

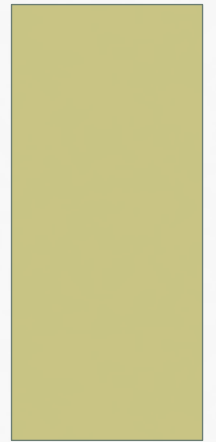


## **ANALISIS KEBUTUHAN FUNGSIONAL**

ANNA DARA ANDRIANA, S.KOM., M.KOM

# DFD

## (DATA FLOW DIAGRAM)





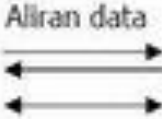
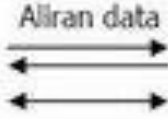
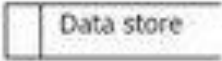



# PENGERTIAN

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 sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

# KOMPONEN DFD

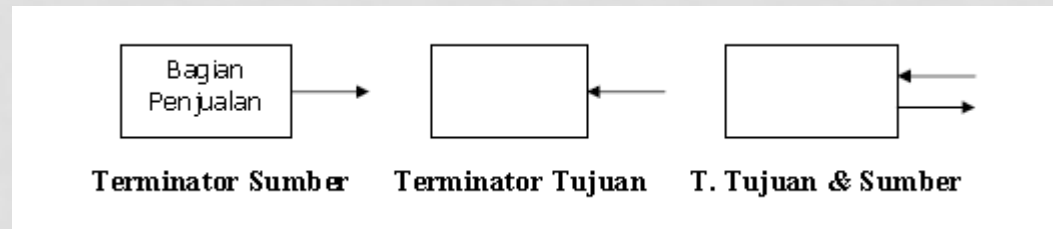
# KOMPONEN DFD

Gane/Sarson	Yourdon/De Marco	Keterangan
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan
		Penyimpanan data atau tempat data direfer oleh proses.

# KOMPONEN TERRMINATOR / ENTITAS EKSTERNAL

# KOMPONEN TERMINATOR

- Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*). Terdapat dua jenis terminator :
  1. Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
  2. Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data



- 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.

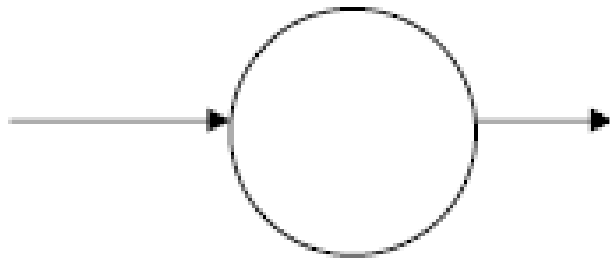
- Ada tiga hal penting yang harus diingat tentang terminator :
- 1. Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
- 2. Profesional sistem tidak dapat mengubah isi atau cara kerja organisasi, atau prosedur yang berkaitan dengan terminator.
- 3. Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.



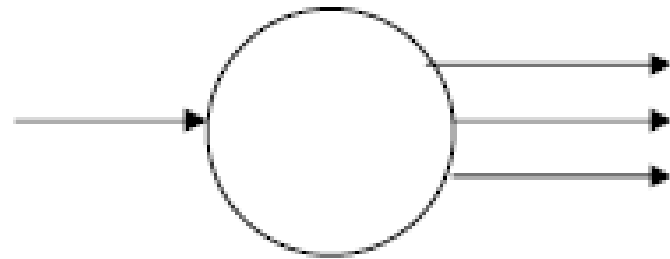
# KOMPONEN PROSES

# KOMPONEN PROSES

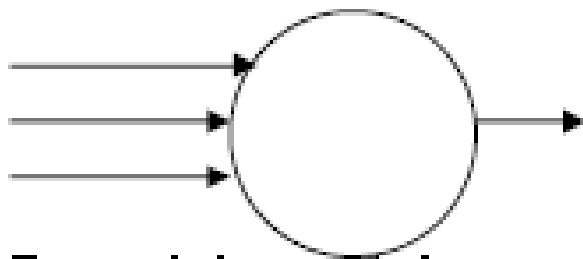
- Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output
- **Proses diberi nama** untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan **kata kerja transitif** (kata kerja yang membutuhkan obyek), seperti **Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS**



