

# **PEMODELAN ANALISIS**

**OLEH: RAHMI HIDAYATI**



# TUJUAN PEMBELAJARAN

- Mahasiswa mampu memahami dan membedakan jenis-jenis metode analisis dan analisis terstruktur
- Mahasiswa mampu memahami pemodelan tingkah laku sistem
- Mahasiswa mampu membuat pemodelan analisis menggunakan ERD

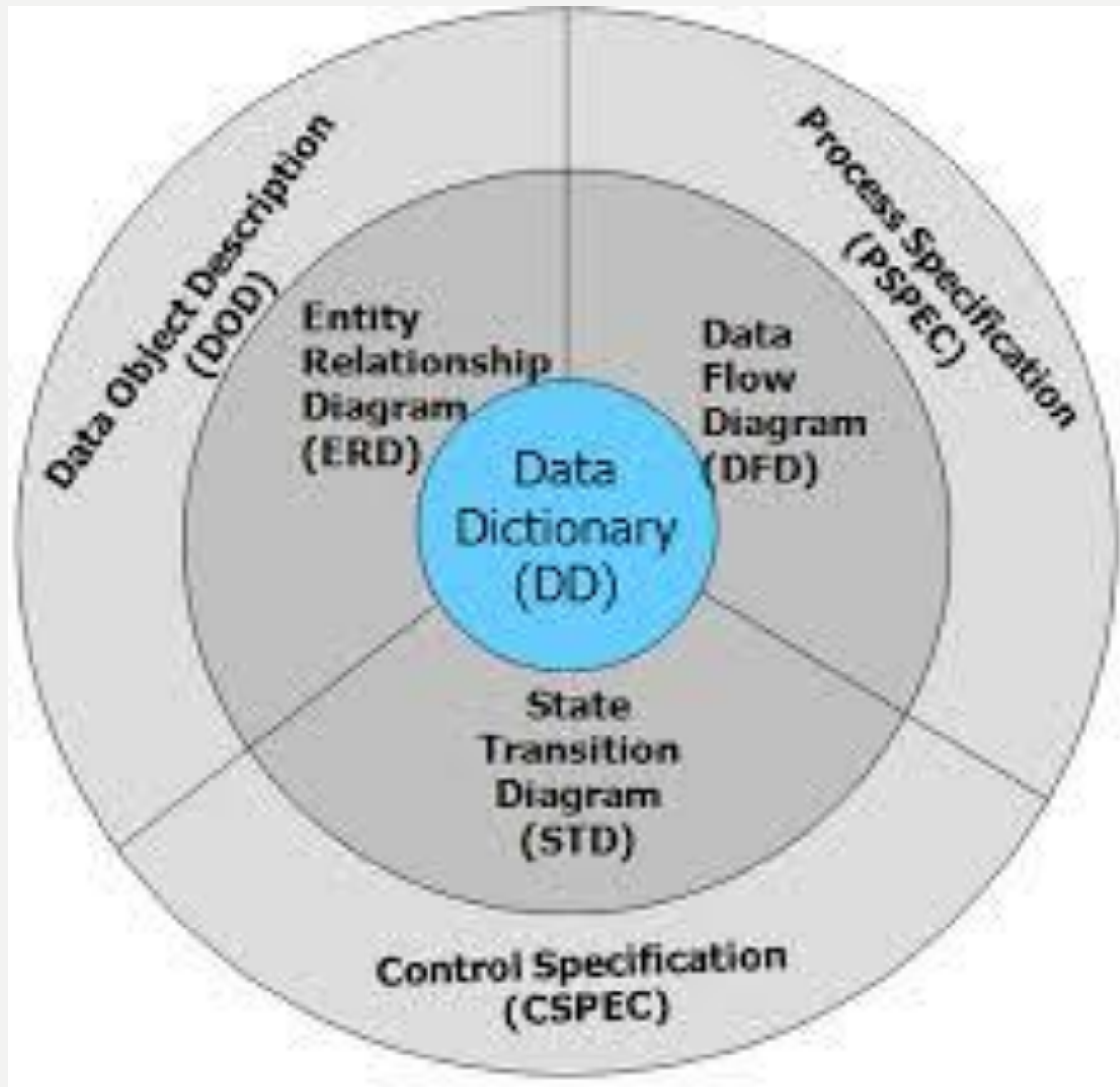
# PENGERTIAN

**Pemodelan analisis** adalah sekumpulan model dan teknik untuk melakukan analisis kebutuhan perangkat lunak.

Sasaran dari model analisis adalah :

1. Menggambarkan apa yang dibutuhkan untuk pelanggan.
2. Membangun dasar bagi pembuatan desain perangkat lunak.
3. Membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun.

