



Tip: Use graphics to set the tone of the speech.

Data Flow Diagram

Analisa dan Perancangan Sistem Informasi
Pertemuan 12

Afriyanti Dwi Kartika, S.Pd., M.T.


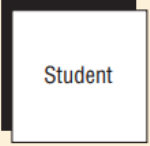
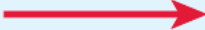
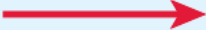
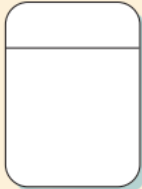
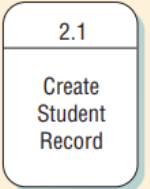

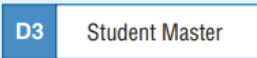
When ***systems analysts*** attempt to understand the information requirements of users, they must be able to ***conceptualize:***

- how ***data move*** through the organization,
- the ***processes or transformation*** that the data undergo, and
- what the ***outputs*** are.



BASIC SYMBOLS

- A double square,
- An arrow,
- A rectangle with rounded corners, and
- An open-ended rectangle (closed on the left side and open ended on the right)

Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	

1. Entity / Terminator

- To depict an **external entity** (another department, a business, a person, or a machine) that can send data to or receive data from the system
- The external entity, or just entity, is also called **a source or destination of data**, and it is considered to be external to the system being described
- An entity should be named with **a noun**.

Terdapat dua jenis terminator :

1. Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
2. Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data / informasi sistem.



Terminator Sumber

Terminator Tujuan

T. Tujuan & Sumber

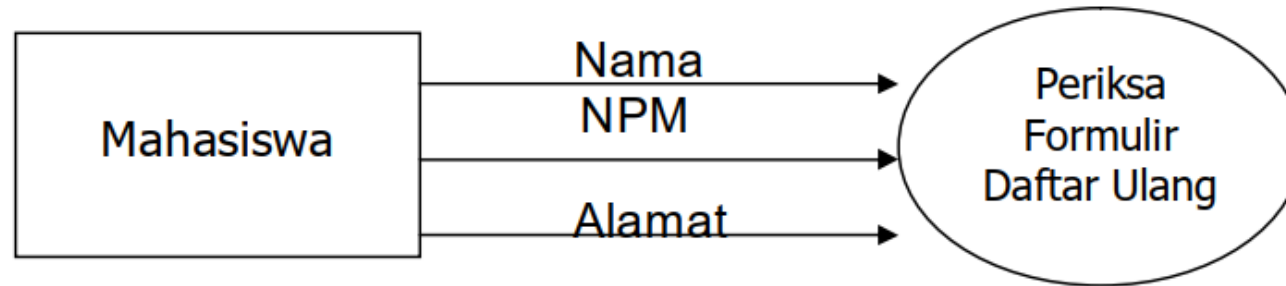
Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya.

Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

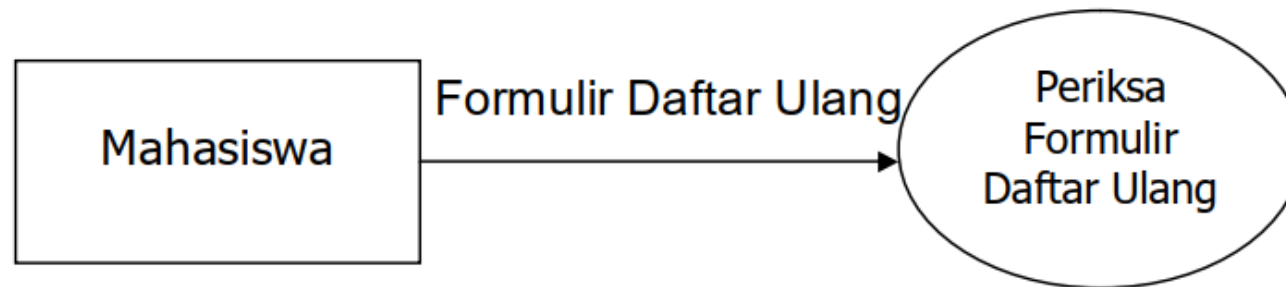
2. Data Flow

- The arrow shows **movement of data** from one point to another, with **the head of the arrow pointing toward the data's destination**.
- Data flows occurring simultaneously can be depicted doing just that through the use of parallel arrows.
- Because an arrow represents data about a person, place, or thing, it too should be described with **a noun**

