

SISTEM BASIS DATA

Pertemuan 2–Model Data

Nur Iksan

Pendidikan Teknik Informatika dan Komputer
UNNES

Model Data

- ▶ Model data merupakan suatu cara untuk menjelaskan tentang data yang tersimpan dalam basis data dan bagaimana hubungan antar data tersebut untuk para pemakai secara logik (edhy sutanta)
- ▶ Model data merupakan sekumpulan konsep untuk menerangkan data, hubungan–hubungan antara data dan batasan–batasan yang terintegrasi didalam suatu organisasi (linda marlinda)

Tujuan Model Data

- ▶ Tujuan model data adalah untuk menyajikan data agar mudah di modifikasi dan di mengerti

Jenis Model Data

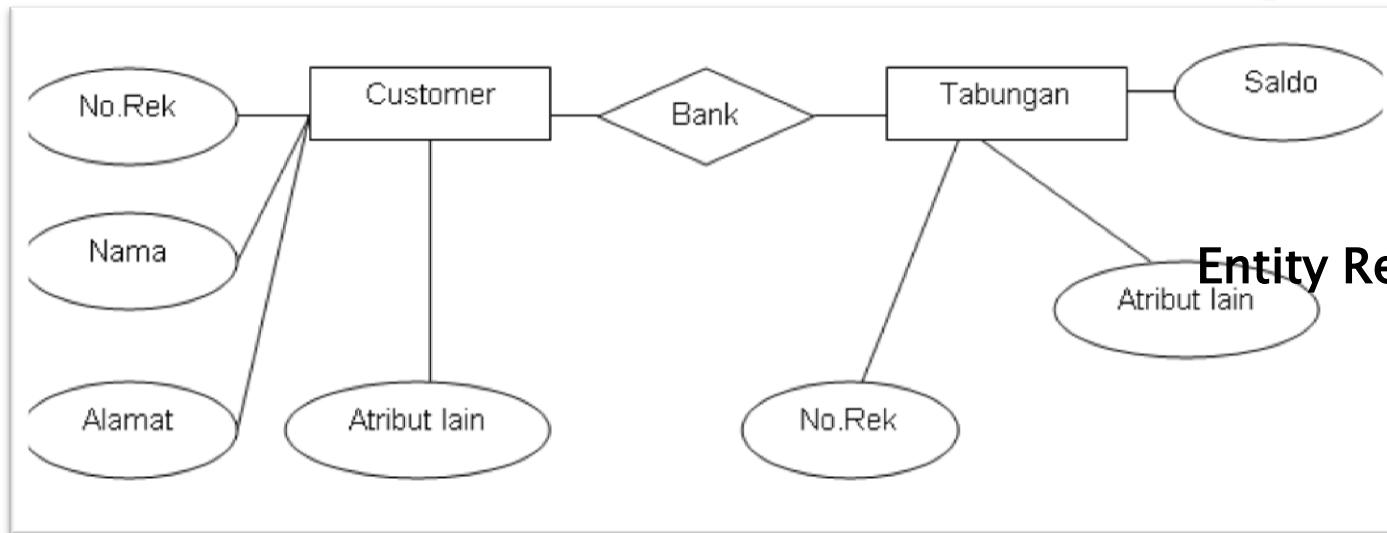
Secara garis besar model data dikelompokkan dalam tiga macam yaitu :

- ▶ Model data berbasis obyek (*object based data model*)
- ▶ Model data berbasis record (*record based data model*)
- ▶ Model data fisik (*physical based data model*)