# STRUKTUR MODEL ANALISIS



# STRUKTUR MODEL ANALISIS

## ***Data Dictionary (Kamus Data)***

- Penyimpanan yang berisi deskripsi dari semua obyek data yang digunakan atau diproduksi oleh perangkat lunak.

## ***Data Object Description (DOD)***

- Deskripsi atribut dari setiap objek data.

## ***Entity Relationship Diagram (ERD)***

- Menggambarkan hubungan antara objek data.

# STRUKTUR MODEL ANALISIS

## *Process Specification (PSPEC)*

- Deskripsi setiap fungsi yang disajikan pada DFD.

## *Data Flow Diagram (DFD)*

- Memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem.
- Menggambarkan fungsi-fungsi dan sub fungsi yang mentransformasikan aliran data.

# STRUKTUR MODEL ANALISIS

## *Control Specification (CSPEC)*

- Informasi tambahan mengenai aspek kontrol dari perangkat lunak.

## *State Transition Diagram*

- Menunjukkan bagaimana sistem bertingkah laku sebagai akibat dari kejadian *eksternal*.

# PEMODELAN DATA

- Pemodelan data menjawab serangkaian data spesifik yang relevan dengan berbagai aplikasi pemrosesan data.

Untuk memodelkan data dapat menggunakan:

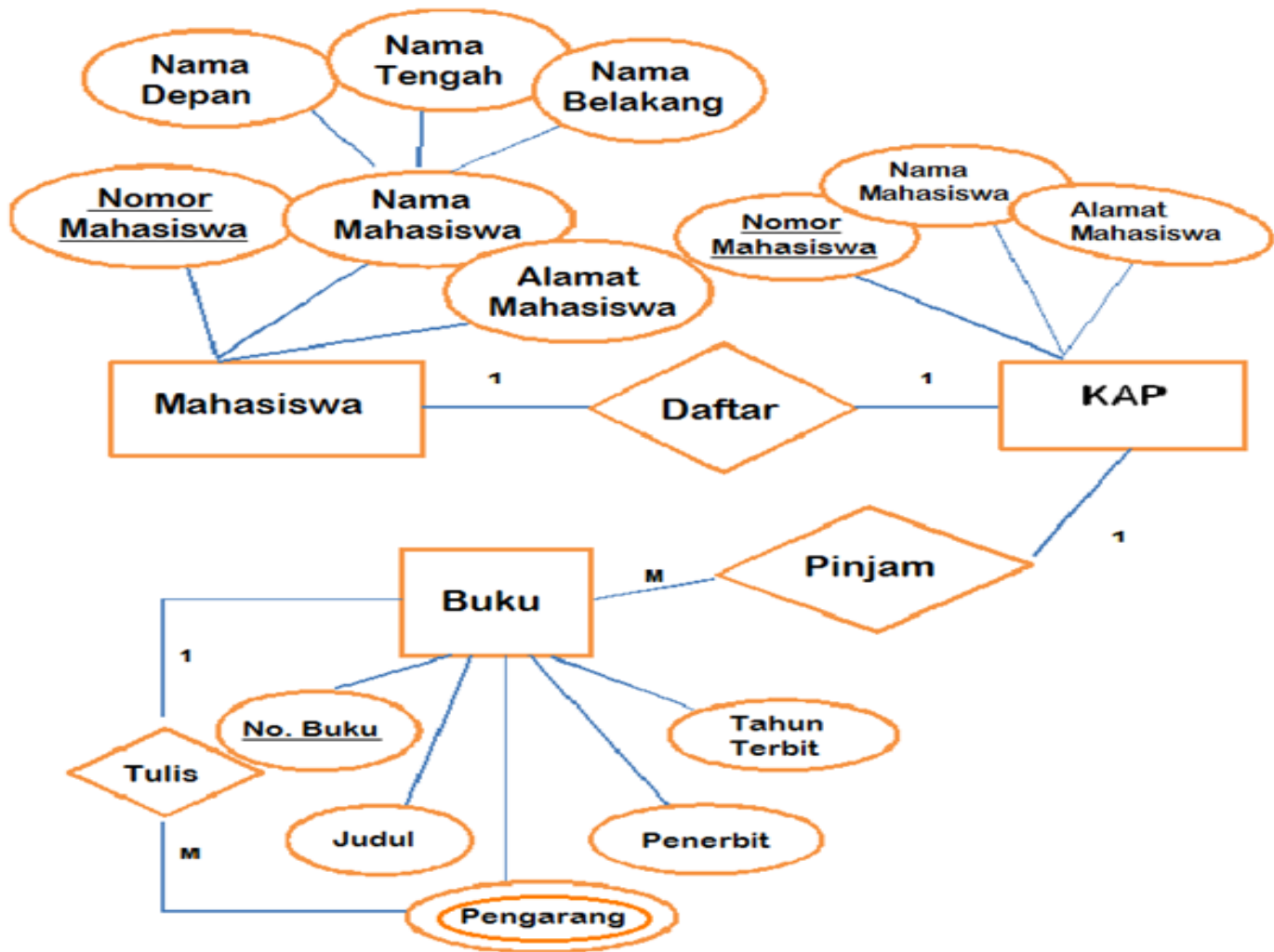
- *Entity Relationship Diagram (ERD).*
- *Conceptual Data Model (CDM).*
- *Physical Data Model (PDM)*



