

# Data Flow Diagram

Sistem Informasi



DIKTISAINTEK  
BERDAMPAK



# Pengertian & sejarah singkat

---

**DFD memetakan bagaimana data bergerak—bukan bagaimana teknologi dibangun.**

Penggunaan DFD sebagai alat pemodelan dipopulerkan oleh DeMarco & Yourdon (1979) dan Gane & Sarson (1979), dengan menggunakan metodologi Structured Systems Analysis.

DFD menggambarkan **aliran data dalam suatu sistem informasi**, baik sistem yang sudah ada maupun sistem baru, dari sudut pandang logis tanpa mempertimbangkan lingkungan fisik tempat data tersebut berada

1979

DeMarco & Yourdon

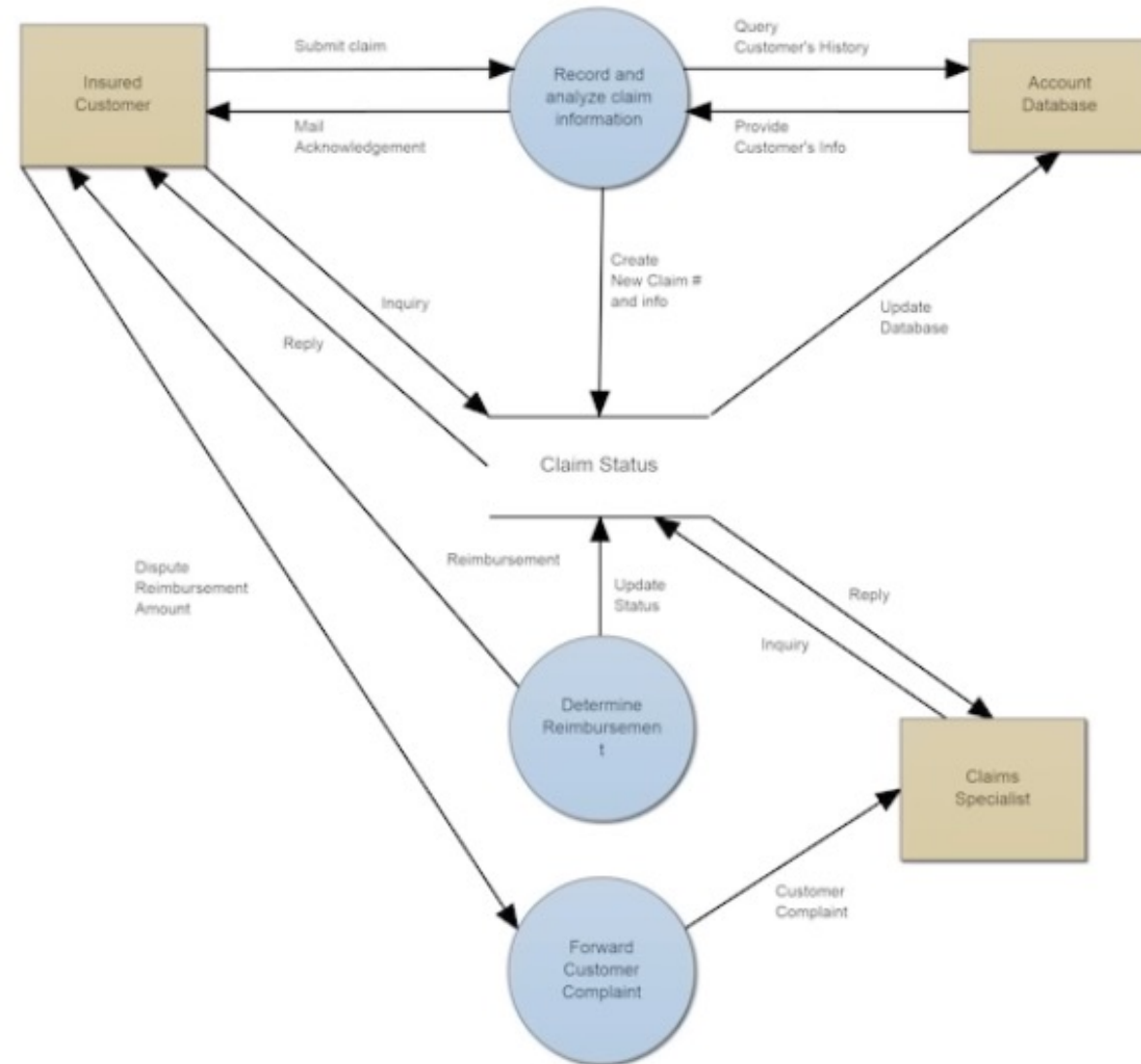


1979

Gane & Sarson

DFD merupakan **alat perancangan sistem** yang berfokus pada **aliran data** dan mudah dipahami serta dikomunikasikan oleh programmer.

