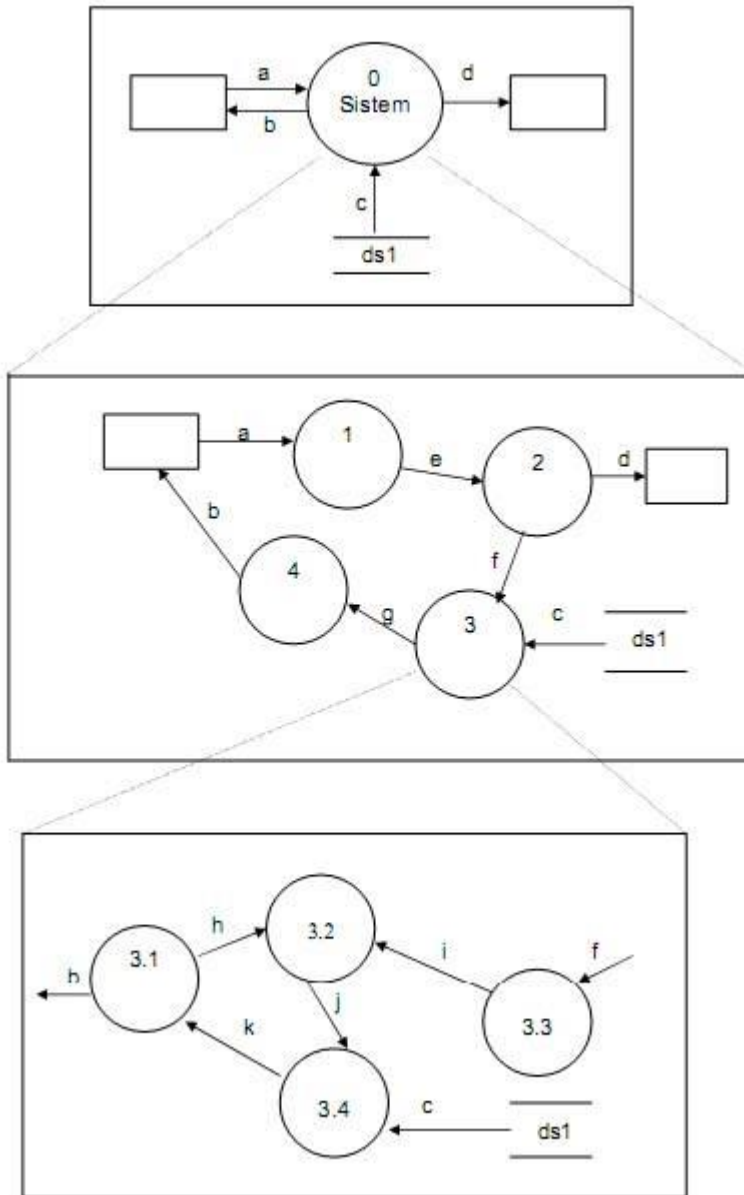


## BUAT DIAGRAM LEVEL SATU

Diagram ini merupakan dekomposisi dari diagram level zero.

