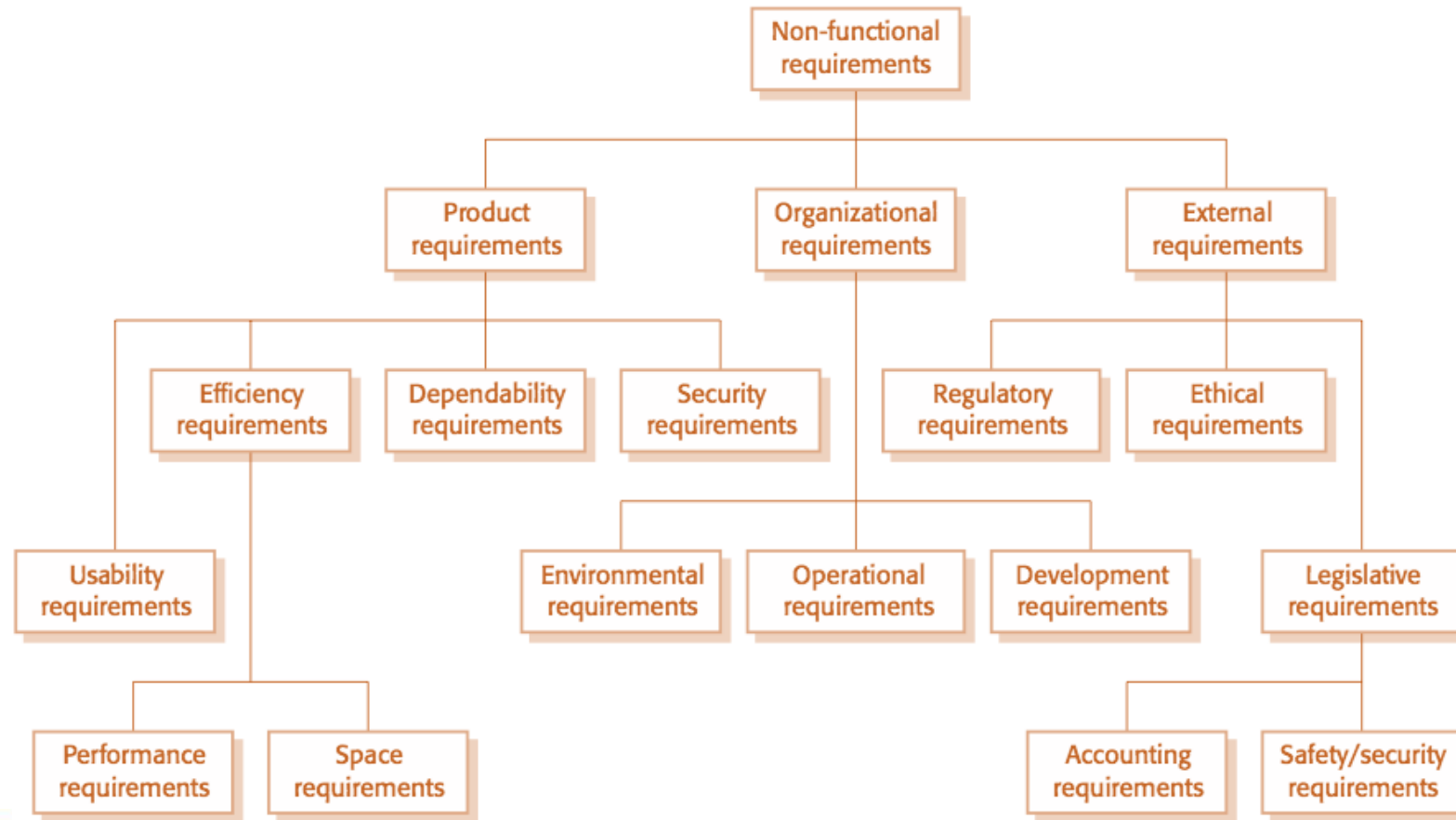


ANALISIS KEBUTUHAN PERANGKAT LUNAK BAG - 2

ANALISIS KEBUTUHAN NON FUNGSIONAL

“*Penguraian kebutuhan-kebutuhan non fungsional* (batasan sistem) menjadi *kebutuhan yang lebih detail* berdasarkan pada *fakta* dan *kebutuhan* yang ada.”

JENIS -JENIS KEBUTUHAN NON FUNGSIONAL



ANALISIS KEBUTUHAN PERANGKAT LUNAK (1)

“Penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi perangkat lunak dan berhubungan dengan proses pembangunan perangkat lunak.”

ANALISIS KEBUTUHAN PERANGKAT LUNAK (2)

Contoh :

- **Kebutuhan Non Fungsional:**

SKPL-NF-001 -Sistem yang dibangun berbasis web dinamis

- **Fakta Perangkat Lunak (Yang ada di lingkungan sistem):**

- Sistem operasi platform windows
- Belum ada web server
- Code Editor berupa notepad.

ANALISIS KEBUTUHAN PERANGKAT LUNAK (3)

- **Kebutuhan Perangkat Lunak:**
 - Sistem operasi (platform bebas)
 - Web Server WAMP versi 2 sebagai web server
 - Code Editor berupa Adobe Dreamweaver CS6.
- **Kesimpulan (Hasil perbandingan fakta dengan kebutuhan):**

Dibutuhkan pengadaan perangkat lunak berupa WAMP versi 2 dan Adobe Dreamweaver CS6 berdasarkan kebutuhan non fungsional dengan kode SKPL-NF-001.

ANALISIS KEBUTUHAN PERANGKAT KERAS (1)

“Penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi perangkat keras dan berhubungan dengan proses pembangunan perangkat lunak.”

ANALISIS KEBUTUHAN PERANGKAT KERAS (2)

Contoh :

- **Kebutuhan Non Fungsional:**

SKPL-NF-002 -Sistem dibangun dengan spesifikasi grafis yang tinggi

- **Fakta Perangkat Keras (Yang ada di lingkungan sistem):**

- Komputer dengan VGA on board 128 MB.
- Monitor LCD dengan resolusi 1366x768

ANALISIS KEBUTUHAN PERANGKAT KERAS (3)

- **Kebutuhan Perangkat Keras:**
 - Komputer dengan VGA dedicated 1 GB
 - Monitor LCD dengan resolusi 1366x768.
- **Kesimpulan (Hasil perbandingan fakta dengan kebutuhan):**

Dibutuhkan pengadaan perangkat keras berupa VGA dedicated 1 GB berdasarkan kebutuhan non fungsional dengan kode SKPL-NF-002.

ANALISIS KEBUTUHAN PENGGUNA (1)

“Penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi pengguna dan berhubungan dengan proses pembangunan perangkat lunak.”

ANALISIS KEBUTUHAN PENGGUNA (2)

Contoh :

- **Kebutuhan Non Fungsional:**

SKPL-NF-003 –Admin sistem minimal memiliki kemampuan untuk mengatasi permasalahan di sistem.

- **Fakta Perangkat Pikir (Yang ada di lingkungan sistem):**

Stakeholder	Tanggung Jawab	Tingkat Pendidikan	Tingkat Keterampilan yang Dimiliki	Pengalaman Menggunakan Komputer

ANALISIS KEBUTUHAN PENGGUNA (3)

- **Kesimpulan (Hasil perbandingan fakta dengan kebutuhan):**
Diisi kebutuhan perangkat pikir apabila terdapat ketidakcocokan spesifikasi pengguna antara fakta dan kebutuhan.
- **Fakta Perangkat Pikir (Yang ada di lingkungan sistem) :**

Pengguna Sistem	Hak Akses	Tingkat Keterampilan yang Harus Dimiliki	Pengalaman yang Harus Dimiliki	Jenis Pelatihan yang Akan diberikan

ANALISIS JARINGAN DAN PENGKODEAN

- Analisis jaringan dilakukan apabila system yang dibangun berbasis jaringan, dimana: *“Identifikasi spesifikasi Jaringan Lokal yang ada maupun yang dibutuhkan dalam pembangunan perangkat lunak.*
- Analisis pengkodean adalah *“Identifikasi pengkodean yang telah digunakan pada sistem berjalan dengan maksud mengetahui format pengkodean.”*

