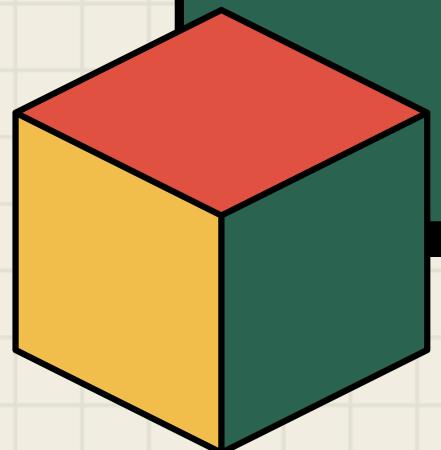


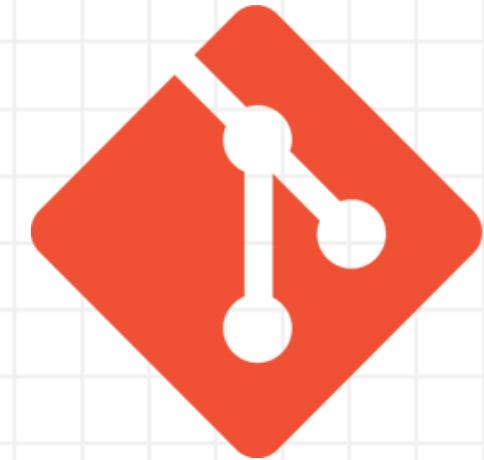
# Learning Progress Review



Git & Github

Datawarehouse and  
Data Modelling

# GIT



# git

GIT adalah sebuah version control system yang telah digunakan oleh para developer untuk dapat mengembangkan software secara bersamaan (Team)

Fungsi utama dari GIT adalah mengatur versi source code program dengan memberikan tanda baris serta code mana yang perlu ditambah ataupun diganti

# Perintah dasar GIT

GIT init -> Membuat sebuah repository yang berada pada file lokal dimana nantinya file tersebut bernama .git

GIT status -> Mengetahui status dari sebuah repository lokal

GIT add -> Menambahkan file baru yang ada pada repository yang telah dipilih.

GIT add ini untuk menambahkan file baru yang ada pada repository yang telah dipilih.

GIT push -> Mengirimkan perubahan file setelah di commit ke remote repository.

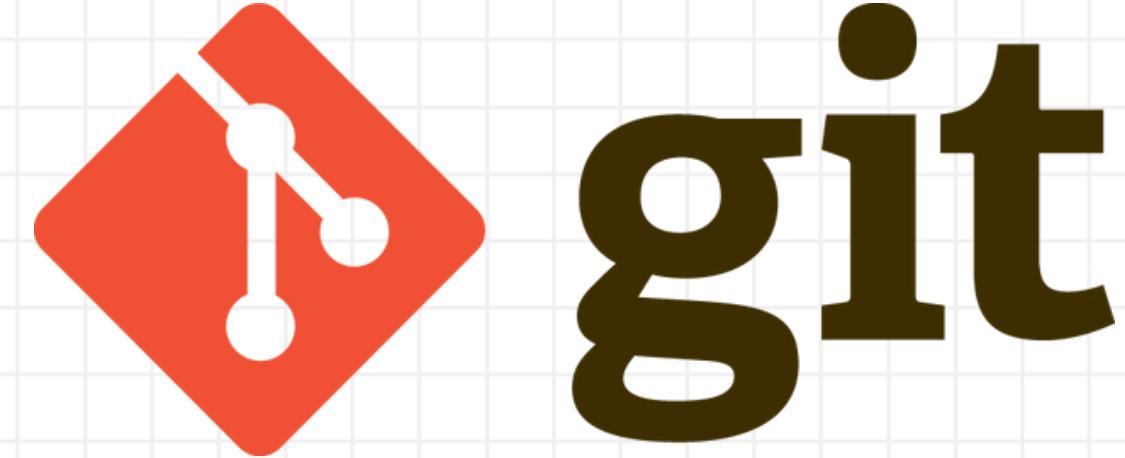
GIT branch -> Melihat seluruh branch yang ada dalam repository

GIT checkout -> Menukar branch yang aktif dengan branch yang telah dipilih

GIT merge -> Menggabungkan branch yang aktif dan yang telah dipilih

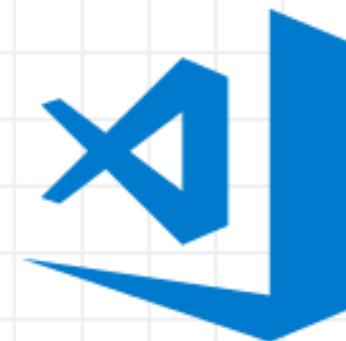
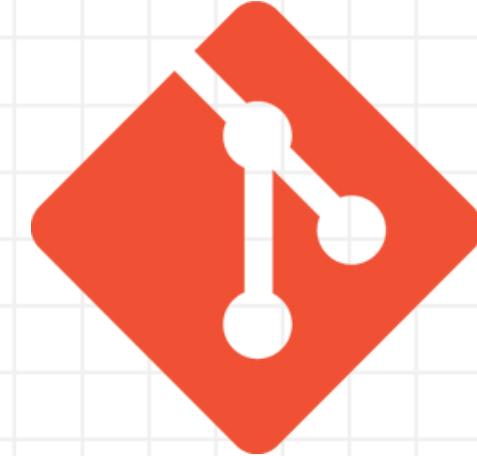
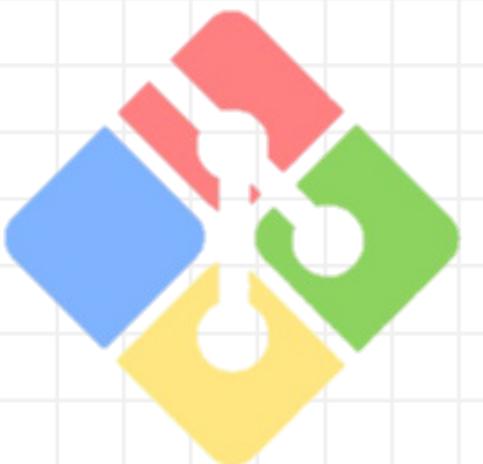
GIT clone -> Membuat salinan repository lokal

# Fitur GIT



- Version control system yang terdistribusi
- GIT memungkinkan developer untuk memiliki branch kode
- GIT bersifat atomic
- Media penyimpanan GIT berada dalam folder .git
- GIT memiliki data model
- GIT memiliki staging area atau index
- GIT sederhana dalam penggunaannya