2.1. Konsep Paket Data (Packets of Data)



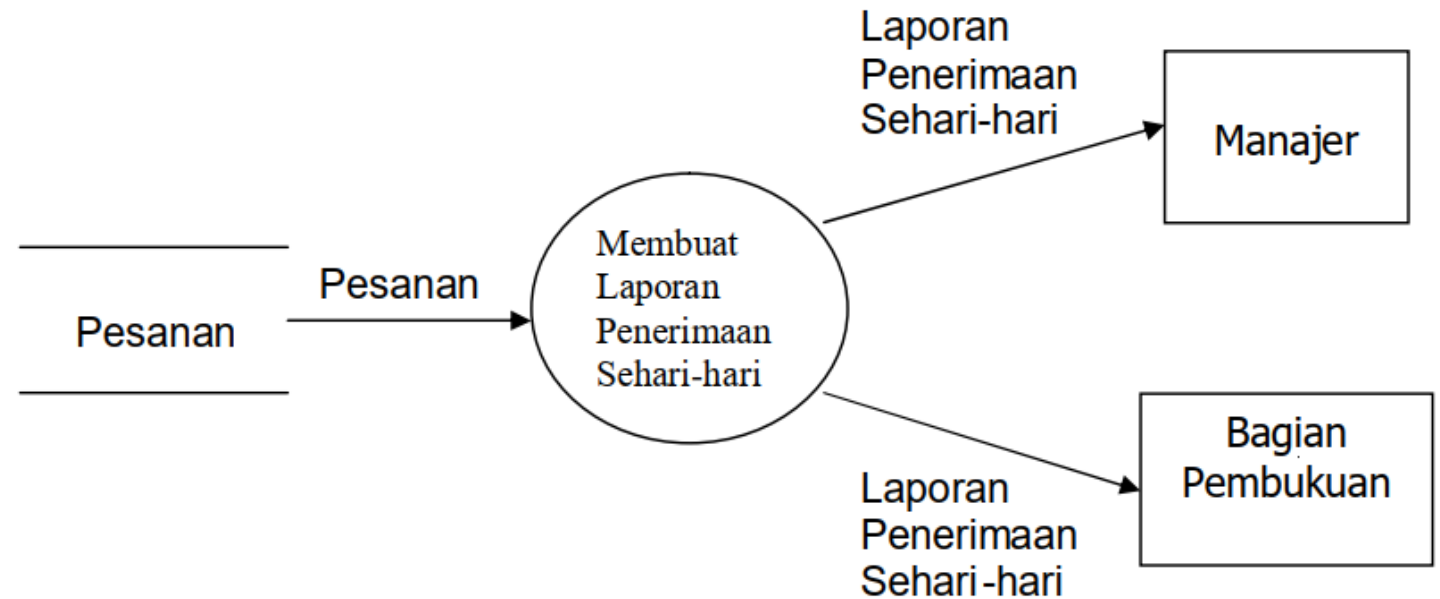
(a) Konsep paket data yang salah



(b) Konsep paket data yang benar

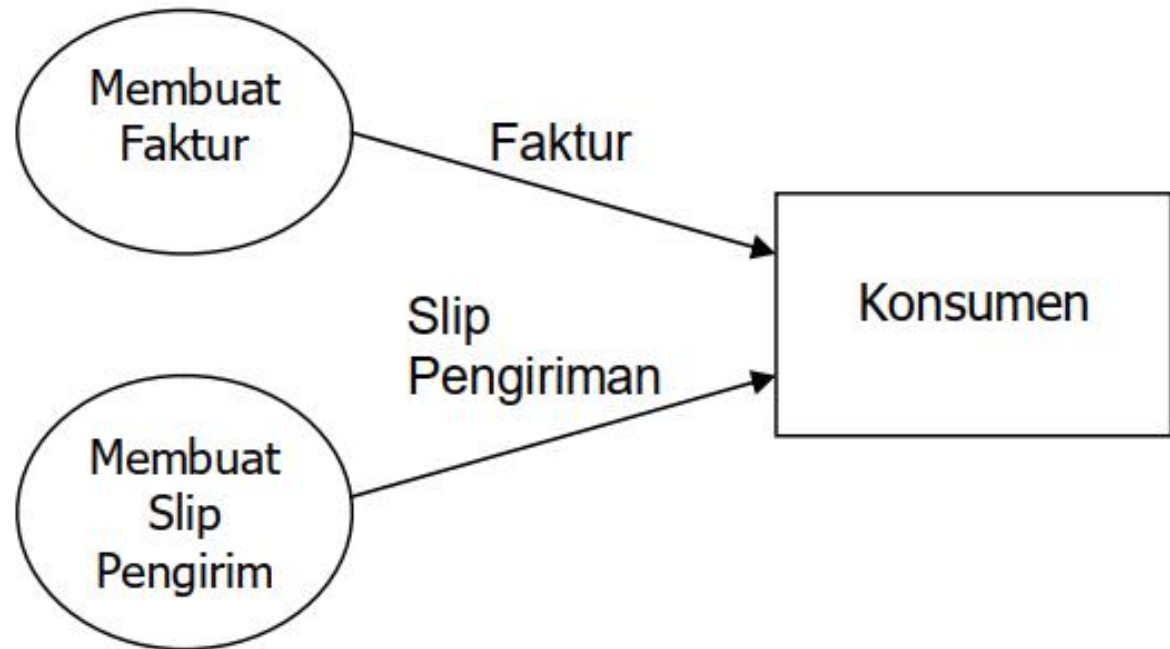
2.2. Konsep Alur Data Menyebar (Diverging Data Flow)

Alur data menyebar menunjukkan sejumlah tembusan paket data yang 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.



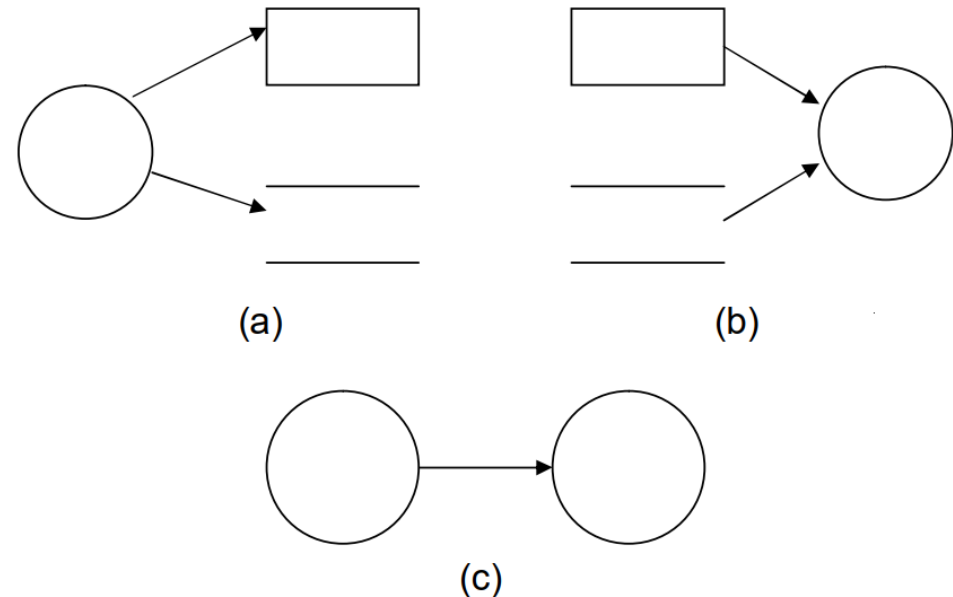
2.3. Konsep Alur Data Mengumpul (Converging Data Flow)

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



2.4. Konsep Sumber atau Tujuan Alur Data

- Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu data store dan/atau terminator (lihat gambar 6 (a)).
- Suatu alur data dihasilkan dari suatu data store dan/atau terminator dan menuju ke suatu proses (lihat gambar 6 (b)).
- Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu proses (lihat gambar 6 (c))



Gambar 6. Konsep sumber atau tujuan alur data

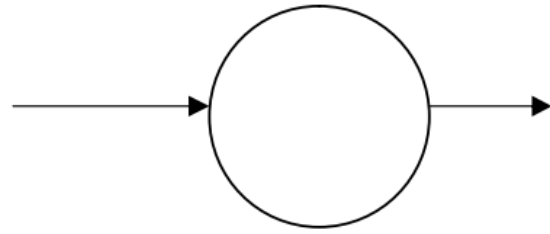
3. Processes

- Processes always denote a ***change in or transformation of data***; hence, the data flow leaving a process is always labeled differently than the one entering it.
- A process must also ***be given a unique identifying number*** that indicates its level in the diagram.

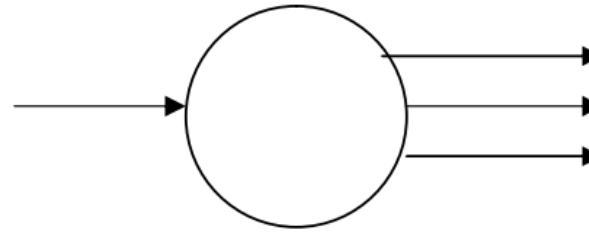
Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

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***.

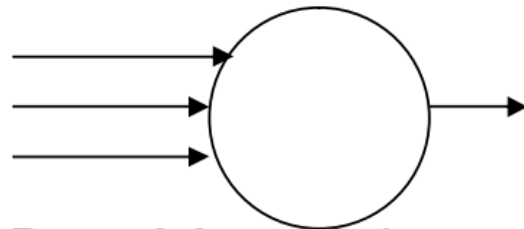
Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



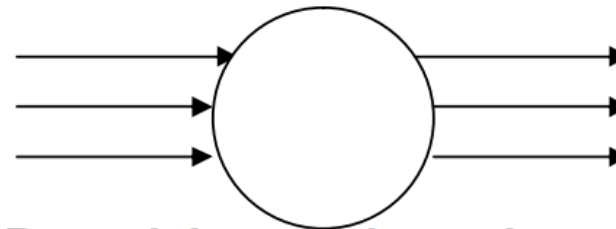
1 input & 1 output



1 input & banyak output



Banyak input & 1 output



Banyak input & banyak output

Ada beberapa hal yang perlu diperhatikan tentang proses :



Proses harus memiliki input dan output.

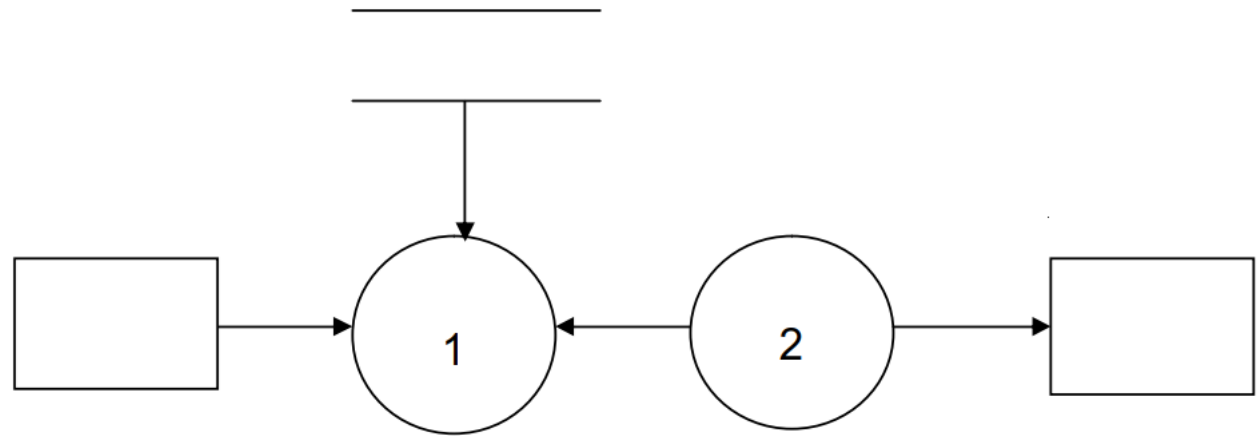


Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.

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*).

