

# Pengantar Basis Data & Pemodelan Data

Nur Muhammad Abdul Falaq

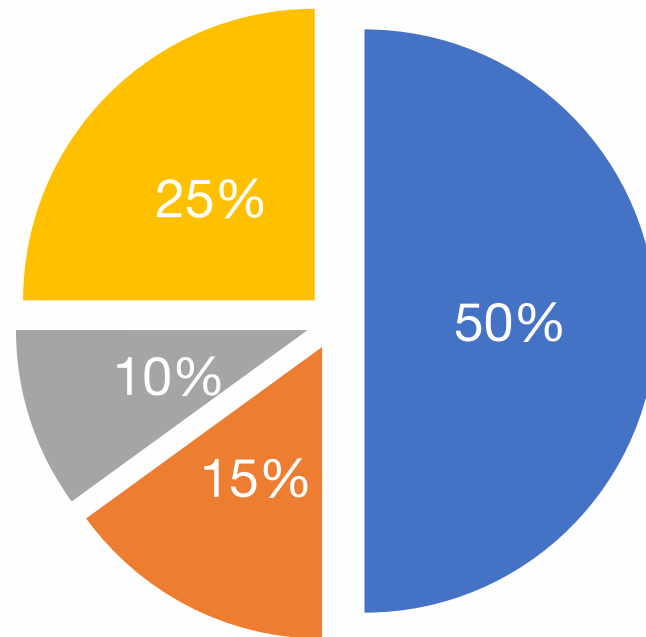


**PINTAR**



# Penilaian

**Komponen Penilaian**



■ Assignment

■ Presentasi

■ Keaktifan

■ Kehadiran

# Scope

- Basis Data

Referensi :

- <https://aws.amazon.com/id/what-is/database/>
- <https://www.youtube.com/watch?v=S4igMZFCvh8&pp=ygUKYmFzaXMgZGF0YQ%3D%3D>

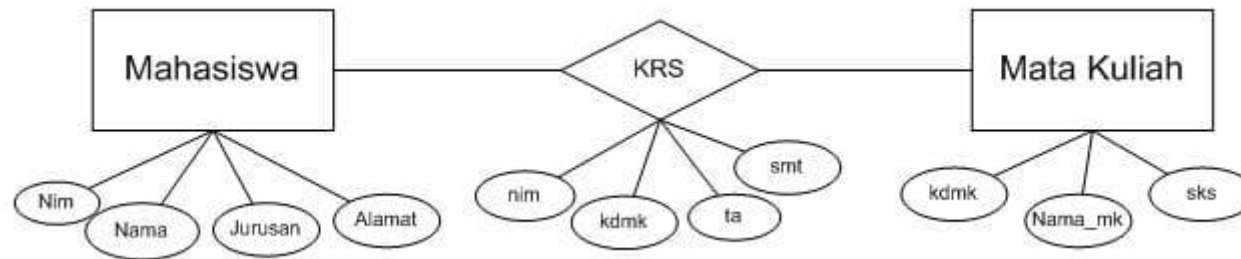


# PreTest – Basis Data

Dikerjakan hari ini – waktu menyesuaikan

Dikumpulkan melalui LMS dalam format **nama.pdf**

1. Menurut kamu, apa yang dimaksud dengan basis data?
2. Mengapa dibutuhkan basis data dan DBMS dalam suatu sistem data?
3. Pada gambar dibawah ini. sebutkan mana entitas, atribut, dan relasi



4. Menurut kamu, mengapa perlu ada keamanan data?

# Data...?

- Data adalah fakta-fakta yang dapat disimpan dan mempunyai arti tertentu. (Fakta yang akan disimpan dinyatakan dengan nilai baik angka, karakter, symbol, gambar, video, suara dan lain sebagainya).

## Data Terdiri Dari...

---

### Elemen Data

Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna.

---

### Rekaman (record)

Rekaman adalah gabungan sejumlah elemen data yang saling berhubungan.

---

### Berkas (file)

Berkas adalah kumpulan dari record yang bertipe sama.

# Informasi

- Informasi adalah sekumpulan data/fakta yang telah diolah dengan cara tertentu sehingga memiliki arti bagi penerimanya. Informasi akan memberikan keterangan/pengetahuan bagi penerimanya.

# Data Vs Informasi

---

## DATA

Material mentah/belum diproses

Belum memiliki arti dan belum bisa memberikan pengetahuan bagi penerimanya

## INFORMASI

Hasil dari penggabungan, perbandingan, analisis atau kalkulasi dari data

Memiliki arti atau fungsi bagi penerimanya serta dapat memberikan pengetahuan.



# Contoh

- Data:
  - Nama
  - NIM
  - Jurusan
  - IPK
- Informasi:
  - Jumlah mahasiswa  $IPK > 3,50$
  - Jumlah mahasiswa per jurusan

# Definisi Basis Data

Basis Data adalah koleksi atau kumpulan data yang mekanis, terbagi (shared), terdefinisi secara formal dan dikontrol pada organisasi.

- *Gordon C. Everest* –

Basis Data adalah koleksi “data operasional” yang tersimpan dan juga dipakai oleh system aplikasi dari suatu organisasi.

- *C.J. Date* –

Basis Data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras computer dan digunakan perangkat lunak untuk memanipulasinya.