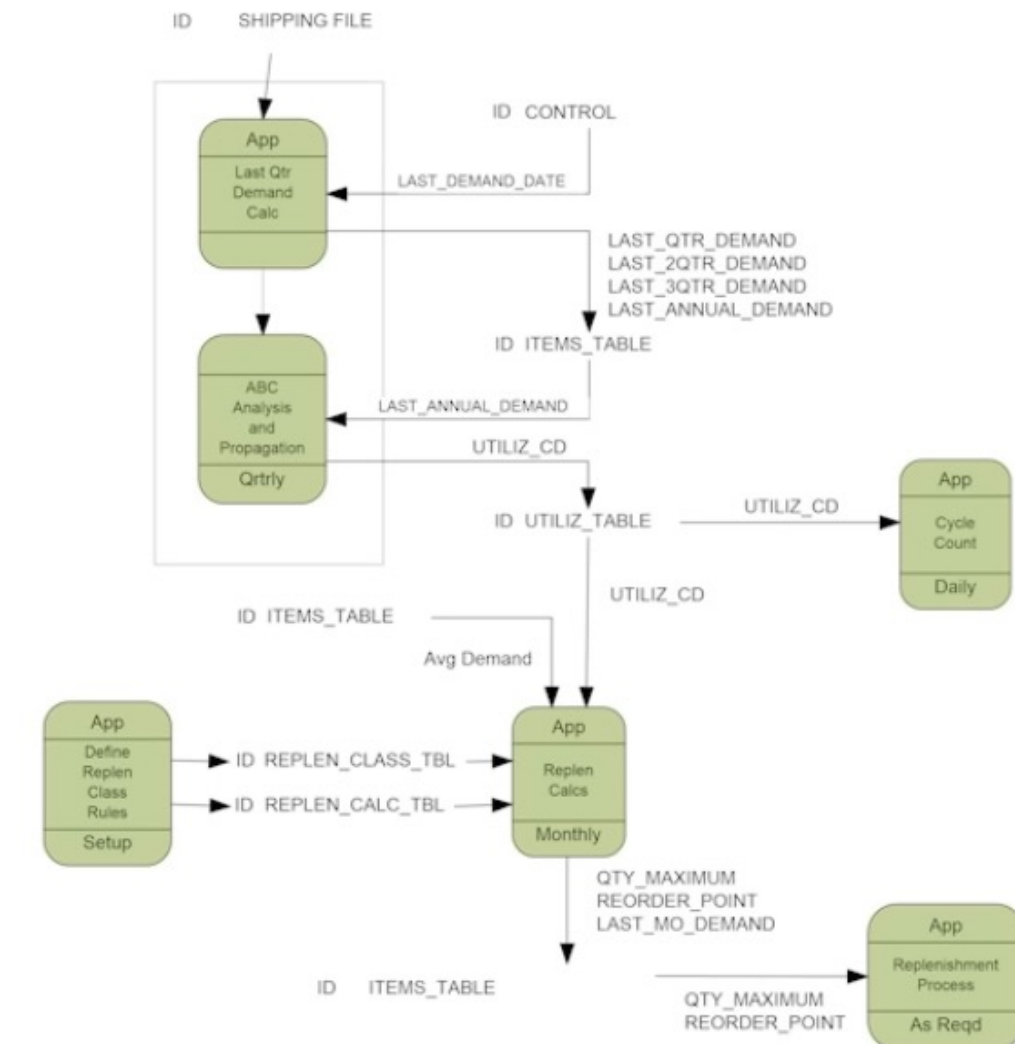
Data Flow Diagram - Insurance Claims



1979

DeMarco & Yourdon

Data Flow Diagram - Cycle Count



1979

Gane & Sarson

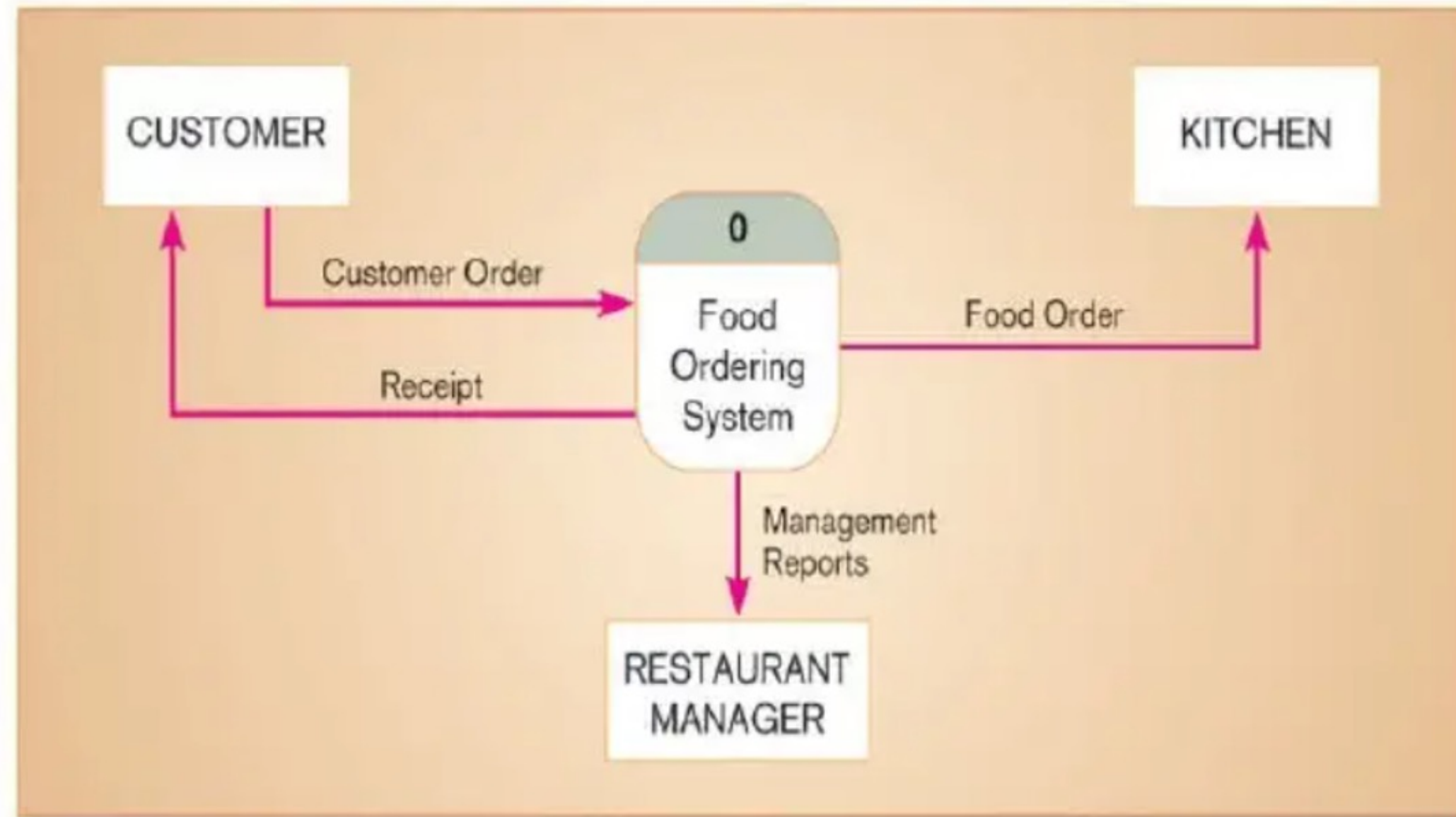
# Symbol-Symbol DFD

|                 | Data Flow Diagram Element                                                                                                                              | Typical Computer-Aided Software Engineering Fields                                                                                | Gane and Sarson Symbol | DeMarco and Yourdan Symbol |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|------------------------|----------------------------|
| Process         | Every <i>process</i> has<br>A number<br>A name (verb phrase)<br>A description<br>One or more output data flows<br>Usually one or more input data flows | Label (name)<br>Type (process)<br>Description (what is it)<br>Process number<br>Process description (Structured English)<br>Notes |                        |                            |
| Data flow       | Every <i>data flow</i> has<br>A name (a noun)<br>A description<br>One or more connections to a process                                                 | Label (name)<br>Type (flow)<br>Description<br>Alias (another name)<br>Composition (description of data elements)<br>Notes         |                        |                            |
| Data store      | Every <i>data store</i> has<br>A number<br>A name (a noun)<br>A description<br>One or more input data flows<br>Usually one or more output data flows   | Label (name)<br>Type (store)<br>Description<br>Alias (another name)<br>Composition (description of data elements)<br>Notes        |                        |                            |
| External entity | Every <i>external entity</i> has<br>A name (a noun)<br>A description                                                                                   | Label (name)<br>Type (entity)<br>Description<br>Alias (another name)<br>Entity description<br>Notes                               |                        |                            |

Typical CASE fields • Gane & Sarson • DeMarco & Yourdon — bentuk berbeda, makna tetap sama.