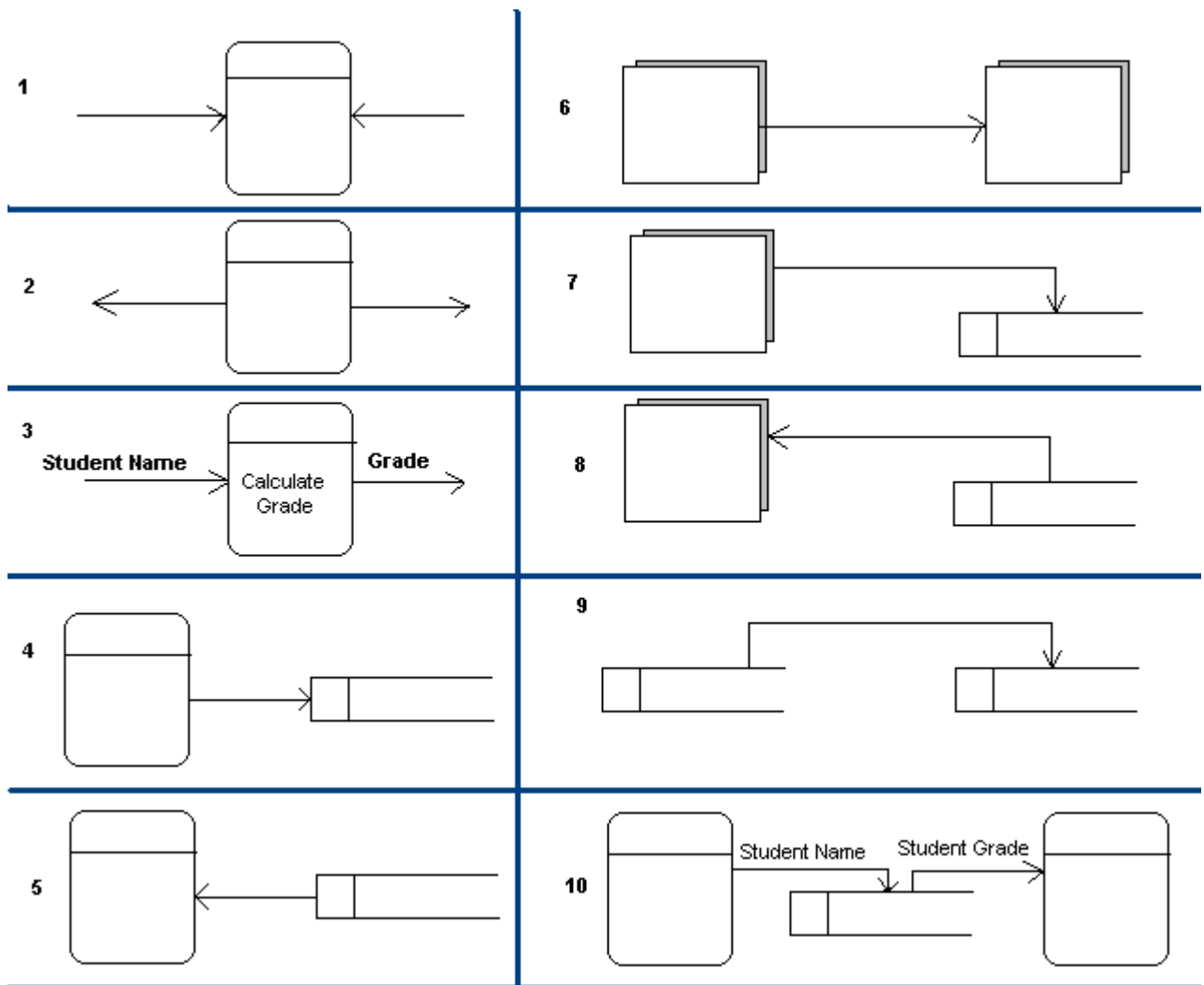
Caranya :

- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
- Tentukan apa yang diberikan/diterima masing-masing sub-proses ke/dari sistem dan perhatikan konsep keseimbangan.
- Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
- Hindari perpotongan arus data.
- Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya. Contoh : 1.1, 1.2, 2