- *John Burch dan Gary Grudnitski* -

# Tujuan Basis Data

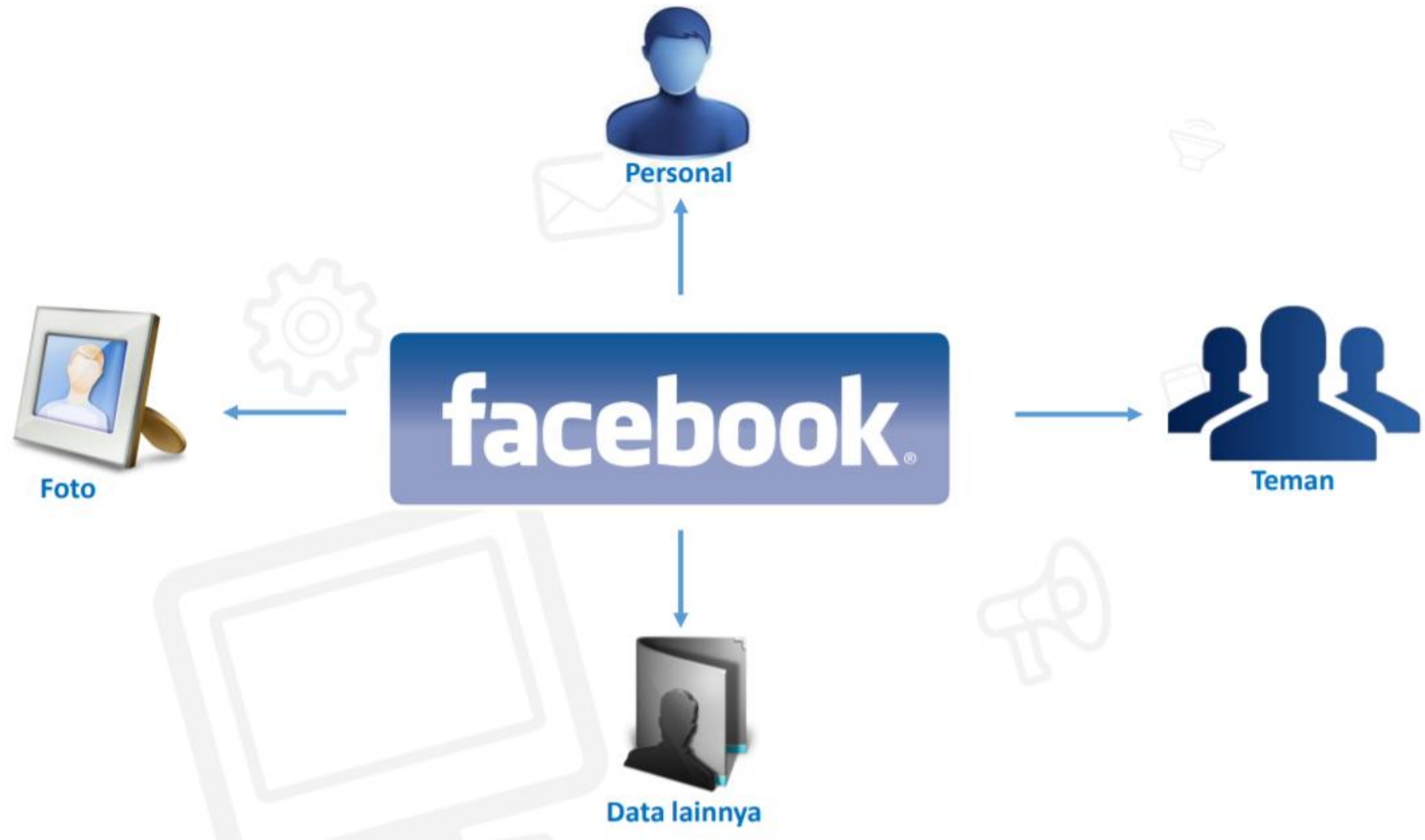
---

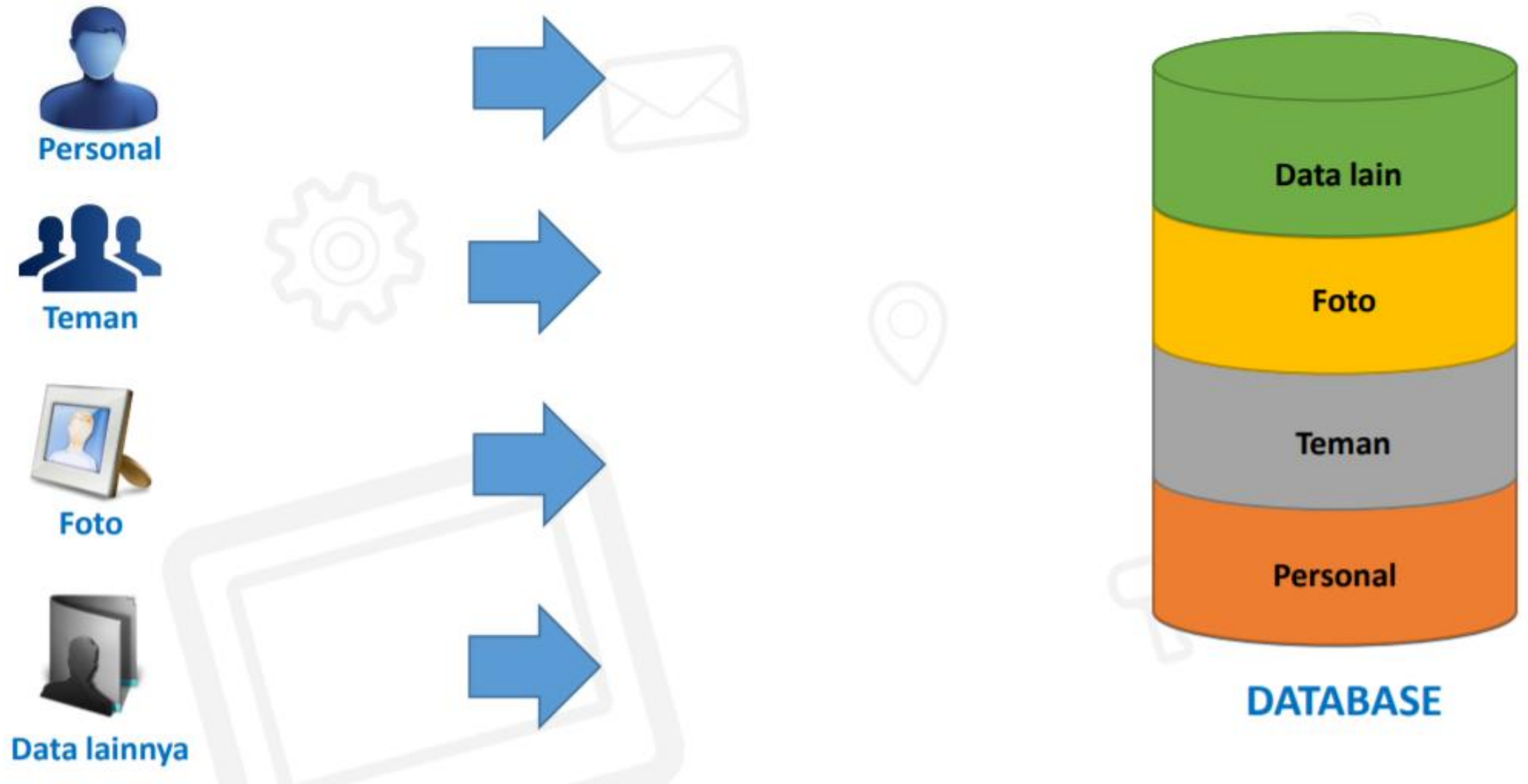
Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali.

---

Untuk mencapai tujuan tersebut, maka basis data harus dibuat dengan baik.

Contoh







What do you think?

Jika anda melihat Facebook tanpa adanya basis data, berapa banyak arsip yang digunakan untuk menyimpan data personal penggunaanya di seluruh dunia ?

# What Do You Think?

Model Penyimpanan Konvensional



# Peranan Basis Data

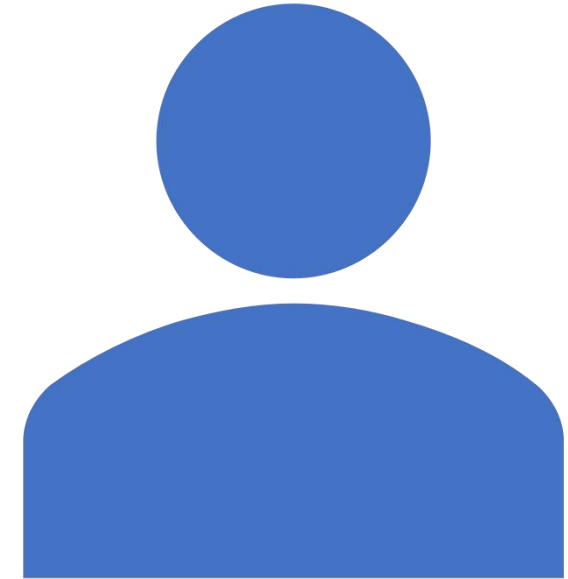
- Mungkin anda tidak menyadari bahwa website yang anda kunjungi sehari-hari dikelola oleh sebuah database.





# Pengguna Database

- Database Administrator
- Security Officer
- Application Developer
- Application Administrator
- Database User
- Network Admin



# Kerugian Basis Data



Membutuhkan storage yang besar



Membutuhkan tenaga yang terampil



Perangkat lunak yang mahal



Kerusakan dapat mempengaruhi department lain

# Syarat Basis Data yang Baik



Tidak ada redundansi  
dan inkonsistensi data



Tidak ada kesulitan  
dalam pengaksesan data



Multiple user

# Redundansi dan Inkonsistensi Data

Tabel 1.1 Mahasiswa

NIM	Nama	Alamat
101	Agung	Klaten
102	Ratna	Bantul

Tabel 1.2 KHS

NIM	Nama	MataKuliah	Nilai
102	Ratna	Basis Data	A
101	Agung	Struktur Data	C
102	Endang	Struktur Data	B
102	Endang	Agama	A

# Kemudahan Pengaksesan Data

Dilengkapi  
dengan fasilitas  
Query pada  
DBMS.

Dapat  
dihubungkan  
dengan program  
lain.

## Multiple User

---

Basis Data memungkinkan penggunaan secara bersama-sama oleh banyak pengguna (user) dalam waktu yang bersamaan.

---

Batasan tiap user dapat diaktur hak aksesnya.

# Pemodelan Data

# Model Data

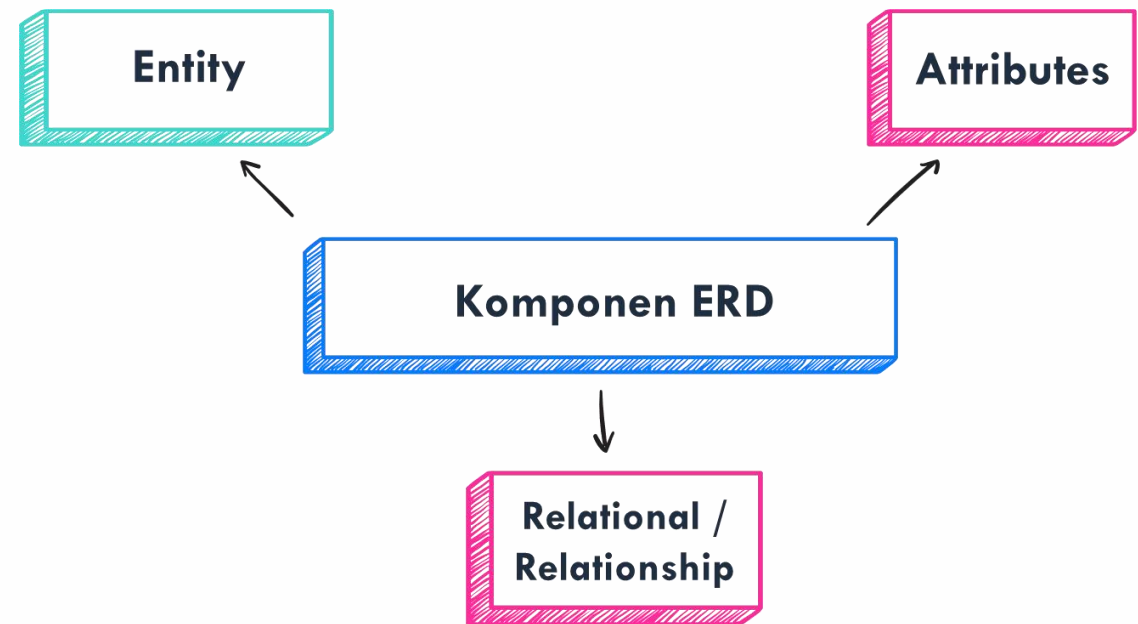
- Sekumpulan konsep-konsep untuk menerangkan data, hubungan-hubungan antara data dan batasan-batasan data yang terintegrasi di dalam suatu organisasi
- Model data berfokus pada data apa yang diperlukan dan bagaimana data tersebut harus diorganisasikan.
- Model data bertindak sebagai jembatan antara konsep yang menyusun dunia nyata dengan proses serta tampilan fisik dari konsep tersebut didalam suatu basis data
- Realisasi Model data berupa ambar perencanaan yang dengan menggunakan konsep ERD (*Entity Relationship Diagram*)





# *Entity Relationship Diagram*

- ERD (Entity Relationship Diagram) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.
- ERD juga merupakan Teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analysts dalam tahapan analisis persyaratan pengembangan system.