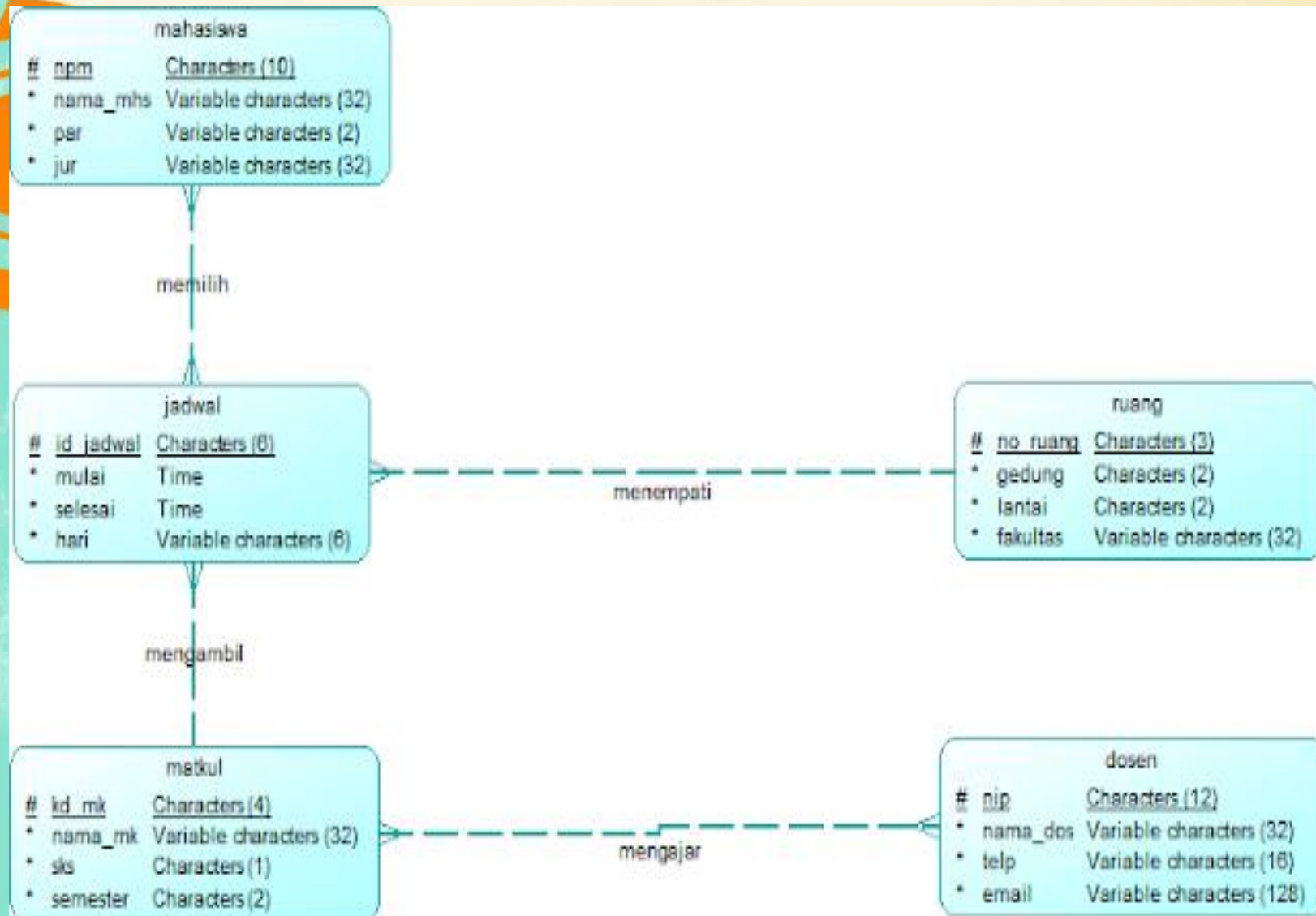
# PEMODELAN DATA

- ERD : umumnya digunakan untuk pemodelan basis data relasional.
- CDM : biasanya dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi basis data.
- PDM : konsep yang menjelaskan secara detail mengenai bagaimana data di simpan di dalam *database*.