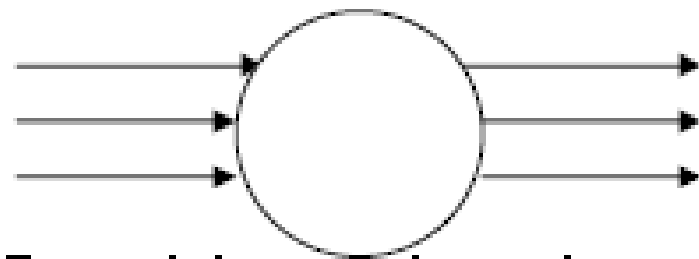
*1 input & 1 output*



*1 input & banyak output*



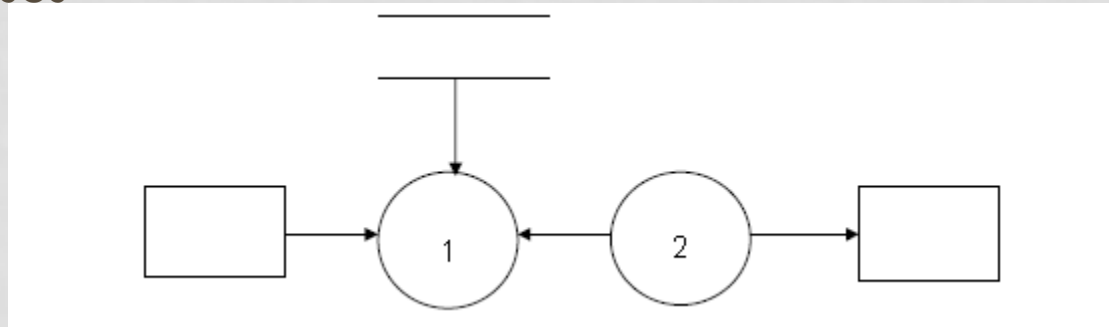
*Banyak input & 1 output*



*Banyak input & banyak output*

- Ada beberapa 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 profesional sistem digambarkan dengan komponen proses.

Berikut ini merupakan suatu contoh proses yang salah :  
Contoh proses



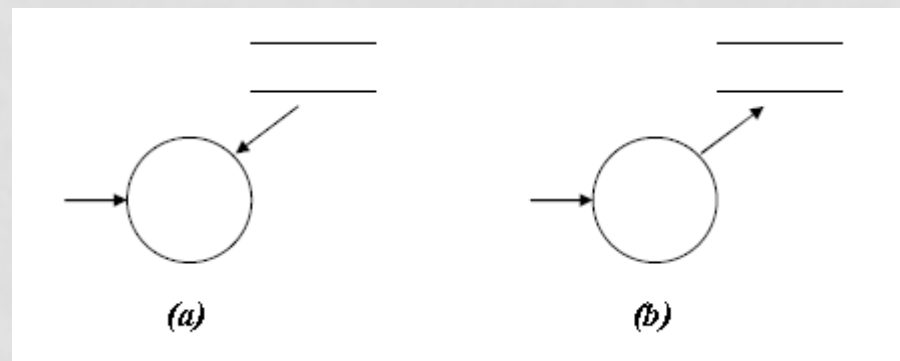
Umumnya kesalahan proses di DFD adalah :

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 (*lihat proses 1*).
2. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan **miracle** (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input (*lihat proses 2*).

# KOMPONEN DATA STORE

# KOMPONEN DATA STORE

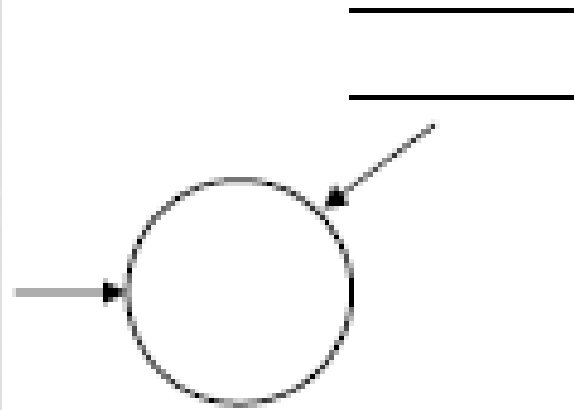
- **Komponen ini** digunakan untuk membuat model sekumpulan paket data dan **diberi nama** dengan **kata benda jamak**, misalnya ***Mahasiswa***