Tools GIT



# Git dan GitHub

## Git

- Dikelola oleh The Linux Foundation
- Meng install software pada penyimpanan lokal
- Berfokus di version control serta code sharing
- Akses secara offline
- Menyediakan desktop interface bernama Git GUI
- Tidak menggunakan fitur user management
- Open sourced licensed
- Bersaing dengan Subversion, Mercurial, Rational Team, IBM, ClearCase, dan Concert

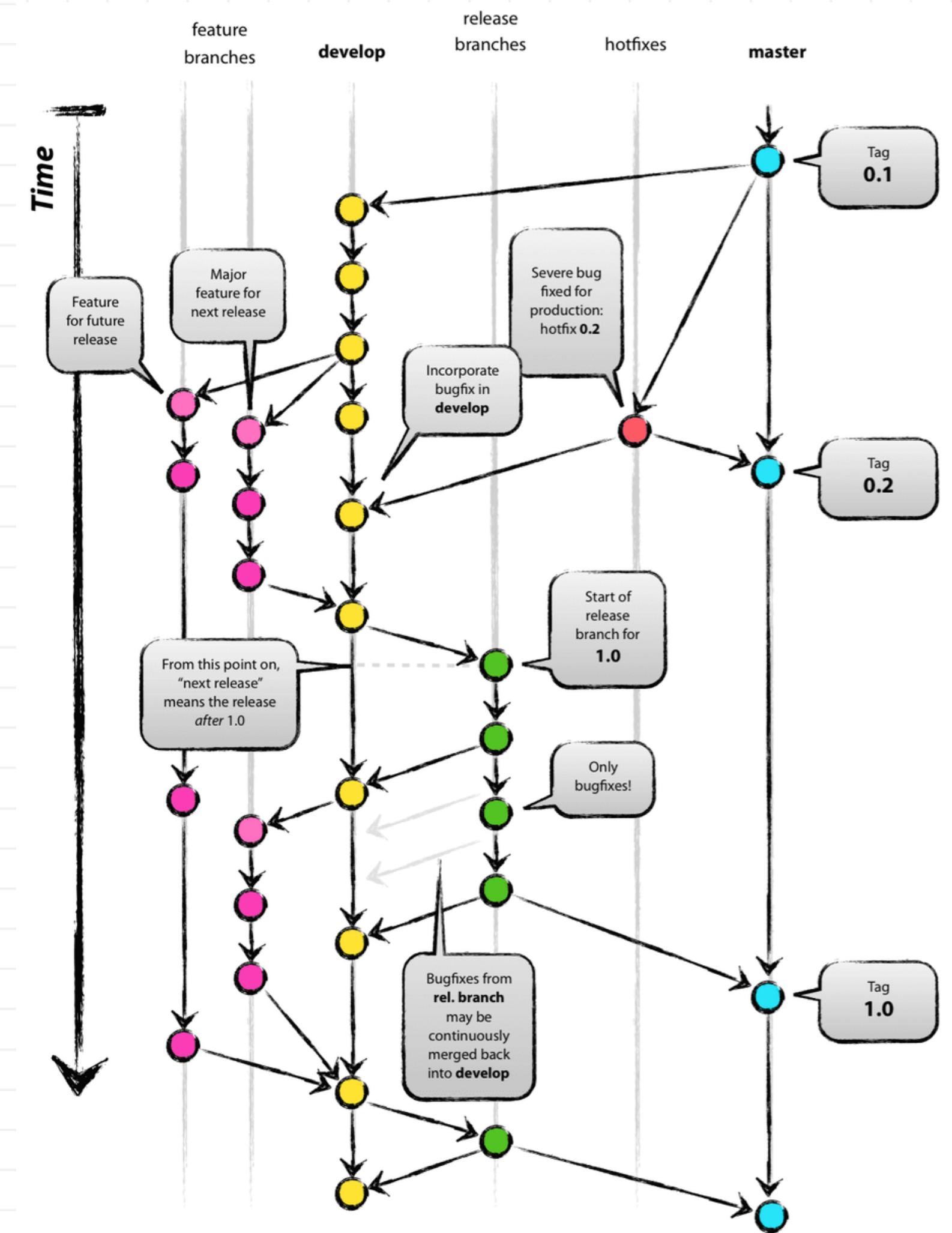
## GitHub

- Diakuisisi Microsoft pada tahun 2018
- Host melalui layanan cloud
- Akses secara online
- Menggunakan user management
- Fokus pada source code hosting
- Menggunakan nama desktop interface GitHub Desktop
- Pilihan bagi para pengguna gratis serta pengguna berbayar
- Bersaing dengan GitLab dan juga Atlassian BitBucket

# GITFLOW

gitflow adalah salah satu branching model yang sangat populer, gampangnya adalah gitflow ini, aturan-aturan yang disepakati oleh tim agar selama proses pengrajaan project terbentuk alur yang jelas

gitflow membagi repo kedalam 2 branch utama, yaitu master dan develop. Branch master adalah branch yang siap atau yang sedang live di production, sedangkan branch develop adalah branch yang masih dalam tahap pengembangan



- Master, branch utama dan tidak boleh ada 1 error pun disini, kadang disebut juga RC Release Candidate, yang berisi fitur-fitur siap dirilis
- Develop, branch tempat development, kontrol ada di Project Manager
- Features, branch untuk setiap fitur baru yang dikerjakan tim, disini tim programmer bebas bereksplorasi, ada error bagaimana pun tidak jadi masalah, karena terisolasi
- Hotfix, branch yang biasa digunakan untuk melakukan patch atau membetulkan error/issue yang terjadi
- Release, branch khusus versioning, biasanya dipakai untuk merilis versi-versi aplikasi yang berisi fitur-fitur stable (layak pakai)

# Data Modelling



## Definisi

Data modeling (pemodelan data) adalah proses menghasilkan skema deskriptif, hubungan antara berbagai jenis informasi untuk disimpan di dalam database.



## Tujuan

Salah satu tujuan data modeling adalah untuk menciptakan metode penyimpanan informasi yang paling efisien sambil memberikan akses dan pelaporan yang komprehensif.



## Fungsi

Data modelling dapat membuat elemen bisnis yang kompleks dan yang terbilang sangat teknis lebih mudah diakses oleh anggota staf (pengguna) yang kurang teknis.

# JENIS-JENIS DATA MODELLING

## Conceptual

Jenis model data ini mendefinisikan data tampilan pengguna tingkat tinggi. Tujuannya adalah untuk mengatur, memperluas, dan mendefinisikan konsep dan aturan bisnis.

## Logical

Pada model ini dapat menentukan bagaimana sistem harus diimplementasikan terlepas dari DBMS . Tujuannya adalah untuk mengembangkan pemetaan teknis regulasi dan struktur data.

