

# UNIVERSITAS GUNADARMA

## ANALISA DAN PERANCANGAN SISTEM INFORMASI

Parno, SKom., MMSI

Email Personal  
[parno@staff.gunadarma.ac.id](mailto:parno@staff.gunadarma.ac.id)

Email Khusus Tugas  
[parno2012@gmail.com](mailto:parno2012@gmail.com)

Personal Website  
<http://parno.staff.gunadarma.ac.id>

Personal Blog  
<http://nustaffsite.gunadarma.ac.id/blog/parno>  
o

# PERANCANGAN SISTEM TERINCI (PROSES)

## DATA FLOW DIAGRAM

# DIAGRAM ARUS DATA (*DATA FLOW DIAGRAM*)

- ◉ 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.
- ◉ DFD 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.

# DEFINISI DFD

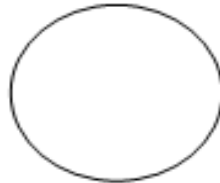
- DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Simbol yang digunakan DFD :

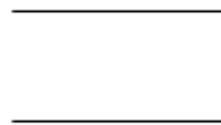
## Menurut Yourdan dan DeMarco



**Terminator**



**Proses**

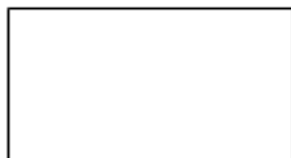


**Data Store**

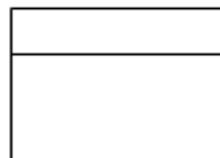


**Alur Data**

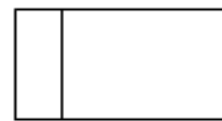
## Menurut Gene dan Serson



**Terminator**



**Proses**



**Data Store**



**Alur Data**

# 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 :
  - Terminator Sumber (*source*) → Terminator yang menjadi sumber.
  - Terminator Tujuan (*sink*) → Terminator yang menjadi tujuan data/informasi sistem.



**Terminator Sumber**

**Terminator Tujuan**

**T. Tujuan & Sumber**

- 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 diluar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.



## Tiga hal penting yang harus diingat tentang terminator :

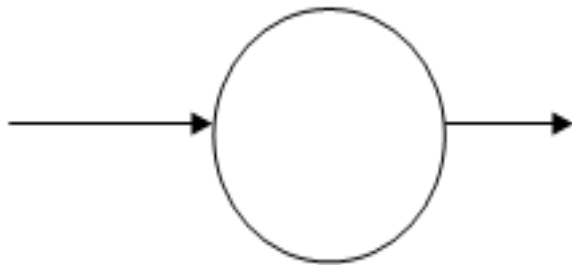
- ◉ Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
- ◉ Profesional Sistem Tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan terminator
- ◉ Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

## PROCESS (PROSES)

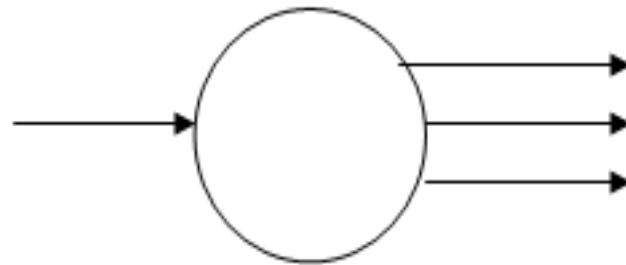
- ◉ Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
- ◉ Proses diberi nama untuk menjelaskan proses/ kegiatan apa yang sedang / akan dilaksanakan.
- ◉ Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek),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

# PROCESS (PROSES)

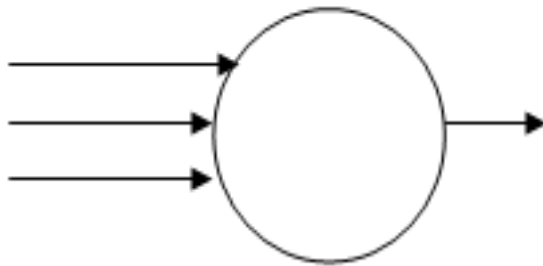
- ◉ Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