# Definisi Entitas

- Entitas adalah sebuah benda/objek yang mewakili sesuatu di dunia nyata yang dapat dibedakan dari objek lainnya (unique). Entitas memiliki atribut yang mendeskripsikan karakter dari objek tersebut.
- Entitas dapat berupa :
  - Makhluk hidup  
contoh : Mahasiswa, Pegawai, Dokter, Dosen, Pasien, Hewan, dsb.
  - Benda mati  
contoh : Mobil, Rumah, Handphone, Ruang Kelas, Laboratorium, dsb.
  - Abstrak / konsep  
contoh : Matakuliah, Pekerjaan, Departemen, Kategori, Jurusan, dsb.
  - Peristiwa  
contoh : Peminjaman, Penjualan, Pembelian, Pembayaran, dsb.

# Varian Entitas

## 1. Strong Entity (Entitas kuat)

Himpunan entitas yang tidak memiliki ketergantungan dengan entitas lain.

Contoh:

- Universitas (Mahasiswa, dosen, pegawai, matakuliah)
- Toko/Swalayan (Pegawai, pelanggan, barang)

Entity Kuat

## 2. Weak Entity (Entitas lemah)

Himpunan entitas yang keberadaannya ketergantungan dengan entitas lain. Himpunan entitas ini tidak memiliki atribut yang berfungsi sebagai kunci yang benar-benar menjamin keunikan entitas.

Contoh

- Universitas (Jadwal perkuliahan, KRS mahasiswa)
- Toko/Swalayan (Struk/catatan pembelian, laporan persediaan)

Entity Lemah

## Studi Kasus : **SISTEM INFORMASI SWALAYAN**

Swalayan ABC ingin menerapkan sebuah system informasi untuk mencatat proses bisnis yang dilakukannya. Selama ini, semua kegiatan jual beli dicatat secara manual dan sering kali data tidak cocok dengan transaksi yang dilakukan.

Pegawai swalayan terkadang salah dalam melakukan pemesanan barang ke supplier karena kesalahan dalam menghitung stok. Kasir seringkali mengalami kesulitan dalam mencari data member swalayan karena tidak tercatat dengan baik, sehingga ada beberapa member swalayan yang tidak mendapatkan fasilitas khusus. Penentuan harga barangpun terkadang terjadi kesalahan karena harga barang dituliskan secara manual oleh pegawai pada masing-masing barang.

Pihak manajer/owner swalayan kesulitan dalam melakukan control dalam aktifitas jual beli karena harus mengecek satu per satu transaksi jual beli yang dilakukan oleh pegawai dan kasir agar data yang diperoleh akurat. Owner juga kurang dapat mengetahui secara pasti data pegawai, data member, data supplier dan data barang. Pembeli kurang mendapat informasi harga yang jelas saat membeli barang karena nota pembelian ditulis secara manual

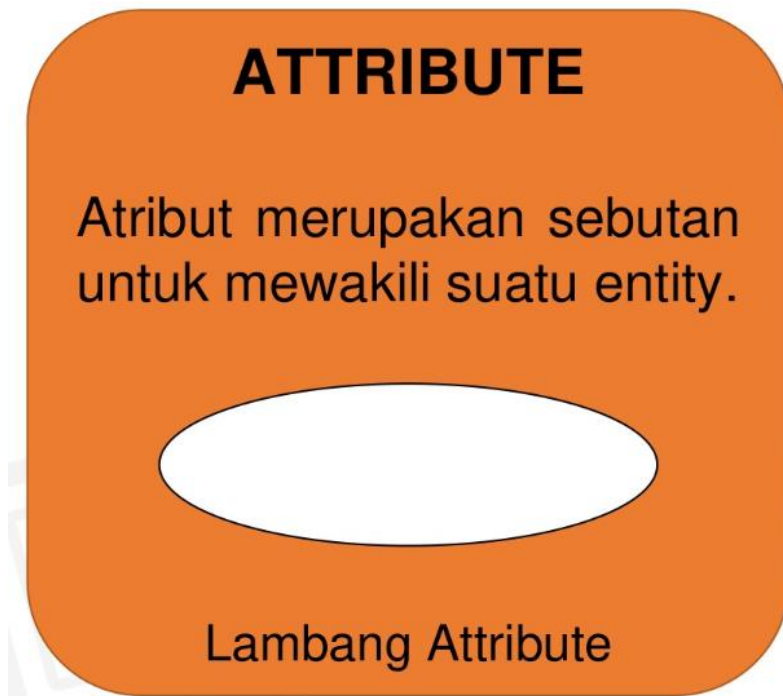
# TASK...!!!

- Tentukan entitas-entitas yang ada pada Sistem Informasi Swalayan ABC sesuai dengan studi kasus yang diberikan!

# Atribut

- Atribut merupakan karakteristik dari sebuah entitas (jika dalam sebuah tabel, berhubungan dengan field).
- Atribut merupakan karakteristik dari entity/relationship yang menyediakan detail tentang entity/relationship sehingga dapat dibedakan dengan yang lain.
- Atribut ditentukan dari relevansinya terhadap entitas tersebut.
- Contoh :
  - Entity : Mahasiswa Memiliki Atribut Nim, Nama, Alamat, Jurusan, Kelas, No\_hp
  - Entity : Ruang Memiliki Atribut Kode\_ruang, Nama\_ruang, Kapasitas
  - Entity : Matakuliah Memiliki Atribut Kodemk, Namamk, Sks, Jeniskuliah

# Notasi / Simbol Atribut



VARIAN  
Atribut

Simple Attribute dan Composite Attribute

Single Valued Attribute dan Multi Valued Attribute

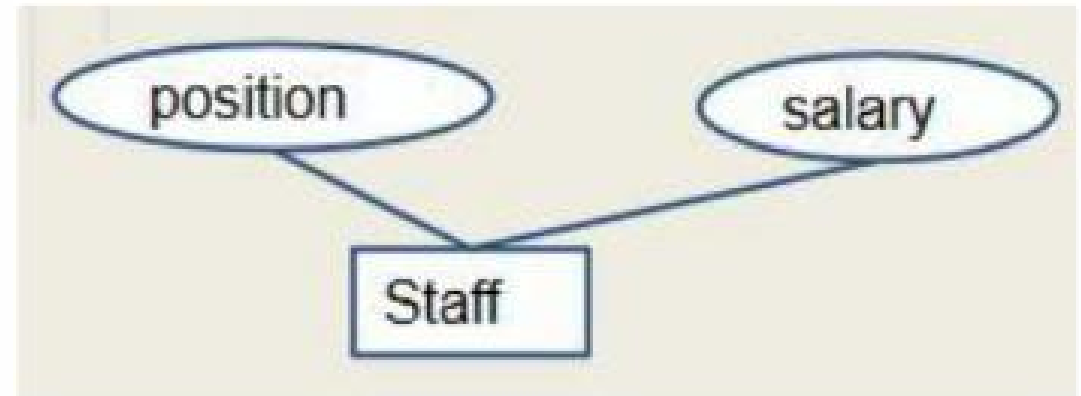
Mandatory Attribute

Derived Attribute (Atribut Turunan)

Key Attribute (Atribut Kunci)