# Context Diagram — lihat batas sistem

**Figure 7-4** Context diagram of Hoosier Burger's food ordering system

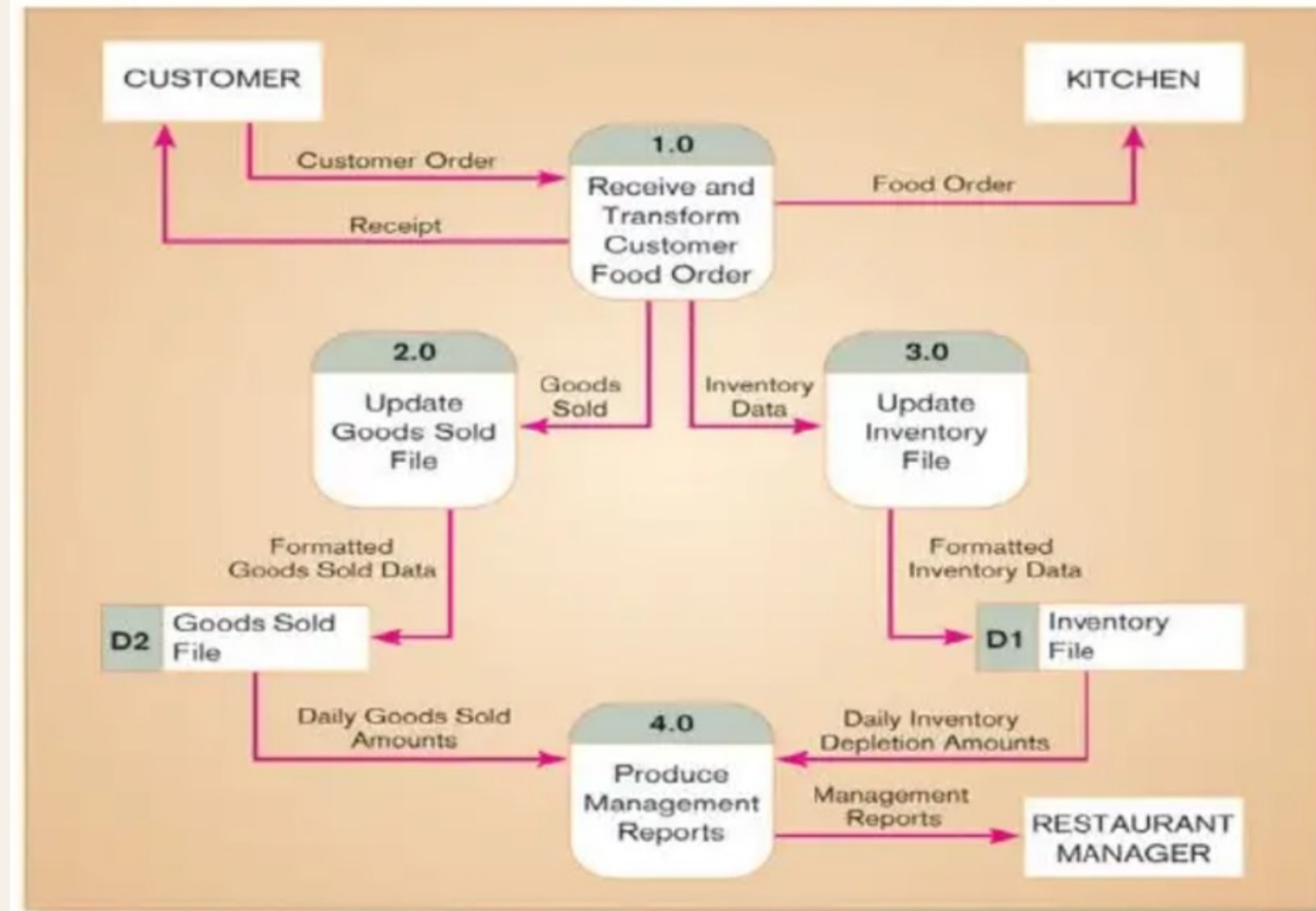


Context diagram shows the system boundaries, external entities that interact with the system, and major information flows between entities and the system.

Satu proses saja • tanpa data store • hanya entitas eksternal dan aliran informasi utama

# Level 0 — pecah sistem menjadi proses utama

**Figure 7-5** Level-0 DFD of Hoosier Burger's food ordering system



Processes are labeled 1.0, 2.0, etc. These will be decomposed into more primitive (lower-level) DFDs.

**Context Diagram menjawab:**

**“Sistem berinteraksi dengan siapa dan melalui data apa?”**

**Level-0 DFD menjawab:**

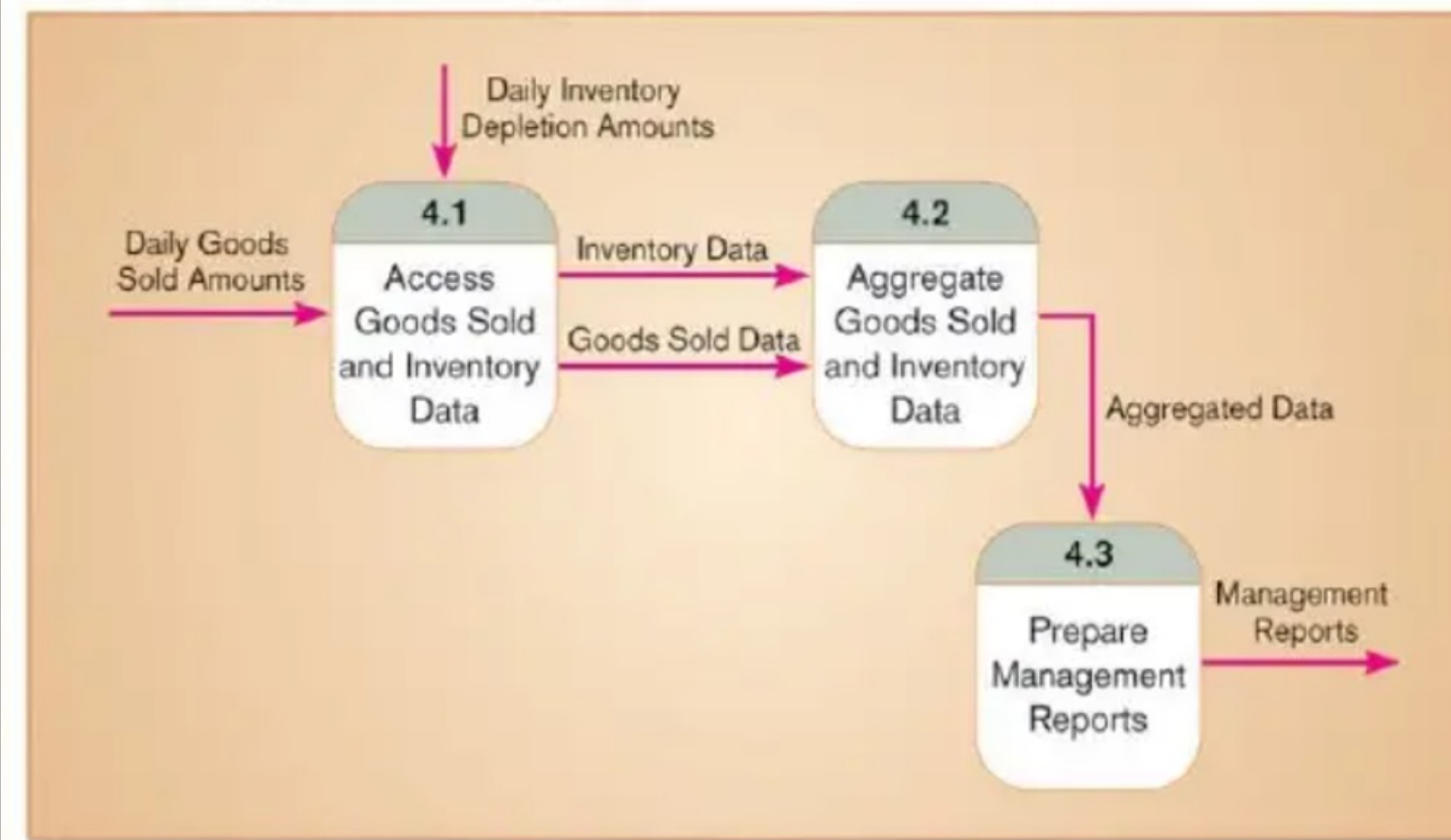
**“Apa saja proses utama yang terjadi di dalam sistem, data apa yang mengalir, dan data apa yang disimpan?”**