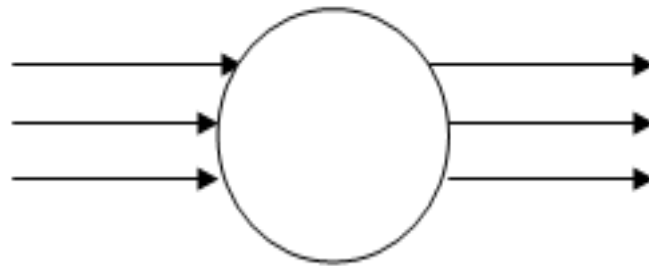
*1 input & 1 output*



*1 input & banyak output*



*Banyak input & 1 output*



*Banyak input & banyak output*

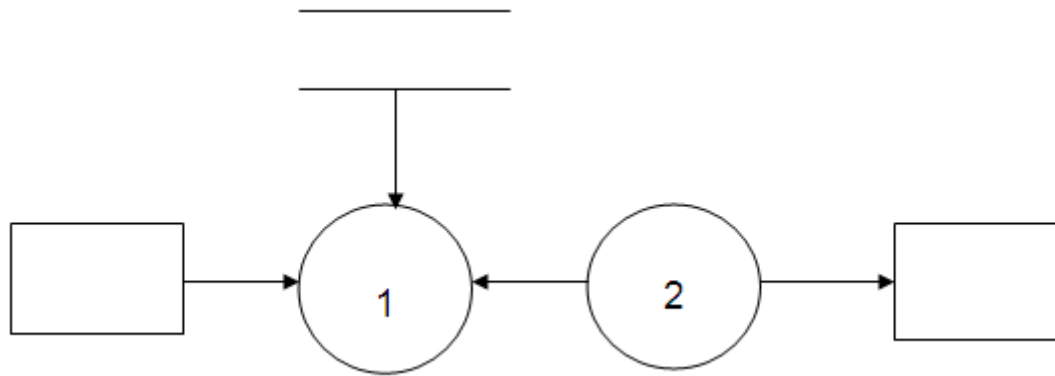


## PROCESS (PROSES)

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.

# PROCESS (PROSES)



- ◉ Gambar Diatas merupakan suatu contoh proses yang salah :  
Umumnya kesalahan proses di DFD adalah :
- ◉ 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*).
- ◉ 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*).

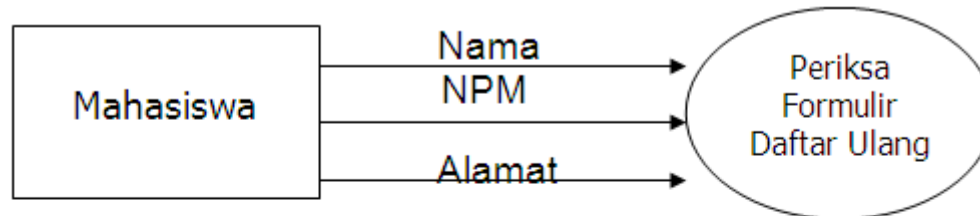
## DATA FLOW (ARUS DATA)

- ◉ 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.
- ◉ Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer.
- ◉ 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*.

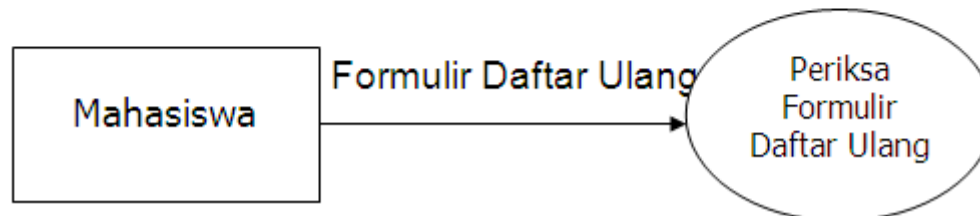
# KONSEP YANG PERLU DIPERHATIKAN DALAM PENGGAMBARAN ALUR DATA, YAITU :