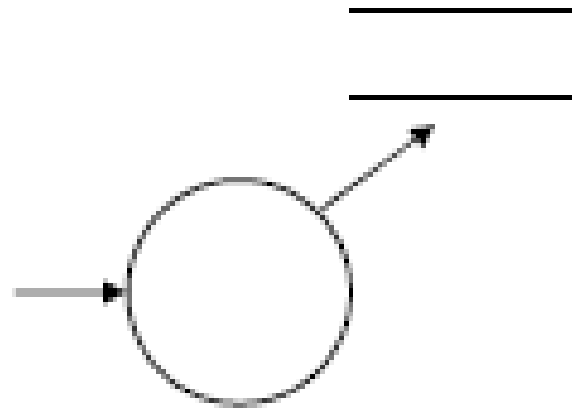
Suatu data store dihubungkan dengan alur data **hanya pada komponen proses**, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

1. **Alur data dari data store** yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses
2. **Alur data ke data store** yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih





**(a)**



**(b)**

# KOMPONEN DATA FLOW

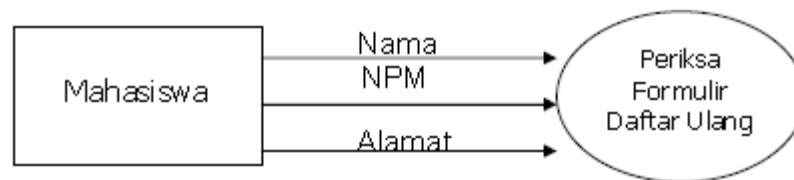
# KOMPONEN DATA FLOW

- Suatu data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.
- **Alur data** perlu **diberi nama** sesuai dengan data/informasi yang dimaksud, biasanya pemberian nama pada alur data dilakukan dengan menggunakan **kata benda**, contohnya ***Laporan Penjualan***

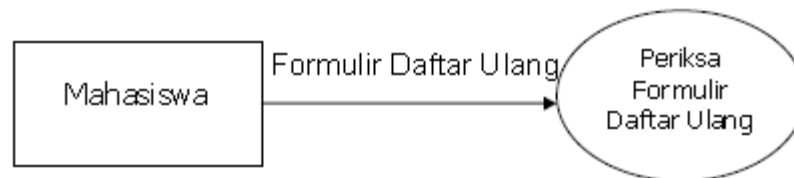
# KONSEP DALAM ALUR DATA

## 1. Konsep Paket Data (*Packets of Data*)

Apabila dua data atau lebih mengalir dari suatu sumber yang sama menuju ke tujuan yang sama dan mempunyai hubungan, dan harus dianggap sebagai satu alur data tunggal, karena data itu mengalir bersama-sama sebagai satu paket.