# Simple Atribut

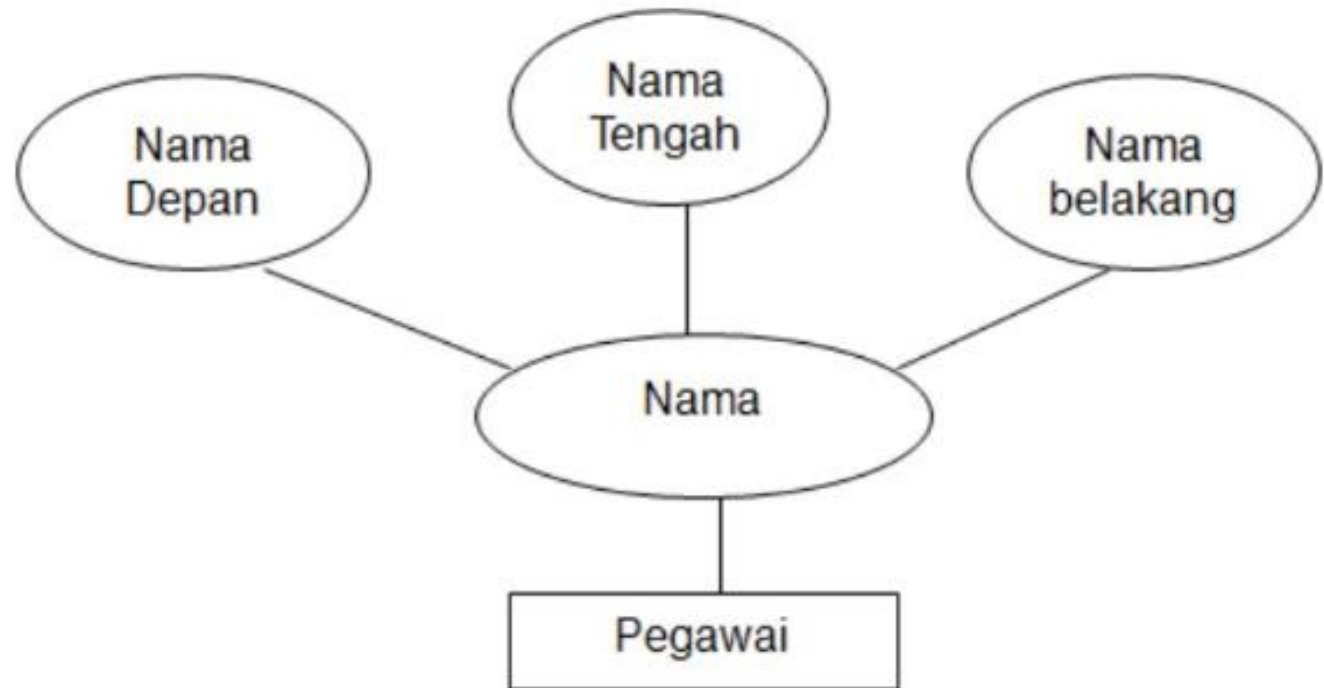
- Simple Attribute (Atribut Sederhana) adalah atribut yang tidak dapat dibagi-bagi menjadi atribut yang lebih mendasar.
- Contoh :
  - Atribut harga pada entitas barang
  - Atribut posisi pada entitas staf





# Composite Atribut

- Atribut yang terdiri dari beberapa Atribut yang lebih mendasar. Contoh attribute dari Entitas “pegawai” adalah “nama”.
- Atribut nama yang terdiri dari
  - nama depan (first name)
  - nama tengah (middle name)
  - nama belakang (last name).



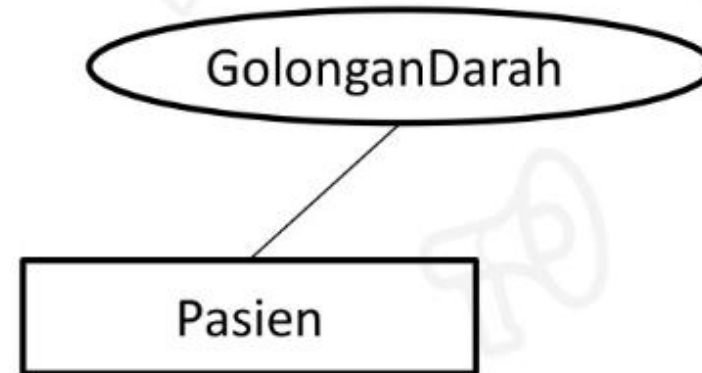
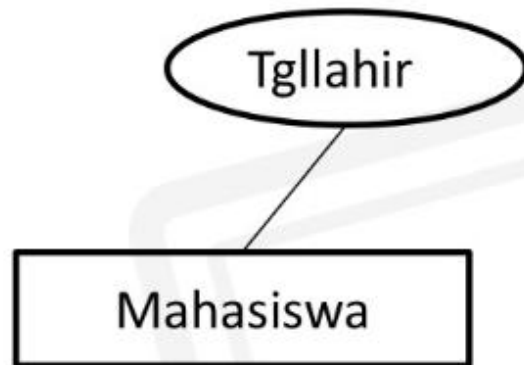
# Single Values Atribut

- Sebuah attribute yang memiliki nilai/value tunggal
- Contoh :
  - Attributes BranchNo mempunyai value B003,
  - Mahasiswa mempunyai hanya satu nim yaitu 13.11.1234



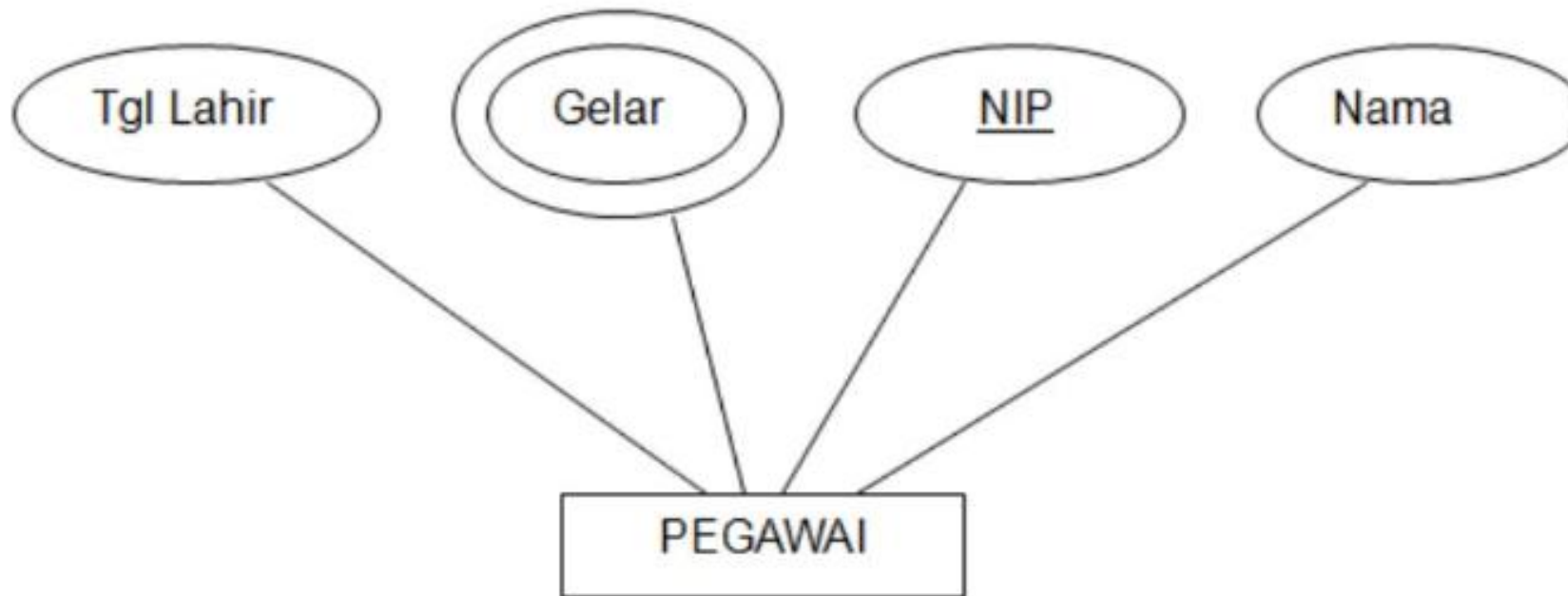
# Discussion

**Bagaimana dengan Atribut ini ....?**



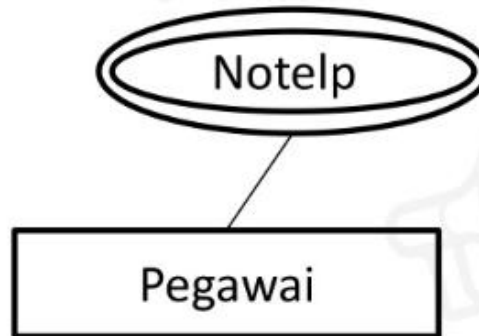
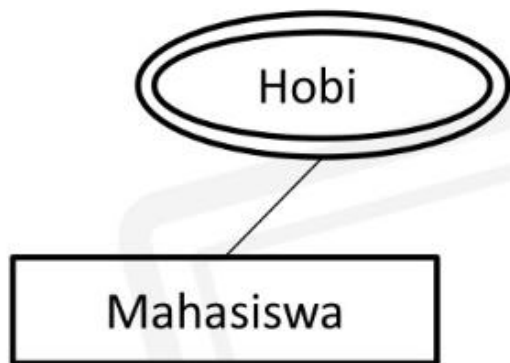
# Multivalues Atribut

- Attribute yang mempunyai nilai lebih dari satu
- Seorang pegawai bisa mempunyai gelar lebih dari satu