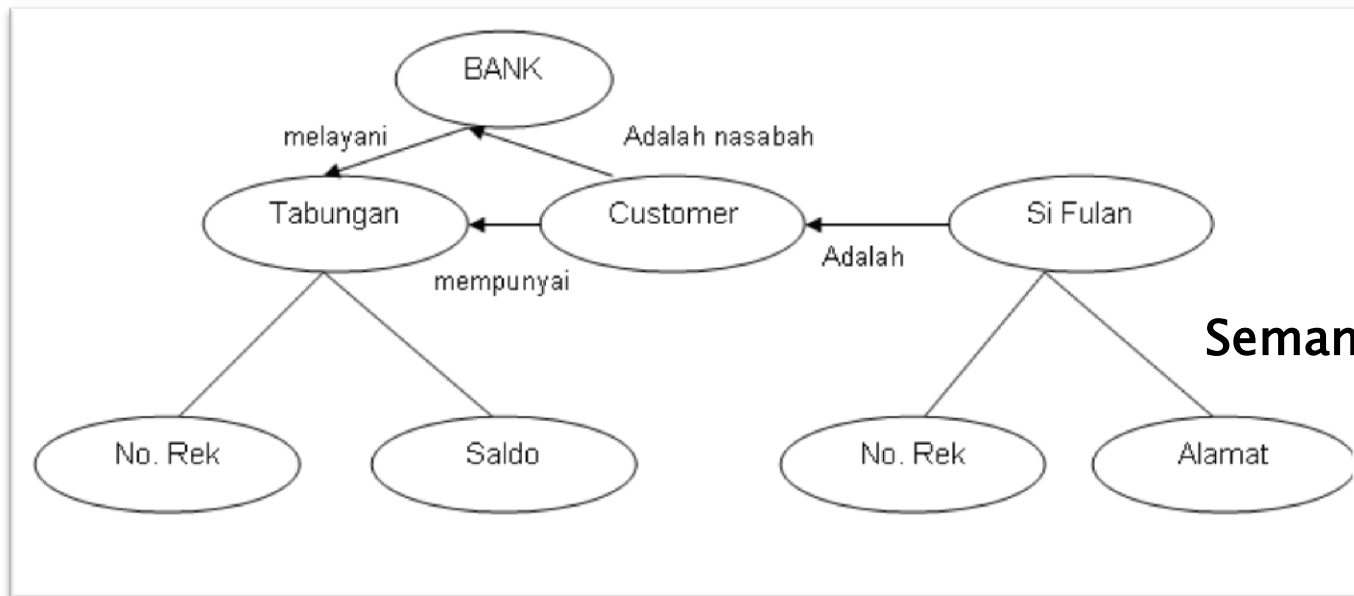
Model data berbasis obyek

- ▶ Merupakan himpunan data dan prosedur/relasi yang menjelaskan hubungan logik antar data dalam suatu basiss data berdasarkan obyek datanya
- ▶ Terdiri atas :
 - *entity relationship model*
 - *Semantic model*
 - *Binary model*

Model data berbasis obyek



Entity Relationship model

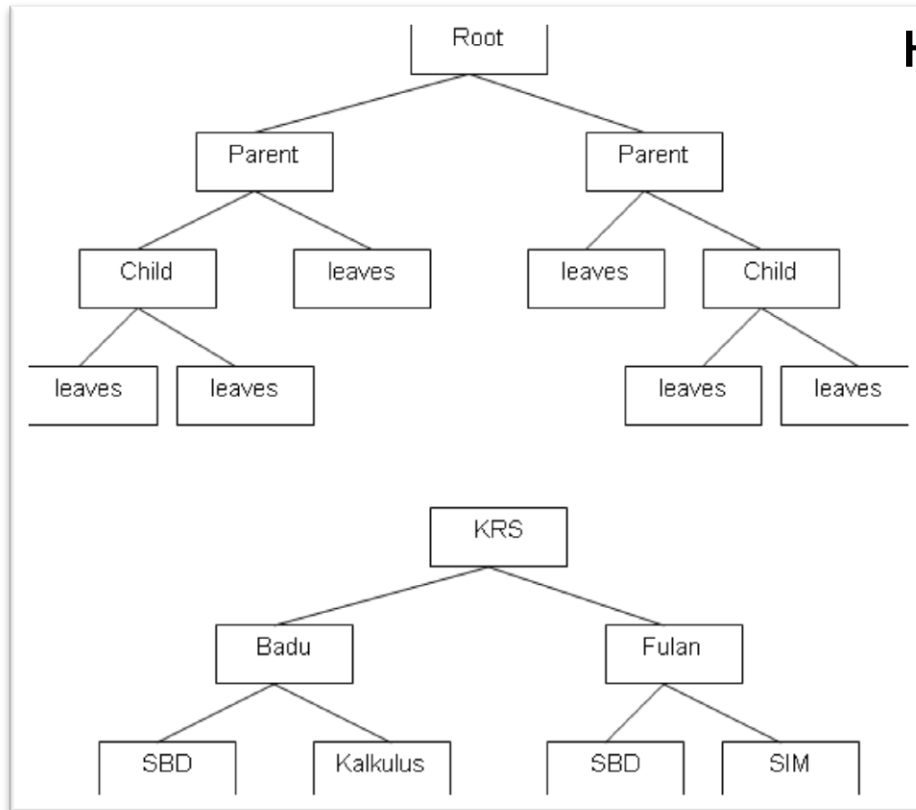


Semantic Model

Model data berbasis record

- ▶ Model ini berdasarkan record/rekaman untuk menjelaskan kepada pemakai mengenai hubungan logik antar data dalam basis data
- ▶ Terdiri dari :
 - *hierarchical model*
 - *Network model*
 - *Relational model*