Berikut ini merupakan suatu contoh proses yang salah :

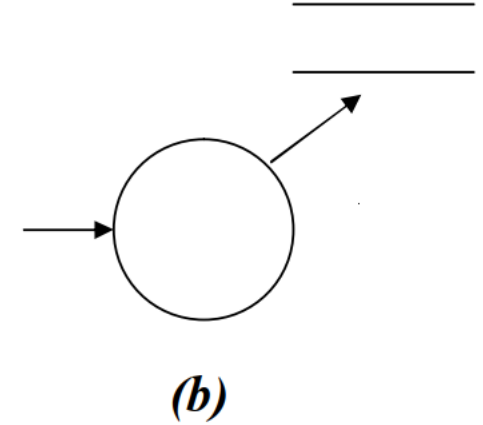
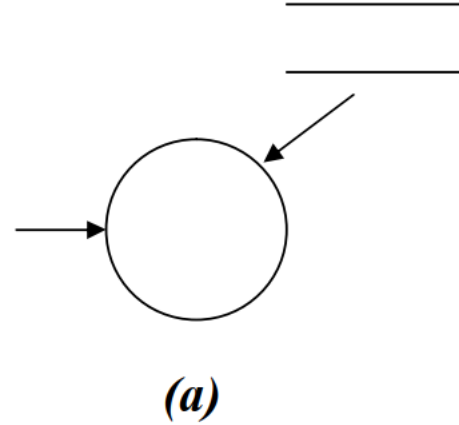


4. Data store

- The rectangle is drawn with two parallel lines that are closed by a short line on the left side and are open ended on the right.
- The data store symbol is simply showing a depository for data that allows examination, addition, and retrieval of data
- The data store may represent a manual store, such as a filing cabinet, or a computerized file or database. Because data stores represent a person, place, or thing, they are named with a noun

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 2 (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 2 (b)*).



Developing Data Flow Diagrams

Developing Data Flow Diagrams Using a Top-Down Approach

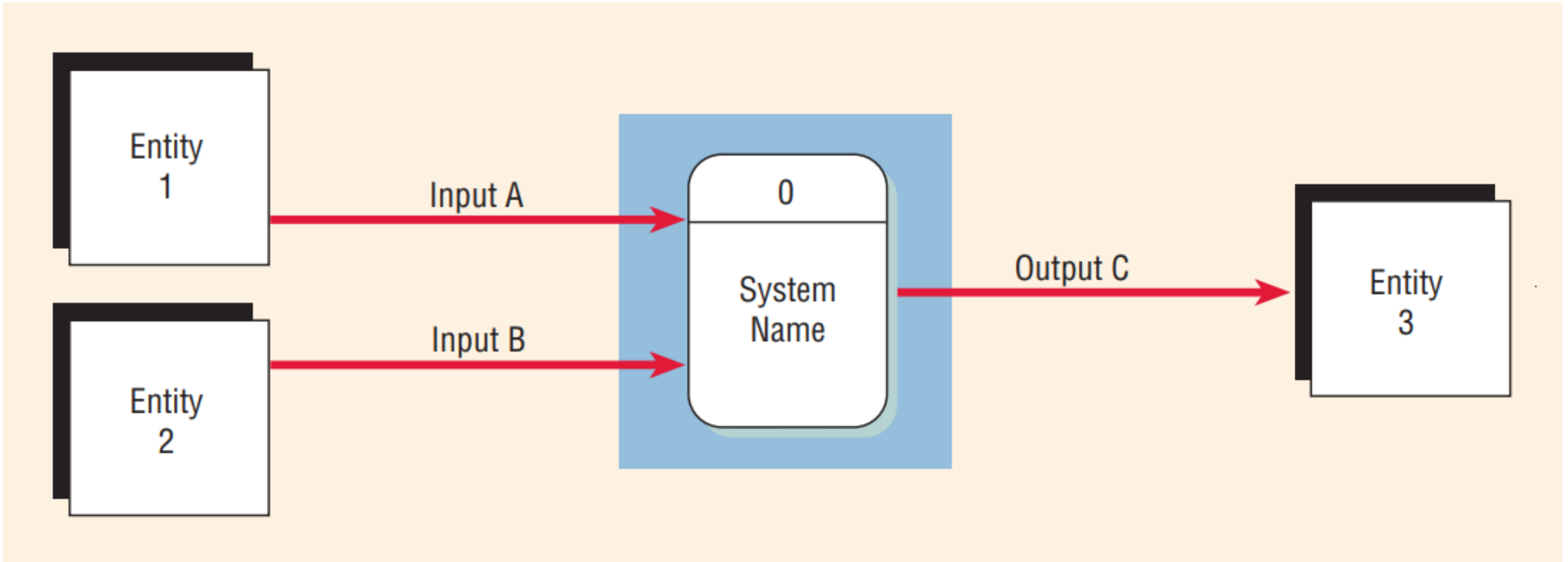
1. Make a list of business activities and use it to determine various
 - External entities
 - Data flows
 - Processes
 - Data stores
2. Create a context diagram that shows external entities and data flows to and from the system. Do not show any detailed processes or data stores.
3. Draw Diagram 0, the next level. Show processes, but keep them general. Show data stores at this level.
4. Create a child diagram for each of the processes in Diagram 0.
5. Check for errors and make sure the labels you assign to each process and data flow are meaningful.
6. Develop a physical data flow diagram from the logical data flow diagram. Distinguish between manual and automated processes, describe actual files and reports by name, and add controls to indicate when processes are complete or errors occur.
7. Partition the physical data flow diagram by separating or grouping parts of the diagram in order to facilitate programming and implementation.

1. Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.
2. Identifikasi semua input dan output yang terlibat dengan entitas luar

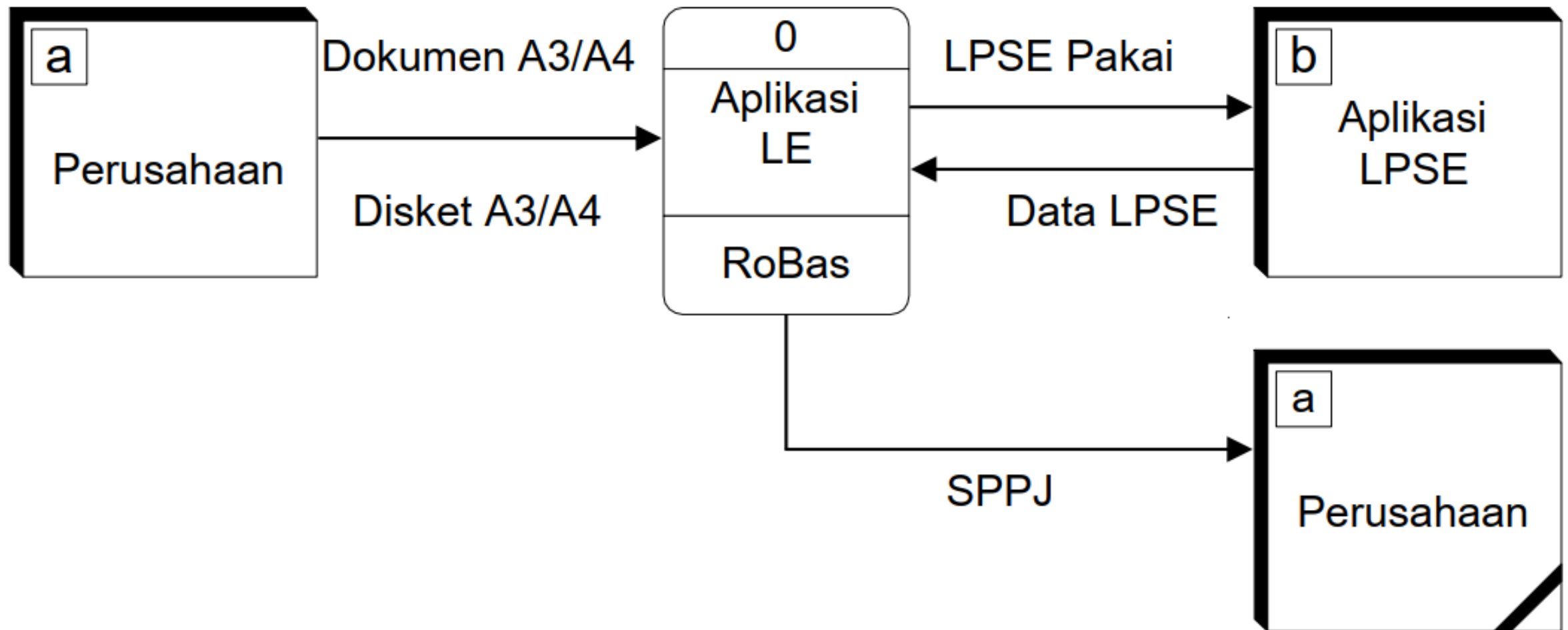
3. Creating the Context Diagram

- The initial context diagram should be an overview, including:
 - **basic inputs,**
 - **the general system, and**
 - **outputs**
- The context diagram is the highest level in a data flow diagram and contains **only one process**, representing the entire system. **The process is given the number zero.**
- **All external entities are shown on the context diagram**, as well as major data flow to and from them.
- The diagram **does not contain any data stores** and is fairly simple to create, once the external entities and the data flow to and from them are known to analysts.