Contoh : seperti format NIM

ANALISIS DATA



Dokumen Manual

Kebutuhan Data
Lainnya yang
Muncul Akibat
Analisis

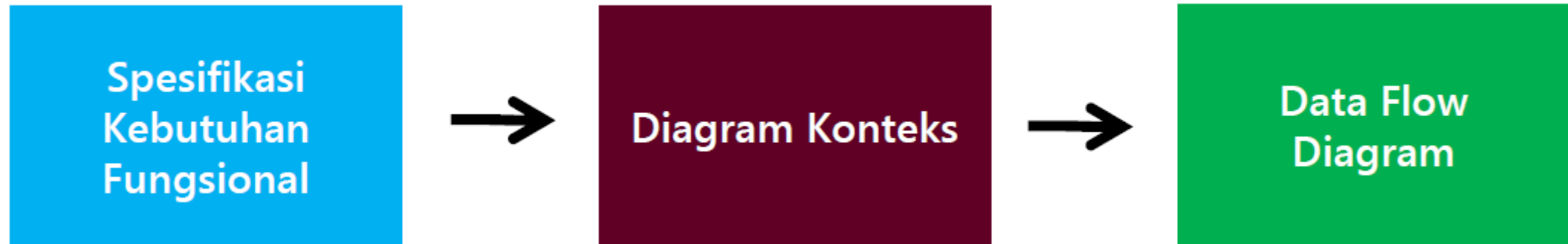


Identifikasi Data



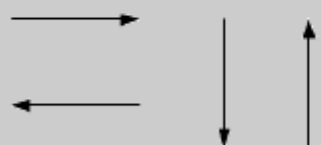



Entity Relationship
Diagram

ANALISIS KEBUTUHAN FUNGSIONAL



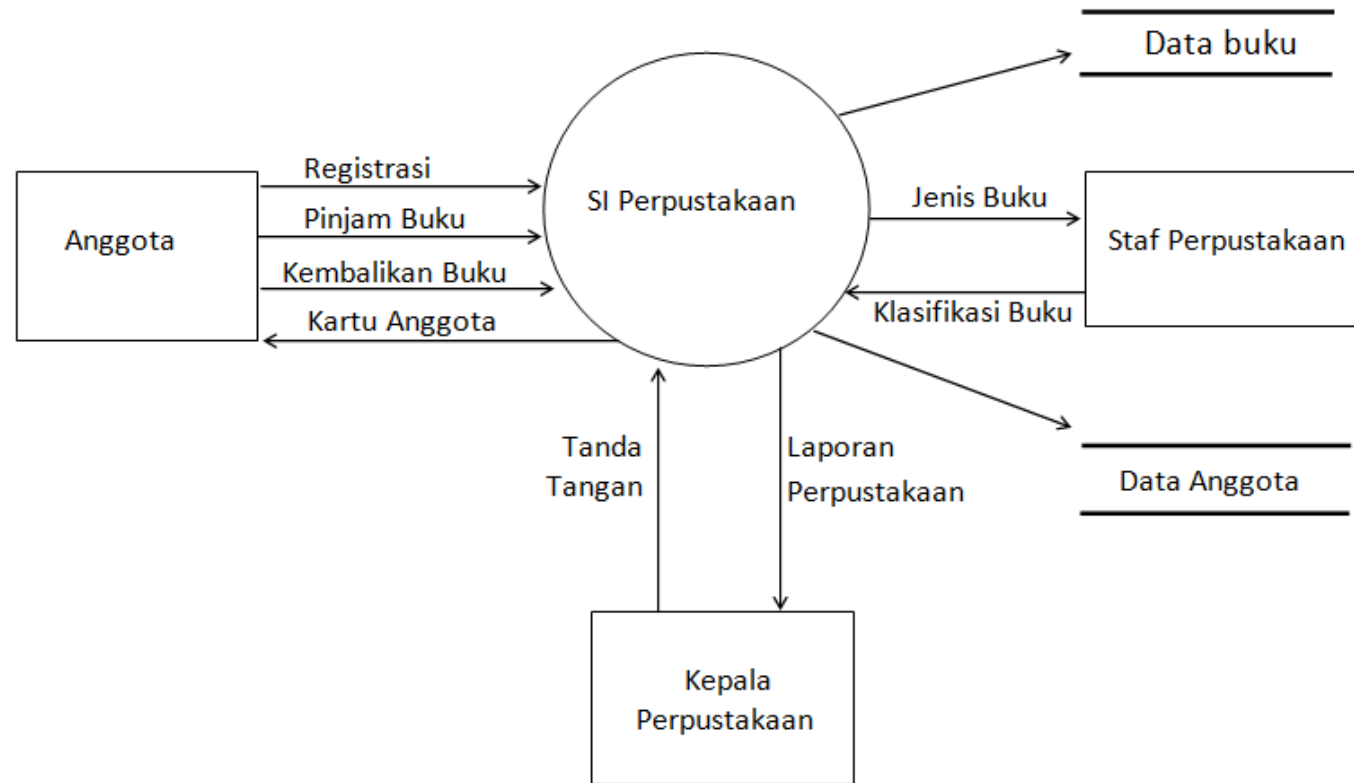
SIMBOL DIAGRAM KONTEKS

SIMBOL	NAMA SIMBOL	FUNGSI
	Entitas Luar	Menggambarkan entitas eksternal yang berhubungan dengan sistem
	Sistem(konteks)/ Proses(DFD)	Menggambarkan proses yang ada dalam suatu sistem
	Aliran Data/Informasi	Menggambarkan aliran data antar proses, data store dan entitas luar
	Data Store	Menggambarkan tempat penyimpanan data di dalam sistem



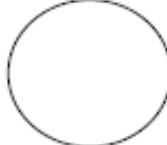

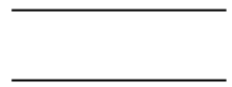


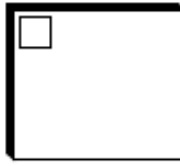

RULE OF THUMB : DIAGRAM KONTEKS

1. Memodelkan **aliran data** dari entitas luar ke dalam sistem.
2. Sistem masih dianggap **kesatuan yang utuh**.
3. Entitas luar bisa berupa **pengguna, mesin**, ataupun **database** yang berada di luar sistem tapi berhubungan dengan sistem.
4. Garis masuk dari entitas luar ke dalam sistem menggambarkan **inputs** sedangkan garis keluar dari sistem ke entitas luar menggambarkan **output**.

CONTOH : DIAGRAM KONTEKS



SIMBOL DATA FLOW DIAGRAM

	Yourdan	C.Gane dan T.Sarson
Aliran data / Data flow		
Proses / Process		
Simpanan data / Data store		
Kesatuan luar, batas sistem / External entity, boundary		
Aliran fisik / Material flow		

RULE OF THUMB : DATA FLOW DIAGRAM

1. Memodelkan **proses beserta aliran data** setiap prosesnya.
2. DFD merupakan **breakdown** dari diagram konteks.
3. Peletakan entitas luar harus **konsisten** supaya mudah dibaca.
4. Data store yang ada pada sistem dimunculkan.
5. Garis aliran data dari entitas luar ke dalam proses harus konsisten baik **secara jumlah maupun penamaan**.
6. DFD bisa dibreakdown sampai level yang “**cukup**”.
7. DFD yang mempunyai level besar merupakan turunan dari DFD dengan level yang lebih kecil.
8. Konsistensi jumlah dan penamaan aliran data harap diperhatikan dari DFD level sebelumnya.

KOMPONEN DFD (1)

1. Proses
2. Sumber (*Source*) dan Tujuan (*sink*) atau *Terminator*
3. Penyimpanan Data (*Data Store/File*)
4. Aliran Data (*Data Flow*)

KOMPONEN DFD (2)

1. Proses

- *Proses* merupakan bagian dari *DFD* yang mengubah (*transform*) satu atau lebih masukan/*inputs* (aliran data yang masuk) menjadi keluaran-keluaran/*outputs* (aliran data yang keluar).
- Nama lain dari *proses* adalah *buble* (bulatan), *function* dan *transform*.
- Notasi dari proses :



Tom De Marco/Yourdon



Gane & Sarson

KOMPONEN DFD (3)

1. Proses

- Pemberian *Nama Proses* :
 - Peorse dapat diberi nama menggunakan sebuah *kata, ungkapan atau kalimat sederhana*.
 - *Nama Proses* menjelaskan *apa yang dikerjakan oleh proses (what the process does)*.
 - Sebuah *Nama Proses* yang bagus umumnya terdiri dari *sebuah ungkapan kata kerja – obyek (a verb - object phrase)* seperti PERIKSA PESANAN atau HITUNG GAJI KOTOR.
 - Dalam beberapa kasus, proses mendeskripsikan **siapa** atau **apa** yang melaksanakan proses dan bukan **apa** proses tersebut. Sebagai contoh SISTEM INFORMASI PEMASARAN.