# Level 1 — buka satu proses

Proses 4.0 diuraikan tanpa mengubah aliran masuk/keluar pada level induk.

## 4.0 PRODUCE MANAGEMENT REPORTS

**Figure 7-8** Level-1 diagram showing the decomposition of Process 4.0 from the level-0 diagram for Hoosier Burger's food ordering system



Level-1 DFD shows the sub-processes of one of the processes in the Level-0 DFD.

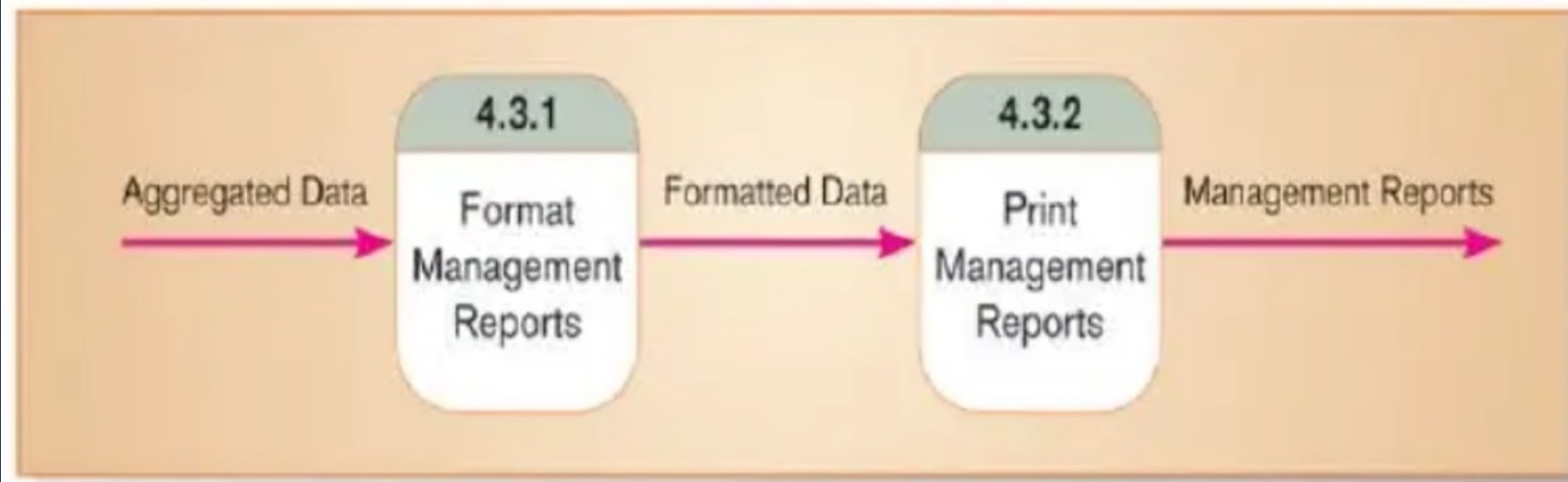
This is a Level-1 DFD for Process 4.0.

Processes are labeled 4.1, 4.2, etc. These can be further decomposed in more primitive (lower-level) DFDs if necessary.

**Akses → agregasi → persiapan laporan**

# Level-n — sampai proses primitif

**Figure 7-9** Level-2 diagram showing the decomposition of Process 4.3 from the level-1 diagram for Process 4.0 for Hoosier Burger's food ordering system



Level- $n$  DFD shows the sub-processes of one of the processes in the Level  $n-1$  DFD.