- ❑ 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.



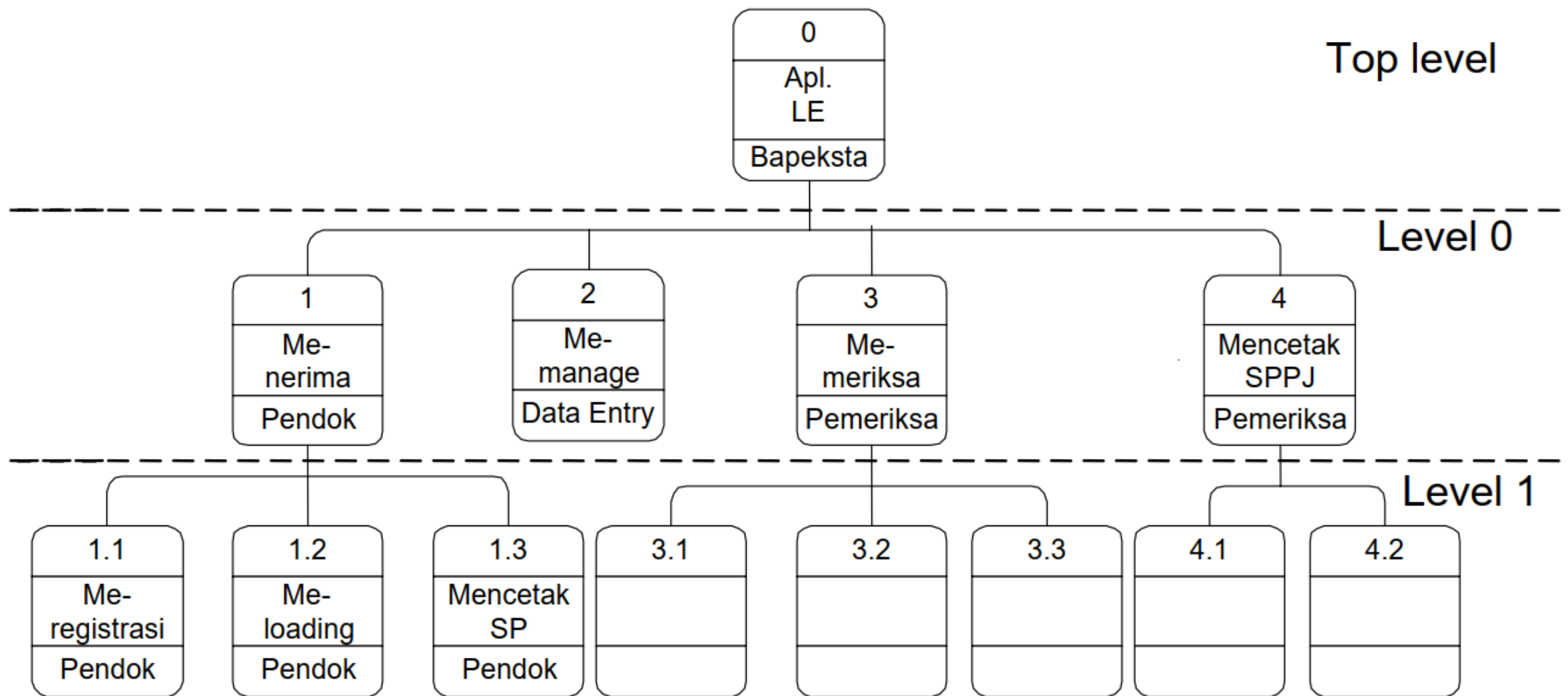
Context Diagram



Context Diagram

4. Gambarkan Bagan Berjenjang

Gambarlah bagan berjenjang untuk semua proses yang ada di sistem terlebih dulu. Bagan berjenjang atau Hierarchy chart digunakan untuk mempersiapkan penggambaran DFD ke level yang lebih bawah lagi.



Bagan Berjenjang

5. Drawing Diagram 0 (The Next Level)

- involving **three to nine processes** and **showing data stores and new lower-level data flows**.
- The **major data stores** of the system (representing master files) and **all external entities** are included on Diagram 0.
 - 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).