Model data berbasis record

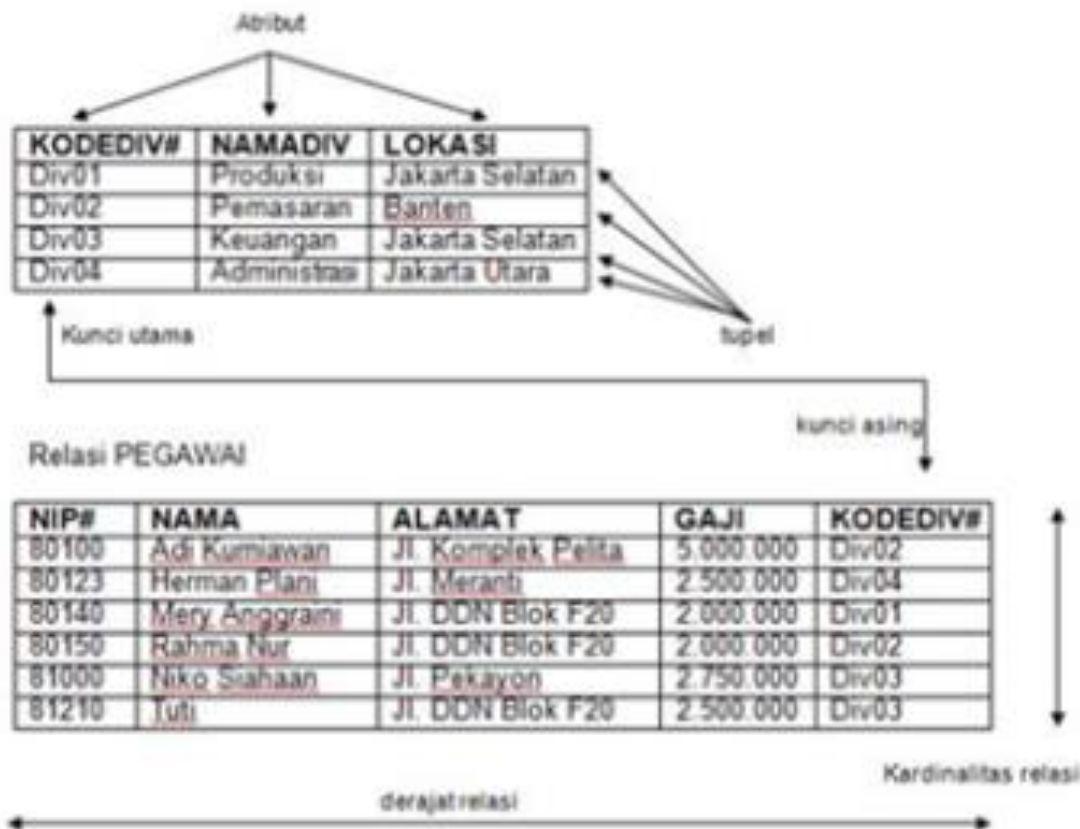


Hirarchycal Model (Tree Structure)

Network Model (Plex Structure)



Model data berbasis record



Model Data Relasional

Model data berbasis record

Model Data Relasional

Tabel : Barang

KdBrg	NmBrg	Stok	Harga

Tabel : Dijual

KdPlg	KdBrg	TglTr	Jumlah

Tabel : Pelanggan

KdPlg	NmPlg	Alamat

Model data fisik

- ▶ Model ini digunakan untuk menguraikan data di tingkat internal atau menjelaskan kepada pemakai bagaimana data-data dalam basis data disimpan dalam media penyimpanan secara fisik. Model ini jarang digunakan karena kerumitan dan kompleksitasnya yang justru menyulitkan pemakai.
- ▶ Model ini terdiri dari :
 - *unifying model*
 - *Frame memory*

Model data fisik

Nama Field	Tipe	Panjang	Keterangan
NIP	Character	10	Nomor Pokok Mahasiswa
NAMA	Character	25	Nama mahasiswa
ALAMAT	Character	35	Alamat mahasiswa
TGL_LAHIR	Date	10	Tanggal lahir

Entity relationship model

- ▶ Digunakan untuk menjelaskan hubungan antar data dalam basis data kepada user secara logik.
- ▶ ER-M didasarkan pada persepsi bahwa real world terdiri atas obyek-obyek dasar yang mempunyai hubungan/kerelasian antar obyek- obyek dasar tersebut.
- ▶ ER-M yang digambarkan dalam bentuk diagram disebut diagram ER dengan menggunakan simbol-simbol grafis tertentu

Mengapa perlu model E-R

- ▶ Dapat menggambarkan hubungan antar entity dengan jelas
- ▶ Dapat menggambarkan batasan jumlah entity dan partisipasi antar entity
- ▶ Mudah dimengerti oleh pemakai
- ▶ Mudah disajikan oleh perancang database

ENTITY-RELATIONSHIP MODEL

ENTITY-RELATIONSHIP MODEL



Entity Relationship Diagram

1. Entity Relationship Model
2. Entity Relationship Diagram (ERD)

Entity Relationship - Model

Model yang digunakan untuk menggambarkan data dalam bentuk **entity**, **attribute**, dan **relationship** antar entitas.

E-R Models are *Conceptual Models* of the database.

Artinya, belum langsung bisa diimplementasikan ke dalam basis data yang sebenarnya.

E-R Model menjembatani antara objek nyata dengan basis data

Entity-Relationship Diagram (ERD)

- ❖ Merupakan diagram yang menggambarkan hubungan (relationship) antar entitas (entity).
- ❖ ER Diagram dapat mengekspresikan struktur logis sebuah basis data dengan simple dan jelas.
- ❖ Terdiri dari tiga bentuk diagram dasar:
 - ❖ Persegi : merepresentasikan entitas
 - ❖ Elips : merepresentasikan attribute
 - ❖ Garis : merepresentasikan hubungan
- ❖ Tiga bentuk dasar tersebut kemudian dikembangkan menjadi bentuk lain, untuk merepresentasikan sifat yang berbeda.

Entitas

- ▶ Entitas adalah objek yang ada dan dapat dibedakan dari objek lain dalam dunia nyata.
- ▶ Sebuah entitas memiliki **atribute** yang membedakan dirinya dengan objek lain, dan seringkali mempunyai hubungan dengan objek lain
- ▶ Example
 - Manusia : MAHASISWA, PASIEN, PEGAWAI
 - Tempat : TOKO, GUDANG
 - Kejadian : ?