## Physical

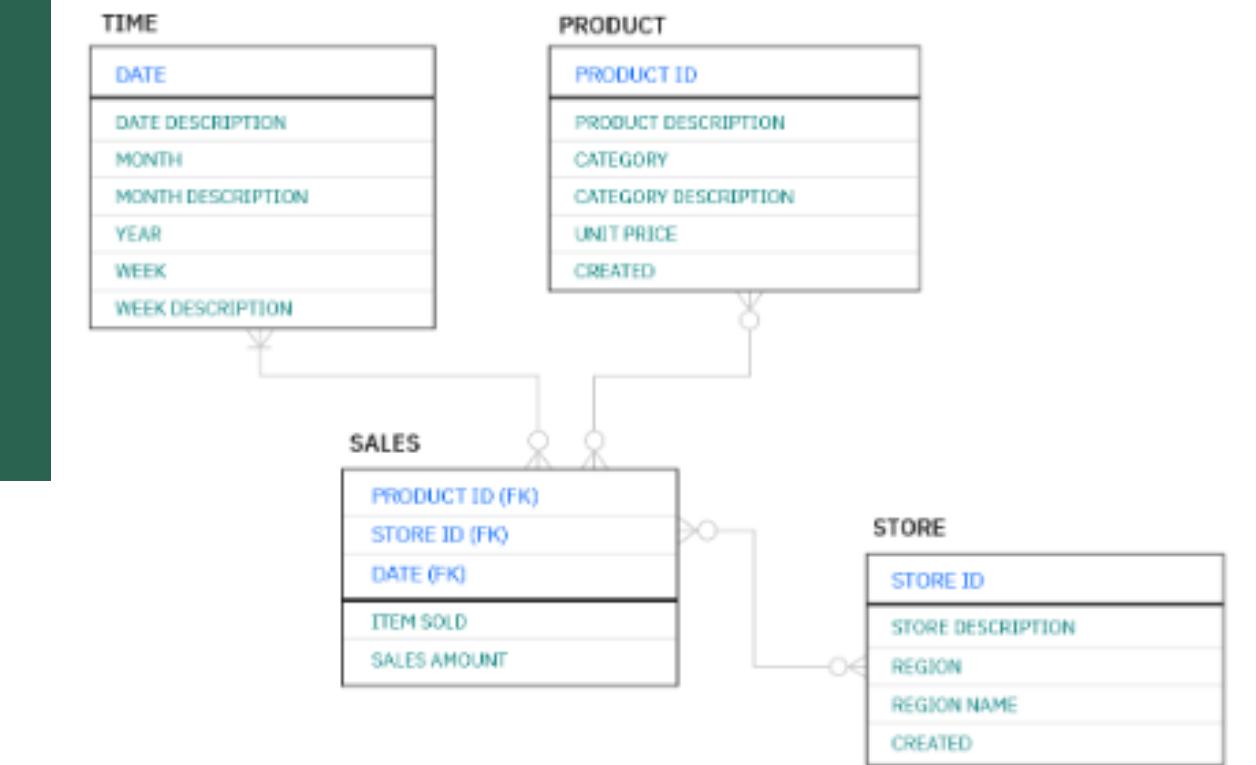
Ada jenis ini akan menjelaskan bagaimana sistem akan diimplementasikan menggunakan sistem DBMS tertentu. Tujuannya adalah untuk mengimplementasikan database yang efektif.

# JENIS-JENIS DATA MODELLING

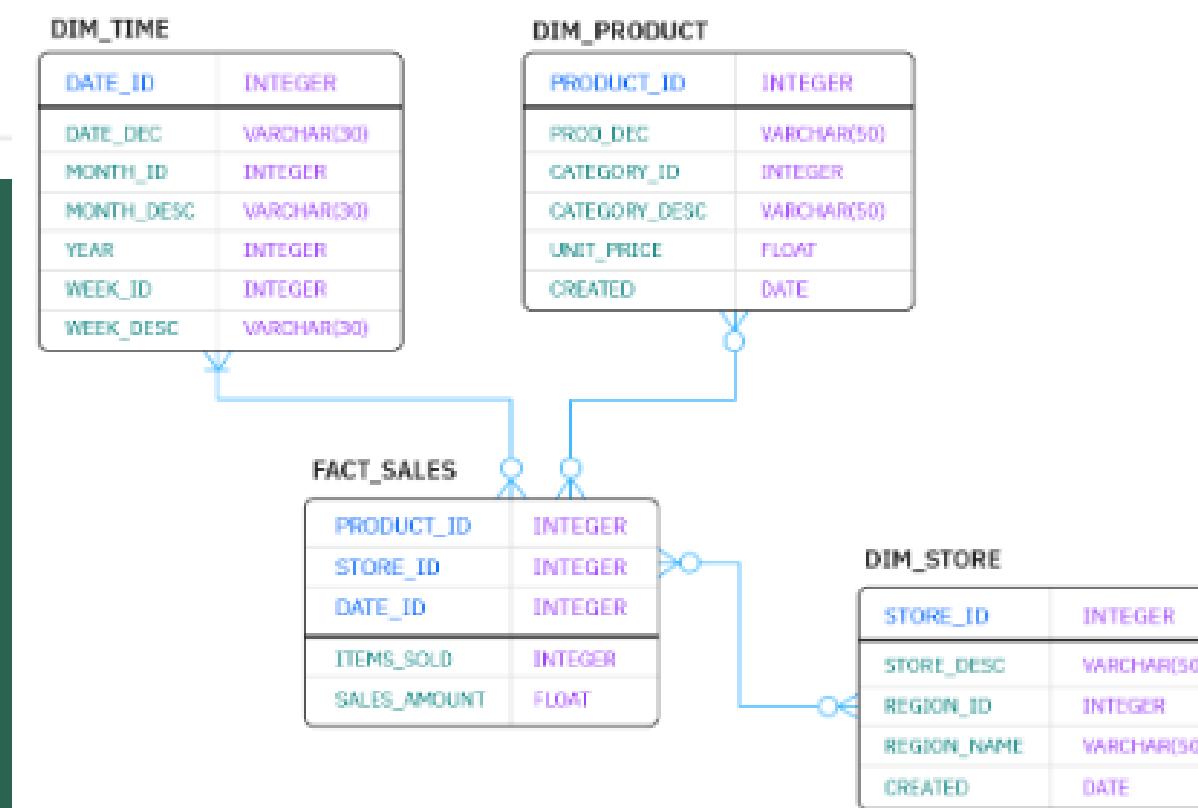
CONCEPTUAL DATA MODELING



LOGICAL DATA MODELING



PHYSICAL DATA MODELING



# Relational Model

## Definisi

Model Data Relasional adalah suatu model basis data yang menggunakan tabel dua dimensi, yang terdiri atas baris dan kolom untuk menggambarkan sebuah berkas data.