This is a Level-2 DFD for Process 4.3.

Processes are labeled 4.3.1, 4.3.2, etc. If this is the lowest level of the hierarchy, it is called a *primitive DFD*.

## Primitive DFD

Tingkat terendah: proses sudah cukup spesifik untuk dijelaskan dengan logika proses atau spesifikasi mini.

# Requirements for Creating a DFD

---

## 01 Mudah dibaca

Visual menarik, kontras, dan jalur jelas.

## 02 Nama tepat

Proses = frasa kerja; data = kata benda.

## 03 Tidak kompleks

Pecah diagram saat kepadatan meningkat.

## 04 Akurat

Model harus setia pada perilaku sistem.

# Aturan 1–3 — proses wajib memediasi

---

## RULE 1

**INPUT SAJA**

TIDAK VALID



**Input → Proses → Output**

## RULE 2

**OUTPUT TANPA SUMBER**

TIDAK VALID



**Data masuk → Proses → Hasil**

## RULE 3

**Store → Store**

TIDAK VALID

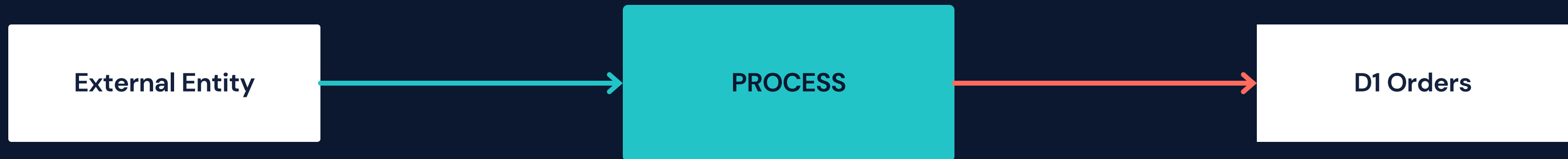


**Store → Proses → Store**

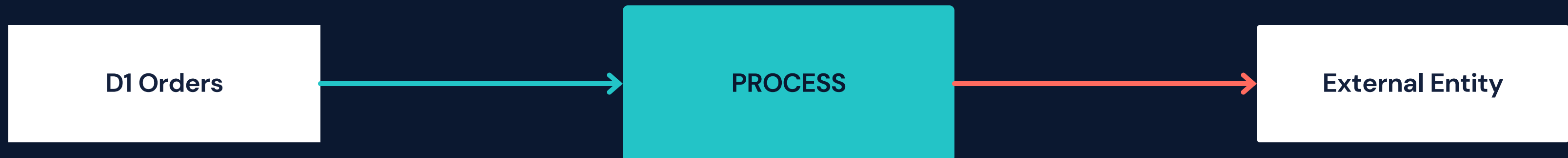
# Aturan 4–6 — jangan lewati proses

---

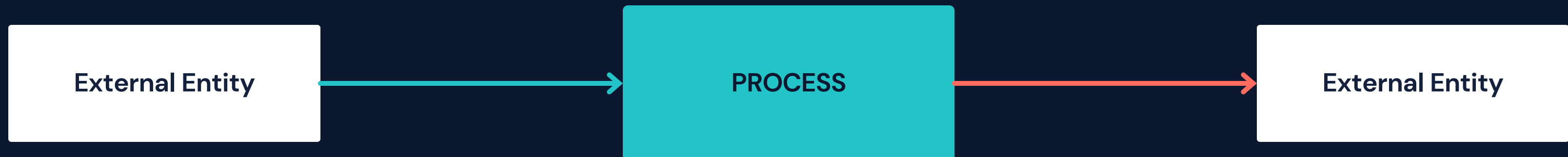
RULE 4 / Entity → Data Store



RULE 5 / Data Store → Entity



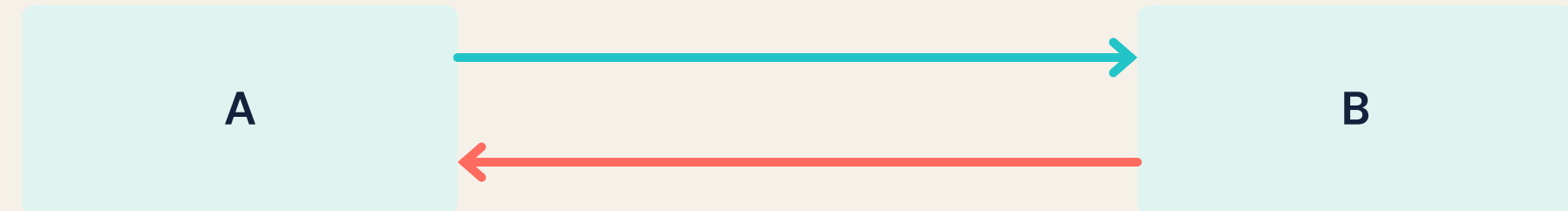
RULE 6 / Source → External Sink



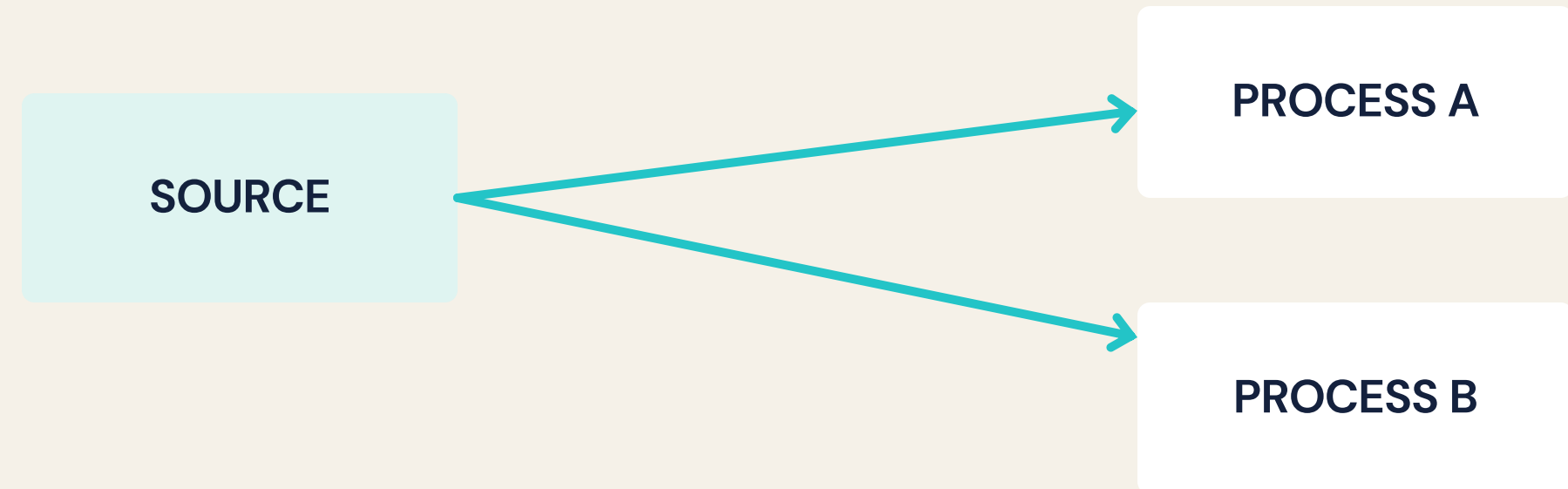
# Aturan 7-9 — arah, fork, dan join

---

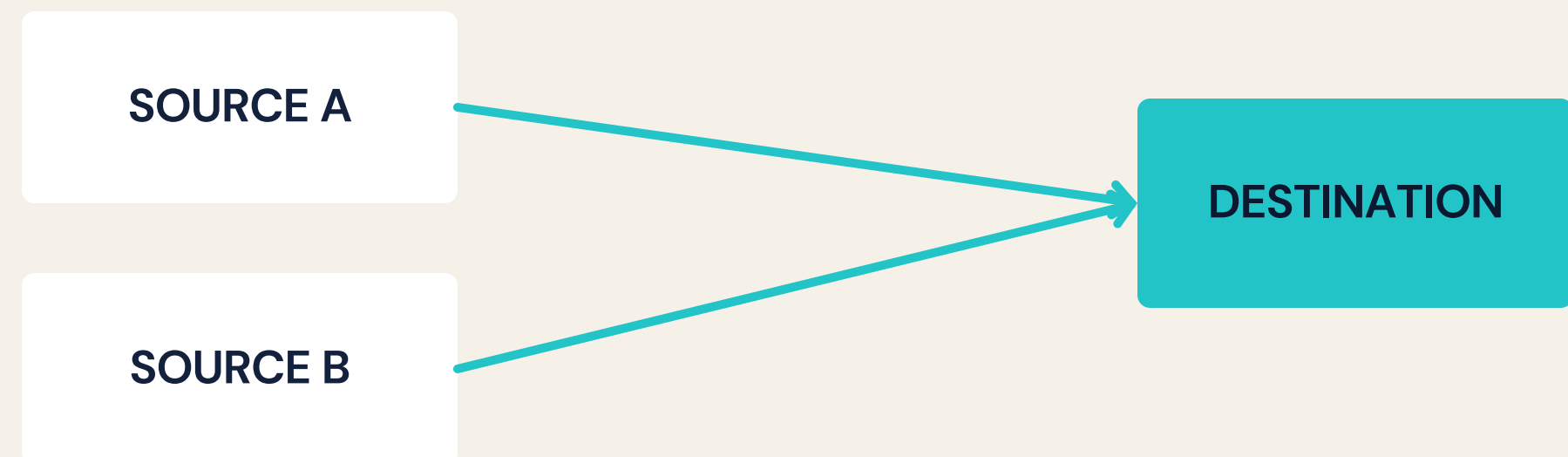
7 DUA ARAH = DUA FLOW



8 FORK — data sama bercabang



9 JOIN — data sama bergabung

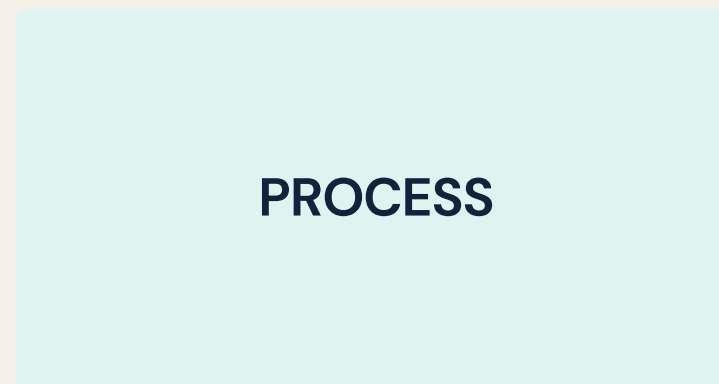


# Aturan 10-12 — recursion, update, retrieve

---

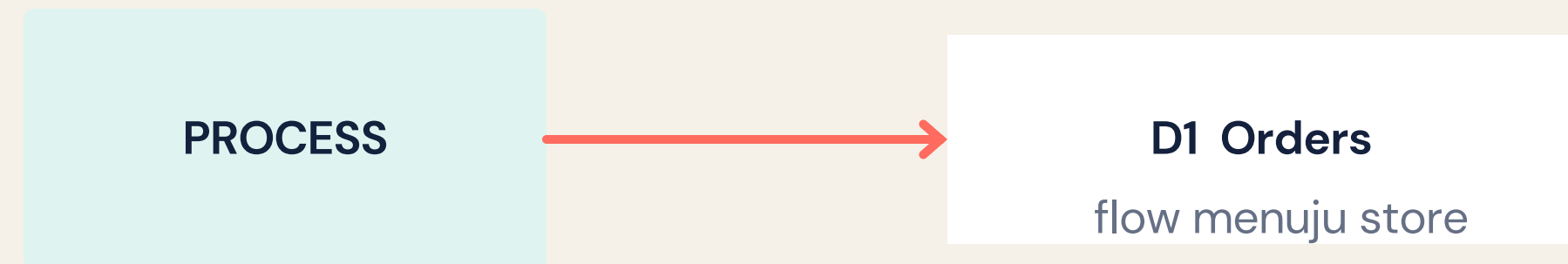