Simbol :

nama_entity

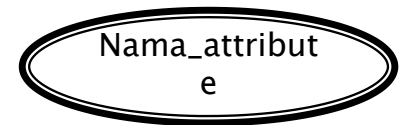
Entity – Set Entitas cont'd

- ▶ *Entity sets* adalah kumpulan entitas yang mempunyai tipe yang sama, contoh kumpulan entitas mahasiswa, kumpulan entitas pegawai, dll.
- ▶ Entity sets dari entitas yang sama tidak perlu diuraikan.
 - Misalkan terdapat set entitas **mahasiswa**, dan set entitas **staff**. Entitas orang mungkin dapat termasuk mahasiswa atau staff atau keduanya atau tidak keduanya.

Attribute

- ▶ Sebuah entitas **direpresentasikan** (dibedakan dengan entitas lain) oleh beberapa atribut.
- ▶ Atribut adalah **properti deskriptif** yang dimiliki oleh semua anggota dari semua set entitas
- ▶ Example
 - MAHASISWA : nim, nama, tgl_lahir, jns_klm
 - PEGAWAI : ?

Simbol :



Attribute cont'd

- ▶ Setiap atribut pada entitas memiliki kunci atribut (key attribute) yang bersifat unik. **Primary key**
- ▶ Example
 - Entitas : Mahasiswa
 - Attribute :

Nim	Nm_mhs	Jns_kl	Tgl_lhr	Alamat
A11.01.019 15	Joarni	Perempua n	12 Des 1991	Jl. Mentok 5 Ujung-ujung Semarang
A11.01.019 89	Ronaldo Iskandar	Laki-laki	4 Jan 1992	Jl. Ngawur 10 Banyukali Semarang
A11.01.019 76	Joarno	Laki-laki	19 Juni 1992	Jl. Kenangan 8 Candirejo Semarang
A11.01.019 77	Joarni	Perempua n	19 Juni 1992	Jl. Kenangan 8 Ksatrian Semarang

VALUE SET (*Domain*) dari Atribut

- ▶ Kumpulan nilai yang dapat dimasukkan pada setiap attribute, Kumpulan nilai yang dapat dimiliki oleh atribut dari suatu entitas.
- ▶ Definisi domain dari suatu atribut akan mencakup :
 - tipe data, panjang, format, nilai yang memungkinkan, keunikan dan kemungkinan data null.
- ▶ Example :
 - Atribut nama pada entitas mahasiswa domainnya nama orang
 - Atribut nama pada entitas barang domainnya nama barang

Jenis Attribute

- ▶ Atribut sederhana - **atomic attribute**
atribut yang terdiri atas satu komponen tunggal dengan keberadaan yang independen, tidak bisa diuraikan lagi.

Example :

atribut *nim* dalam entitas mahasiswa.

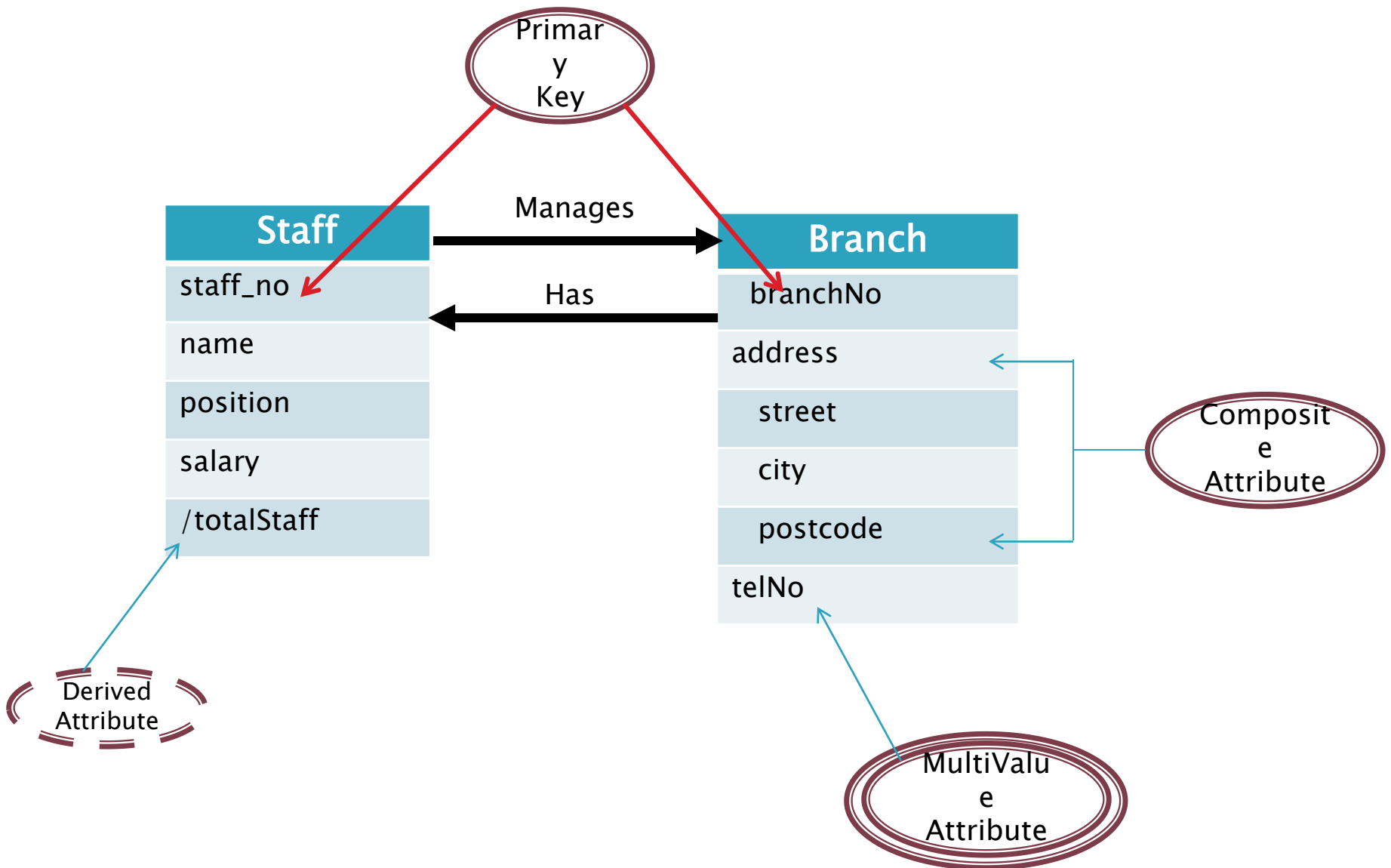
- ▶ Atribut komposit – **composite attribute**
atribut yang terdiri dari beberapa atribut yang lebih mendasar, bisa diuraikan.