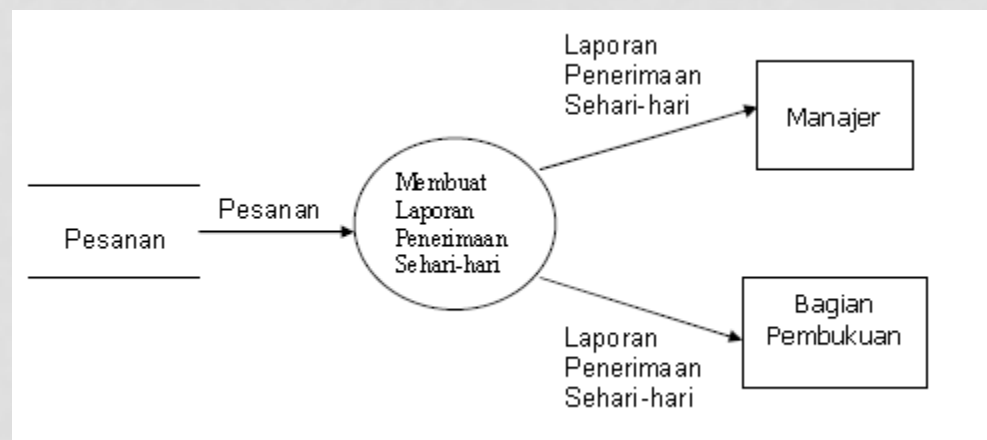
(a) Konsep paket data yang salah



(b) Konsep paket data yang benar

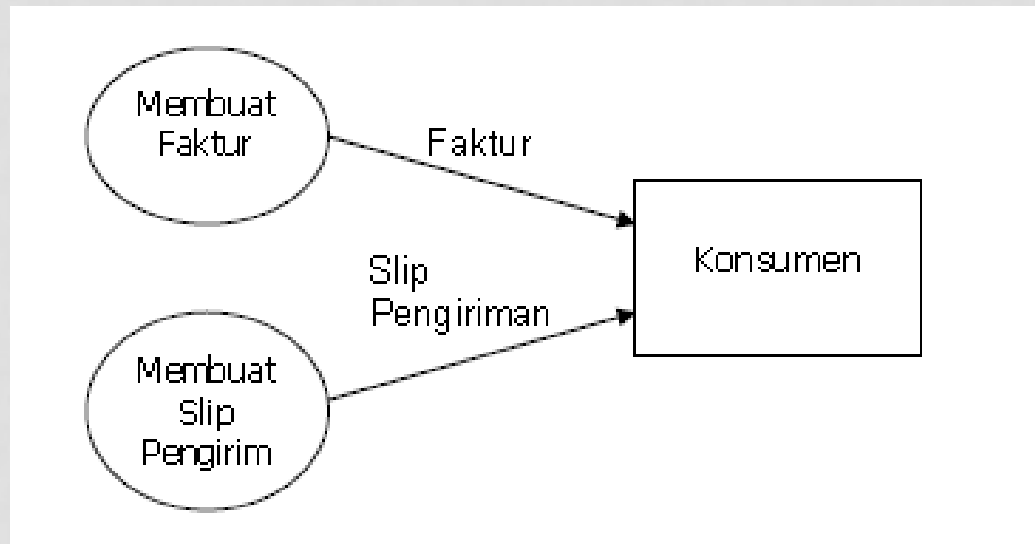
## 2. Konsep Alur Data Menyebar (*Diverging Data Flow*)

Alur data menyebar menunjukkan sejumlah tembusan paket data yang berasal dari *sumber yang sama* menuju ke *tujuan yang berbeda*, atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda, atau alur data ini membawa paket data yang memiliki nilai yang berbeda yang akan dikirim ke tujuan yang berbeda.



### 3. Konsep Alur Data Mengumpul (*Converging Data Flow*)

Beberapa alur data yang *berbeda sumber* bergabung bersama-sama menuju ke *tujuan yang sama*

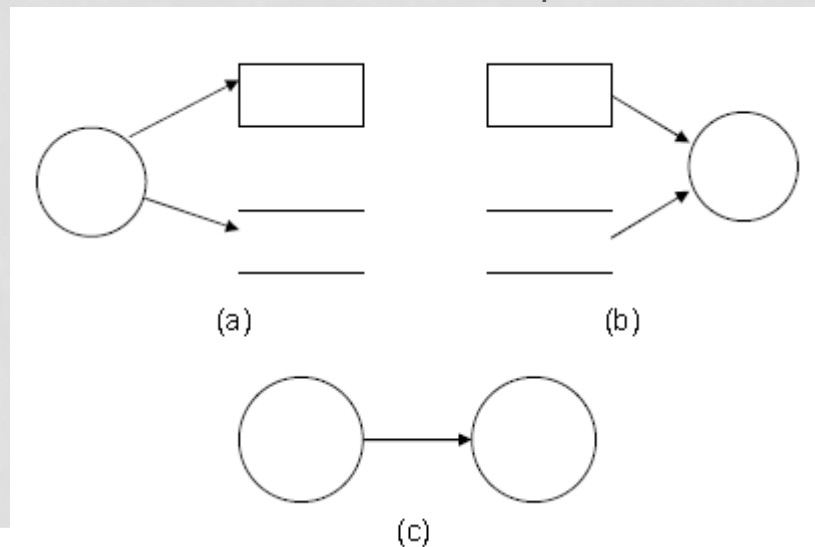


- 4. Konsep Sumber atau Tujuan Alur Data

Semua alur data harus **minimal mengandung satu proses**.