KOMPONEN DFD (4)

2. Sumber dan Tujuan

- *Sumber atau Tujuan* adalah seseorang atau sekelompok orang atau suatu organisasi luar, atau suatu kelompok atau departemen di dalam perusahaan/organisasi yang sama, tetapi di luar kendali dari sistem yang dibuat modelnya.
- Nama lain dari *Sumber atau Tujuan* ialah *terminator* dan *external entity*.
- Notasi untuk sumber atau tujuan :



KONSUMEN

KOMPONEN DFD (5)

2. Sumber dan Tujuan

- *Terminators* berada di luar sistem yang dimodelkan; aliran-aliran yang menghubungkan *terminators* dengan berbagai proses di dalam sistem menunjukkan *interface* antara sistem dengan dunia luar.
- *Terminators* mendefinisikan “*boundary*” atau ruang-lingkup dari sistem yang dimodelkan.
- Akibatnya, baik *analisis sistem* atau *perancang sistem* tidak boleh/tidak seharusnya mengubah isi dari *terminator* atau bagaimana *terminator* bekerja.
- Ketika *ruang lingkup berubah*, *terminators* bisa berubah menjadi proses-proses dan sebaliknya.
- Setiap hubungan yang ada di antara *terminators* tidak akan ditunjukkan dalam model DFD.

KOMPONEN DFD (6)

3. Penyimpanan Data

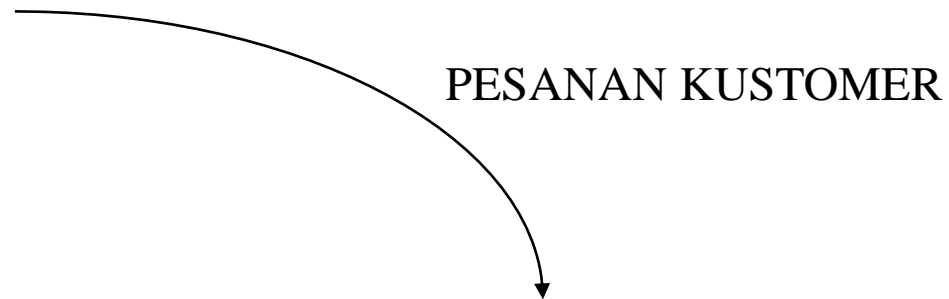
- *Penyimpanan data* merupakan bagian dari DFD yang digunakan untuk menunjukkan suatu *kumpulan dari paket data* yang diam.
- *Penyimpanan data* juga disebut : *data* store atau *file*
- *Penyimpanan data* diperlukan sebagai *time-delayed storage area* di antara 2 proses yang terjadi pada waktu yang berbeda.
- Notasi Penyimpanan Data :



KOMPONEN DFD (7)

4. Aliran Data

- *Aliran data* digunakan untuk menunjukkan *pergerakan dari paket data* atau *informasi* dari satu bagian sistem ke bagian yang lain.
- Notasi dari Aliran Data :



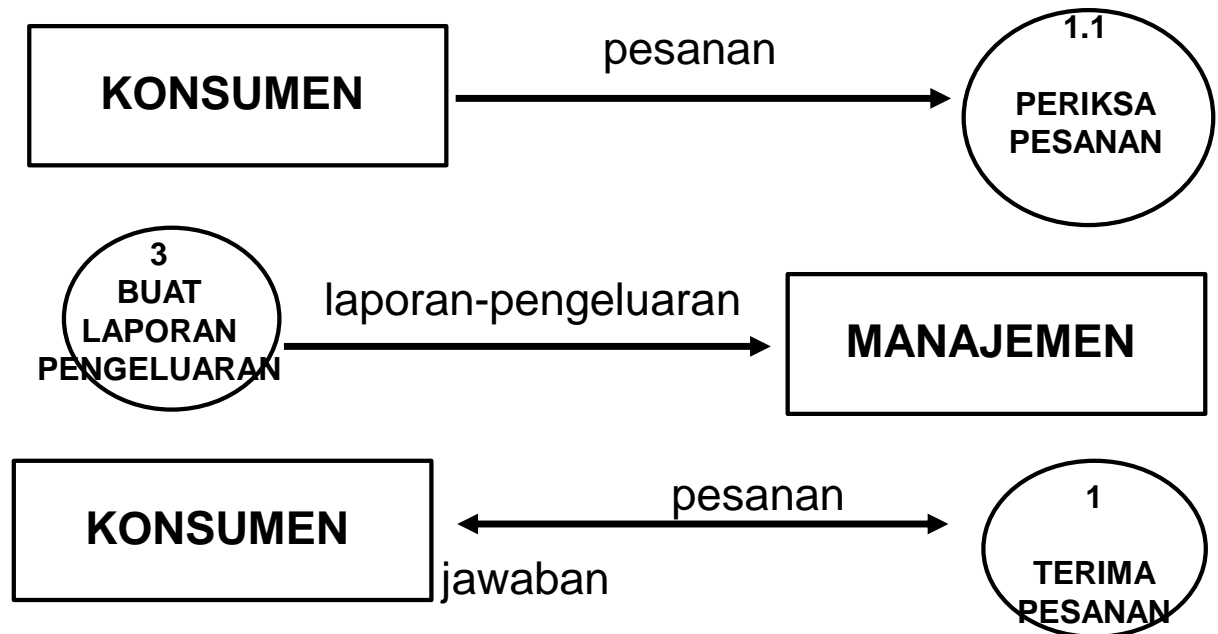
Contoh sebuah data flow dengan nama:
PESANAN KUSTOMER

KOMPONEN DFD (8)

4. Aliran Data

- *Interface Aliran Data* dengan *Terminator*.

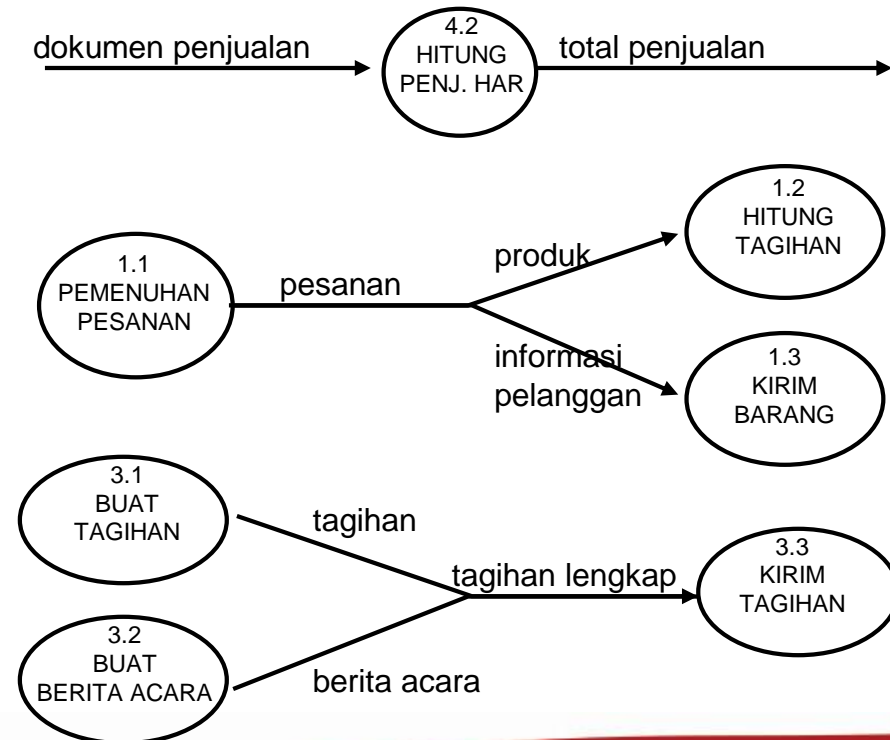
- *Input Data* ke *sistem* dari *sumber diluar sistem*.
- *Output Data* ke *tujuan diluar sistem*.
- *Interaktif Dialog* antara *sistem sumber* atau *tujuan*.