Example :

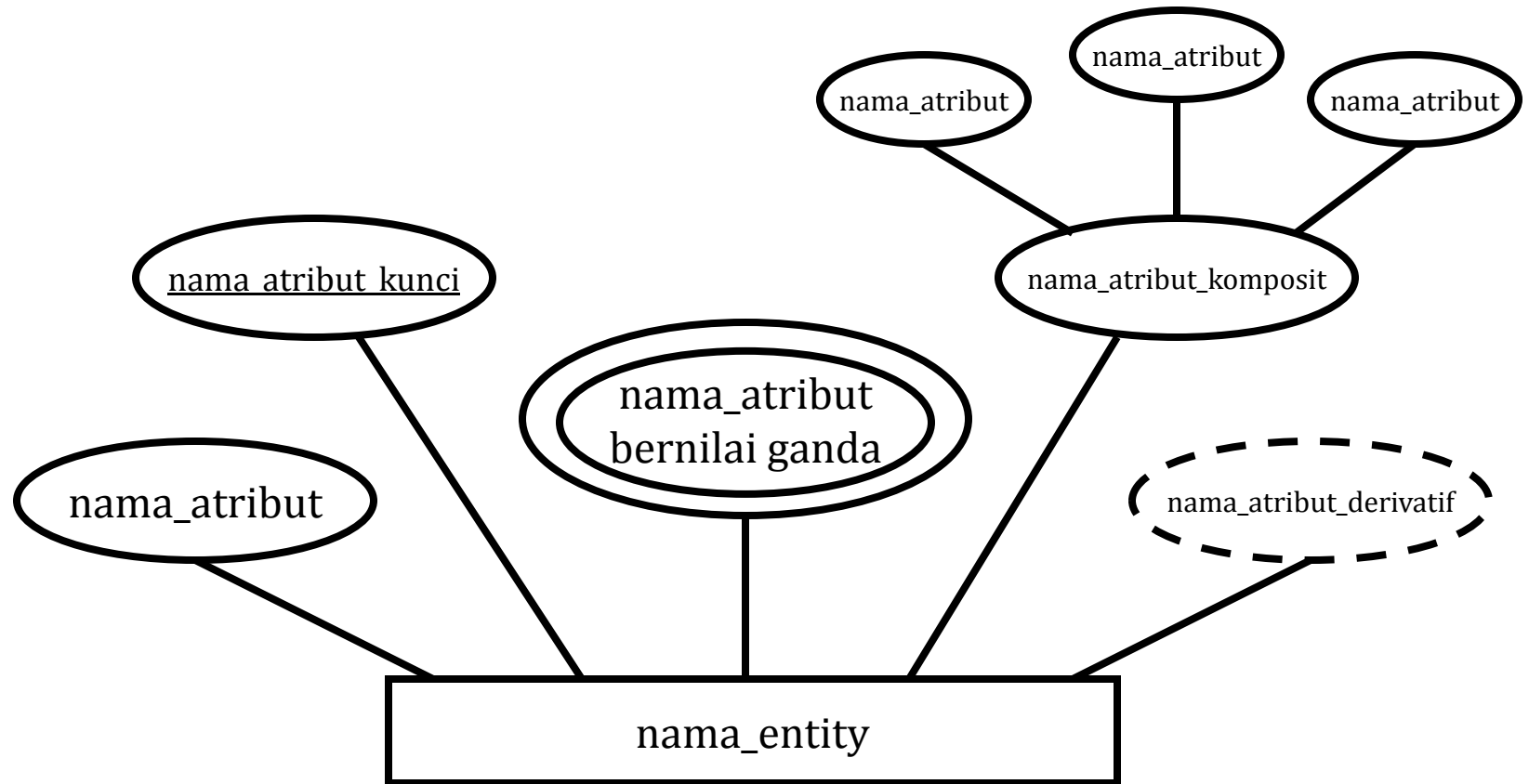
Entity mahasiswa memiliki atribut *alamat_mhs* yang terdiri dari *alamat*, *nama_kota*, dan *kode_pos*.