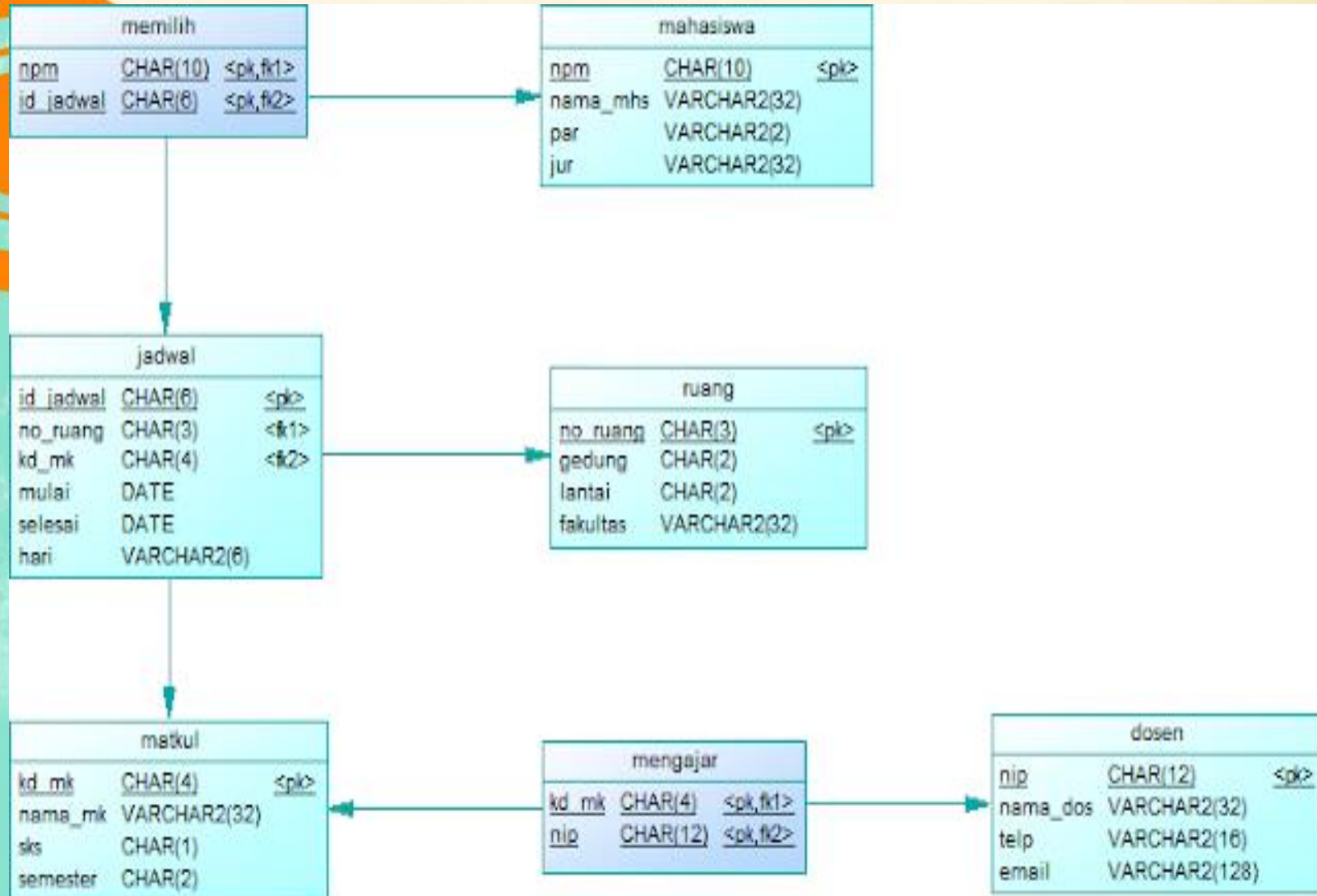
# CONTOH ERD



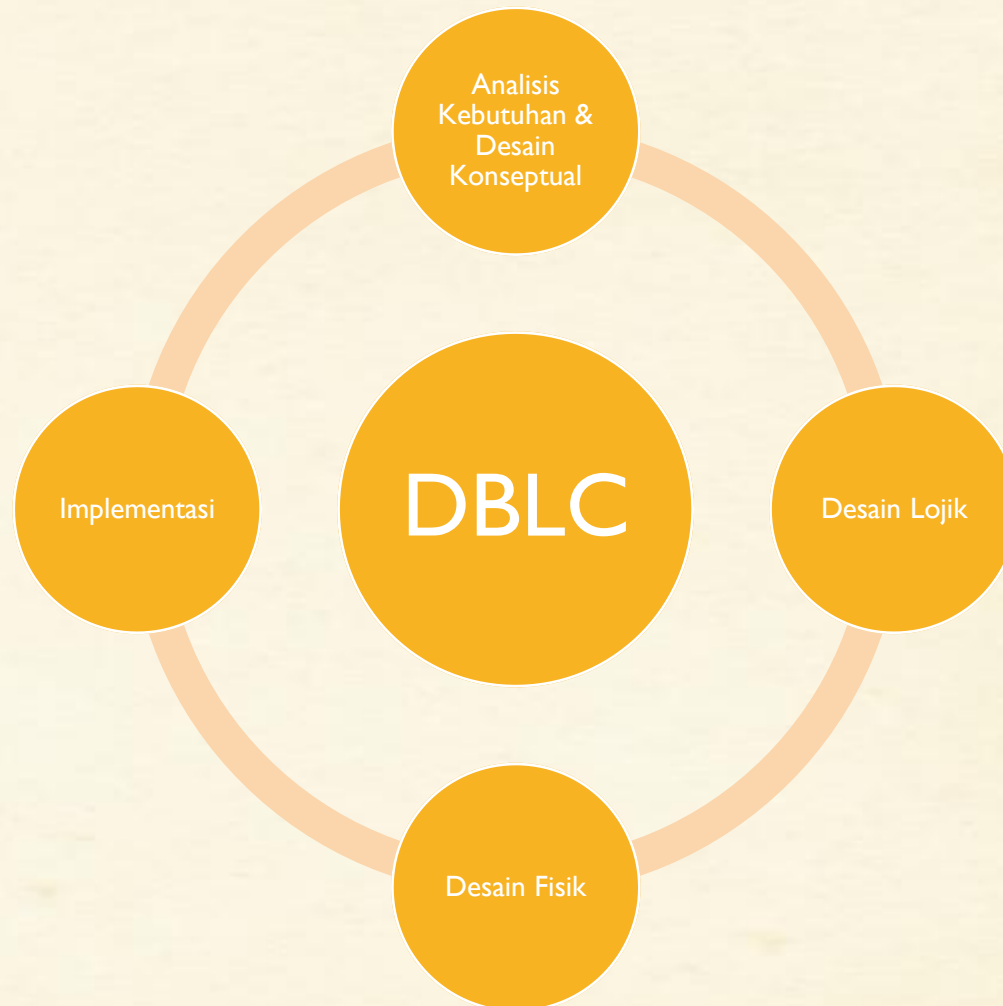
# CONTOH CDM



# CONTOH PDM



# ALUR HIDUP BASIS DATA - *DATABASE LIFE CYCLE*





# ***DATABASE LIFE CYCLE***

- Analisis kebutuhan : melalui wawancara antara produsen dan pemakai data, data apa saja yang butuh untuk di simpan, membuat kontrak spesifikasi basis data, ERD.
- Desain logik : membuat *Conceptual Data Model* (CDM).
- Desain fisik : membuat *Physical Data Model* (PDM).
- Implementasi : membuat query SQL, aplikasi ke DBMS.

# PEMODELAN DATA - ERD

Pada model data ERD ada 3 informasi yang saling berhubungan yaitu :

- Objek data atau entitas
- Atribut
- Hubungan

# OBJEK DATA, ATRIBUT dan HUBUNGAN



Karyawan

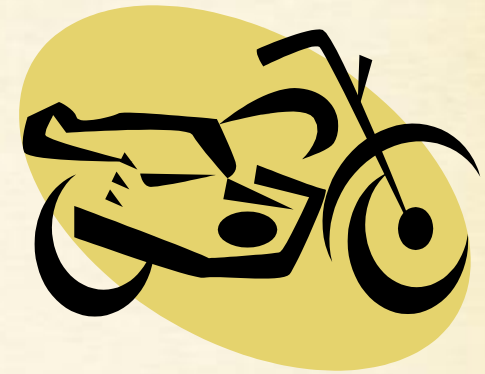
**Nama**

**NIP**

**Alamat**



Memiliki



Kendaraan

**Model**

**Plat nomor**

**No. Rangka**





# OBJEK DATA

- Representasi semua informasi gabungan yang harus dipahami perangkat lunak.

Contoh:

Karyawan adalah objek data. Gabungan informasi yang dimiliki karyawan adalah nama, nip, alamat, golongan.

# ATRIBUT

- Informasi yang dimiliki oleh objek data.