KOMPONEN DFD (9)

4. Aliran Data

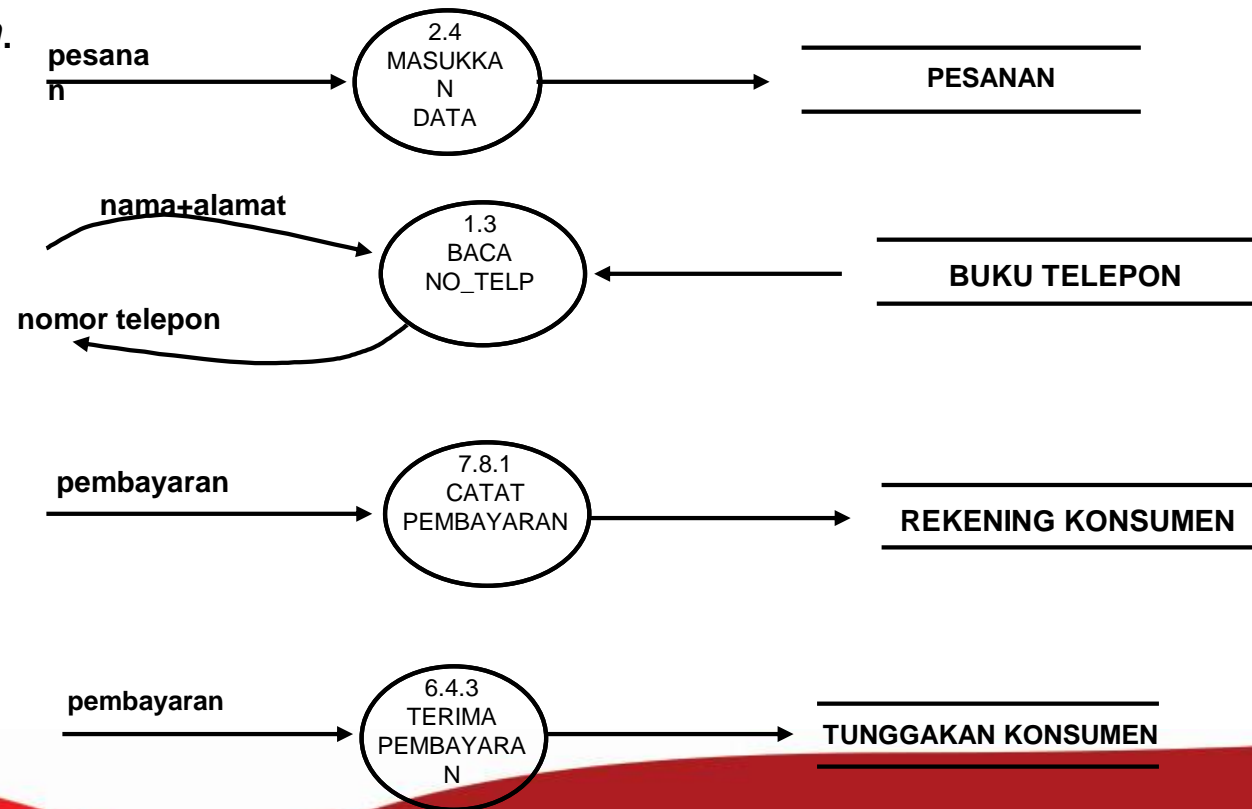
- *Interface di dalam sistem*
 - *Pemrosesan Aliran Data*
(Pengulangan, Decision, dll-nya).
 - Aliran Data *Divergen*.
 - Aliran Data *Konvergen*.



KOMPONEN DFD (10)

4. Aliran Data

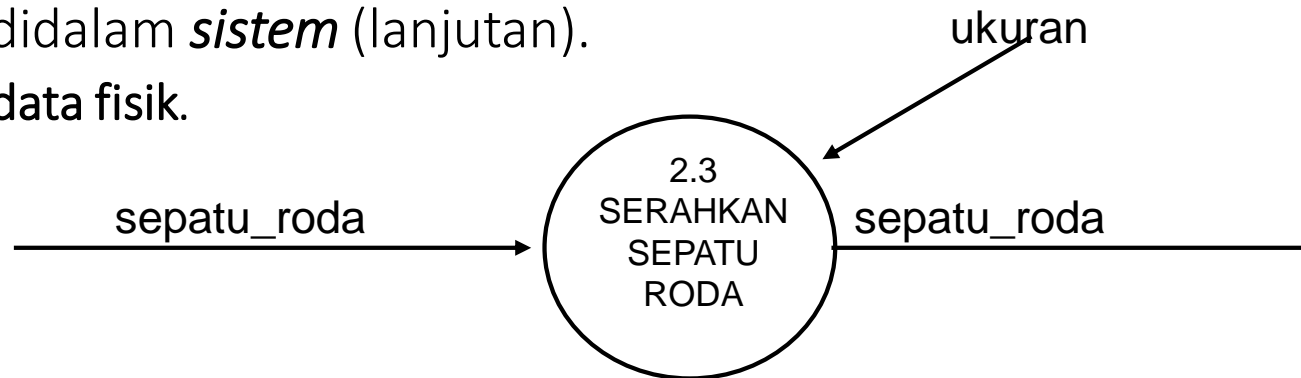
- **Interface** terhadap **penyimpanan data**.
 - Suatu proses yang **write data** ke penyimpanan data.
 - Suatu proses yang **read data** dari penyimpanan data.
 - Suatu proses yang **update data** didalam penyimpanan data.
 - Suatu proses yang **delete data** didalam penyimpanan data.



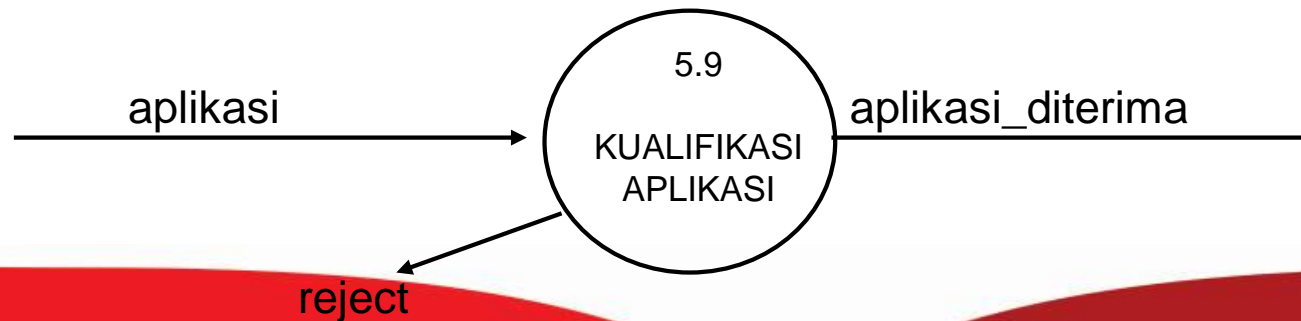
KOMPONEN DFD (11)

4. Aliran Data

- *Interface* didalam *sistem* (lanjutan).
 - Aliran data fisik.



- *Aliran data yang ditolak (reject)* oleh *sistem* dan *tidak diproses lebih lanjut*.

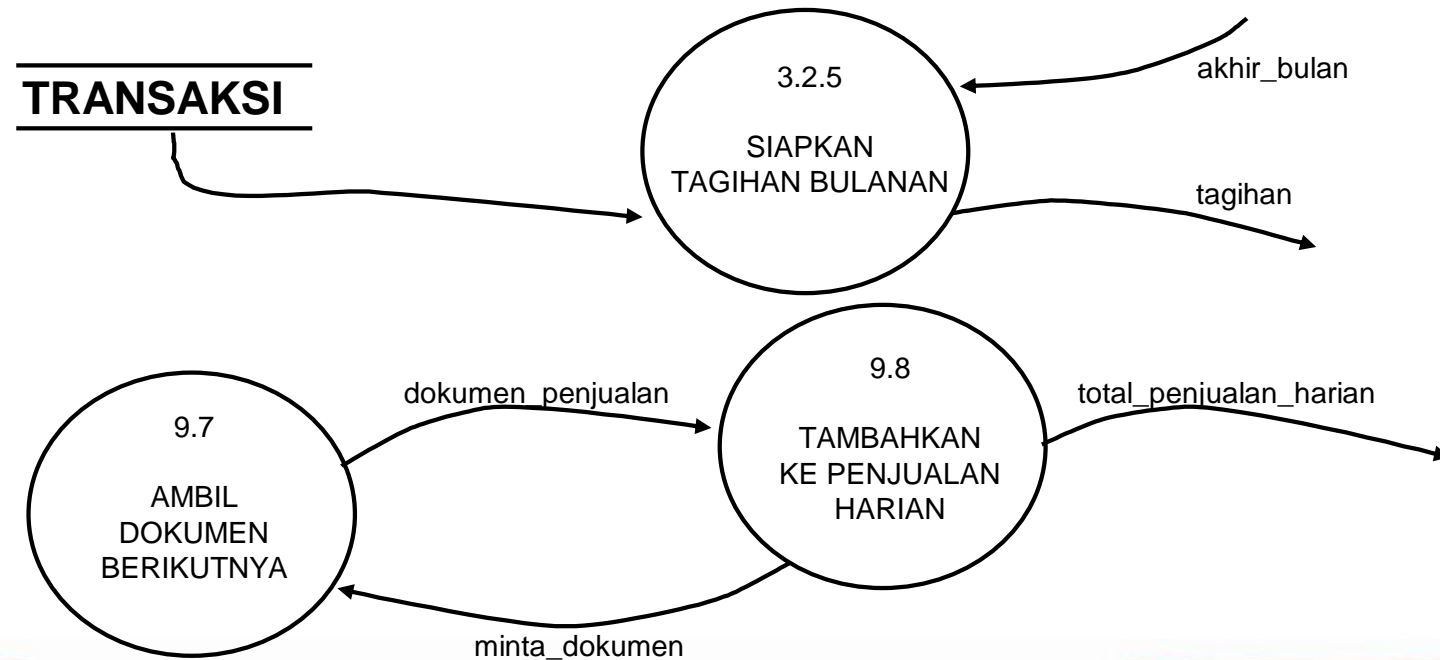


KOMPONEN DFD (12)

4. Aliran Data

- Aliran Data yang *tidak memenuhi syarat* untuk sistem yang *non real time*.