Jenis Attribute cont'd

- ▶ Atribut Berharga Tunggal - **Single-valued Attribute**
atribut yang hanya mempunyai satu nilai untuk suatu entitas tertentu.
Example : atribut *jns_klm* pada entity mahasiswa
- ▶ Atribut Berharga Banyak - **Multi-valued Attribute**
atribut yang dapat terdiri dari sekumpulan nilai untuk suatu entitas tertentu.
Example : atribut *hobi* pada entity mahasiswa
- ▶ Atribut Derivatif - **Derived Attribute**
suatu atribut yang dihasilkan dari atribut lain yang tidak berasal dari satu entitas
Example : atribut *umur* yang dapat dihasilkan dari atribut *tgl_lahir*.



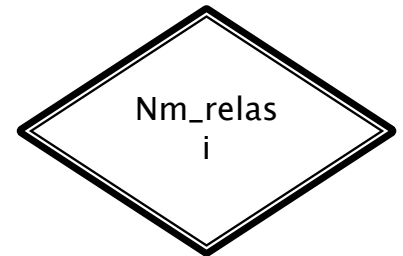
Simbol-Simbol Atribut (oval)



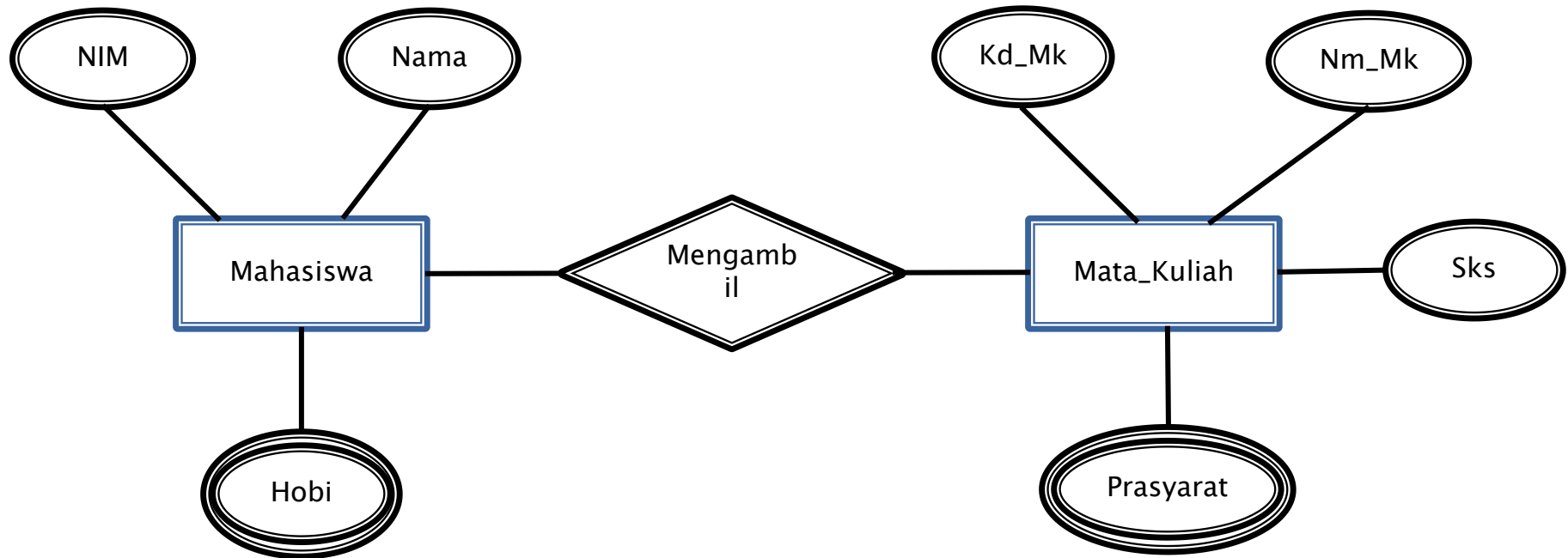
RELATIONSHIP

- ▶ **Relationship** adalah hubungan yang terjadi antara satu atau lebih entity.
- ▶ Dapat memiliki atribut, dimana terjadi adanya transaksi yang menghasilkan suatu nilai tertentu.
- ▶ Example :
 - Entity MAHASISWA dengan
nim="99002" dan nama="Azhary"
 - **berelasi** dengan entity sebuah MATA KULIAH
kd_mk="BD02" dan nama_kul="Basis Data".
 - Relasi diantara kedua entity :
Mahasiswa sedang **mengambil** mata kuliah tersebut

Simbol :