- Satu atribut atau lebih harus jadi pengidentifikasi atau kunci yang unik ketika ingin menemukan sebuah objek data.



# HUBUNGAN

- Menggambarkan hubungan suatu objek data ke objek data lainnya.
- Contoh :
  - Karyawan **memiliki** kendaraan
  - Karyawan **mempunyai** komputer
  - Karyawan **mengambil** pinjaman
  - Karyawan **bekerja pada** divisi

# KARDINALITAS

- **Kardinalitas** : mendefinisikan jumlah maksimum dari hubungan objek yang ikut serta dalam sebuah hubungan.
- Contoh :  
orang tua punya **banyak** anak dan anak hanya punya **satu** nama

# KARDINALITAS

- Model data harus dapat merepresentasikan jumlah peristiwa dari obyek didalam hubungan yang diberikan.
- Satu ke satu (1:1)
- Satu ke banyak (1:N)
- Banyak ke banyak (M:N)

# KARDINALITAS

- Contoh **Satu ke satu (1:1)**: Sebuah negara memiliki satu ibukota.
- Contoh **Satu ke banyak (1:N)**: Seorang ibu dapat memiliki banyak anak tetapi seorang anak hanya dapat memiliki satu ibu kandung.
- Contoh **Banyak ke banyak (M:N)**: Seorang paman dapat memiliki banyak keponakan, sementara itu seorang keponakan dapat memiliki banyak paman.