Maksud kalimat ini adalah :

1. Suatu alur data dihasilkan dari suatu *proses* dan menuju ke *suatu data store* dan/atau *terminator*
2. Suatu alur data dihasilkan dari suatu *data store* dan/atau *terminator* dan menuju ke suatu *proses*
3. Suatu alur data dihasilkan dari suatu *proses* dan menuju ke suatu *proses*



# SYARAT PEMBUATAN DFD



# 1. PEMBERIAN NAMA

- Pemberian nama pada komponen proses lebih baik menunjukkan aturan-aturan yang akan dilaksanakan oleh seseorang dibandingkan dengan memberikan nama atau identitas orang yang akan melaksanakannya
- Karena nama untuk komponen proses lebih baik menunjukkan tugas/fungsi sistem yang akan dilaksanakan, maka lebih baik pemberian nama ini menggunakan kata kerja
- Pemberian nama untuk komponen data store menggunakan kata benda, karena data store menunjukkan data apa yang disimpan untuk kebutuhan sistem dalam melaksanakan tugasnya. Jika sistem sewaktu-waktu membutuhkan data tersebut untuk melaksanakan tugasnya, maka data tersebut tetap ada, karena sistem menyimpannya.
- Begitu pula untuk komponen alur data, namanya lebih baik diberikan dengan menggunakan kata benda. Karena alur data ini menunjukkan data dan informasi yang dibutuhkan dan yang dikeluarkan oleh sistem dalam pelaksanaan tugasnya.

## 2. PEMBERIAN NOMOR PADA KOMPONEN PROSES

- Maksud pemberian nomor pada proses yang penting adalah untuk menunjukkan referensi terhadap skema penomoran secara hirarki pada levelisasi DFD. Dengan kata lain, nomor proses ini merupakan dasar pemberian nomor pada levelisasi DFD

### 3. PENGGAMBARAN DFD SESERING MUNGKIN

- Penggambaran DFD dapat dilakukan berkali-kali sampai secara teknik DFD itu benar, dapat diterima oleh pemakai, dan sudah cukup rapih sehingga profesional sistem tidak merasa malu untuk menunjukkan DFD itu kepada atasannya dan pemakai.

## 4. PENGHINDARAN PENGGAMBARAN DFD YANG RUMIT

Tujuan DFD adalah untuk membuat model fungsi yang harus dilaksanakan oleh suatu sistem dan interaksi antar fungsi. Tujuan lainnya adalah agar model yang dibuat itu mudah dibaca dan dimengerti tidak hanya oleh profesional sistem yang membuat DFD, tetapi juga oleh pemakai yang berpengalaman dengan subyek yang terjadi. Hal ini berarti DFD harus mudah dimengerti, dibaca, dan menyenangkan untuk dilihat.

## 5. PENGGAMBARAN DFD KONSISTEN

- Penggambaran DFD harus konsisten terhadap kelompok DFD lainnya. Profesional sistem menggambarkan DFD berdasarkan tingkatan DFD dengan tujuan agar DFD yang dibuatnya itu mudah dibaca dan dimengerti oleh pemakai sistem. Hal ini sesuai dengan salah satu tujuan atau syarat membuat DFD.