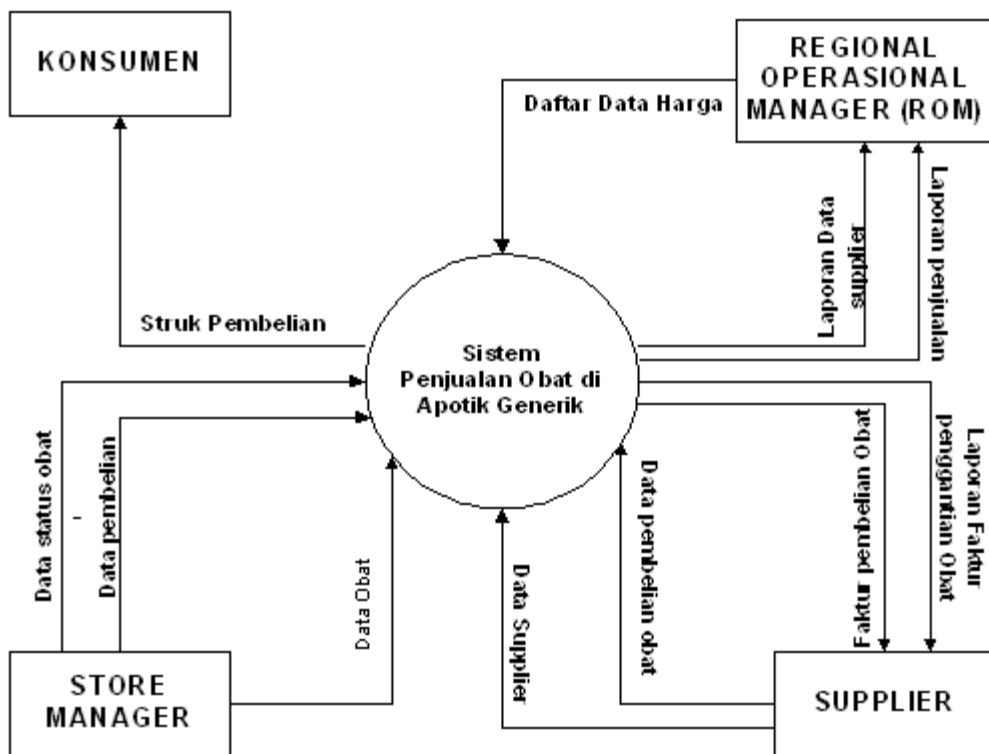
Kesalahan dalam pembuatan DFD

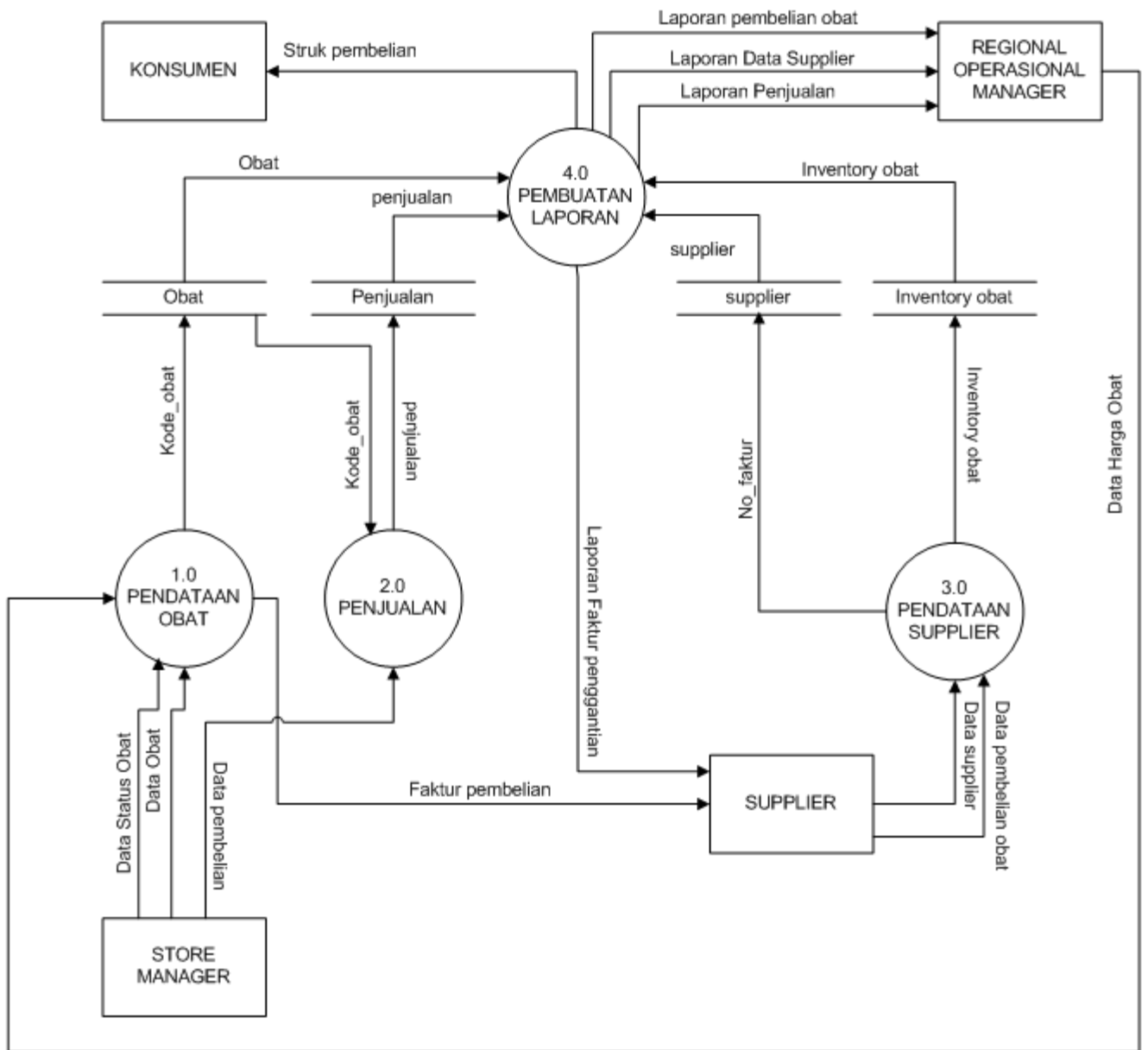
Umumnya kesalahan dalam pembuatan DFD adalah :



1. Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan *black hole* (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.
2. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan *miracle* (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.
3. Input yang masuk tidak sesuai dengan kebutuhan proses
4. Data Store tidak memiliki keluaran
5. Data Store tidak memiliki masukan
6. Hubungan langsung antar entitas luar
7. Masukan langsung entitas data store
8. Keluaran langsung dari data store ke Entitas luar
9. Hubungan langsung antar data store
10. Data masukan dan keluaran yang tidak bersesuaian dalam data store

# CONTOH DIAGRAM KONTEKS





Contoh diagram 0

Refrensi :

[Kendal System Analysis and Design 8th Edition](#)