# MODALITAS

- Modalitas dari suatu hubungan adalah **noI** bila tidak ada kebutuhan eksplisit untuk hubungan yang terjadi atau hubungan itu bersifat **opsional**.
- Modalitas bernilai **satu** jika suatu kejadian dari hubungan merupakan **perintah**.

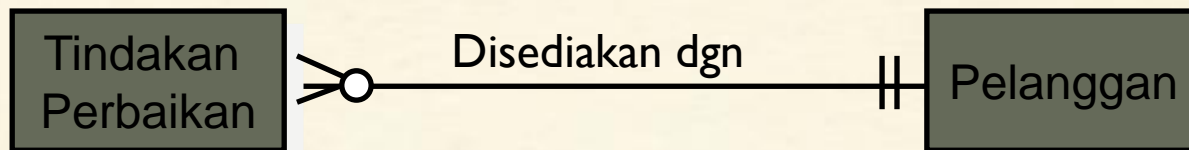
# KARDINALITAS & MODALITAS

**Kardinalitas:**

Ada banyak tindakan perbaikan

**Kardinalitas:**

Pelanggan tunggal menunggu tindakan perbaikan



**Modalitas**

Ada situasi dimana tindakan perbaikan tidak diperlukan

**Modalitas :**

Untuk melakukan tindakan perbaikan harus ada pelanggan



# CONTOH NOTASI ERD YANG LAIN



Tujuan Utama dari ERD adalah mewakili objek data dan hubungan antara objek data

# PEMODELAN FUNGSIONAL DAN ALIRAN INFORMASI

- Informasi ditransformasikan pada saat informasi mengalir melalui sebuah sistem berbasis komputer.
- Sistem tersebut menerima *input* dengan berbagai cara dan menghasilkan suatu *output*.
- Akibatnya pengguna dapat menciptakan suatu model aliran bagi setiap sistem berbasis komputer tanpa melihat ukuran dan kompleksitasnya.