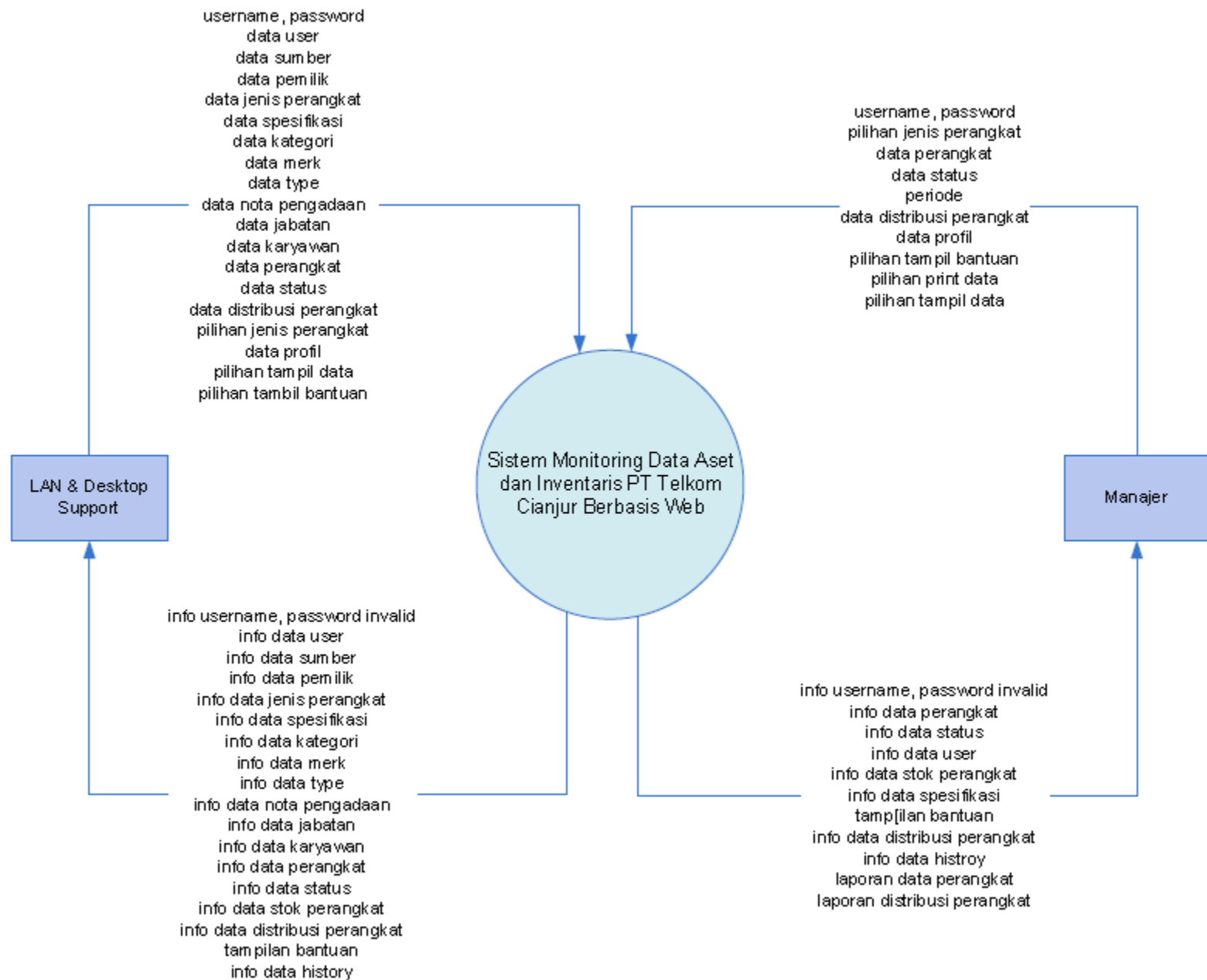
# PENGGAMBARAN DFD

1. Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.
2. Identifikasi semua input dan output yang terlibat dengan entitas luar.
3. ***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 terminator dari/ke sistem.
- Gambarkan diagram konteks.