# Discussion

**Bagaimana dengan Atribut ini ....?**

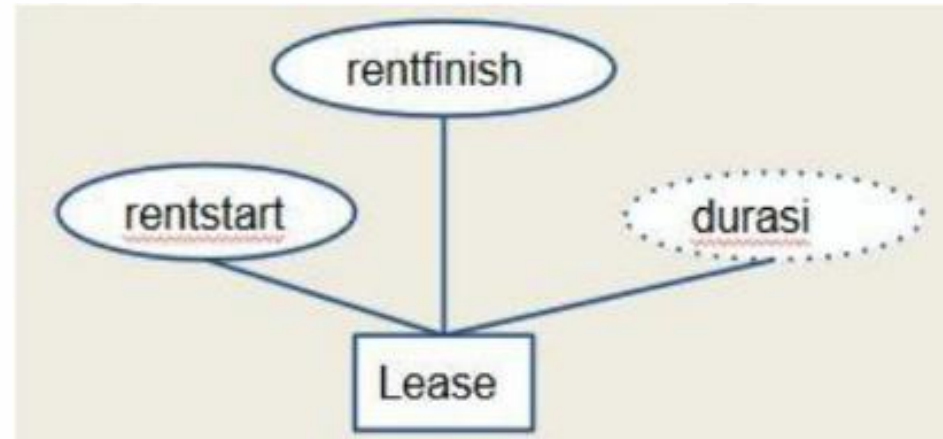
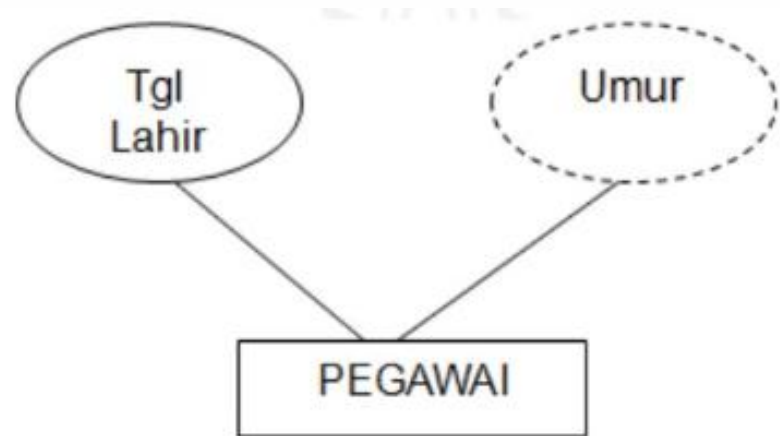


# Mandatory Atribut

- Atribut Mandatory merupakan Atribut yang harus diisi dan tidak boleh dikosongkan nilainya. Atribut mandatory biasanya merupakan Atribut yang menjadi kunci (key) dalam suatu entitas.
- Contoh :
  - Atribut nim untuk entitas mahasiswa
  - Atribut kode\_barang untuk entitas barang

# Derived Atribut

- Atribut yang dihasilkan dari atribut lain.
- Disebut juga attribute turunan.



# Key Atribut

ISTILAH  
LAIN YANG  
HARUS  
DIPAHAMI

SUPER KEY

KANDIDAT KEY

Kunci Relasi (*Relation key*)/Kunci Utama (*Primary Key*)

Kunci Alternatif (*Alternate Key*)

KOMPOSIT KEY

FOREIGN KEY



# SUPER KEY

- Super key merupakan satu atau lebih Atribut (kumpulan Atribut) yang dapat membedakan setiap baris data dalam table secara unik.

- Contoh :

NIM	NAMA	ALAMAT	JURUSAN
14.11.4561	Andhika	Yogyakarta	TI
14.61.0020	Rara	Jakarta	TI
14.02.2495	Yoga	Yogyakarta	MI
14.12.5723	Kartika	Solo	SI

- Dari table diatas yang mungkin menjadi super key adalah Atribut NIM, Atribut NIM dan NAMA, Atribut NIM dan ALAMAT, Atribut NIM dan JURUSAN.
- Atribut yang tidak dapat menjadi super key adalah ALAMAT, NAMA atau JURUSAN.

# KANDIDAT KEY

Kandidat key merupakan kumpulan Atribut minimal yang membedakan setiap baris data dalam table secara unik. Untuk menjadi kandidat key, suatu Atribut harus memenuhi syarat sebagai berikut :

1. Untuk satu nilai hanya mengidentifikasikan satu baris dalam satu relasi (unique)
2. Tidak memiliki subset yang juga merupakan kunci relasi
3. Tidak dapat bernilai null

Dengan kata lain, kandidat key adalah super key yang paling sedikit jumlah Atributnya. Dari contoh table mahasiswa sebelumnya, maka yang termasuk dalam kandidat key adalah Atribut NIM.

# Primary Key

Nilai dari kunci relasi harus mengidentifikasikan sebuah baris yang unik di dalam sebuah relasi. Kunci relasi dapat terdiri dari satu atau lebih Atribut relasi. Agar dapat menjadi kunci utama (Primary Key) maka sebuah Atribut harus memenuhi syarat sebagai kandidat key.

Seperti contoh table mahasiswa sebelumnya, maka yang dapat menjadi kunci utama (Primary Key) adalah Atribut **NIM**, sehingga dapat disimpulkan bahwa **NIM** adalah kunci relasi.

# Alternate Key

- Kunci alternatif merupakan kunci yang tidak ada dalam dunia nyata namun diadakan dan dijadikan sebagai Primary Key.
- Kunci alternatif dibuat ketika tidak ada satupun Atribut dalam sebuah relasi yang dapat mewakili relasi tersebut / ada yang dapat menjadi Kandidat Key tetapi tidak cukup efektif untuk menjadi Primary Key.

# Alternate Key

## Contoh: Tabel Hobi

NAMA	KETERANGAN
Membaca	Membaca buku, novel, majalah, surat kabar
Membuat program	
Makan	Suka kuliner dan mencoba makanan baru

Kandidat key untuk tabel Hobi adalah NAMA, namun agar lebih efektif

ditambahkan `KODE_HOBI`. `KODE_HOBI` sebenarnya tidak ada dalam

dunia nyata.

# KOMPOSIT KEY

- Komposit key merupakan primary key yang terdiri lebih dari 1 Atribut.
- Contoh : Tabel Nilai

NIS	NAMA	TAPEL	SEMESTER	MAPEL	NILAI
12.0086	Rudy	2014/2015	Genap	Matematika	89
12.0086	Rudy	2013/2014	Ganjil	Matematika	75
13.0167	Santi	2015/2016	Ganjil	IPA	90

Dari contoh table diatas ketika ingin mendapatkan informasi mengenai nilai Matematika dari Rudy maka perlu disebutkan Tahun Pelajaran (TAPEL) dan Semester selain NIS dan MAPEL. Hal tersebut yang dimaksud dengan Komposit Key.

# FOREIGN KEY

Foreign Key adalah sekumpulan Atribut dalam suatu relasi (misal A) sedemikian sehingga kumpulan Atribut ini bukan kunci relasi A namun merupakan kunci dari relasi lain.

- Contoh : Entitas Mahasiswa dan Entitas Jurusan

NIM	NAMA	KD_JUR
14.11.4561	Andhika	11
14.61.0020	Rara	61
14.02.2495	Yoga	02
14.12.5723	Kartika	12

KODE_JURUSAN	JURUSAN
11	S1 TI
61	S1 IF
02	S1 SI
12	D3 SI

- Maka ketika dua entitas tersebut direlasikan maka yang akan menjadi FOREIGN KEY adalah KD\_JUR

# TASK

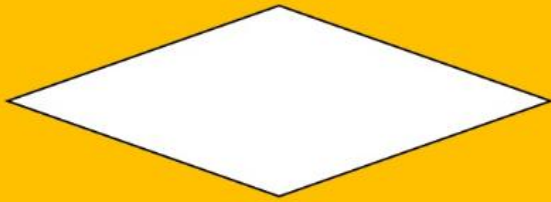
- Tentukan Atribut dari Entitas yang telah ditemukan pada tugas sebelumnya!
- Tentukan Atribut kunci (Primary Key) dari masing-masing entitas yang dapat dijadikan pembeda antara data yang satu dengan data yang lainnya!



# Relationship

## RELATIONSHIP

Relationship adalah hubungan diantara beberapa entity.



Lambang Relationship Set

Relasi merupakan hubungan diantara beberapa entitas. Dengan adanya relasi maka dapat menggambarkan hubungan antara entitas satu dengan entitas lainnya.