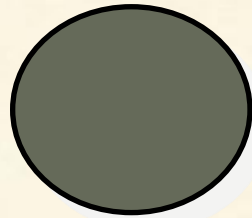
# ***DATA FLOW DIAGRAM (DFD)***



# NOTASI DASAR DFD



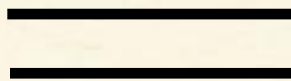
**external entity**



**process**



**data flow**



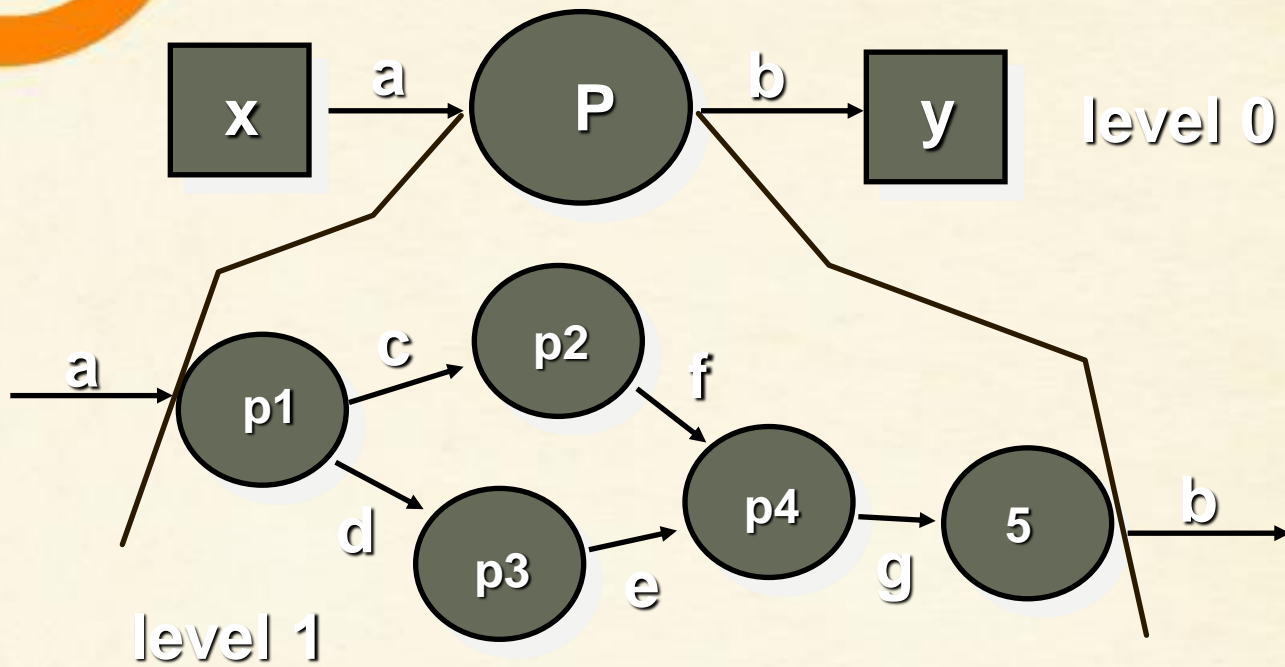
**data store**



# LEVELISASI DFD

- DFD tingkat 0 atau konteks diagram mewakili seluruh elemen sistem sebagai sebuah *bubble* tunggal dengan data *input* dan *output* yang ditunjukkan oleh anak panah masuk dan keluar secara berurutan.
- DFD tingkat satu merepresentasikan sub fungsi dari seluruh sistem yang digambarkan dalam konteks diagram.

# HIRARKI PADA DFD





# PEMODELAN TINGKAH LAKU

- Keadaan (*state*) adalah model tingkah laku yang dapat diobservasi.
- Perubahan keadaan (*State transistion*) adalah pergerakan dari satu keadaan ke keadaan lain.
- *Event* adalah kejadian yang membuat sistem menampilkan suatu bentuk tingkah laku.
- *Action* adalah proses yang terjadi sebagai konsekuensi dari adanya transisi.



# LANGKAH UNTUK PEMODELAN TINGKAH LAKU

- Buat daftar sejumlah keadaan yang berbeda dari sistem.
- Analisa bagaimana sistem membuat sebuah transisi ke transisi berikutnya.
- Gambarkan dalam sebuah *State Transition Diagram (STD)*.