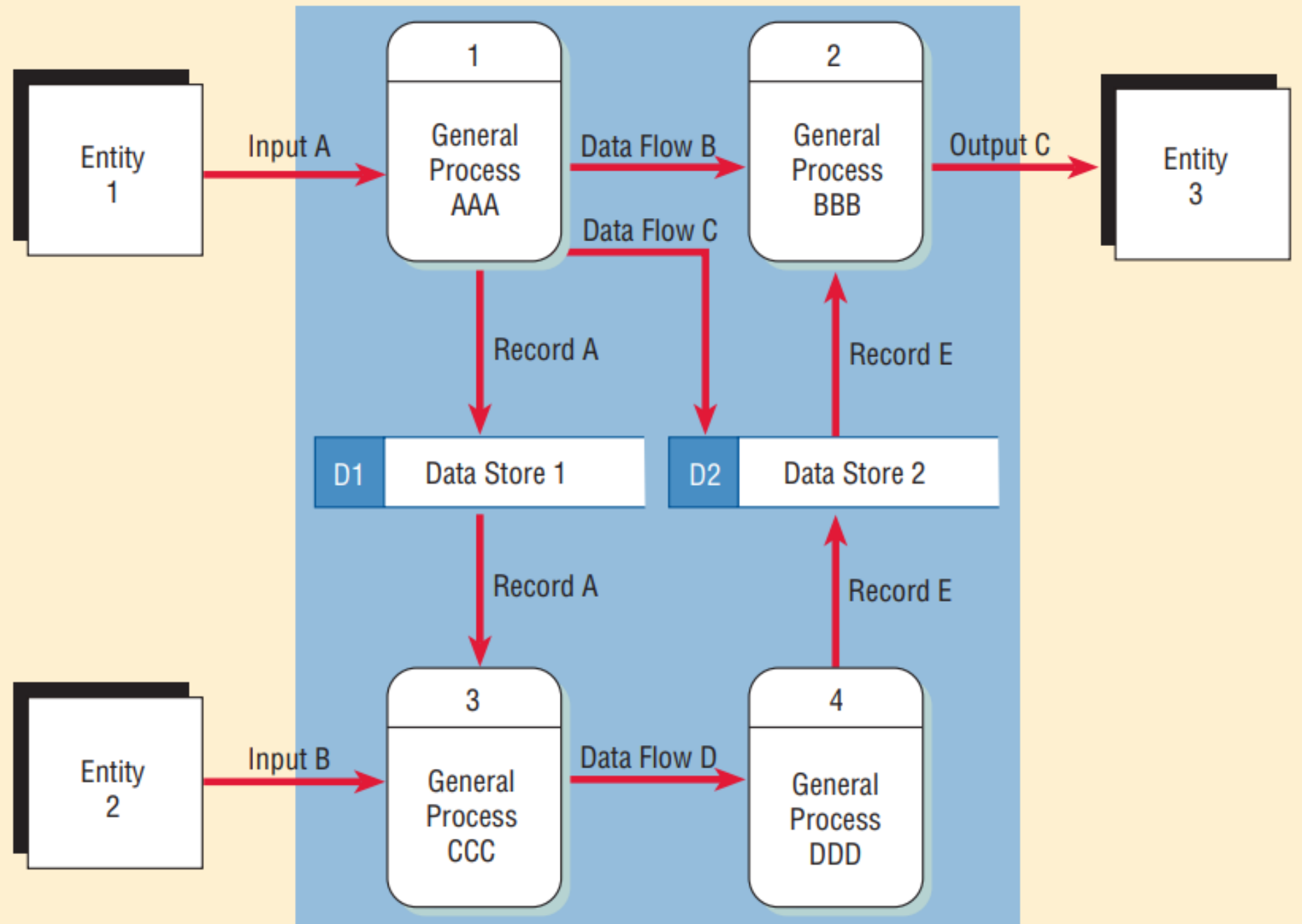


Diagram 0

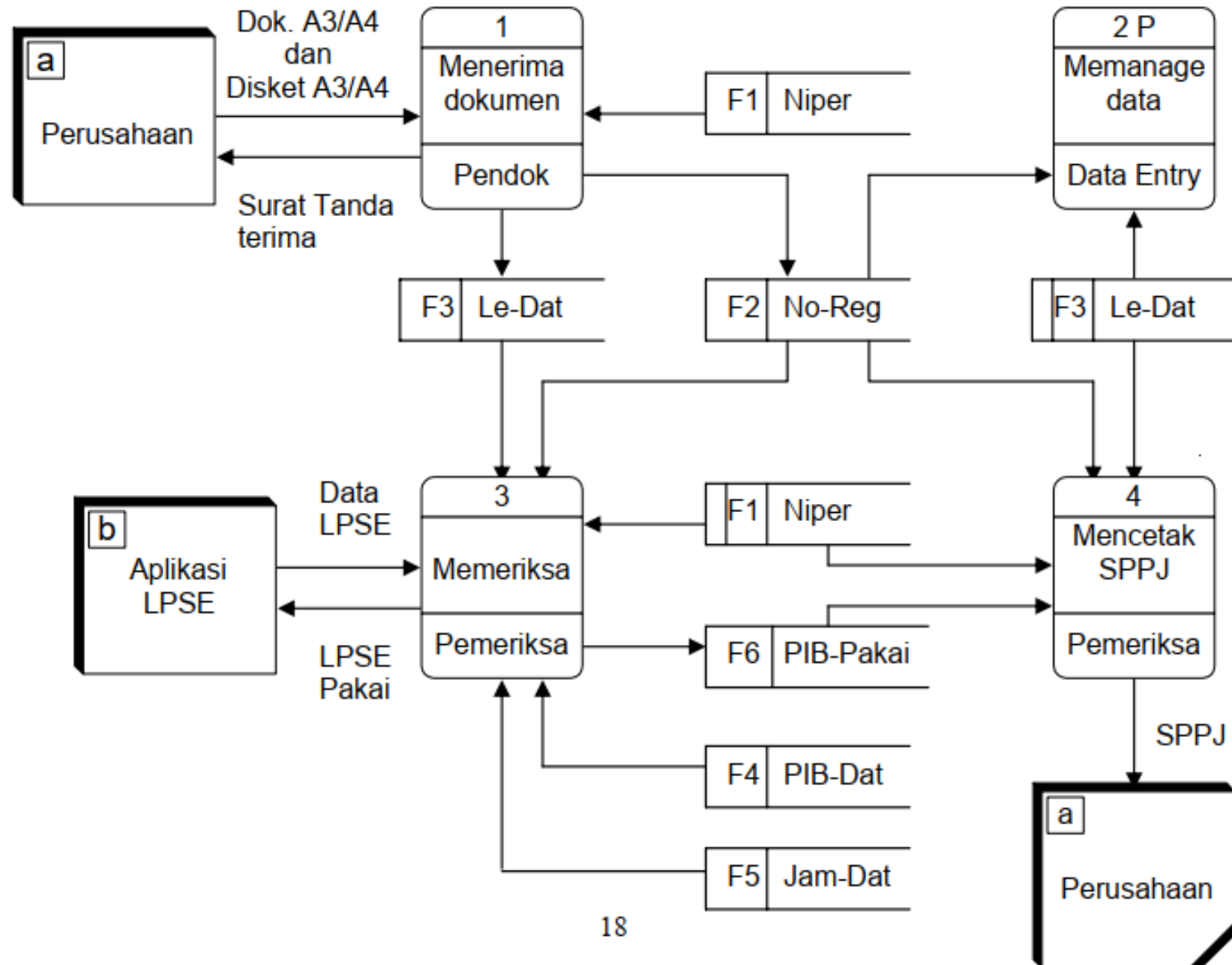


Diagram 0

6. Creating Child Diagrams (More Detailed Levels)

- ❑ 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.
- ❑ 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

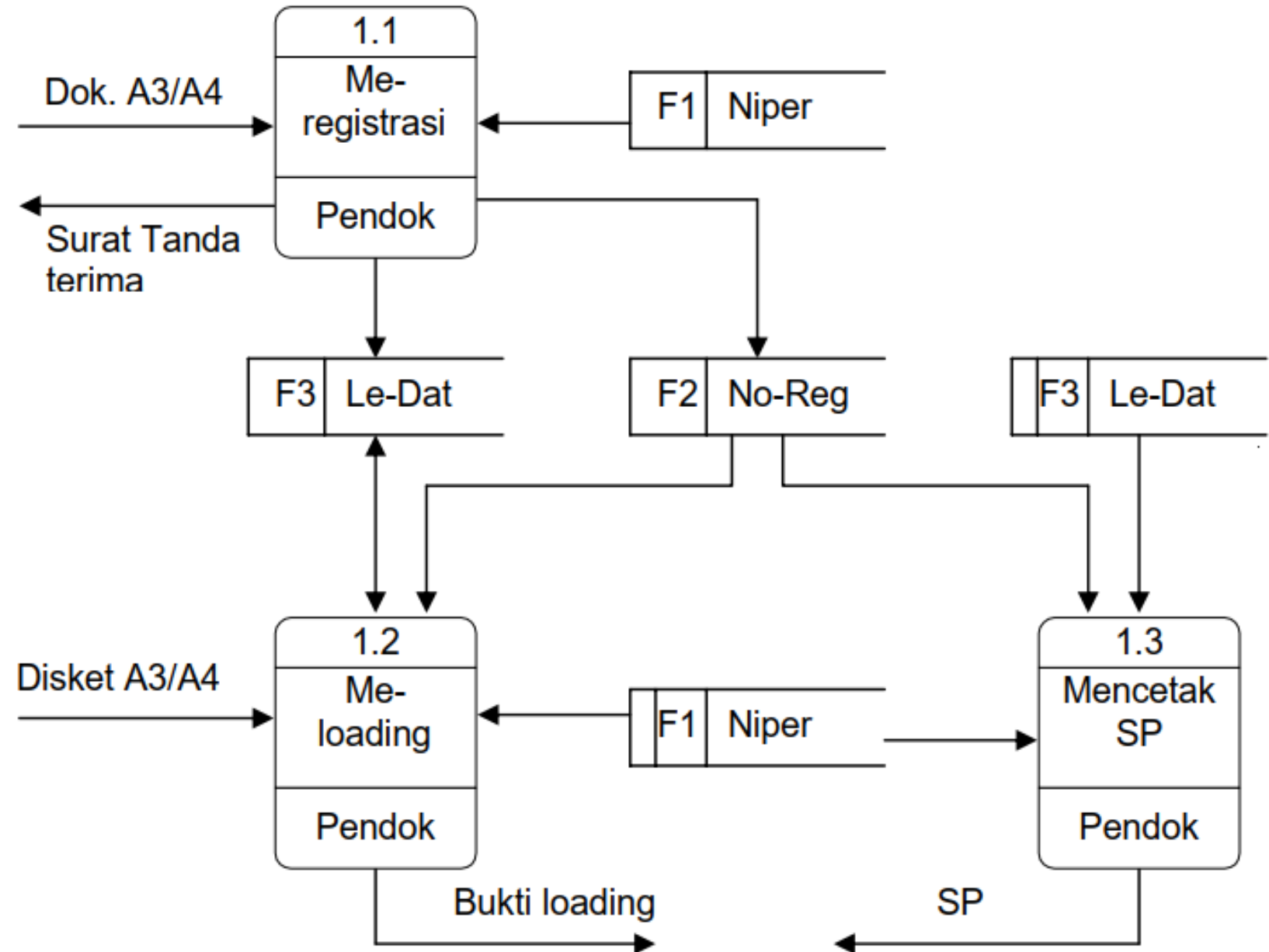
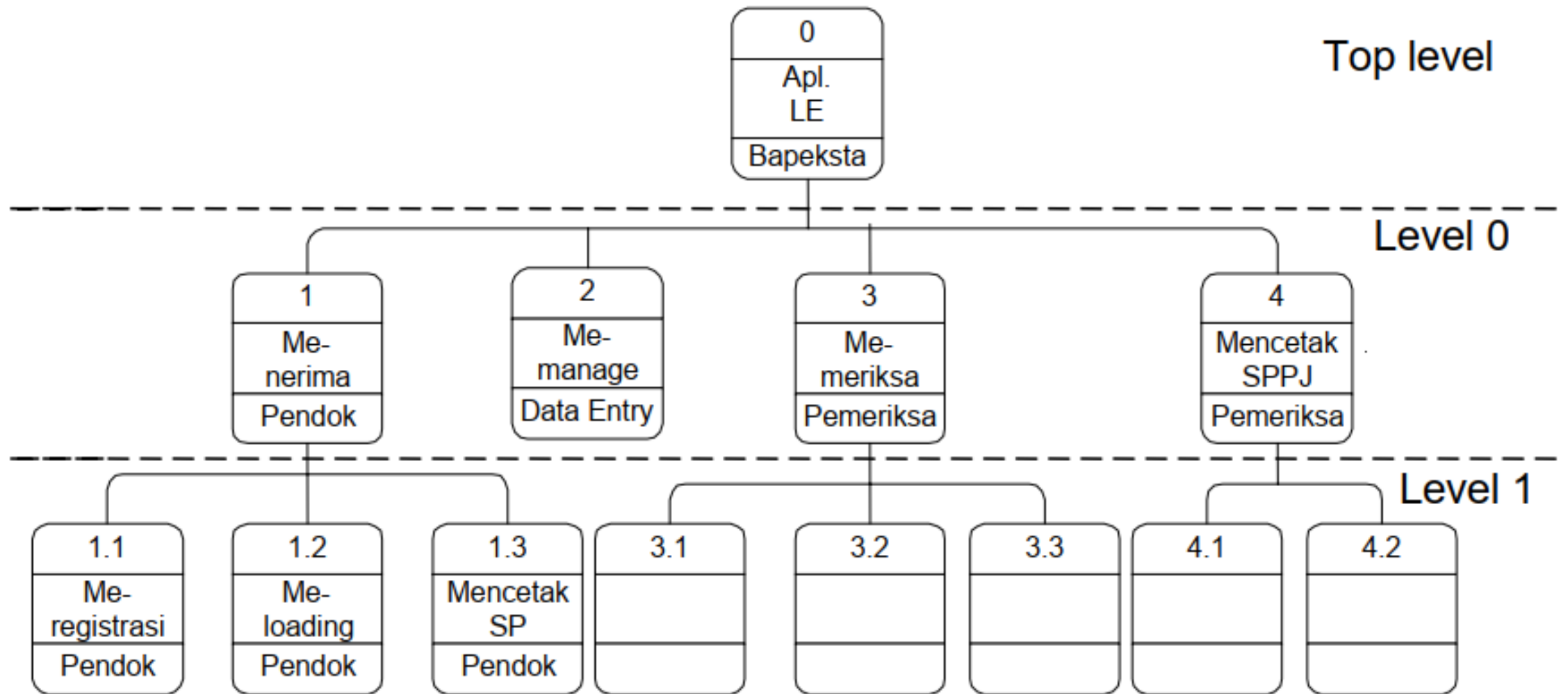