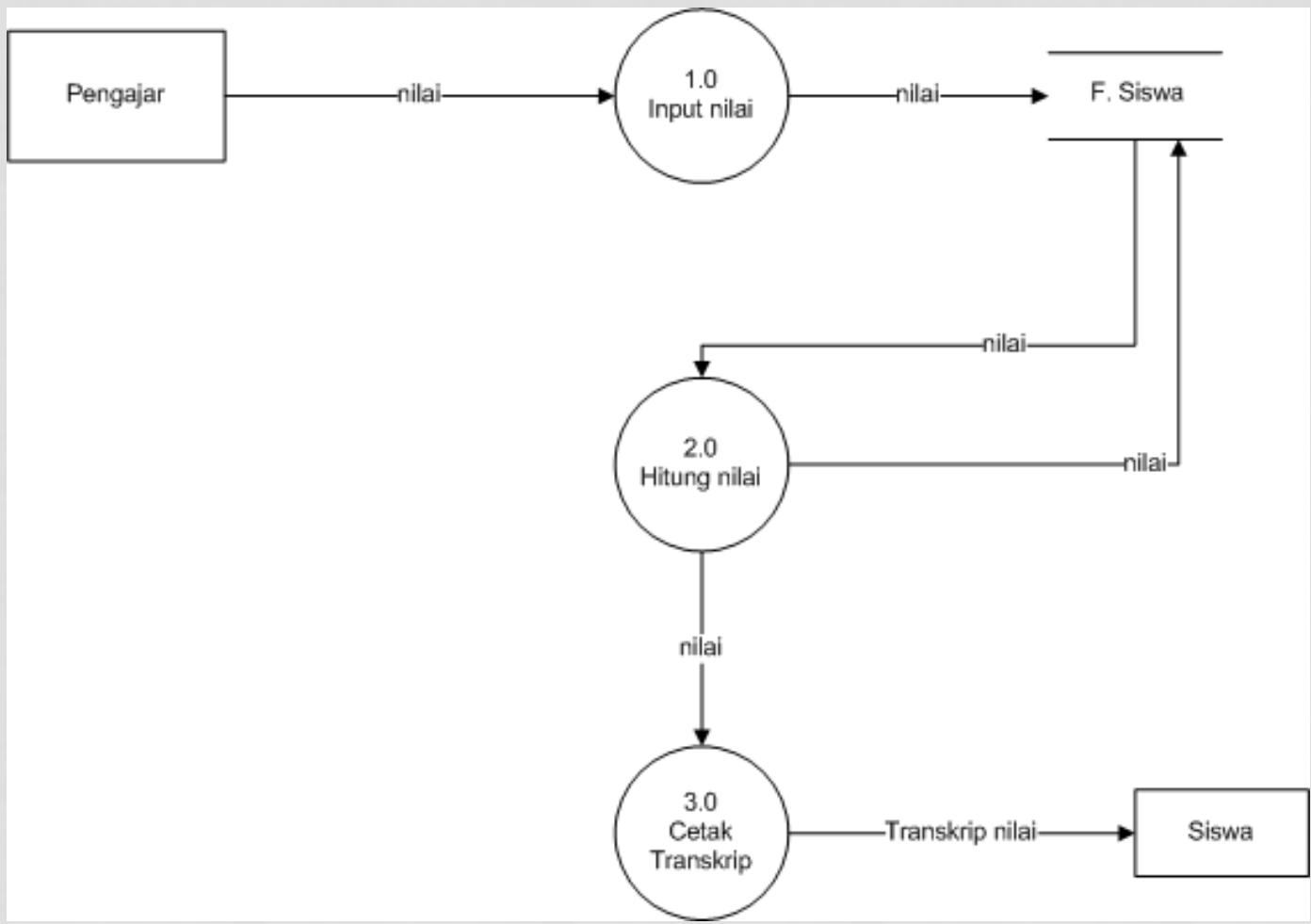


#### **4. Buat Diagram Level Zero**

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



## **5. Buat Diagram Level Satu**

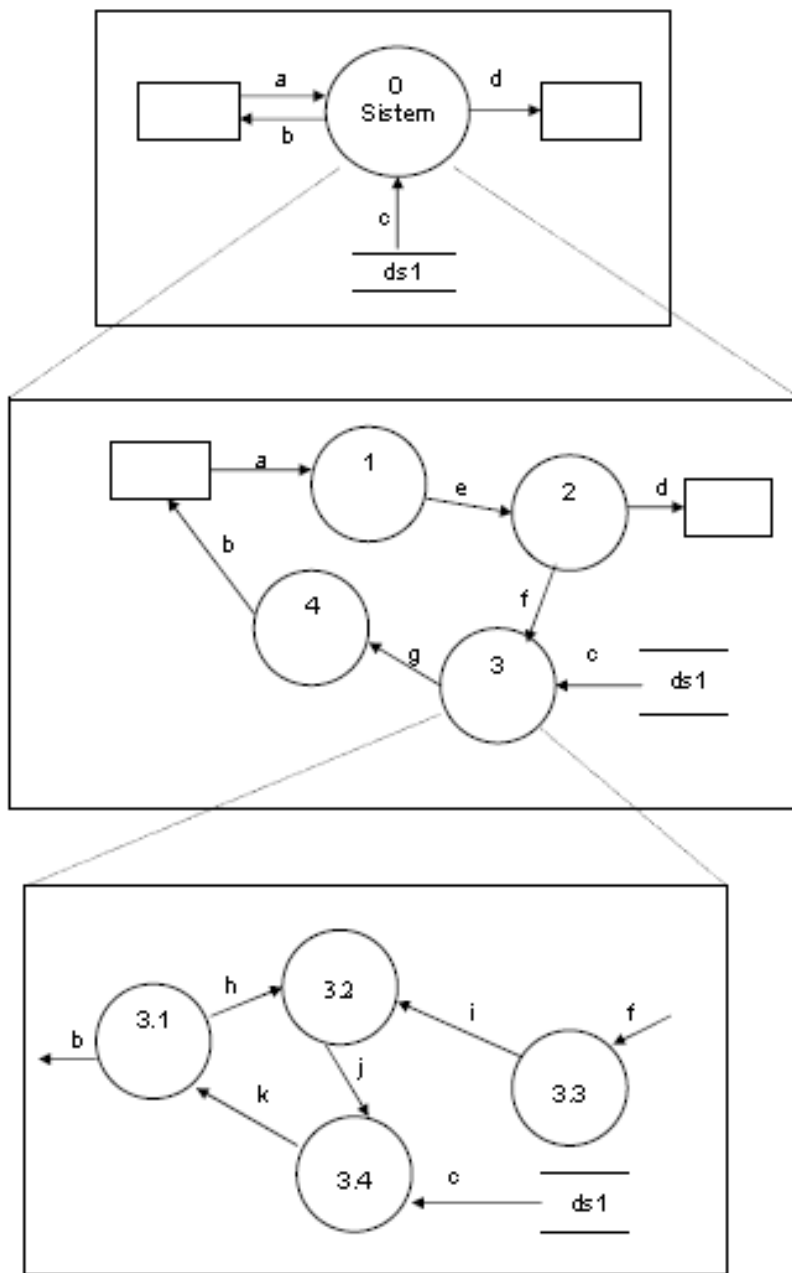
Diagram ini merupakan dekomposisi dari diagram level zero.

Caranya :

- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
- Tentukan apa yang diberikan/diterima masing-masing subproses ke/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

## 6. **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.



Gambar 11. Levelisasi DFD

# SPESIFIKASI PROSES

# FORMAT SPESIFIKASI PROSES

No Urut.	Proses	Keterangan
	No. Proses	
	Nama Proses	
	Source (sumber)	
	Input	
	Output	
	Destination (tujuan)	
	Logika Proses	

# CONTOH

No	Proses	Keterangan
1	No. Proses	1
	Nama Proses	Login
	Source	LAN&Desktop Support, Manajer
	Input	<i>username, password</i>
	Output	info <i>username, password</i> invalid
	Destination	LAN&Desktop Support, Manajer
	Logika Proses	<ol style="list-style-type: none"><li>1. LAN&amp;Desktop Support, Manajer memasukkan <i>username, password</i></li><li>2. Sistem mengecek apakah <i>username, password</i> yang dimasukkan kosong.</li><li>3. Apabila <i>username, password</i> yang dimasukkan</li></ol>



# KAMUS DATA DFD

# FORMAT KAMUS DATA

Nama	
Where used / how used	
Deskripsi	
Struktur Data	
[Penjelasan per struktur data]	

# CONTOH

Nama Aliran data	Data pemilik
Digunakan pada	LAN&Desktop Support Proses 2.2, 3.1
Deskripsi	Berisi data pemilik perangkat
Struktur data	ID_Pemilik + Pemilik
ID_Pemilik	[0-9]
Pemilik	[A-Z a-z]

# TUGAS

Buat analisis kebutuhan fungsional terhadap sistem informasi yang akan anda buat

(buat DFD, SPESIFIKASI PROSES dan KAMUS DATA )

TERIMAKASIH