Macam-macam relasi :

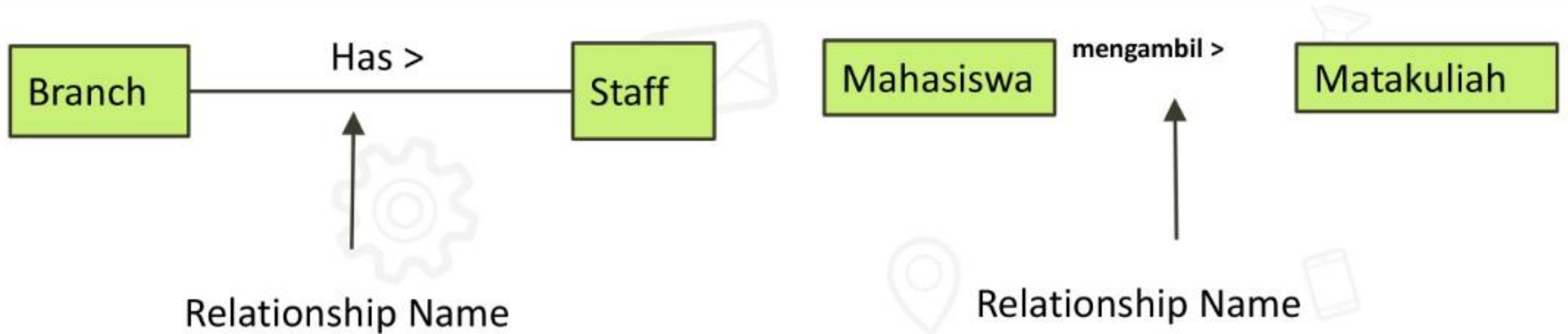
1. Relasi Ku



2. Relasi Lema



# Relationship

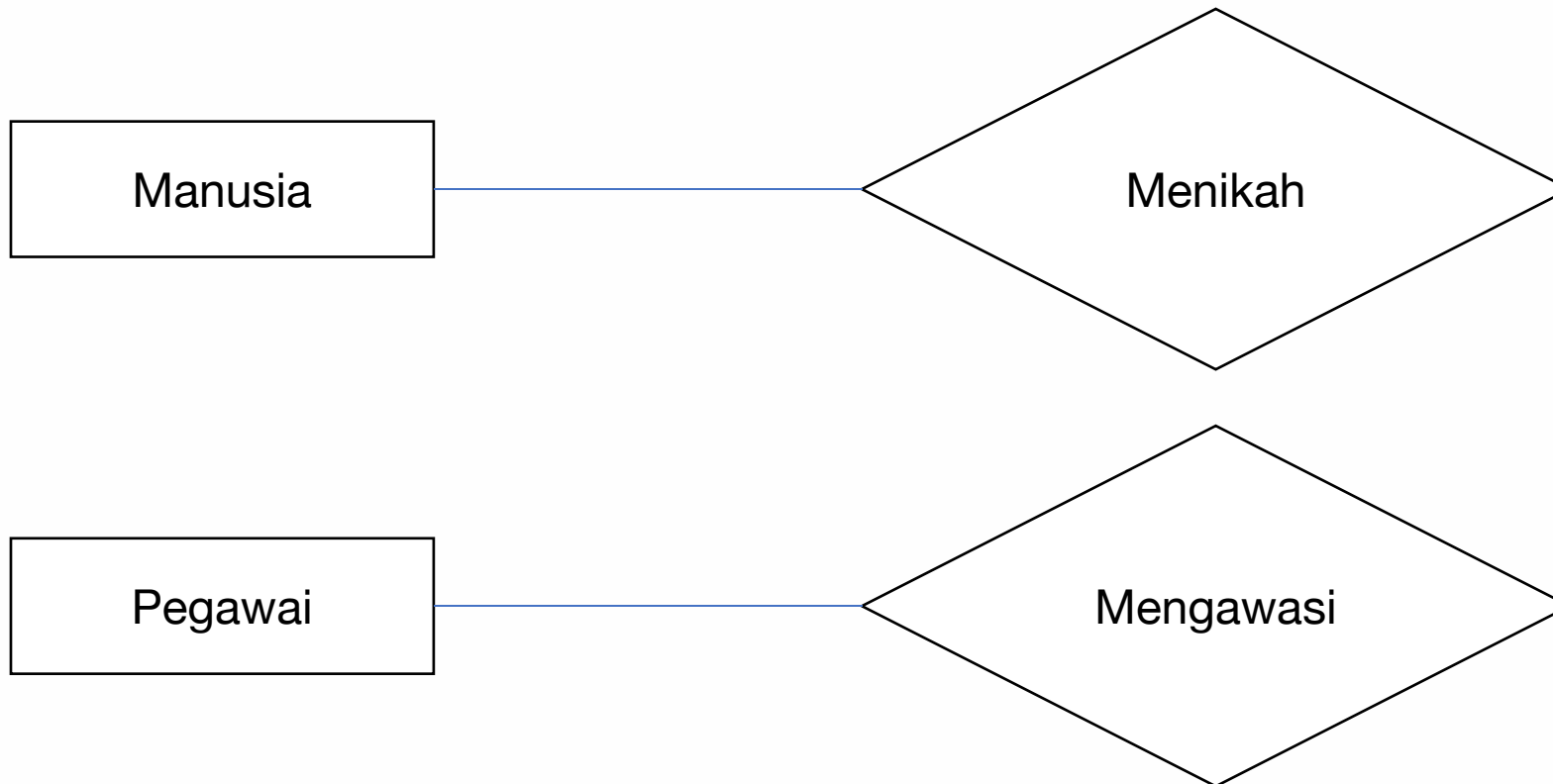


# Relationship Degree

- Jumlah entitas yang berpartisipasi dalam suatu relationship
- Relationship degree :
  1. Unary Degree (derajat satu)
  2. Binary Degree (derajat dua)
  3. Ternary Degree (derajat tiga)
  4. Quaternary Degree / Complex Relationship