## Keuntungan Model Data Relasional

- Dapat mengakomodasi berbagai kebutuhan pengelolaan database yg ada di dunia nyata (real word).
- Merupakan model paling sederhana, dibandingkan dengan model jaringan/network atau model hirarki.
- Mudah melakukan berbagai operasi data (query, update/edit, delete).

# Terminologi Model Relasional

## Relasi

Relasi direpresentasikan sebagai tabel yang terdiri dari baris dan kolom.

## Tuple

Tupel adalah sebuah baris pada tabel.

## Derajat

Derajat (degree) adalah jumlah atribut yang ada didalam tabel relasi/ (banyaknya atribut/kolom pada tabel).

## Atribut

Atribut adalah kepala/header dari setiap kolom yang ada dalam tabel relasi.

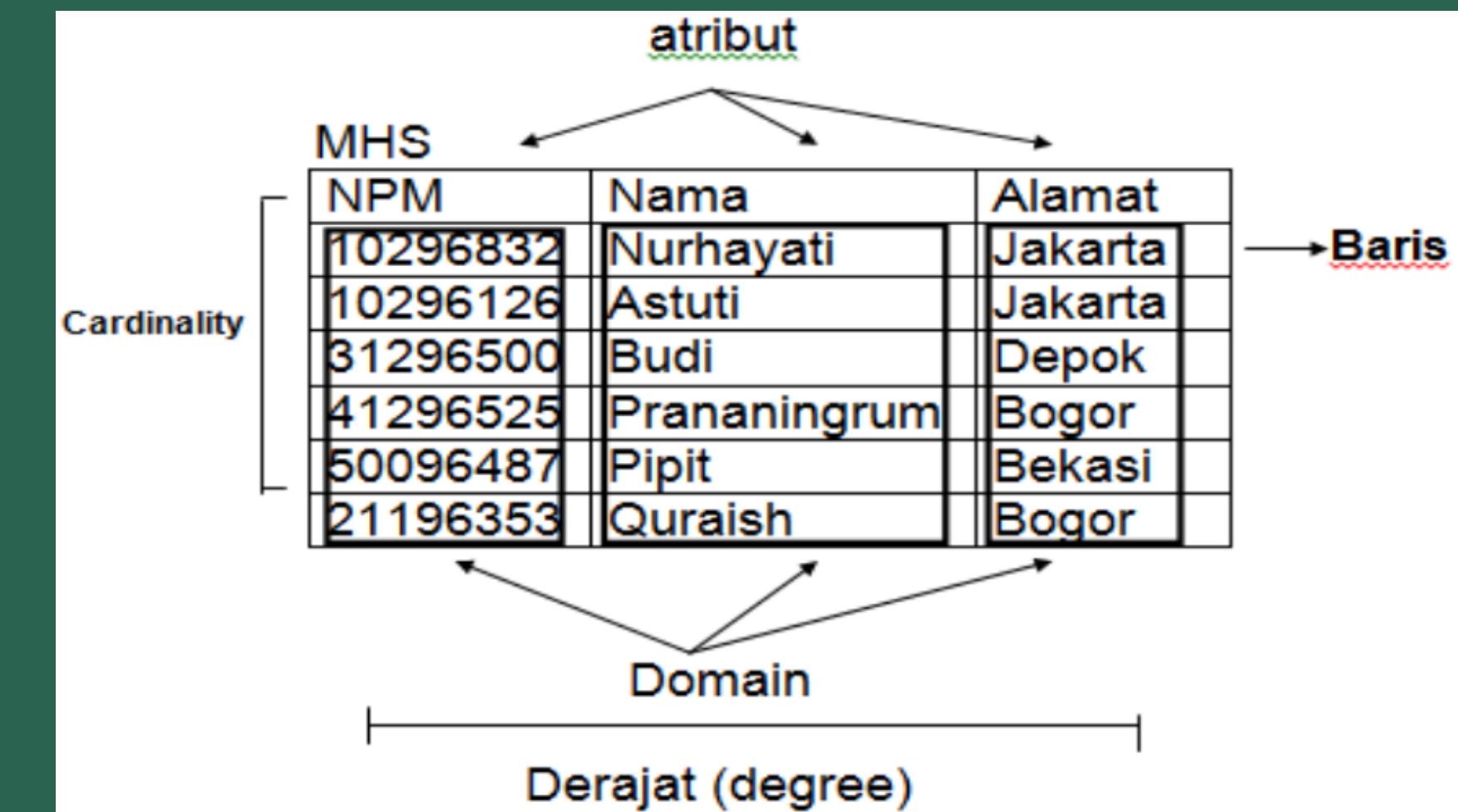
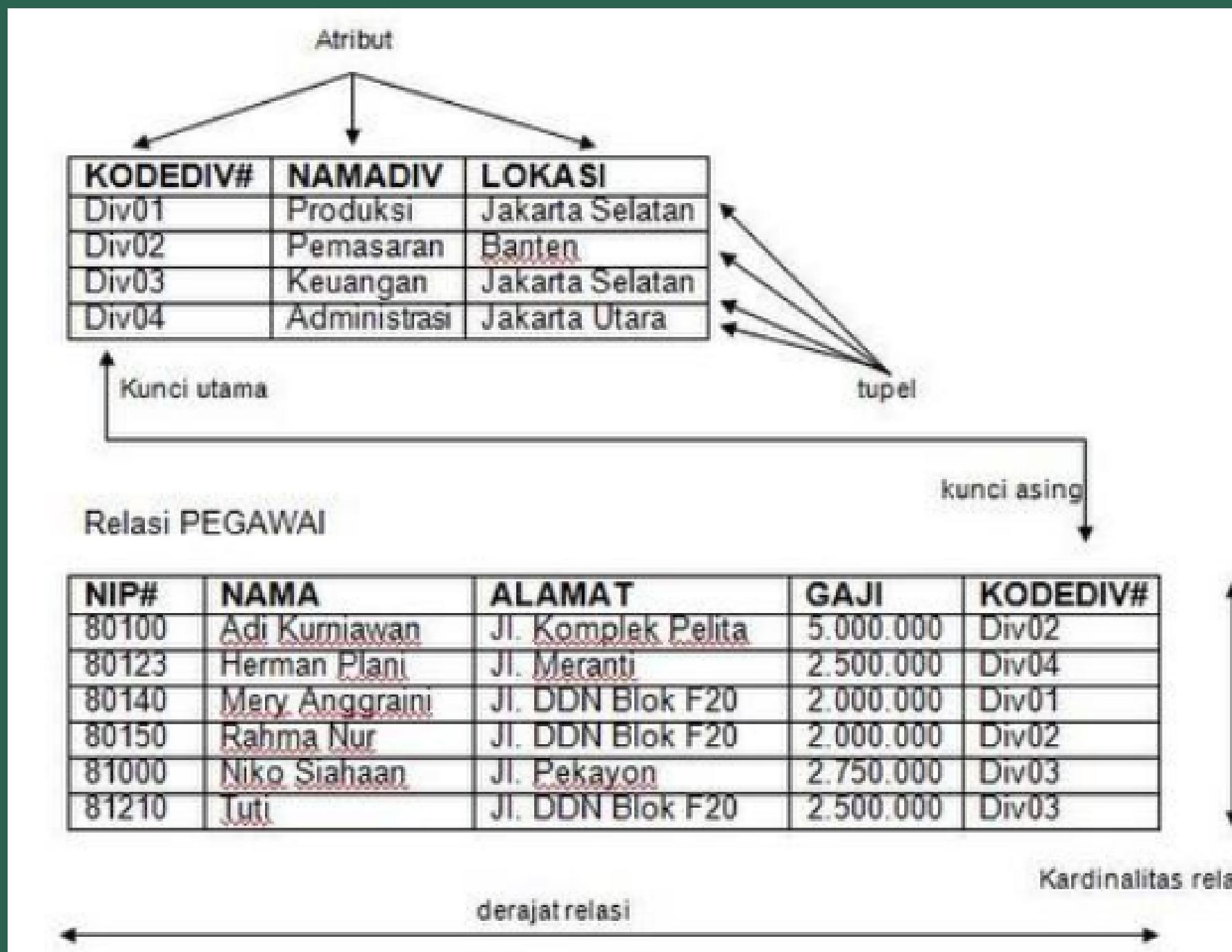
## Domain