## ⦿ 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.



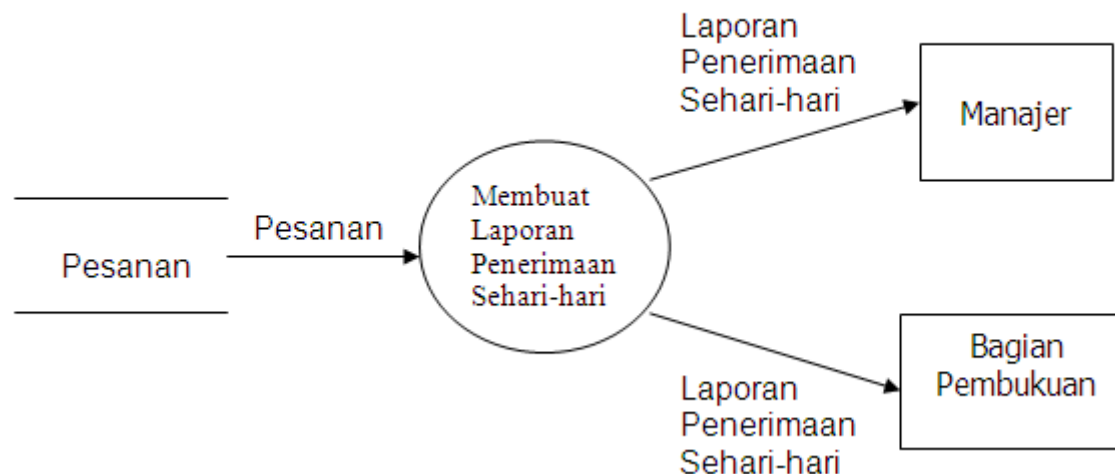
(a) Konsep paket data yang salah



## KONSEP YANG PERLU DIPERHATIKAN DALAM PENGGAMBARAN ALUR DATA, YAITU :

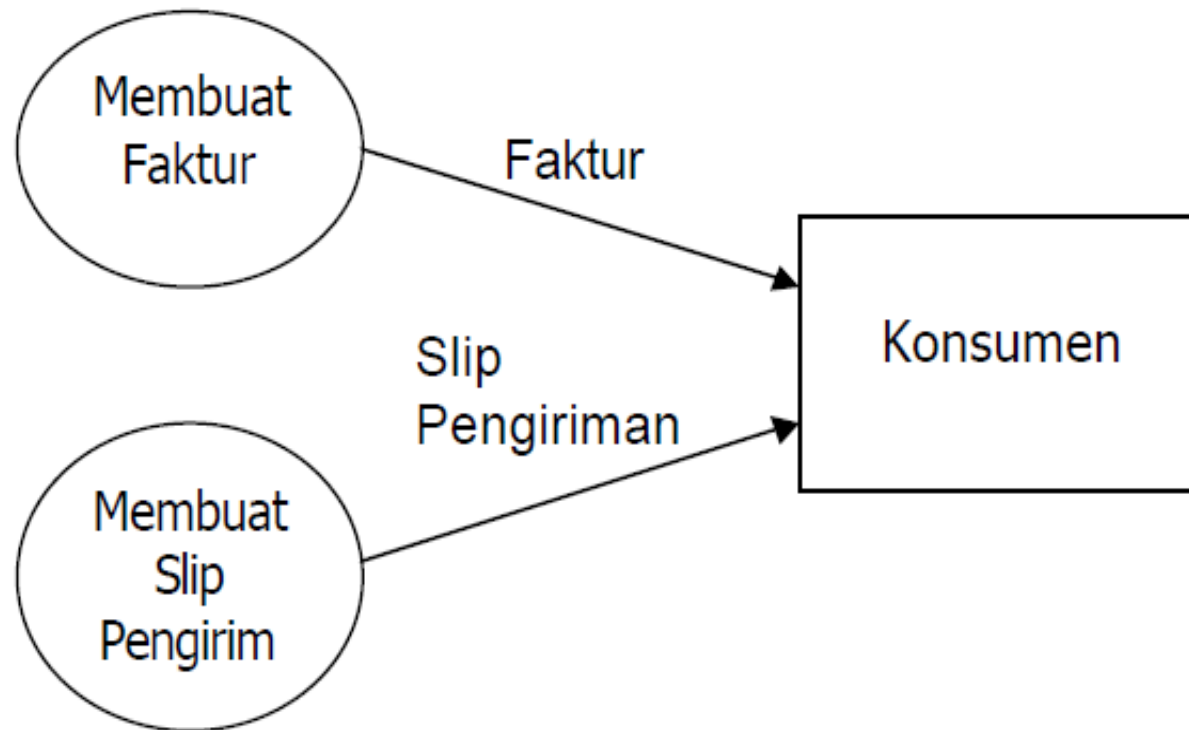
- ◉ **Konsep Alur Data Menyebar (*Diverging Data Flow*)**

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.