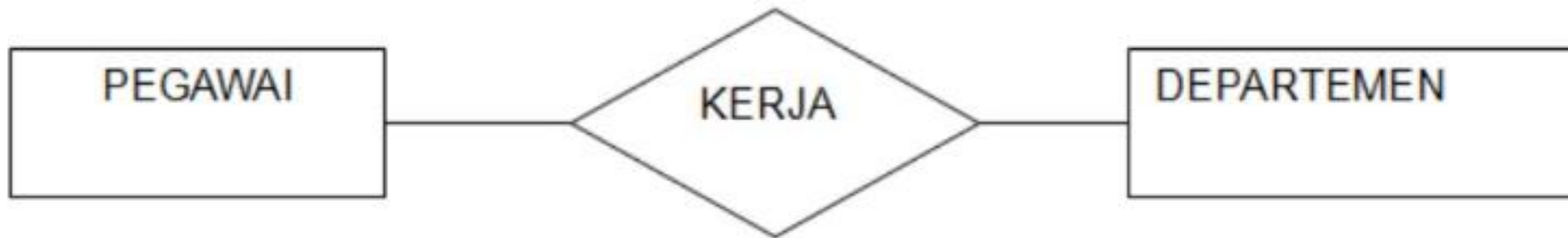
# Relationship Degree – Unary Degree

- Jika dalam sebuah relasi terdapat 1 buah entitas yang berpartisipasi



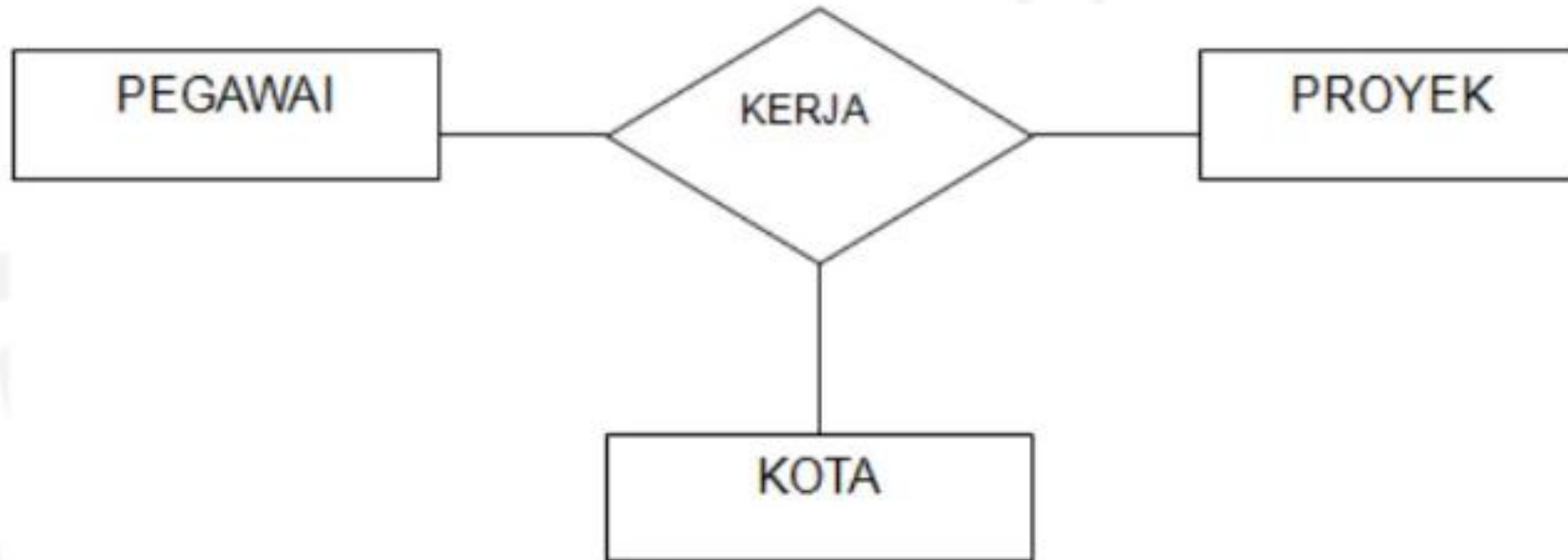
# Relationship Degree – Binary Degree

- Jika dalam sebuah relasi terdapat 2 buah entitas yang berpartisipasi



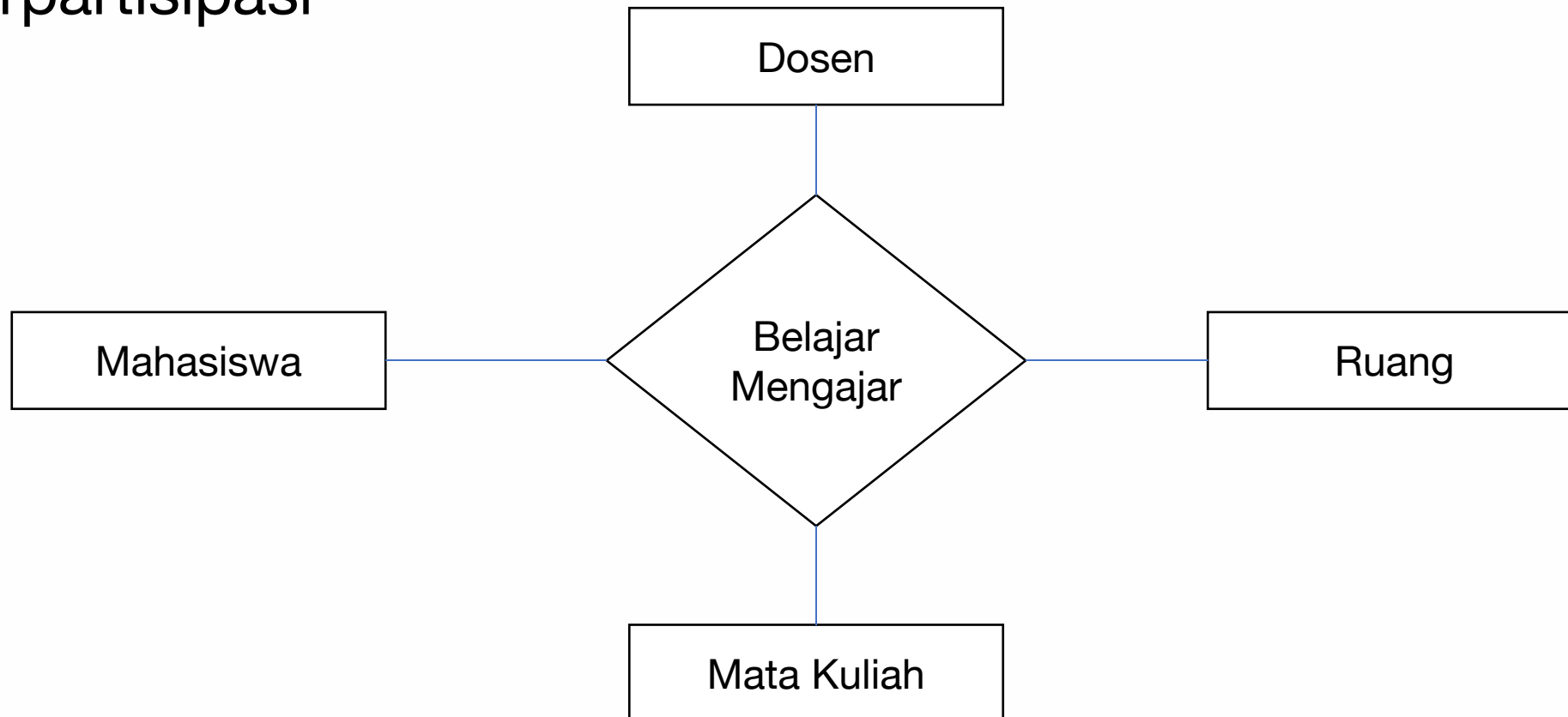
# Relationship Degree – Ternary Degree

- Jika dalam sebuah relasi terdapat 3 buah entitas yang berpartisipasi



# Relationship Degree - Quaternary Degree

- Jika dalam sebuah relasi terdapat 4 buah entitas yang berpartisipasi



# Cardinality

Kardinalitas relasi menunjukkan jumlah maksimum himpunan entity yang dapat berelasi dengan himpunan entity yang lain

- one - to - one (1:1)
- one - to - many (1:n)
- many - to - many (n:n)



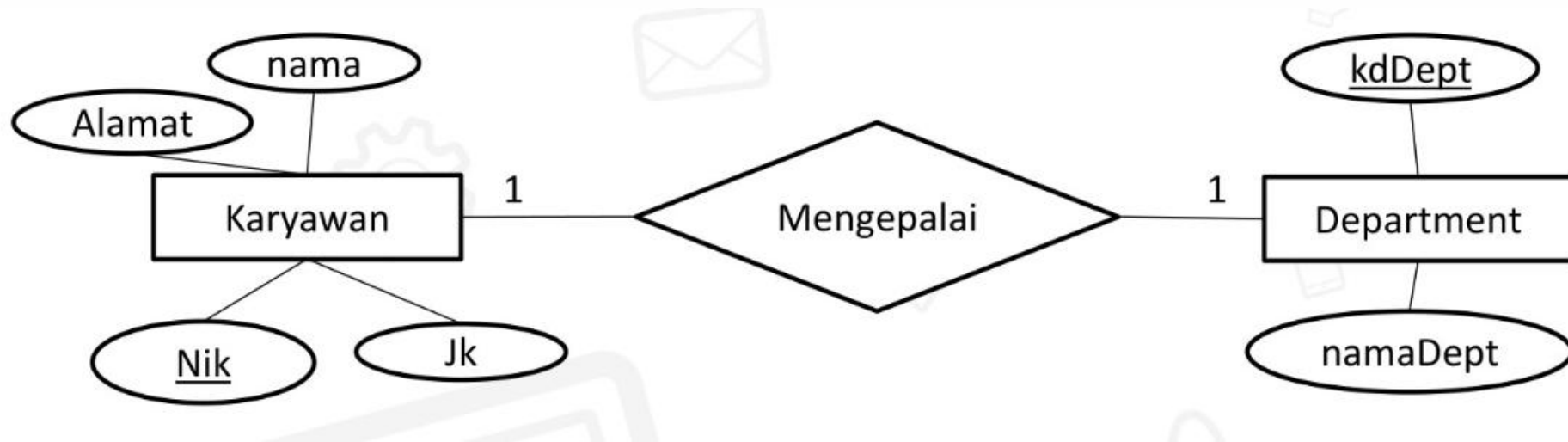
# One - to – one (1:1)

Setiap data value pada himpunan entitas A berhubungan paling banyak satu data value pada himpunan entitas B, begitu pula sebaliknya.

Contoh : A adalah Entitas Karyawan, B adalah entitas department, memiliki data value sebagai berikut :



# One - to - one (1:1)



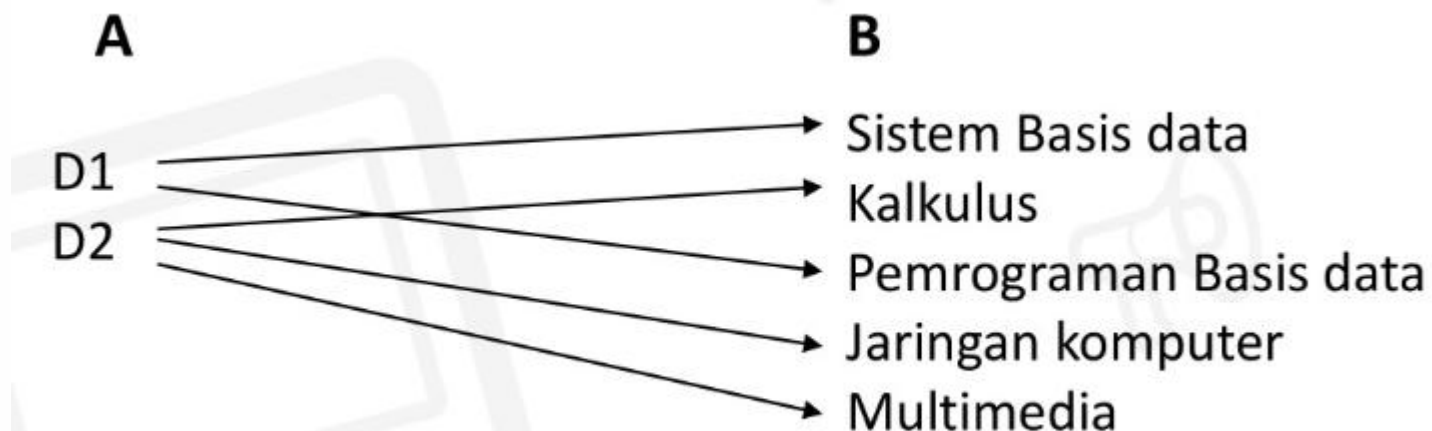
- Setiap karyawan paling banyak mengepalai 1 department.
- Setiap department dikepalai oleh paling banyak 1 karyawan.