Kumpulan nilai yang valid untuk satu atau lebih atribut Setiap domain memiliki deskripsi, tipe data dan format data tertentu.

## Kardinalitas

Kardinalitas (Cardinality) adalah jumlah tuple yang ada dalam tabel (jumlah record)/ (banyaknya tuple/baris pada tabel).

# Terminologi Model Relasional



# Karakteristik Relasi

Harus mempunyai nilai tunggal (single value) dan nilai yang tidak dapat dibagi lagi (atomic value)

Semua elemen data harus mempunyai tipe dan ukuran sama

Masing-masing atribut (kolom) dalam sebuah relasi mempunyai nama yang unik

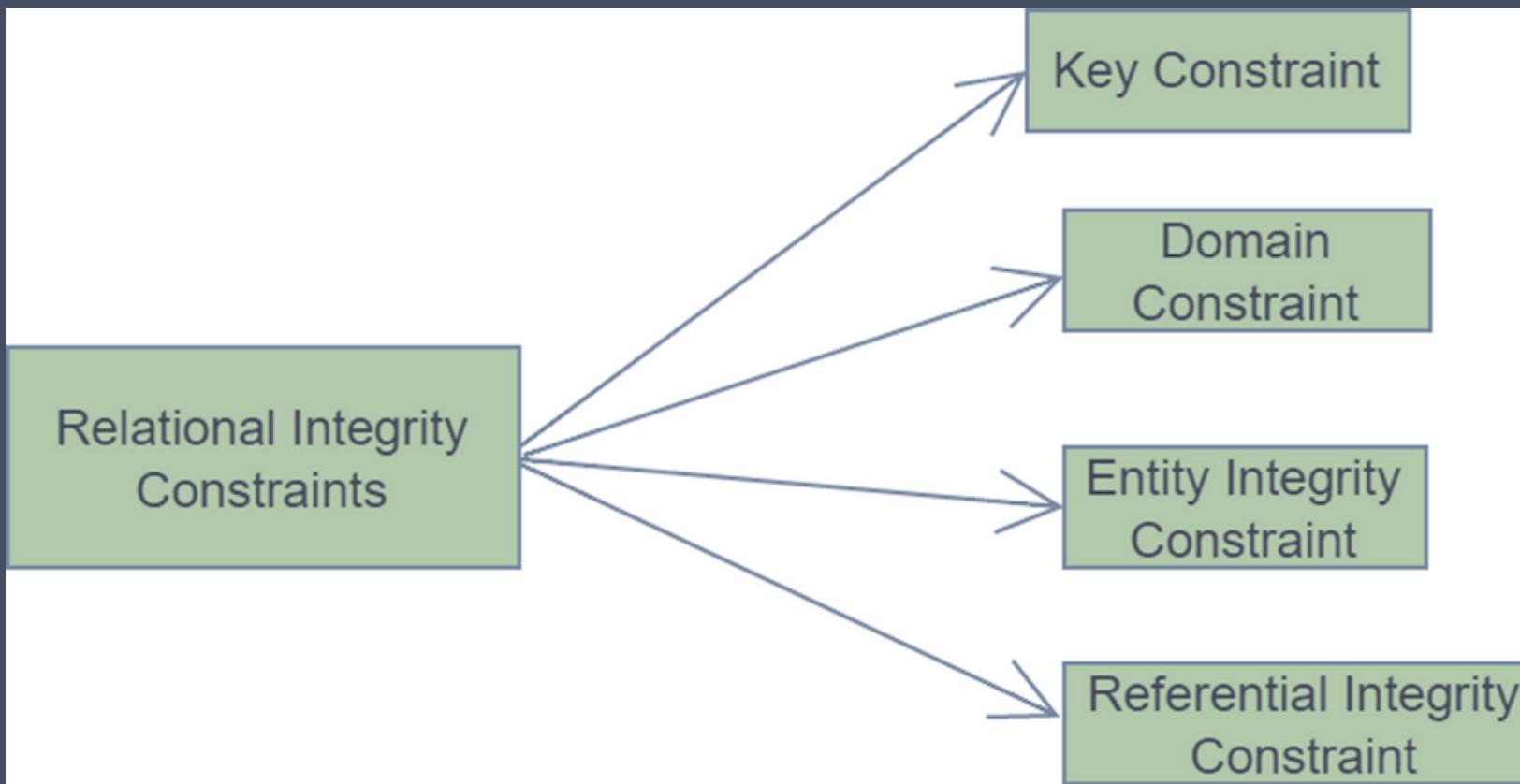
Dalam sebuah relasi tidak ada dua record data yang identik

Urutan atribut (kolom) bebas

Urutan tuple (baris) bebas

# Relational Integrity Constraint

Merupakan set rules yang dapat digunakan untuk menjaga integritas data selama operasi insert, delete, dan update, pada sebuah tabel