Diagram 1

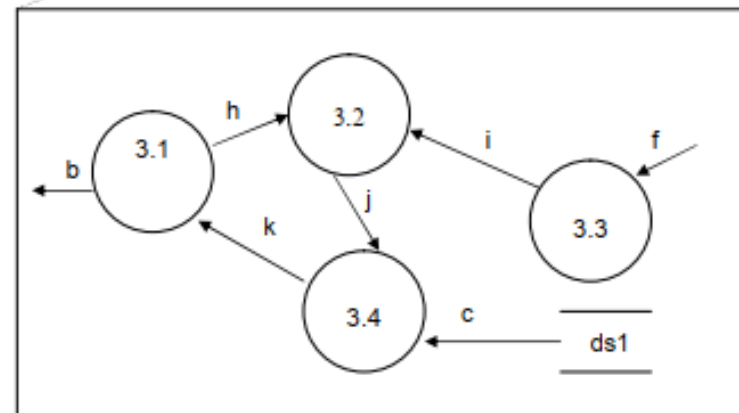
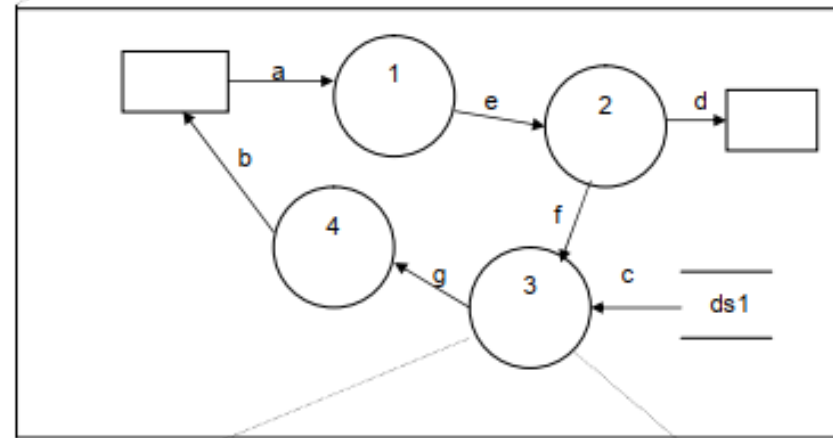
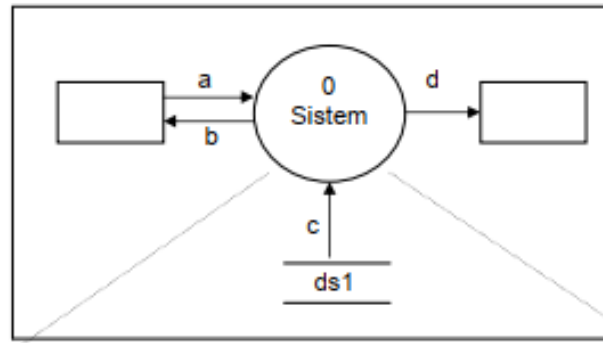
7. DFD Level Dua, Tiga, «