## 10 NO RECURSION

Proses tidak boleh mengirim data kembali langsung ke dirinya sendiri.



**HINDARI**

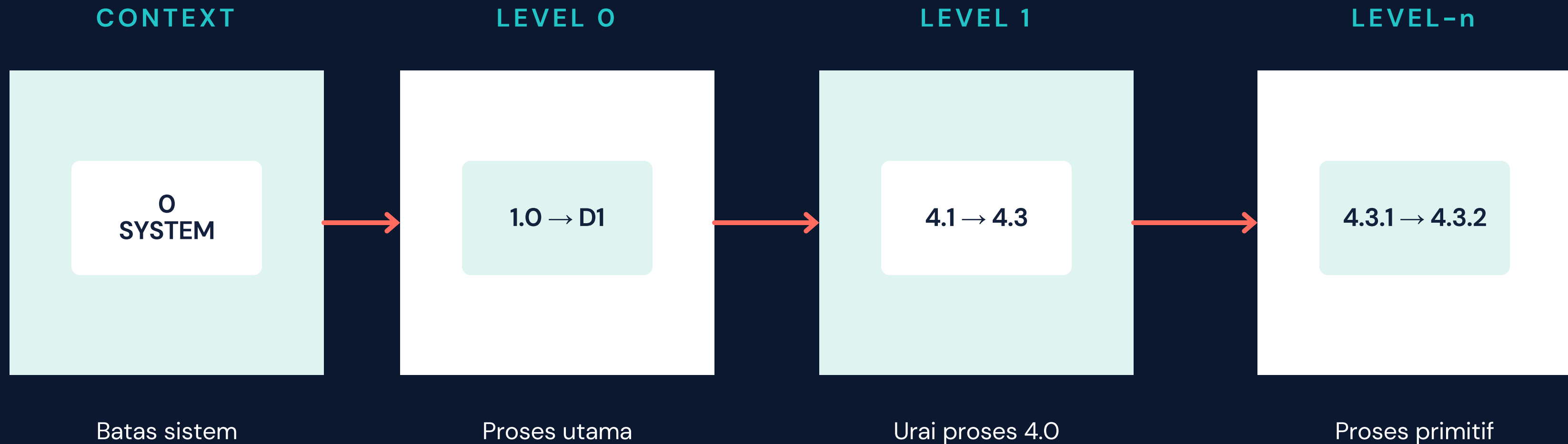
## 11 UPDATE



## 12 RETRIEVE



# Satu sistem, empat tingkat detail



**Balancing antarlevel • proses memediasi flow • update ≠ retrieve**

# Tiga hal yang perlu diingat

---

## 01

### ALIRAN

DFD menjelaskan bagaimana data bergerak di dalam batas sistem.

## 02

### HIRARKI

Context → Level 0 → Level 1 → Level-n menjaga kompleksitas tetap terbaca.

## 03

### VALIDITAS

Simbol, penamaan, balancing, dan aturan flow menentukan kualitas model.



ANALISIS DESAIN SISTEM INFORMASI

# THANK YOU



---

Pertanyaan, tanggapan, atau mari kita uji sebuah DFD bersama.