- Key Constraints
  - Memberi identitas pada setiap record sehingga bersifat unik dengan menggunakan primary key
- Domain Constraints
  - Atau attribute constraints, menggunakan tipe data sebagai batasan.
- Entity Integrity Constraint
  - Menjaga konsistensi nilai suatu primary key agar tidak bernilai null.
- Referential Integrity Constraint
  - Menjaga konsistensi data pada dua table yang saling berhubungan dengan foreign key

# Basic Operation in Relational Database

## Insert

Memasukan data kedalam table

## Truncate

Menghapus record data pada sebuah tabel

## Update

Mengubah atau memperbaharui data record dari pada kolom dan baris existing

## Select

Memilih, menampilkan, mengambil data spesifik dari suatu tabel

# Other Operation in Relational Database

GROUP BY : untuk mengelompokkan data dalam sebuah kolom yang ditunjuk

COUNT: menghitung jumlah baris sesuai kondisi yang diterima

MIN, MAX, AVG : mencari nilai maximal, minimal, dan rata-rata

GROUP\_CONCAT : mengembalikan nilai berupa string yang digabungkan dalam suatu grup data dengan separator yang diberikan

SUM : mengembalikan nilai dari jumlah total kolom dengan tipe data numerik (penjumlahan record numerik)

IF : digunakan untuk mengetes suatu kondisi

CASE : menampilkan nilai sesuai dengan syarat yang diberikan. Dalam penggunaanya, digunakan bersama SELECT

IN : menampilkan nilai record dari beberapa kriteria. Dapat digunakan untuk menyederhanakan operator OR

UNION: menggabungkan hasil dari dua atau lebih query SELECT

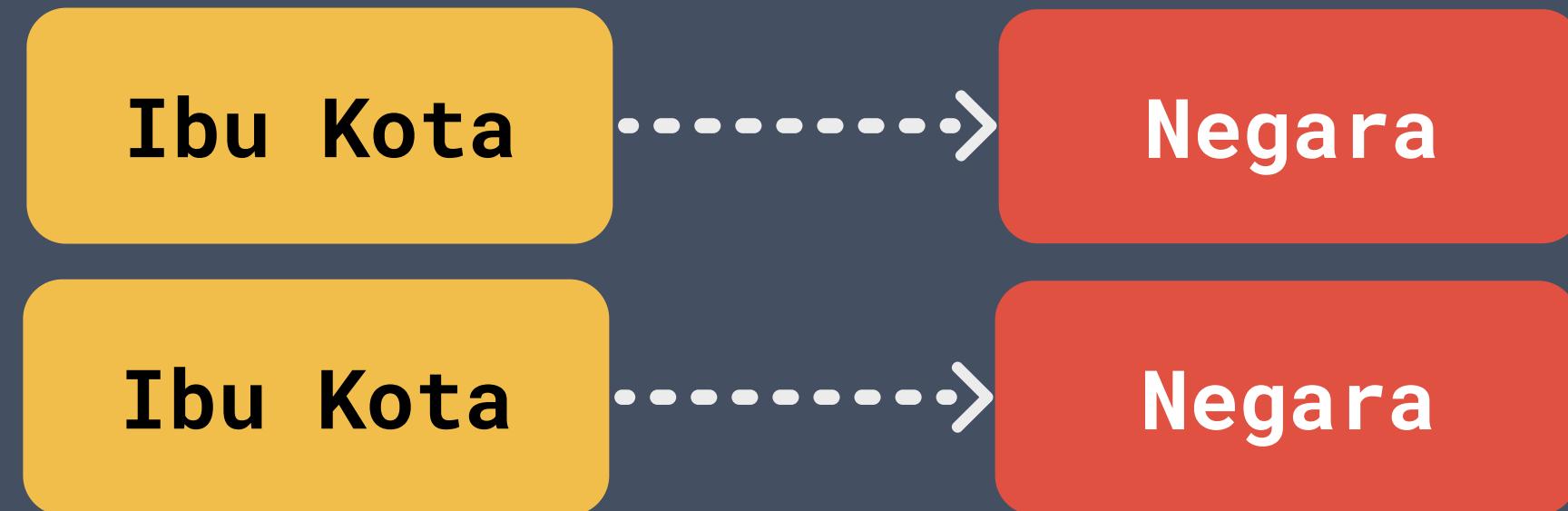
# Relasi One to One

Merupakan sebuah relasi dimana satu baris data pada tabel pertama hanya berhubungan dengan satu baris data pada tabel kedua

Ibu Kota		
id	nama	id_negara
0001	Jakarta	01
0002	Kuala Lumpur	02
0003	Bangkok	03
0004	Hanoi	04

→

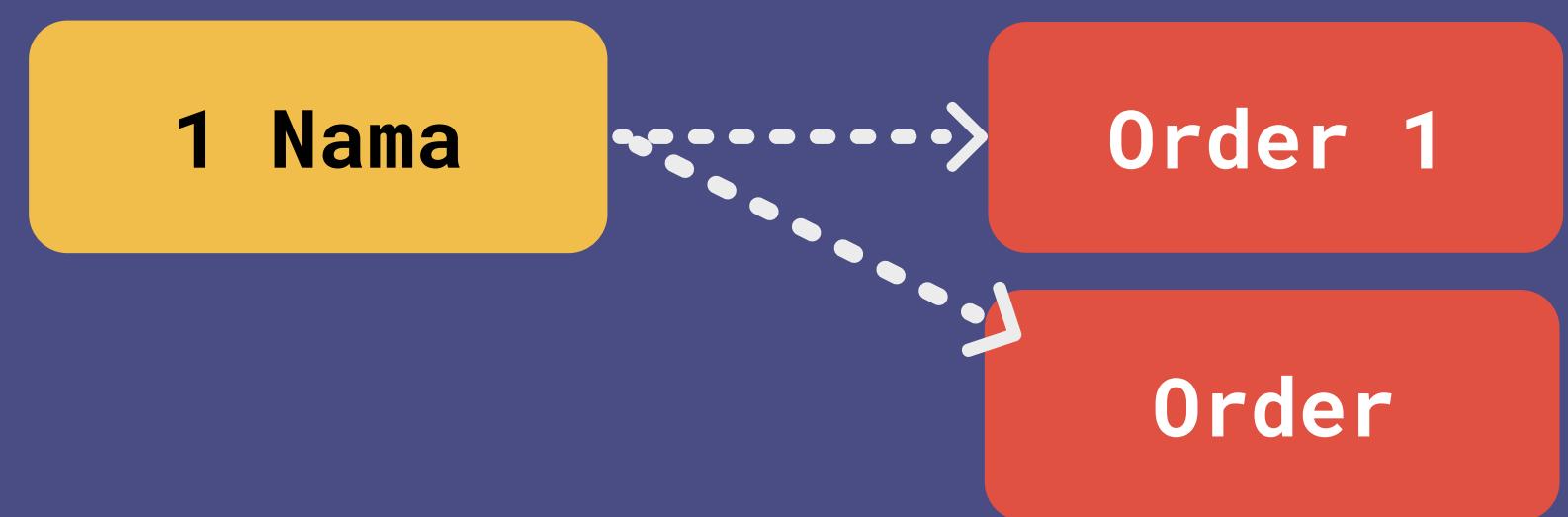
Negara	
id	nama
01	Indonesia
02	Malaysia
03	Thailand
04	Vietnam