- 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

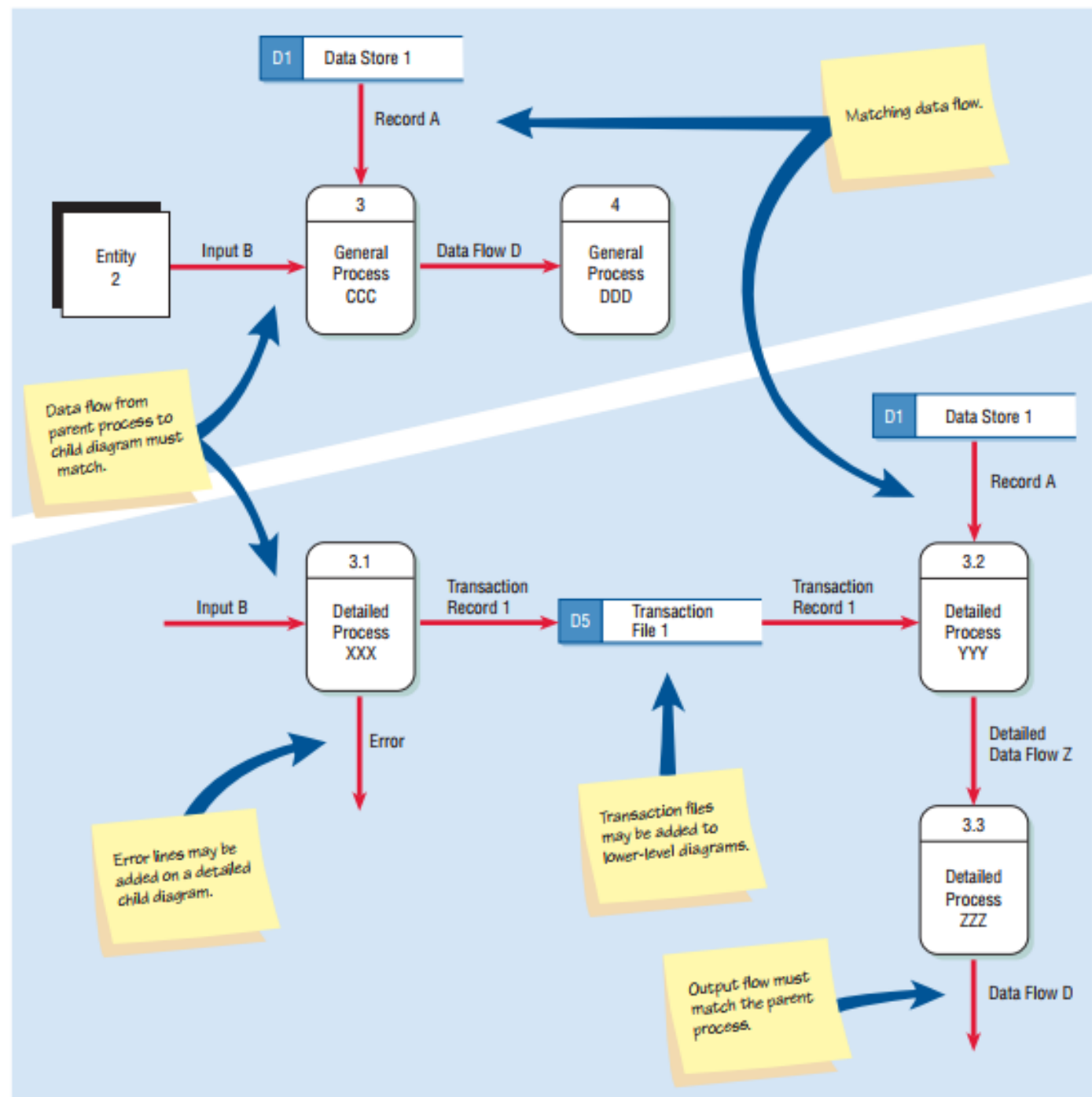
Leveling pada DFD



Level pada DFD

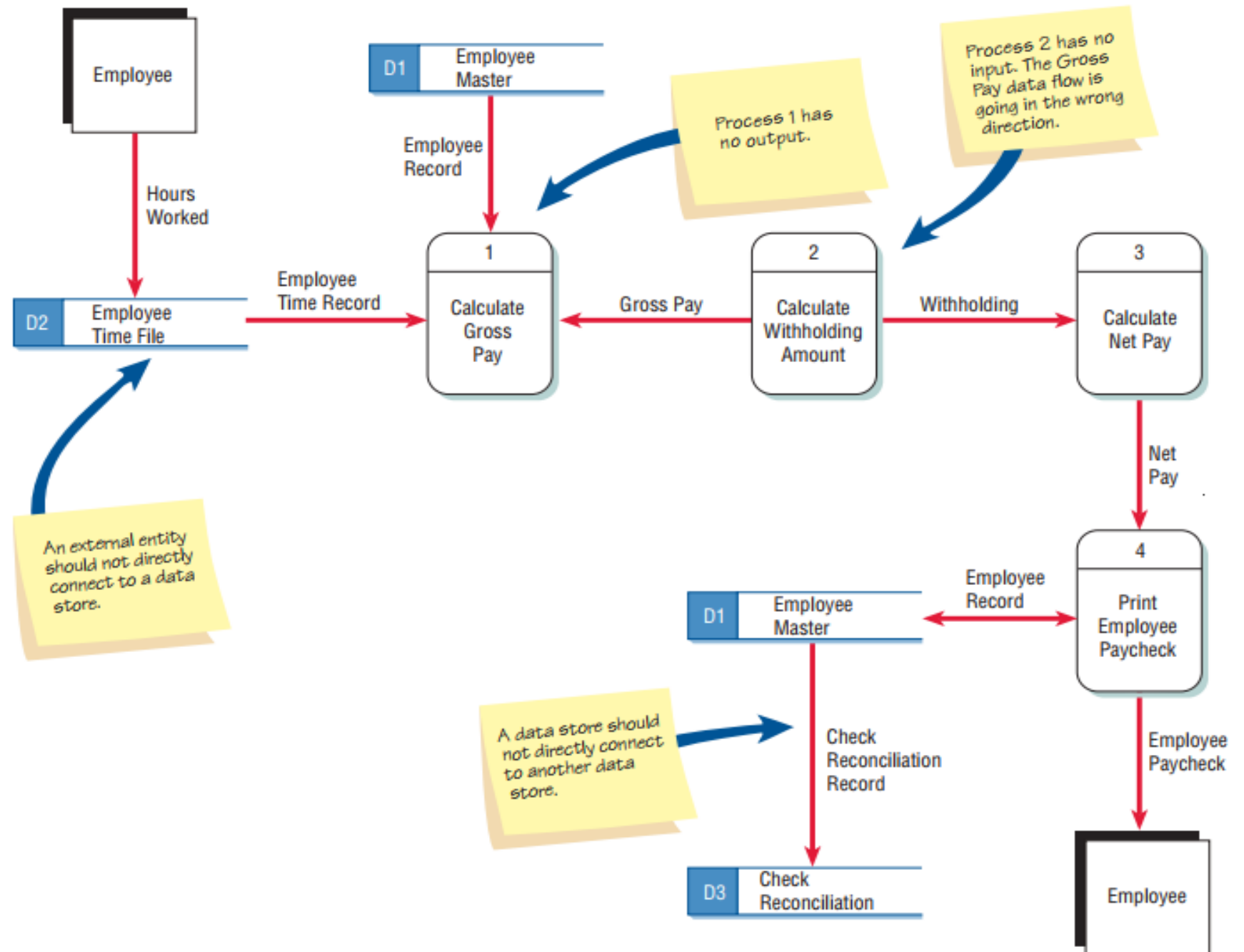


Level pada DFD



Level pada DFD

Typical errors that can occur in a data flow diagram (payroll example).

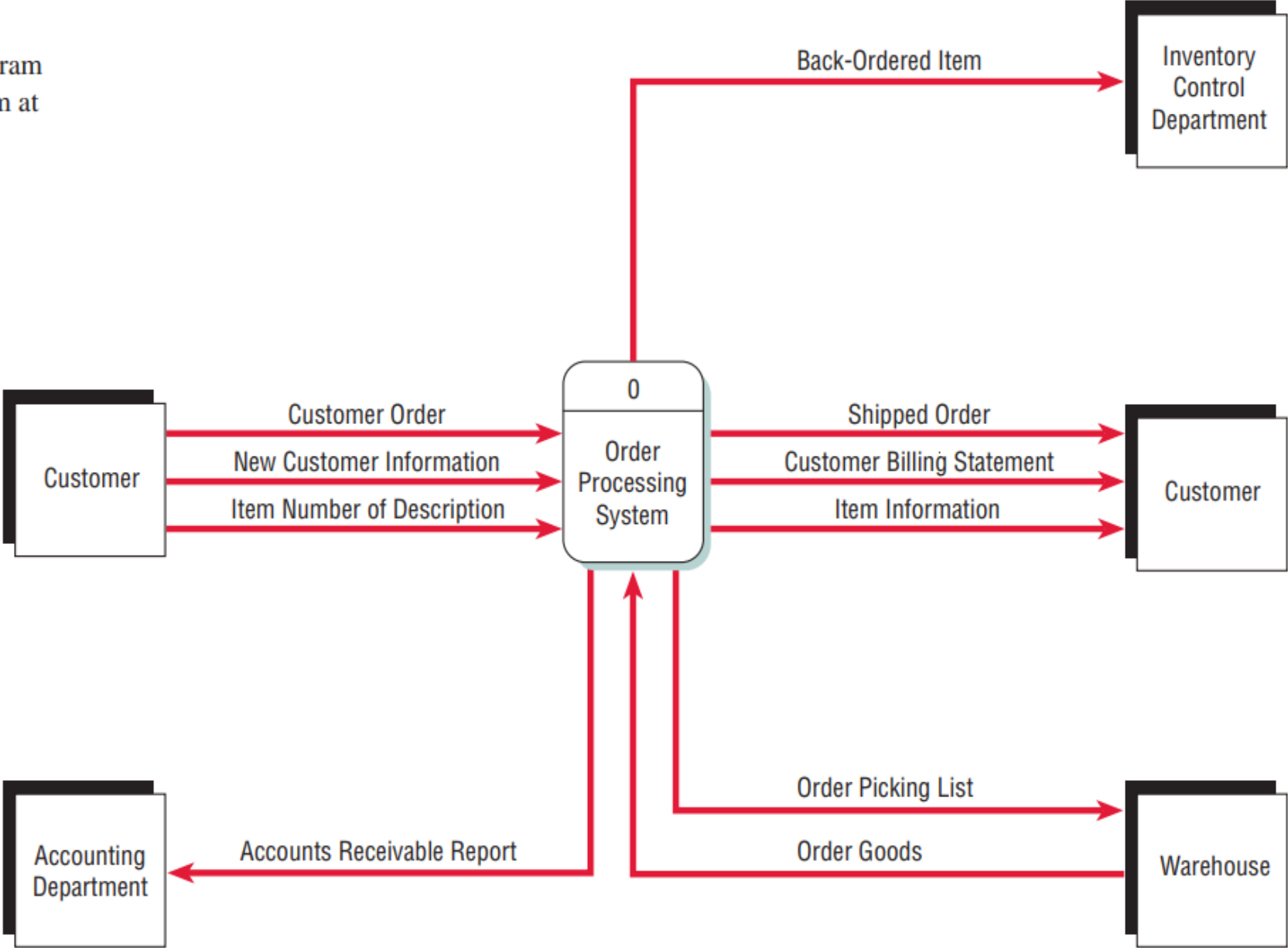


Typical errors that can occur in a data flow diagram (payroll example).