- Konsep Alur Data Mengumpul (*Converging Data Flow*)

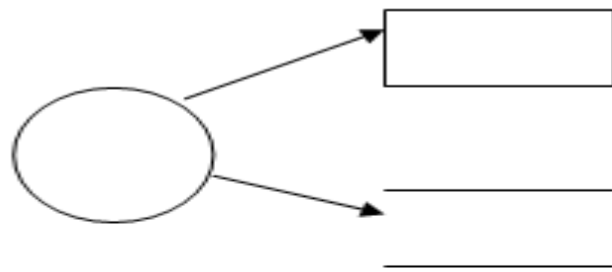
Beberapa alur data yang *berbeda sumber bergabung bersama-sama* menuju ke *tujuan yang sama*



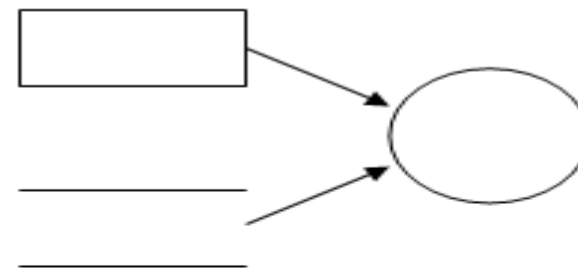


- **Konsep Sumber atau Tujuan Alur Data**

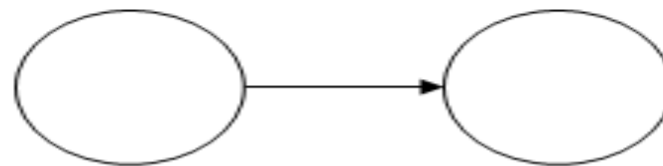
Semua alur data harus ***minimal mengandung satu proses***  
Suatu alur data dihasilkan dari suatu *proses* dan menuju ke *suatu data store* dan/atau *terminator* (*lihat gambar a*).  
Suatu alur data dihasilkan dari suatu *data store* dan/atau *terminator* dan menuju ke suatu *proses* (*lihat gambar b*).  
Suatu alur data dihasilkan dari suatu *proses* dan menuju ke suatu *proses* (*lihat gambar c*).



(a)



(b)



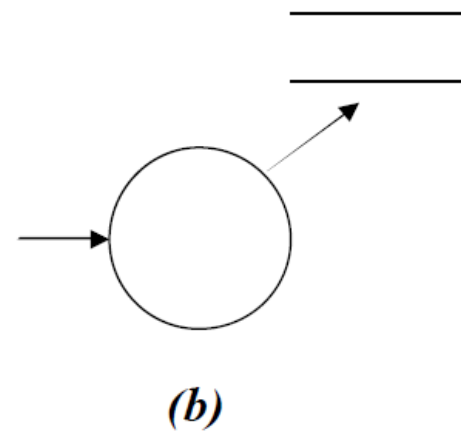
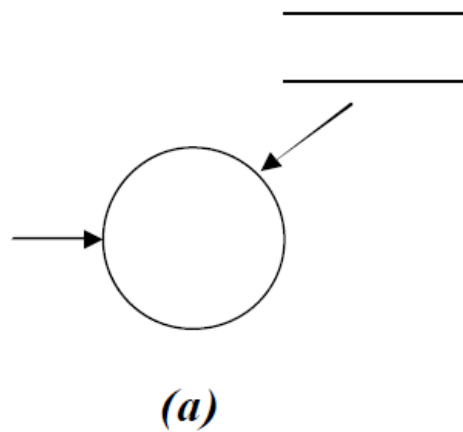
(c)



## *DATA STORE( SIMPANAN DATA)*

- ◉ Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.
- ◉ 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 :

- ◉ 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 (*lihat gambar a*).
- ◉ 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 (*lihat gambar b*).