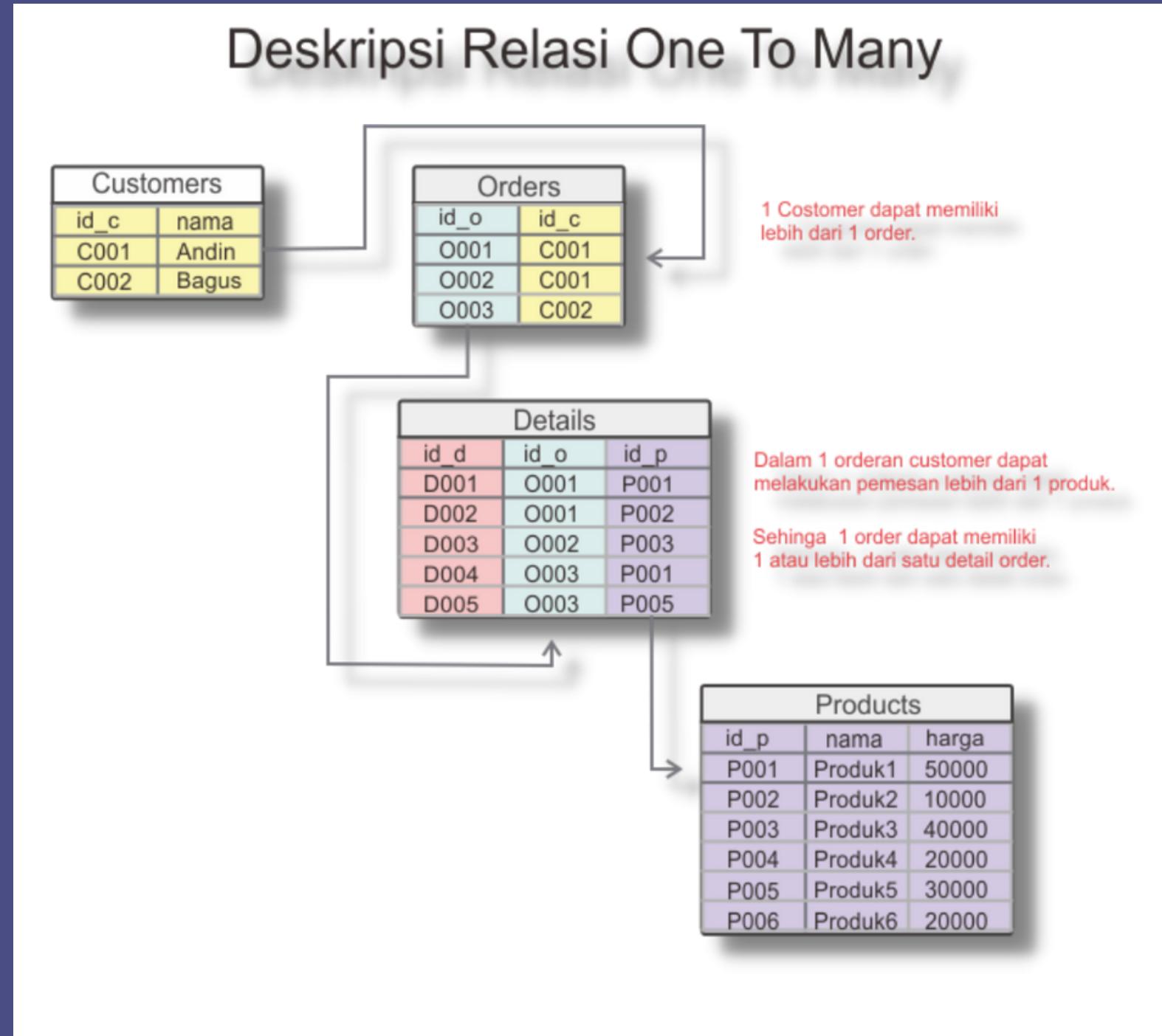
# Relasi One to Many

relasi yang berbentuk satu baris data di sebuah tabel dapat berhubungan dengan satu atau lebih baris data di tabel lain

## Contoh Relasi One To Many



# Relasi One to Many



Dalam proses order produk di Samping telah terjadi beberapa hubungan relasi yaitu one to many dan many to many. Relasi one to many terjadi pada tabel customers dan orders. Selanjutnya relasi kedua orders dan product adalah relasi many to many, tabel details disini menjadi pivot tabel atau tabel penghubung yang mengakibatkan terjadinya many to many realationship

# Query Join

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Sebuah query Join digunakan untuk menggabungkan rows lebih dari banyak tabel yang berhubungan dengan colom mereka

Contoh 2 Table ini dihubungkan melalui Colom CustomerID

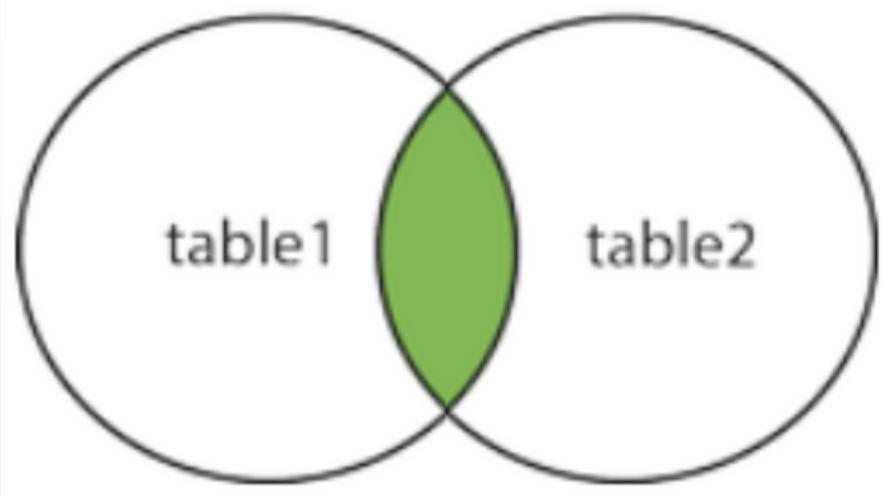
kita dapat menggabungkan dua tabel berikut dengan cara INNER JOIN