RELATIONSHIP cont'd



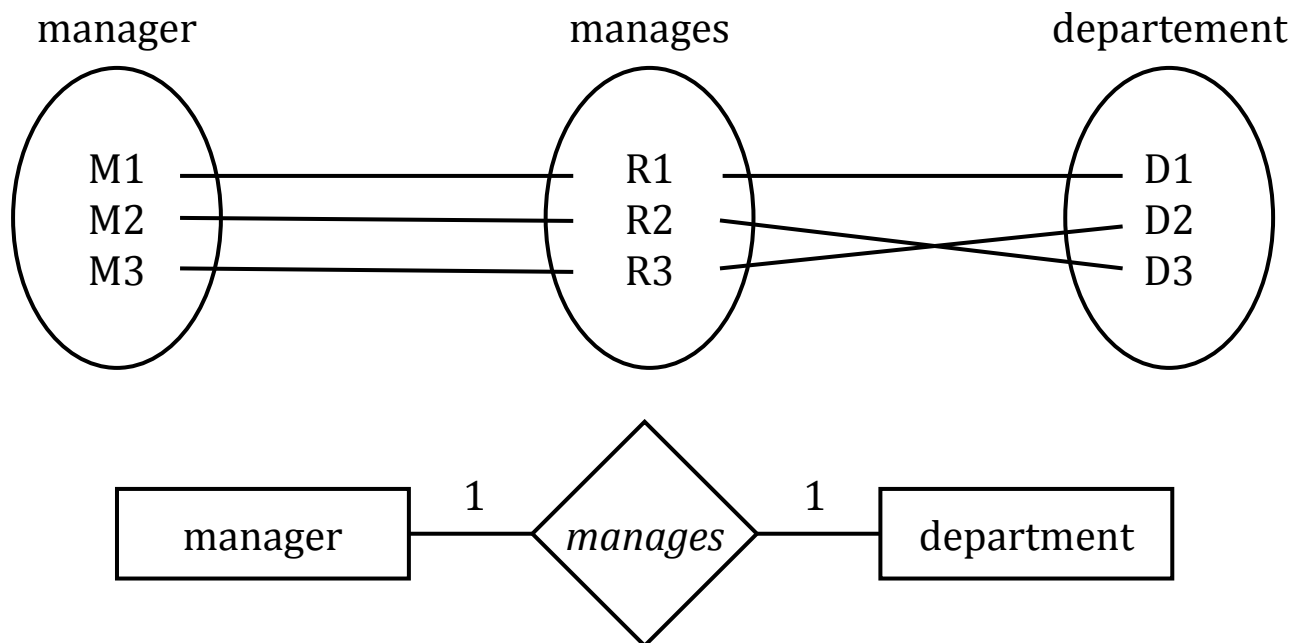
► Mahasiswa *mengambil* mata kuliah

Cardinality Ratio

- ▶ Menjelaskan jumlah **keterhubungan** satu entity dengan entity yang lainnya.
- ▶ Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.
- ▶ Maximum Cardinality
 - One to One
 - One to Many or Many to One
 - Many to Many

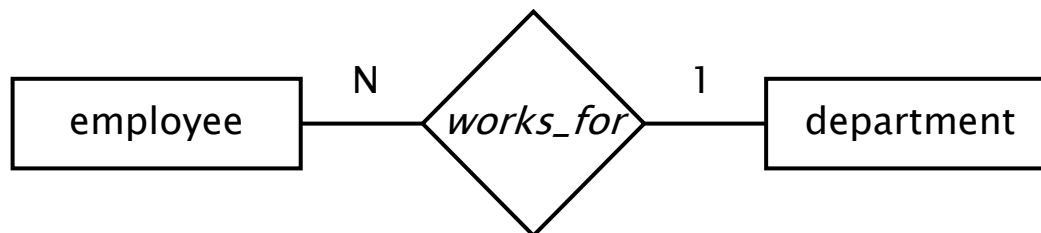
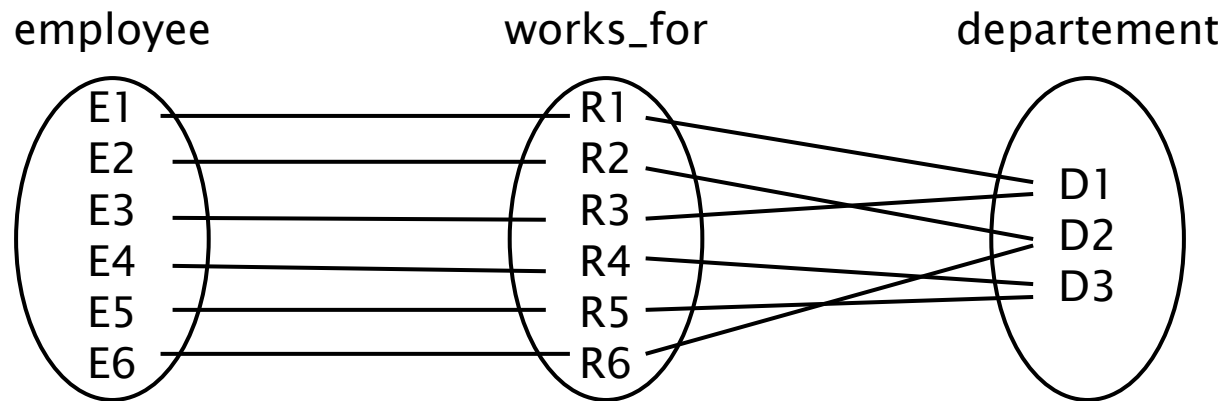
Cardinality Ratio – One to One

- ▶ (1 : 1) : satu entitas pada entitas A berhubungan dengan paling banyak satu entitas pada entitas B dan juga sebaliknya.
 - Contoh : seorang manager hanya memimpin satu departemen dan begitu sebaliknya.