Example

FIGURE 7.16

A context-level data flow diagram for the order processing system at World's Trend.



Context Diagram

Level 0 DFD

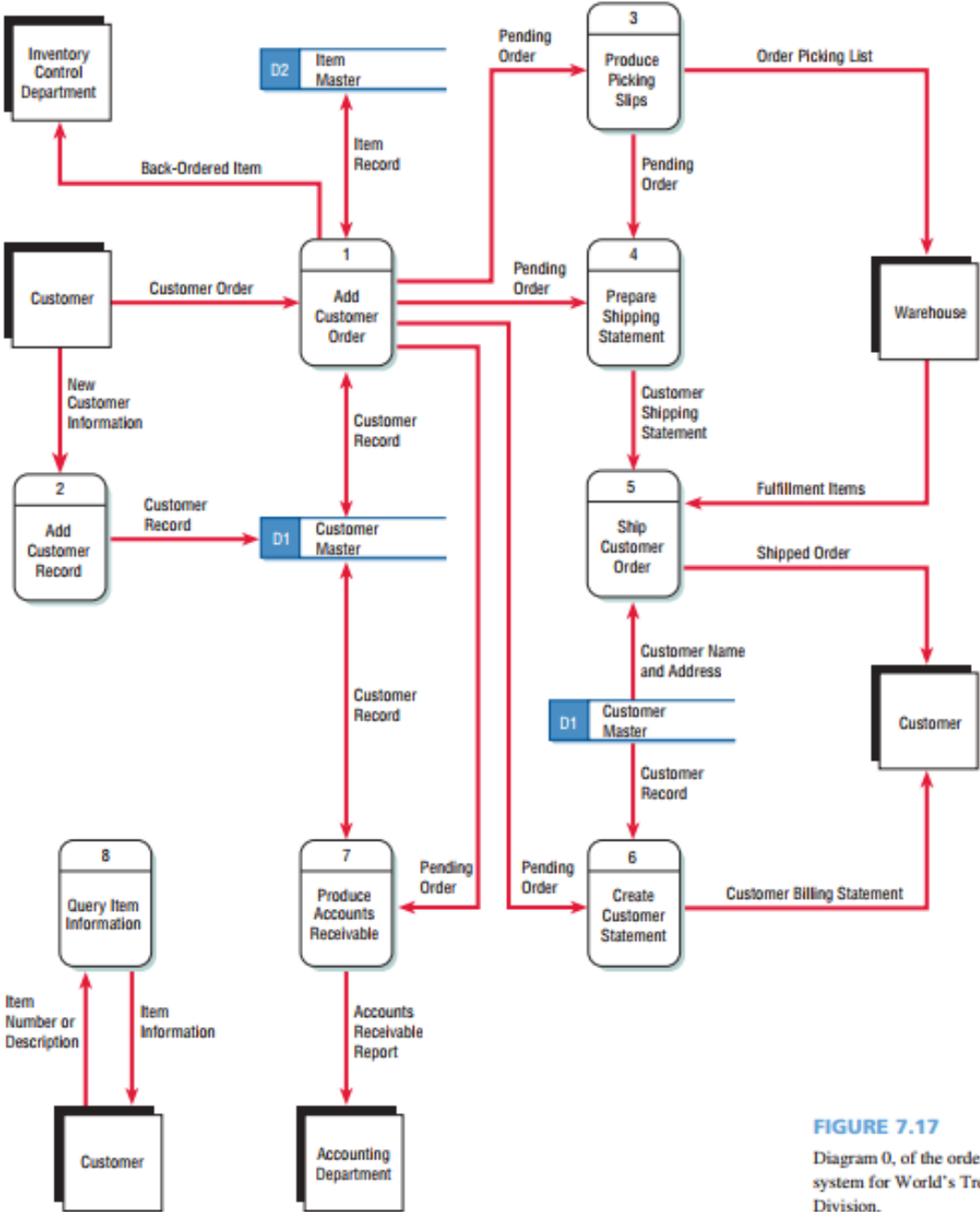


FIGURE 7.17
Diagram 0, of the order processing
system for World's Trend Catalog
Division.

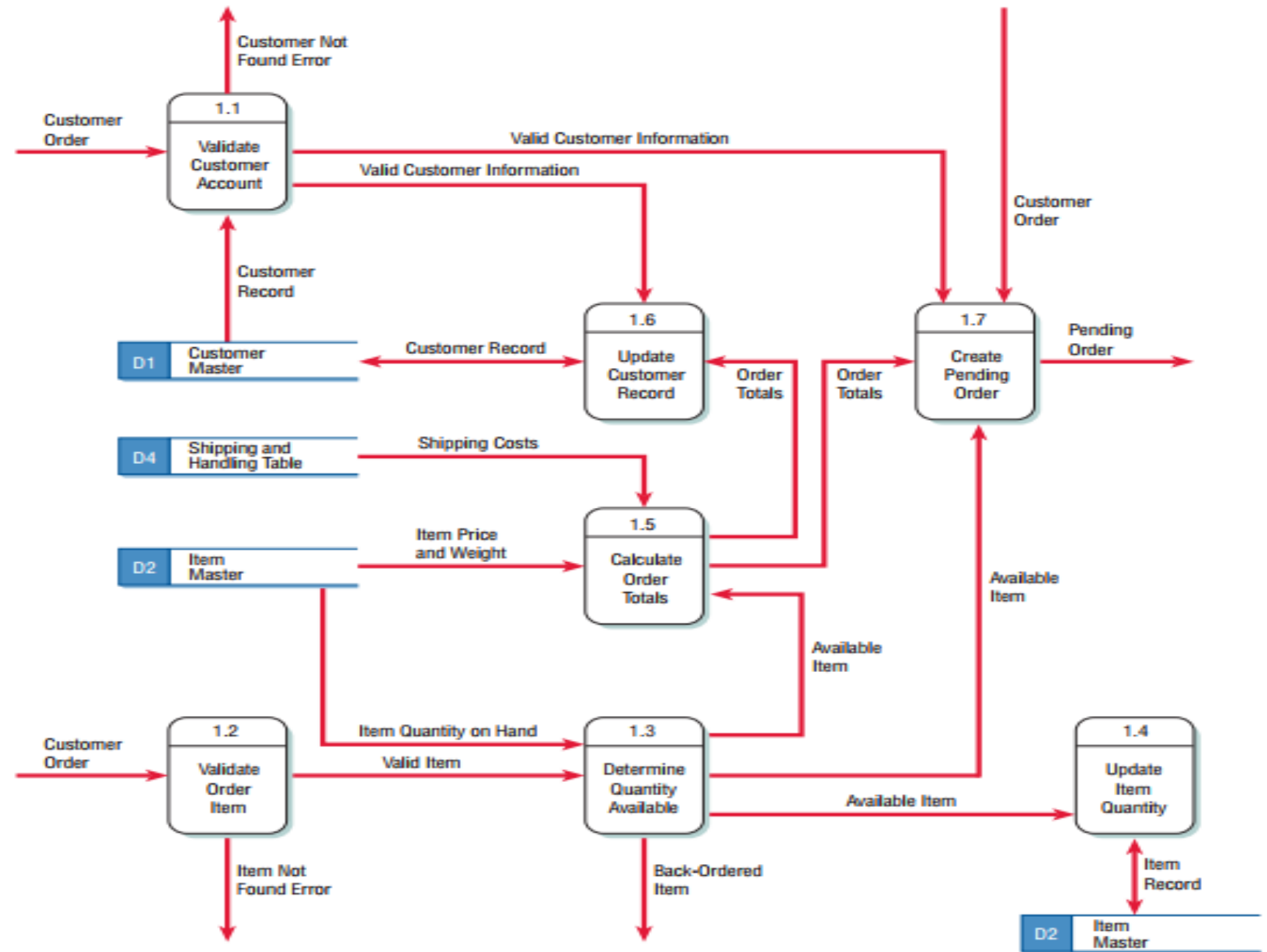


FIGURE 7.18

Diagram 1, of the order processing system for World's Trend Catalog Division.