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

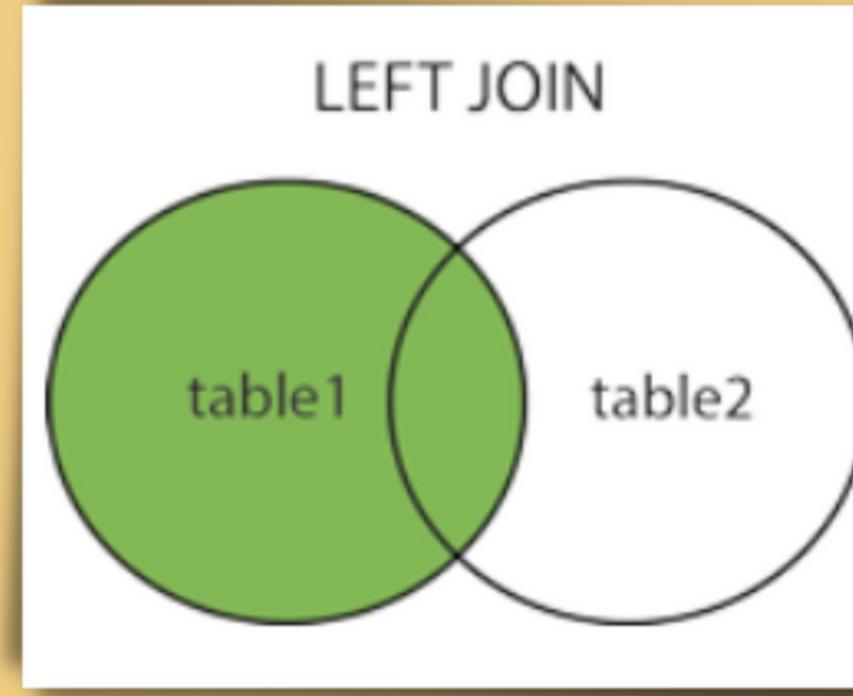
```
SELECT Orders.OrderID,  
Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON  
Orders.CustomerID=Customers.Cu  
stomerID;
```

# Different Type of SQL Join

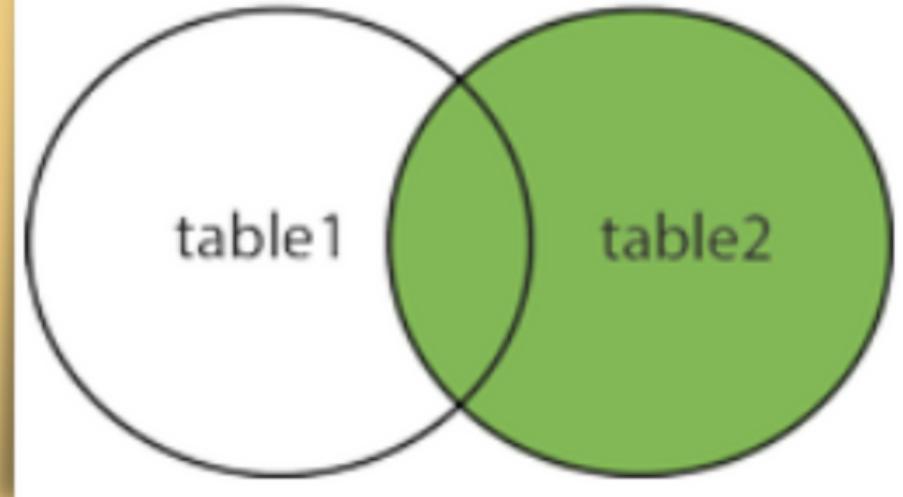
INNER JOIN



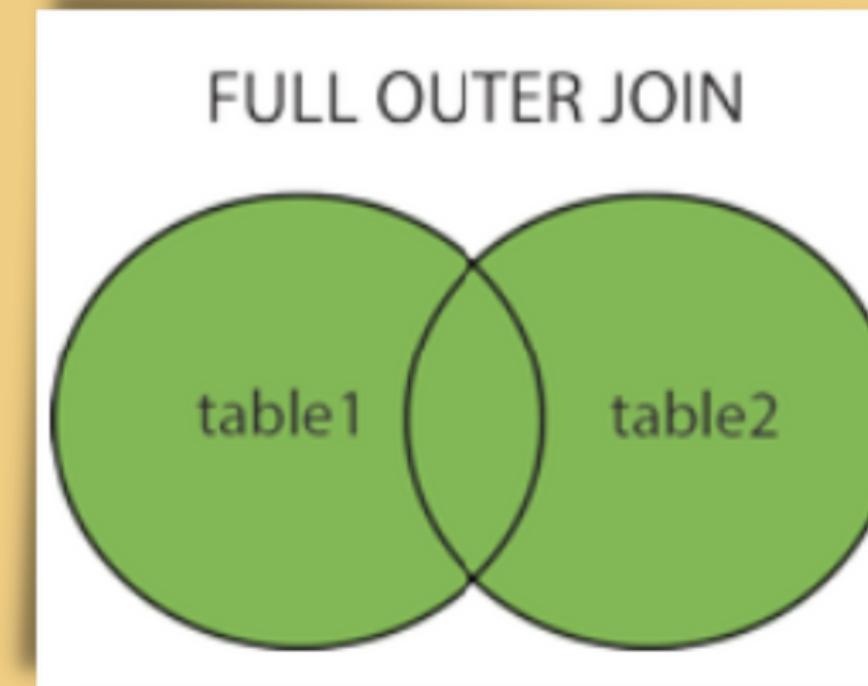
LEFT JOIN



RIGHT JOIN

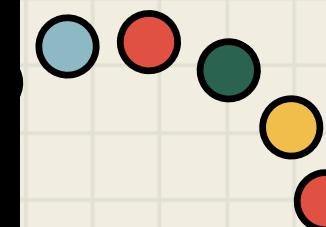


FULL OUTER JOIN



- (INNER) JOIN: Menggabungkan data yang memiliki nilai yang cocok di kedua tabel
- LEFT (OUTER) JOIN: Menghubungkan data dari tabel kiri, dan record yang cocok dari tabel kanan
- RIGHT (OUTER) JOIN: Menghubungkan data dari tabel kanan, dan record yang cocok dari tabel kiri
- FULL (OUTER) JOIN: Menghubungkan data ketika ada kecocokan di kedua kiri atau meja kanan

# TERIMA KASIH



Sampai jumpa di  
Learning Progress Review  
kami berikutnya!

Hilda Meiranita Prastika Dewi

Nur Indrasari

Rezha Sulvian

Thasha Dinya Ainsha