- *Input signal*



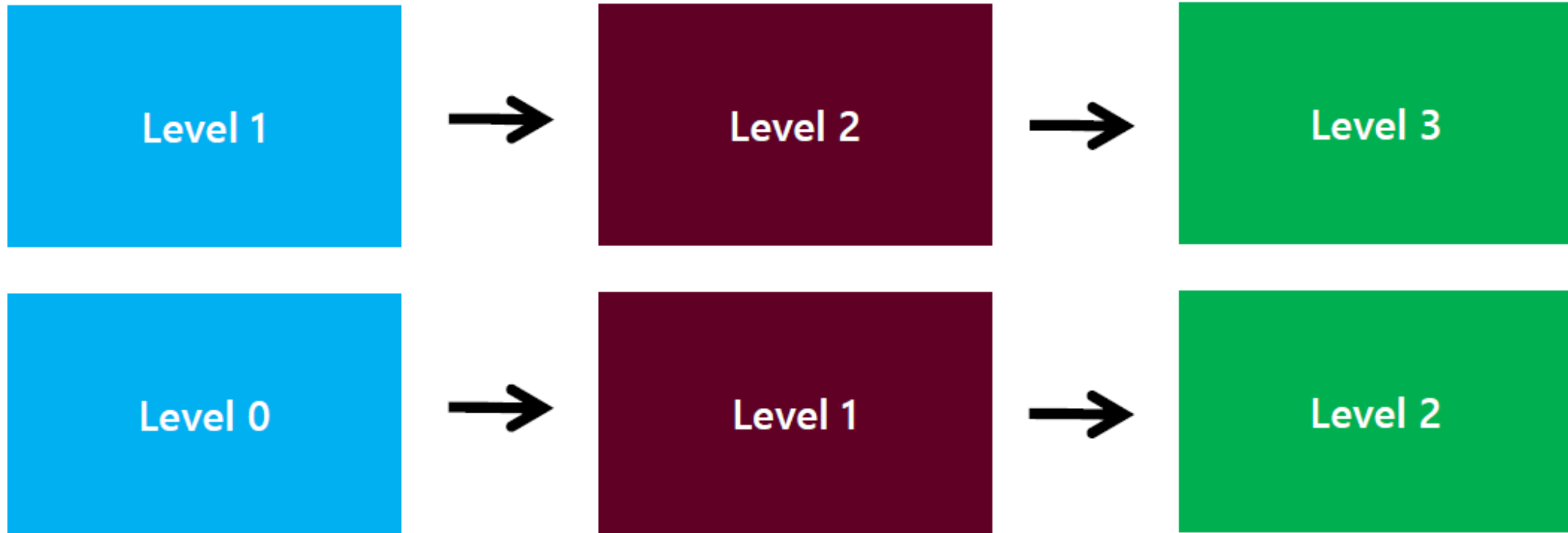
- *Loop*

RULE OF THUMB : DATA FLOW DIAGRAM (2)



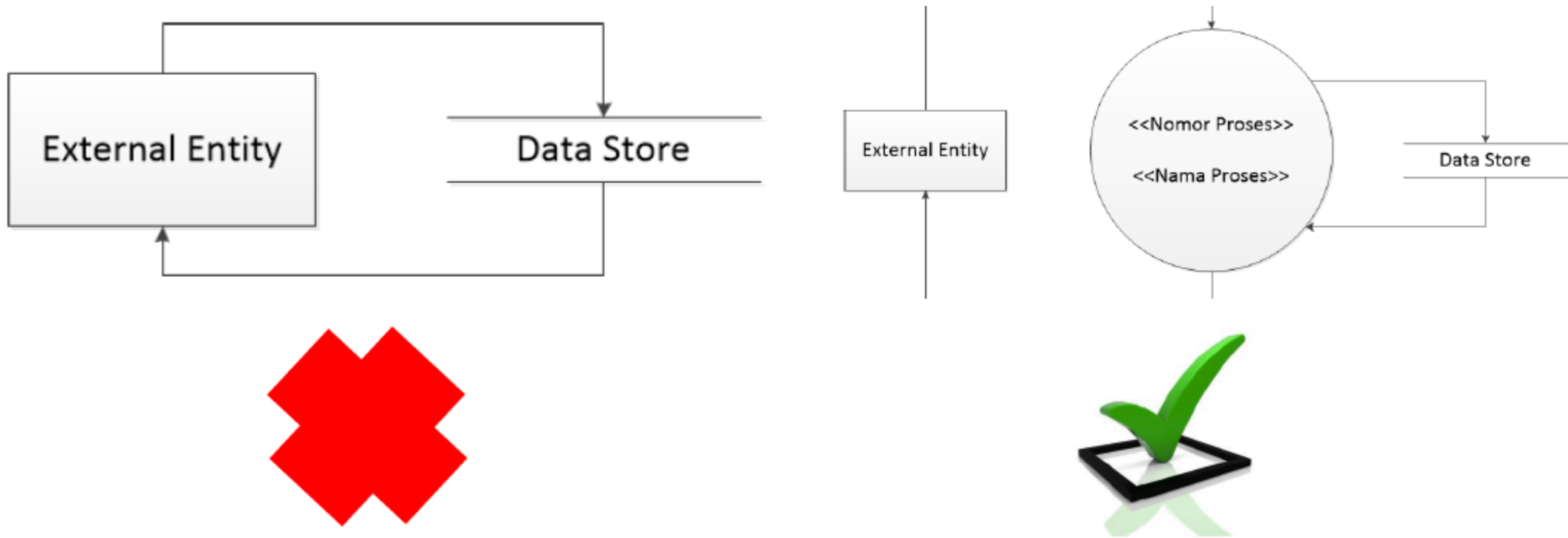
Proses Diberi Nomor yang Jelas

RULE OF THUMB : DATA FLOW DIAGRAM (3)



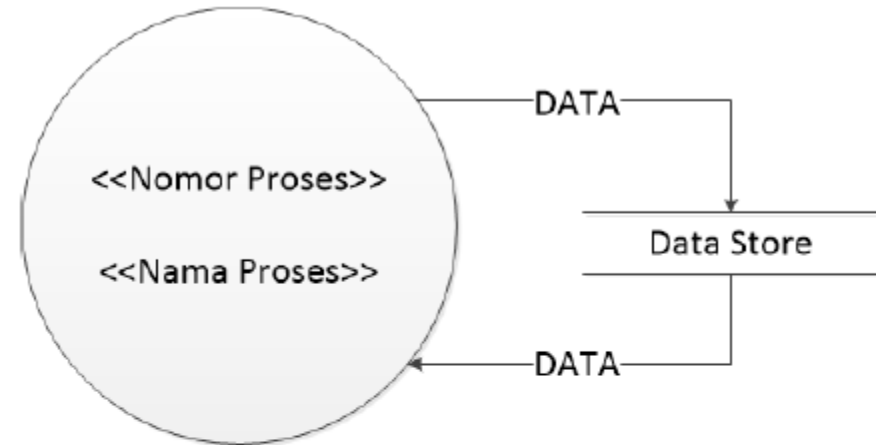
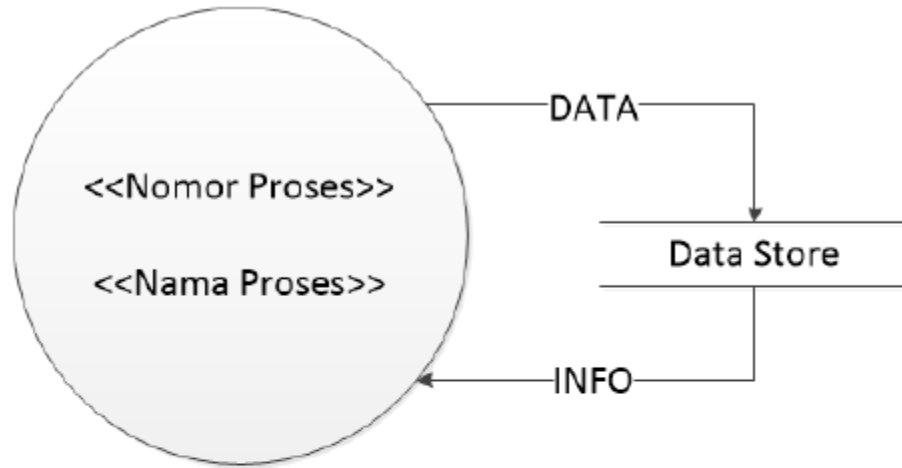
Pelevelan DFD Harus Runut (Start Level Optional)