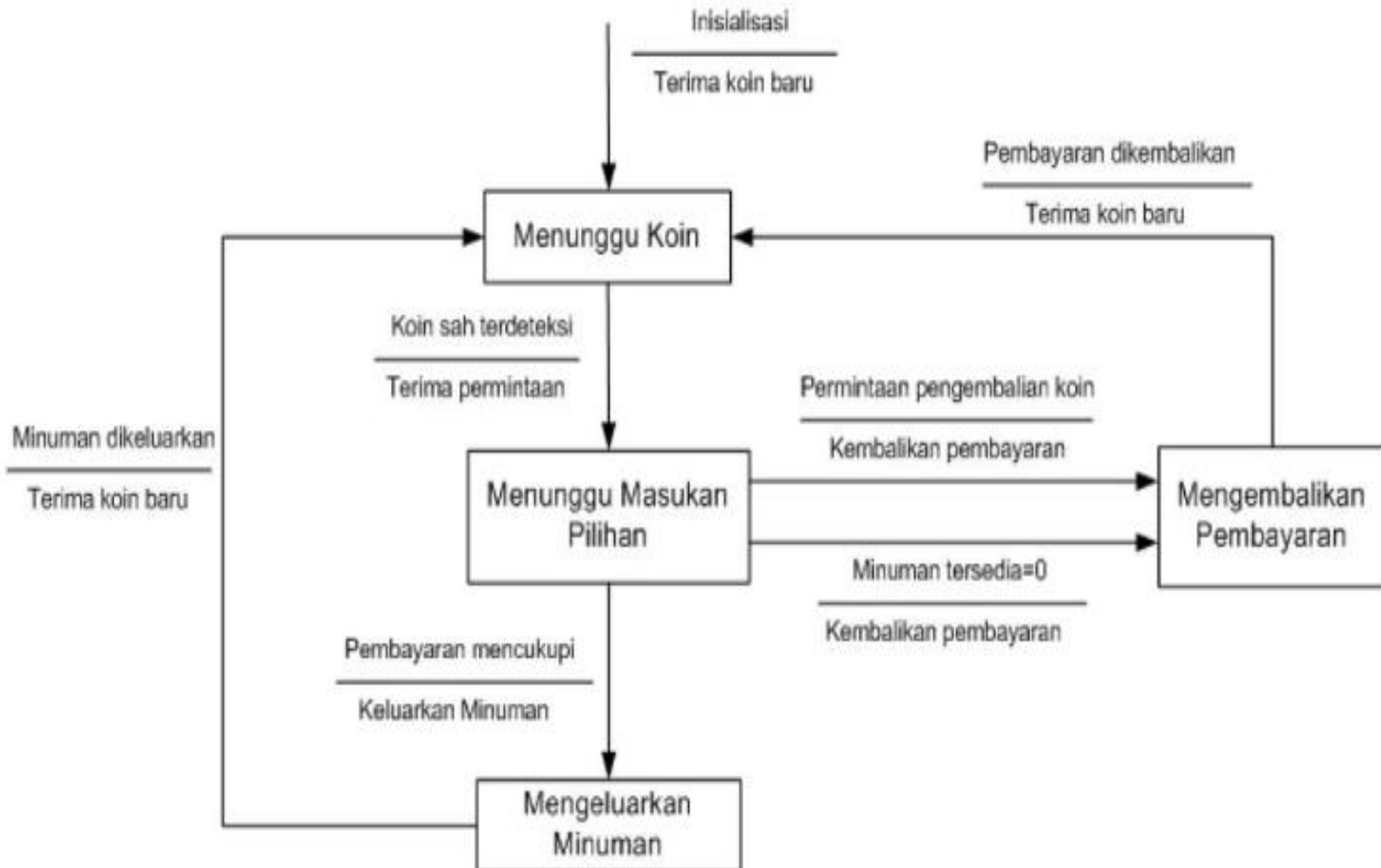
# NOTASI STD

**Keadaan**

Event yang menyebabkan transisi  
Action yang terjadi

**Keadaan  
akhir**

# CONTOH STD : MESIN OTOMATIS PENJUALAN MINUMAN

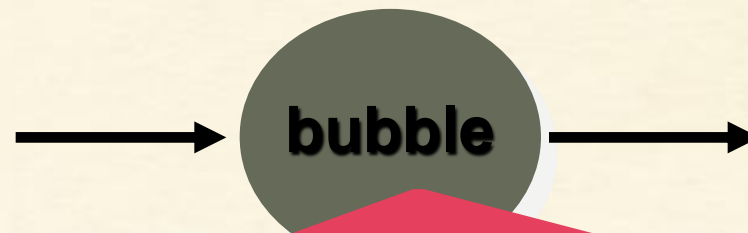




# ***PROCESS SPECIFICATION***

- Digunakan untuk menggambarkan semua proses aliran yang nampak pada tingkat akhir penyaringan.
- Spesifikasi bisa berupa teks naratif, bahasa desain program, persamaan matematika, tabel, diagram atau bagan.

# ***NOTASI PROCESS SPECIFICATION***



## **PSPEC**

- narrative**
- pseudocode (PDL)**
- equations**
- tables**
- diagrams and/or charts**

# CONTOH *PROCESS SPECIFICATION*

Dimensi sisi  
segitiga



## **PSPEC: Naratif Pemrosesan analisa segitiga**

Prosedur analisa segitiga;

Membaca dimensi sisi segitiga

Jika semua dimensi negatif maka terjadi pesan error

Jika dimensi terbesar kurang dari jumlah yang lain

.....



# KAMUS DATA

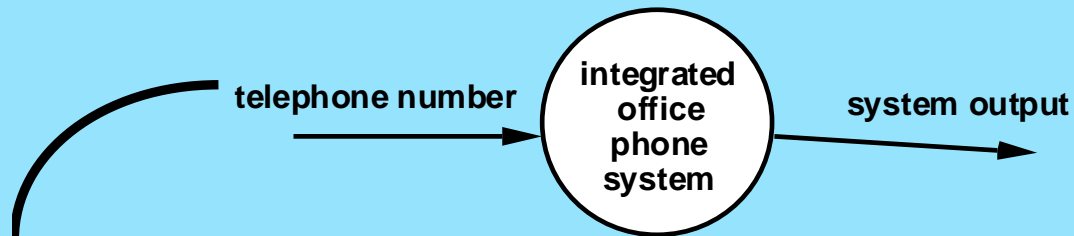
- Merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang jelas dan teliti sehingga pemakai dan analis sistem memahami secara umum mengenai *input*, *output* dan komponen penyimpanan.



# KONTEN KAMUS DATA

- **Nama**  
nama data, kontrol, penyimpan data atau entitas *eksternal*
- **Alias**  
nama lain yang digunakan
- ***Where used/how used***  
proses yang menggunakan data dan bagaimana ia digunakan
- ***Content desc***  
notasi untuk merepresentasikan isi
- ***Supplementary information***  
informasi lain mengenai tipe data, harga

# CONTOH



***Build the requirements dictionary:***

Name:	telephone number
Aliases:	phone number, number
Where/How used:	read-phone-number (input) display-phone-number (output) analyze-long-distance-calls (input)
Description:	telephone no. = [ local extension   outside no.   0 ] outside no. = 9 + [ service code   domestic no. ] service code = [ 211   411   611   911 ] domestic no. = ( ( 0 ) + area code ) + local number area code = *three numeral designator*
Format:	alphanumeric data





**TERIMA KASIH**