# One – to – many (1:n)

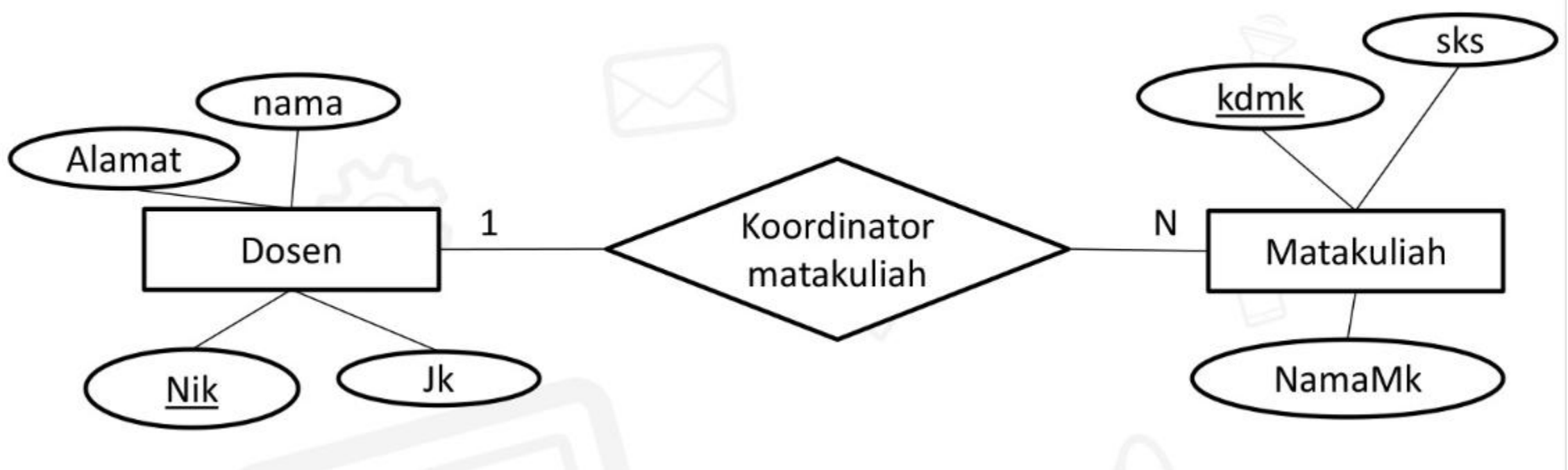
Setiap data value pada himpunan entitas A dapat berhubungan banyak ke data value pada himpunan entitas B.

Contoh :

A adalah Entitas dosen, B adalah entitas matakuliah



# One – to – many (1:n)



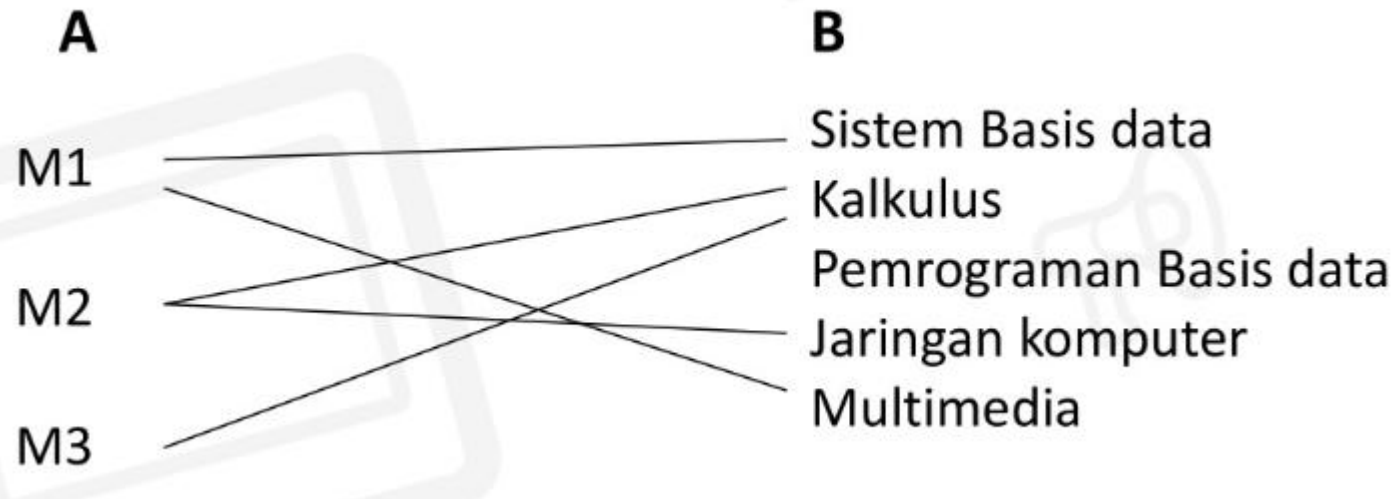
- Setiap dosen bisa menjadi koordinator beberapa matakuliah
- Setiap matakuliah hanya memiliki 1 dosen koordinator.

# Many-to-many (n:n)

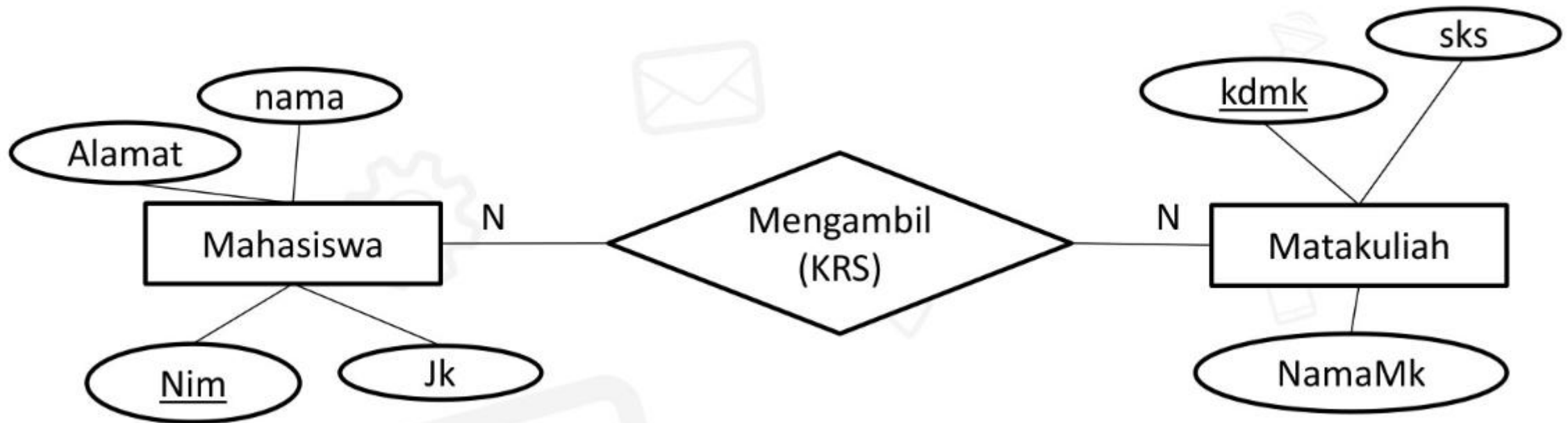
Setiap data value pada himpunan entitas A dapat berhubungan banyak ke data value pada himpunan entitas B begitu pula sebaliknya

Contoh :

A adalah Entitas Mahasiswa R adalah entitas matakuliah

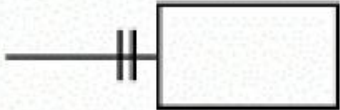
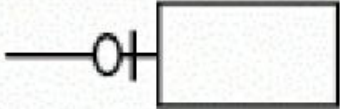
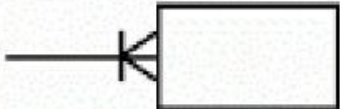
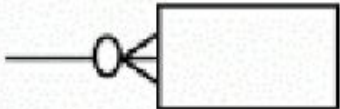
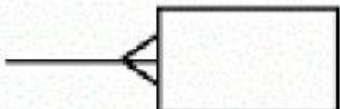


# Many – to – many (n:n)



- Setiap mahasiswa bisa mengambil beberapa matakuliah
- Setiap matakuliah diambil oleh banyak mahasiswa

# Notasi Lain dari Derajat Cardinalitas

Interpretasi	Min	Max	Notasi
Satu dan hanya satu ( <i>exactly one</i> )	1	1	
Nol atau satu ( <i>zero or one</i> )	0	1	
Satu atau lebih ( <i>one or more</i> )	1	>1	
Nol, satu atau lebih ( <i>zero, one, or more</i> )	0	>1	
Lebih dari satu ( <i>more than one</i> )	>1	>1	

# Assignment

- Tentukan relasi dari entitas pada task sebelumnya
- Buat diagram ERD-nya



Terima Kasih....