RULE OF THUMB : DATA FLOW DIAGRAM (4)



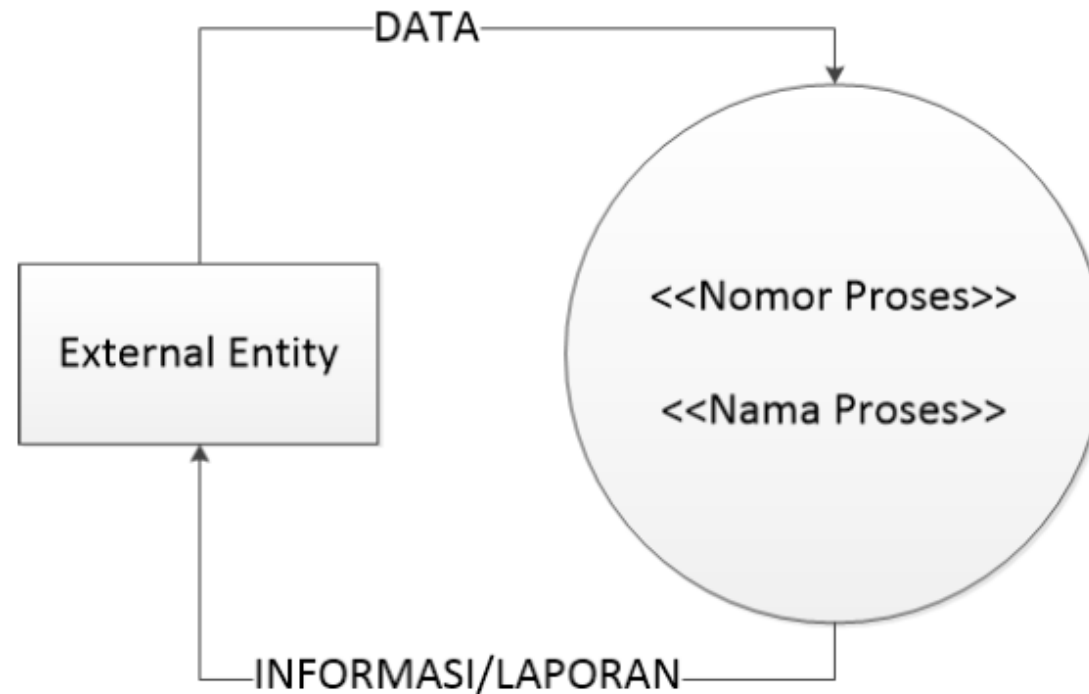
Entitas luar tidak boleh berhubungan langsung dengan data store begitu pun sebaliknya

RULE OF THUMB : DATA FLOW DIAGRAM (5)



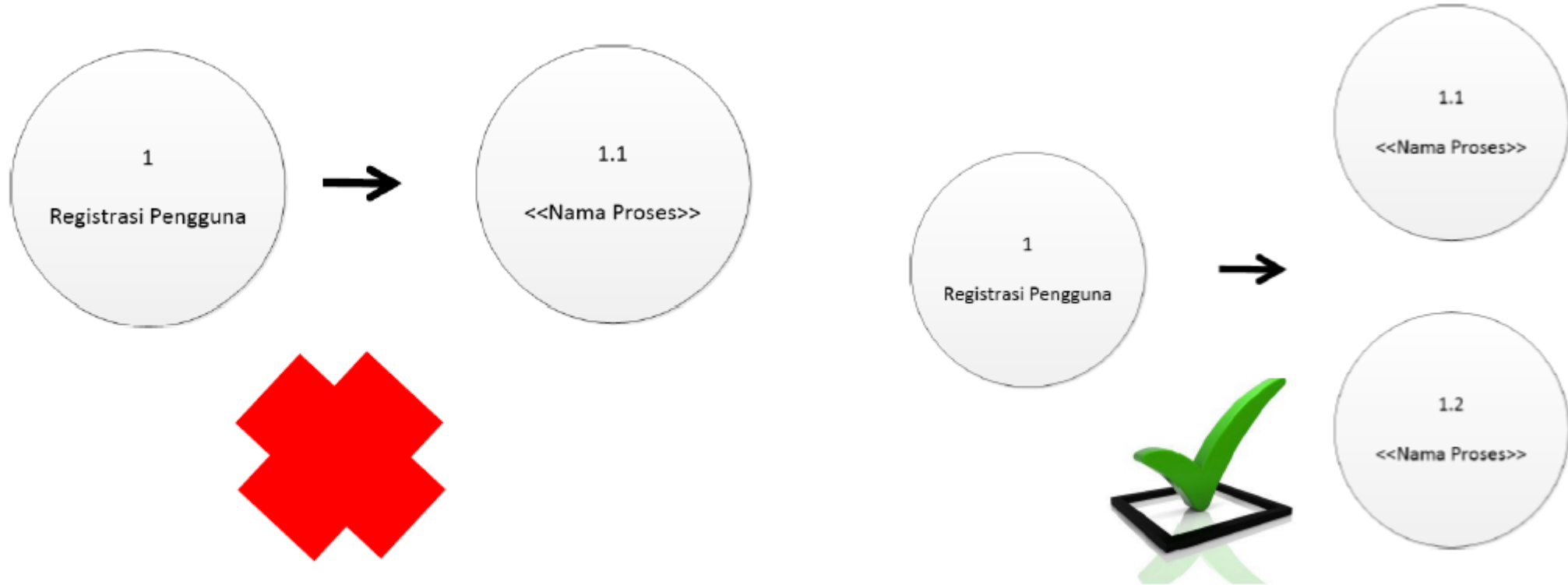
Hubungan antara proses dan data store dan sebaliknya berupa data bukan informasi

RULE OF THUMB : DATA FLOW DIAGRAM (6)



Hubungan antara entitas luar dan proses berupa input dan output

RULE OF THUMB : DATA FLOW DIAGRAM (7)



Tidak boleh membreakdown jika turunannya hanya satu proses.

RULE OF THUMB : DATA FLOW DIAGRAM (8)



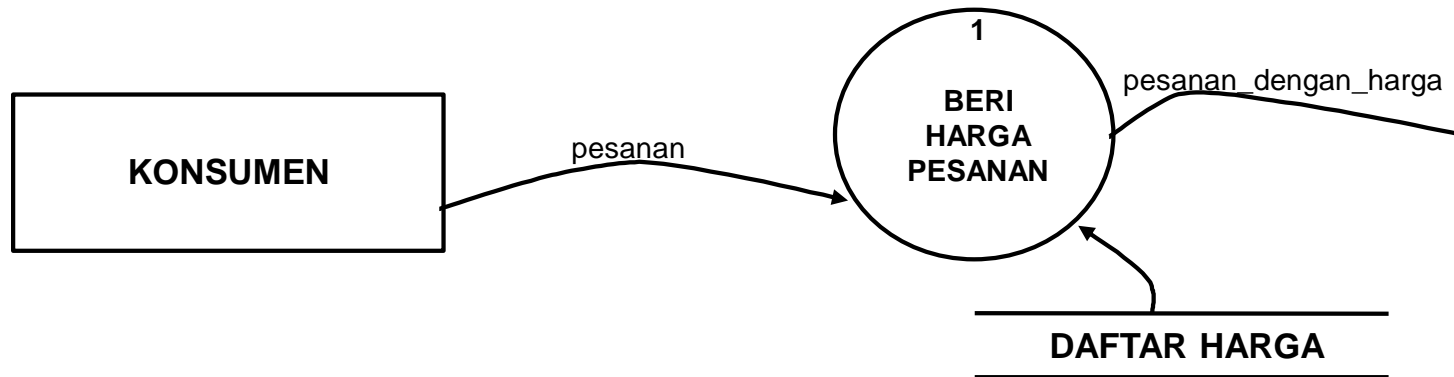
Penomoran proses pada DFD level kecil akan mempengaruhi penomoran pada DFD level berikutnya

CONTOH PEMBUATAN DFD (1)

TOKO SEMBAKO MINI (TSM) adalah sebuah grosir sembako dengan layanan delivery. TSM menerima pesanan-pesanan dari konsumen yang sudah terdaftar. Setiap konsumen mempunyai sebuah rekening konsumen dengan batas kredit untuk setiap pesanan. Semua pesanan yang diterima terlebih dahulu diberi harga, kemudian diperiksa terhadap rekening konsumen. Jika kredit tidak cukup konsumen diberi informasi dan diminta memperbaiki pesannya. Pesanan yang diterima diproses lebih lanjut dengan memeriksa barang di persediaan. Jika barang tersedia maka pengiriman segera dilakukan. Tetapi jika tidak tersedia konsumen diberitahu.