## **Syarat-syarat pembuatan DFD:**

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 konsiten secara logika

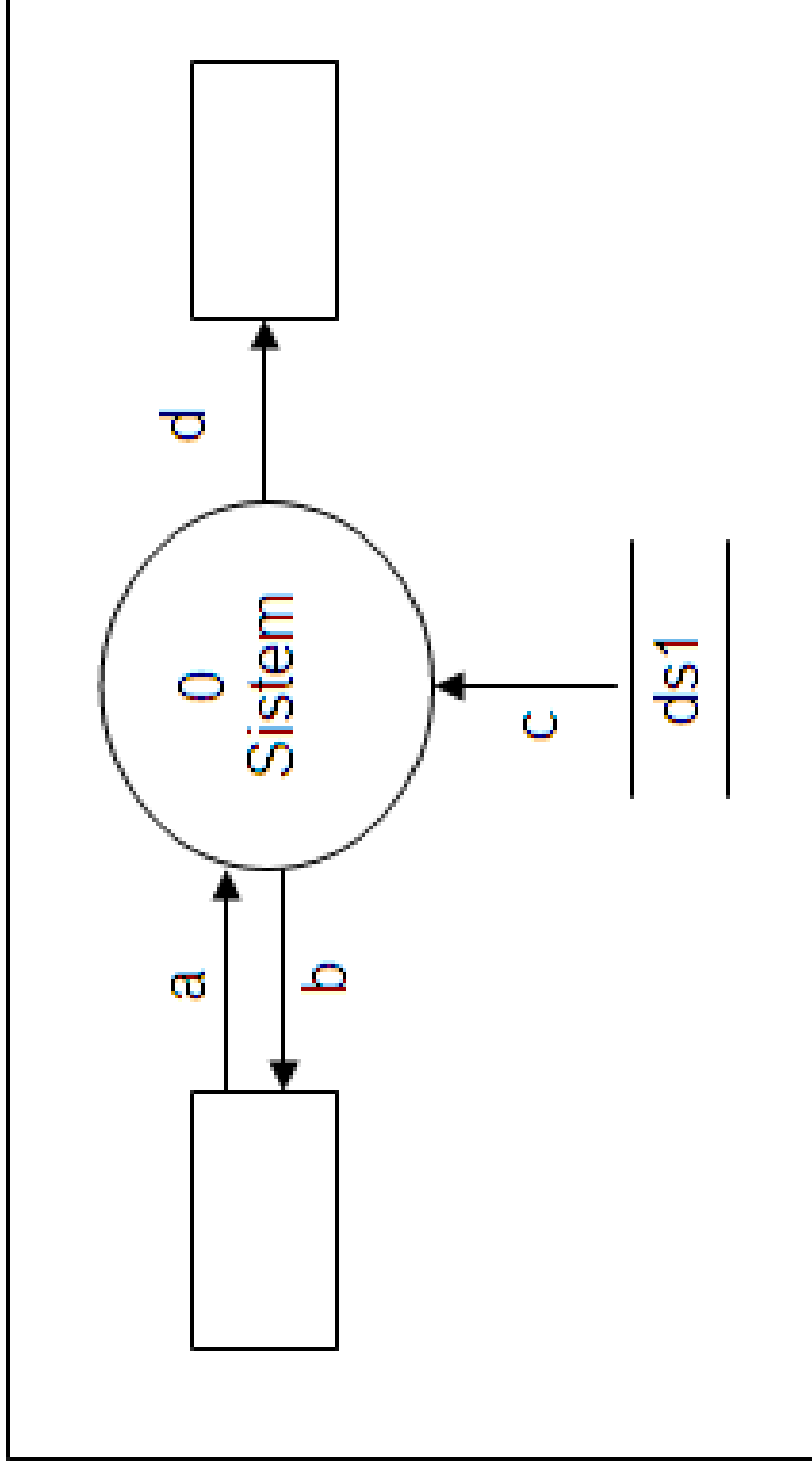


# PENGGAMBARAN DFD

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

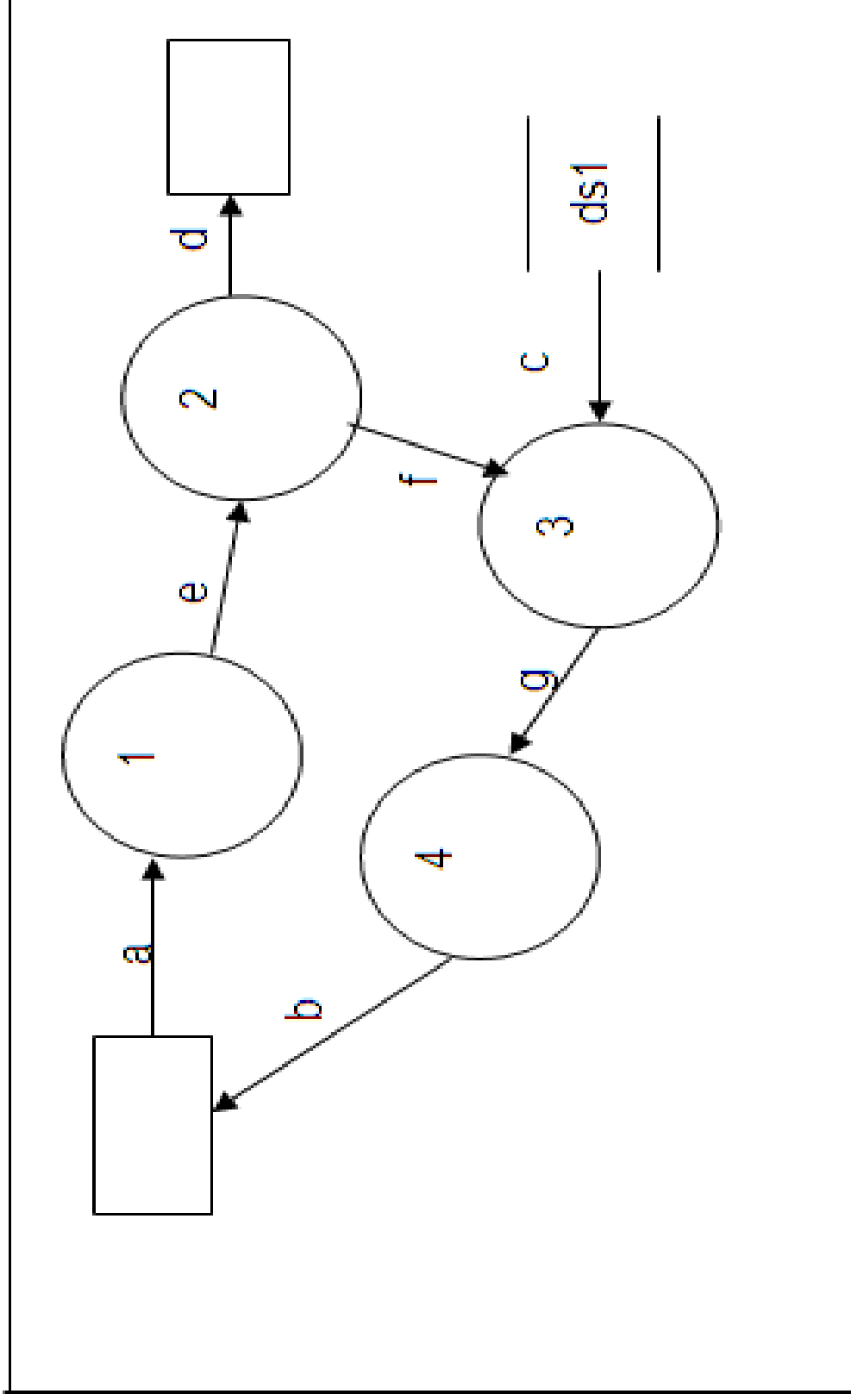
- ◉ Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem
- ◉ Identifikasi semua input dan output yang terlibat dengan entitas luar
- ◉ Buat Diagram Konteks (*diagram context*)
- ◉ Buat bagan berjenjang untuk semua proses yang ada di sistem terlebih dahulu
- ◉ Gambar sketsa DAD untuk *overview diagram* (level 0) berdasarkan proses dibagian berjenjang
- ◉ Gambar DAD untuk level-level berikutnya (level1 dan seterusnya)
- ◉ Setelah semua level DAD digambar, gambar DAD untuk pelaporan manajemen yang digambar terpisah

# LEVELISASI DFD (LEVEL KONTEKS)





# LEVELISASI DFD (LEVEL ZERO)



# LEVELISASI DFD (LEVEL 1)