CONTOH PEMBUATAN DFD (2)

1. *Pesanan diterima dan diberi harga.*

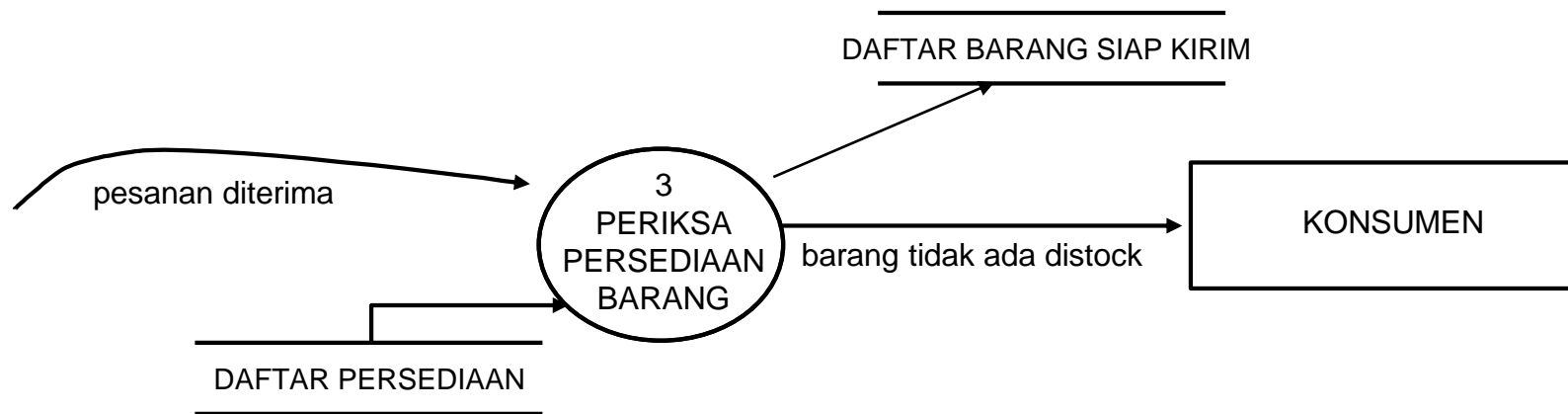


2. *Pemeriksaan batas kredit konsumen.*



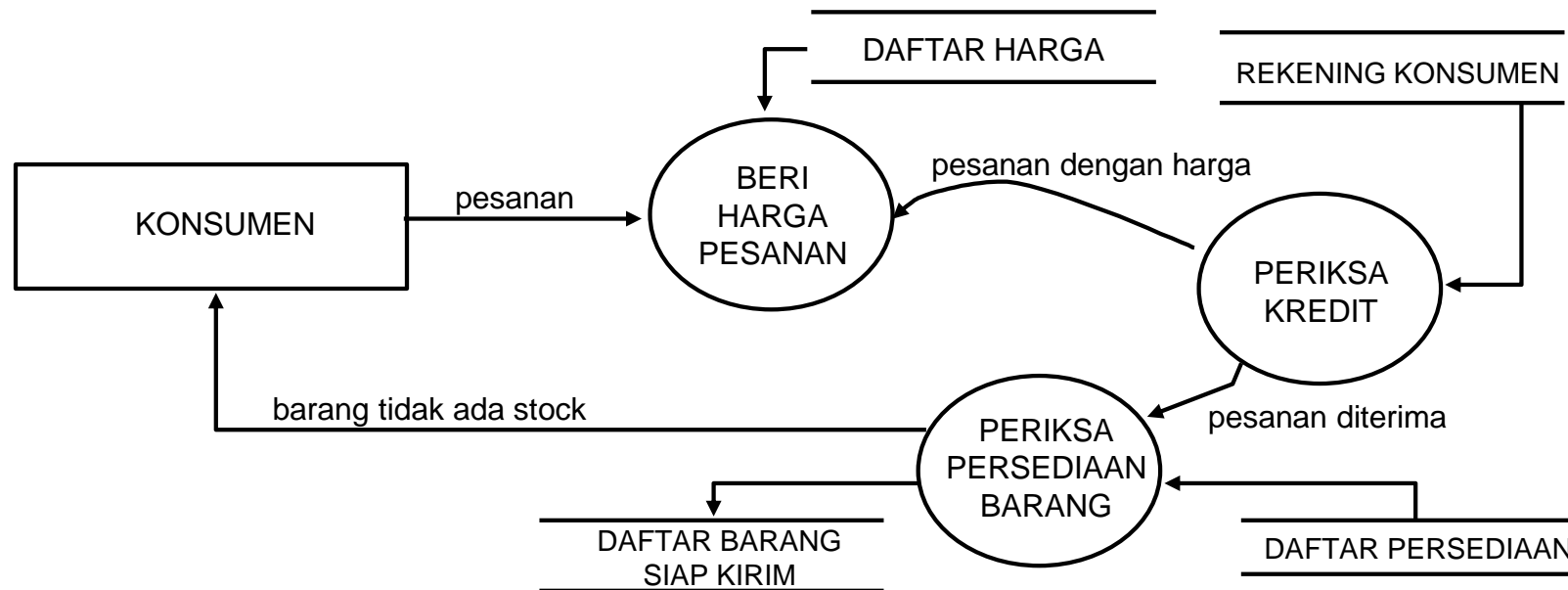
CONTOH PEMBUATAN DFD (3)

3. *Periksa barang di persediaan*



CONTOH PEMBUATAN DFD (3)

- Gabungan 1, 2, dan 3

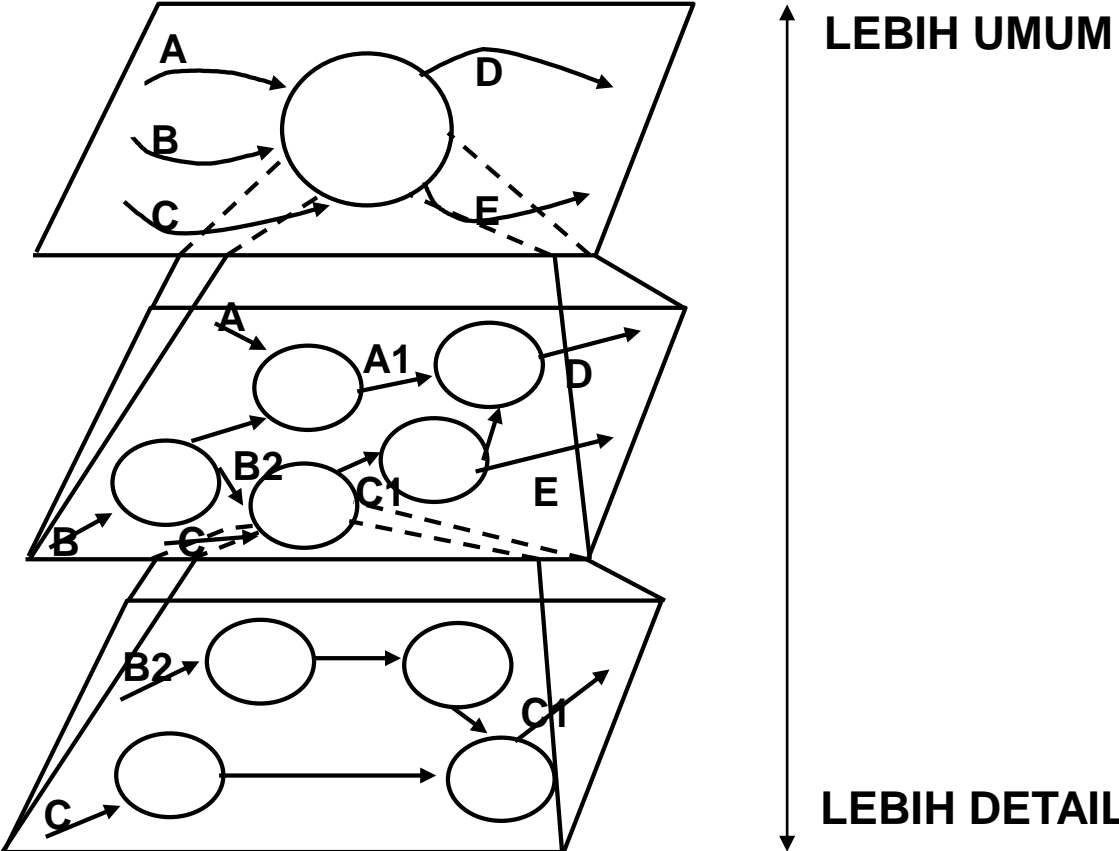


DFD BERTINGKAT (1)

Tujuannya adalah Agar mampu mengatur kerumitan dalam model aliran data dan kerumitan sistem dapat diatasi dengan :

- **PARTITIONING** - membagi proses yang rumit menjadi bagian yang kecil yang lebih banyak.
- **DEKOMPOSISI BERURUTAN** - membagi proses yang rumit menjadi bagian yang terkecil sampai setiap bagian dapat dengan mudah dimengerti, dan kemudian menyusun bagian secara hirarki.

DFD BERTINGKAT (2)



DFD BERTINGKAT (3)

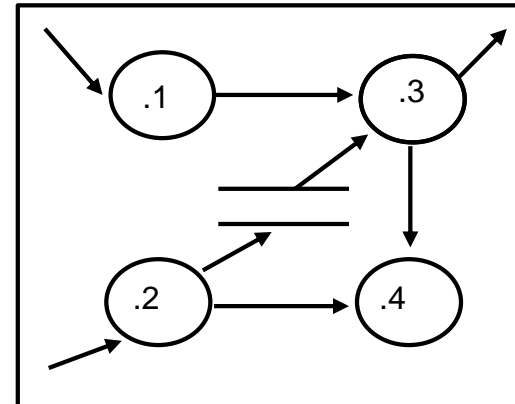
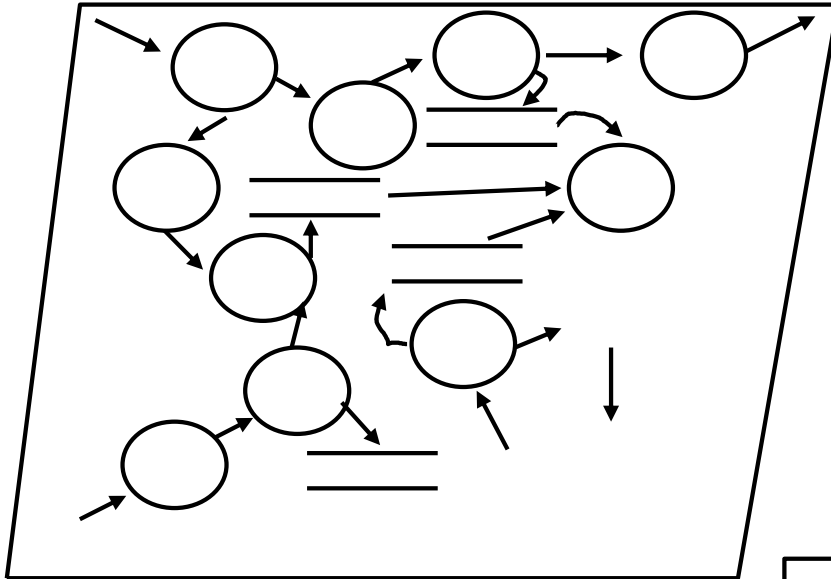


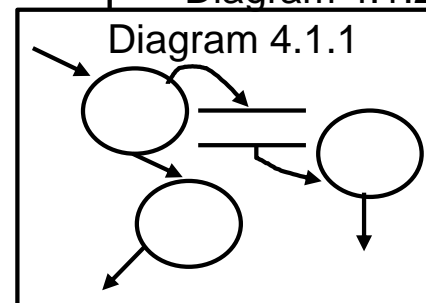
Diagram "Orang Tua"

Diagram 4.1.4

Diagram 4.1.3

Diagram 4.1.2

Diagram 4.1.1



Setiap Diagram "Anak" digambarkan tersendiri

PENYUSUNAN DFD BERTINGKAT

Jenis DFD dalam DFD bertingkat :

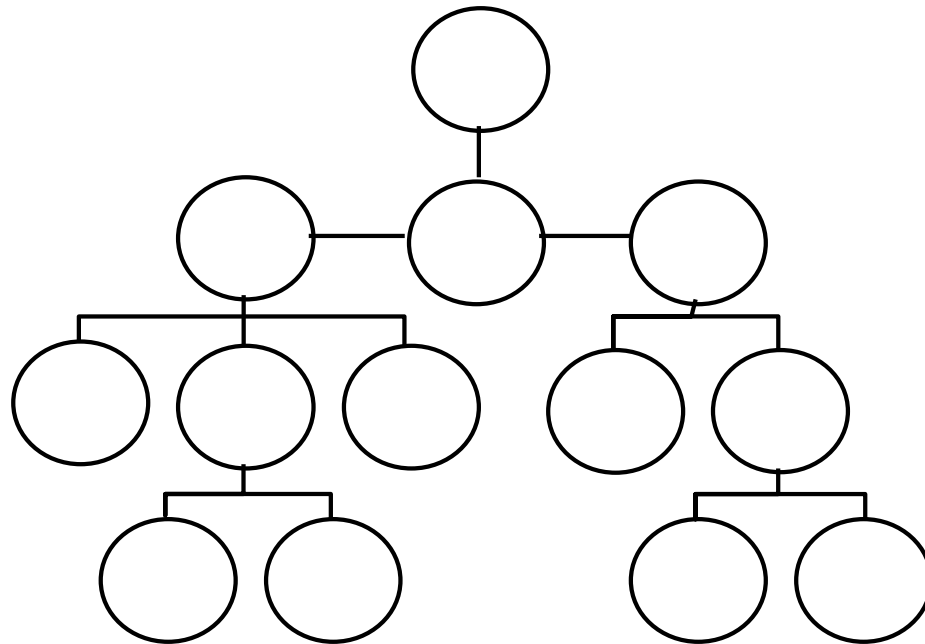
- **Diagram konteks (Context Diagram);** diagram paling atas, terdiri dari suatu proses dan menggambarkan ruang lingkup sistem.
- **Diagram Primitif Fungsional (Functional Primitive);** diagram- diagram paling bawah; yang tidak dapat dibagi lagi atau memiliki masukan tunggal dan keluaran tunggal atau telah sangat sederhana (narasi untuk deskripsi dapat dituliskan secara singkat).
- **Diagram Tengah;** diagram-diagram yang terletak diantara diagram konteks dan primitif fungsional. Diagram tengah tingkat paling atas sering disebut sebagai *Diagram 0*.

DEKOMPOSISI PROSES (1)

- Dekomposisi berurutan (*sequential decomposition*) adalah *membagi proses yang rumit* menjadi bagian yang terkecil sampai setiap bagian dapat dengan *mudah dimengerti*, dan kemudian menyusun bagian-bagian tersebut *secara hirarki*.
- Dekomposisi proses ini banyak digunakan dalam kaitannya dengan *dekomposisi fungsional dari sistem/bisnis (Functional Decomposition Diagram/FDD)*.
- Di dalam *menganalisis suatu sistem/bisnis*, mula-mula dilakukan *dekomposisi* menurut *fungsi* yang ada didalamnya kemudian fungsi tersebut di dekomposisi ke dalam proses dan selanjutnya ke proses-proses tingkat lebih bawah .

DEKOMPOSISI PROSES (2)

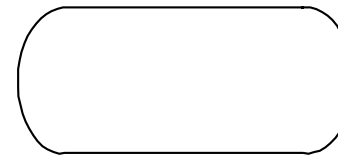
Bentuk Diagram



Bisa menggunakan notasi lain, seperti:



atau

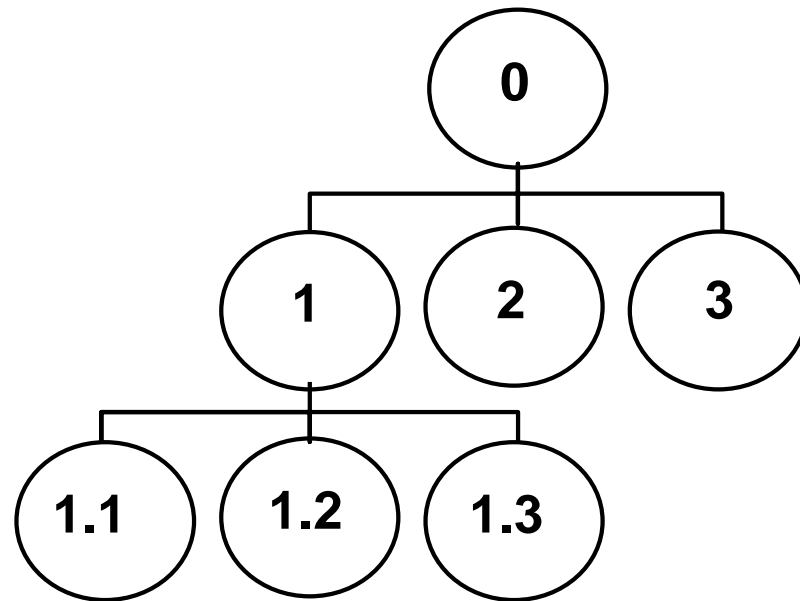


atau



DEKOMPOSISI PROSES (3)

Diagram Dekomposisi



Data Flow Diagram