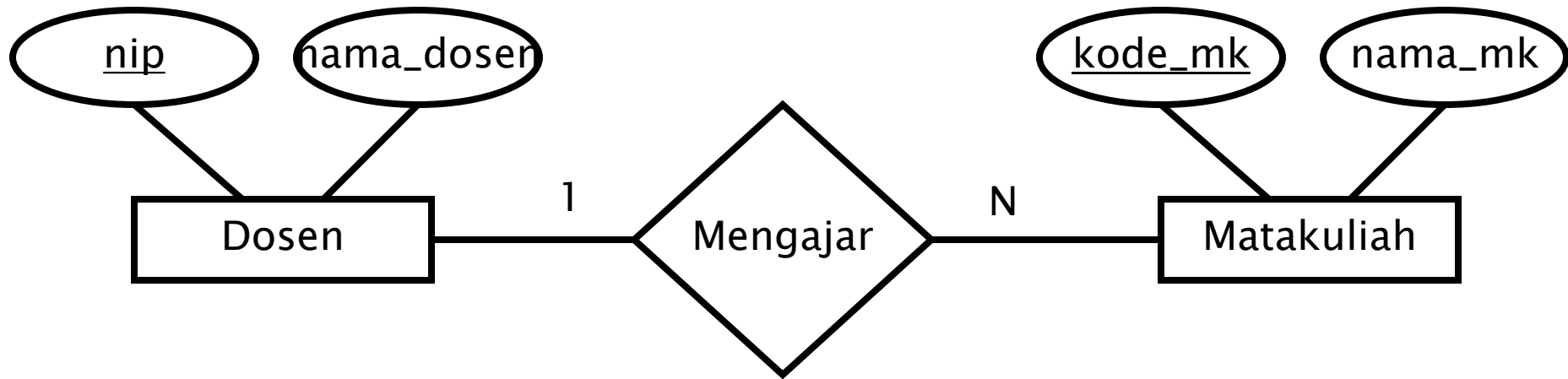


Cardinality Ratio – One to Many

- ▶ $(1 : N / N : 1)$: suatu entitas di A dihubungkan dengan sejumlah entitas di B.
 - Contoh : banyak karyawan berkerja untuk satu departement atau satu departement memiliki banyak karyawan yang bekerja untuknya.

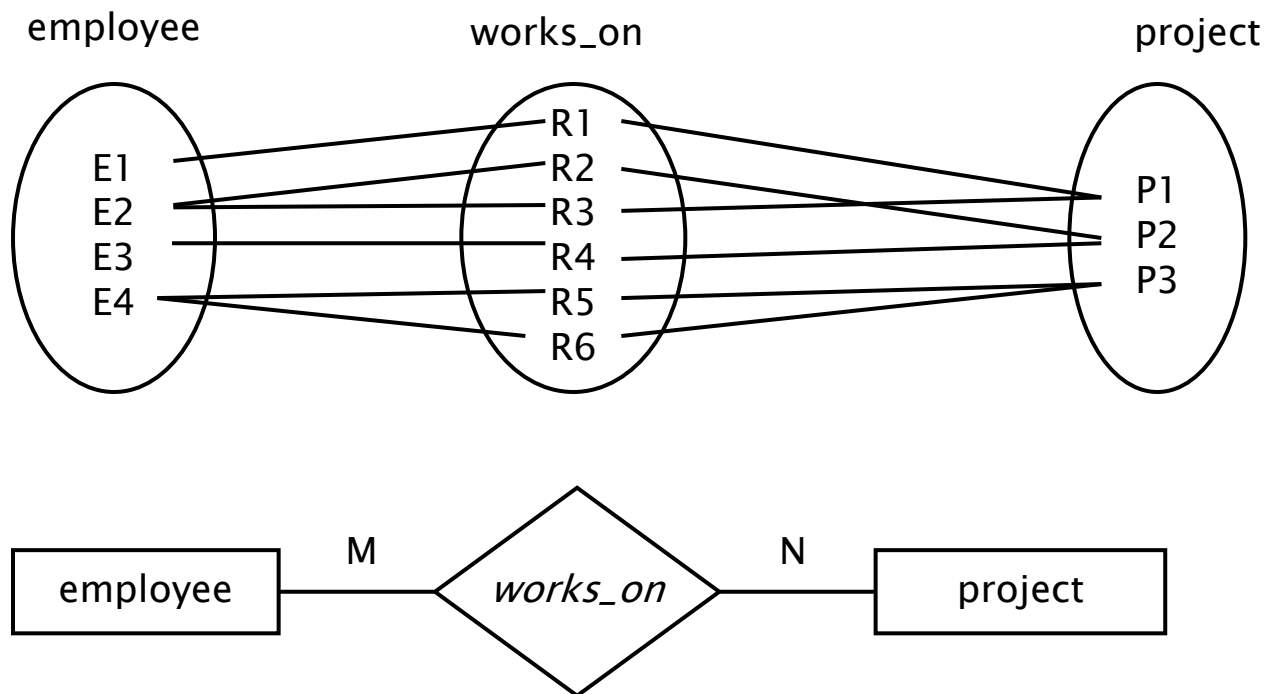


Contoh : ERD One to Many

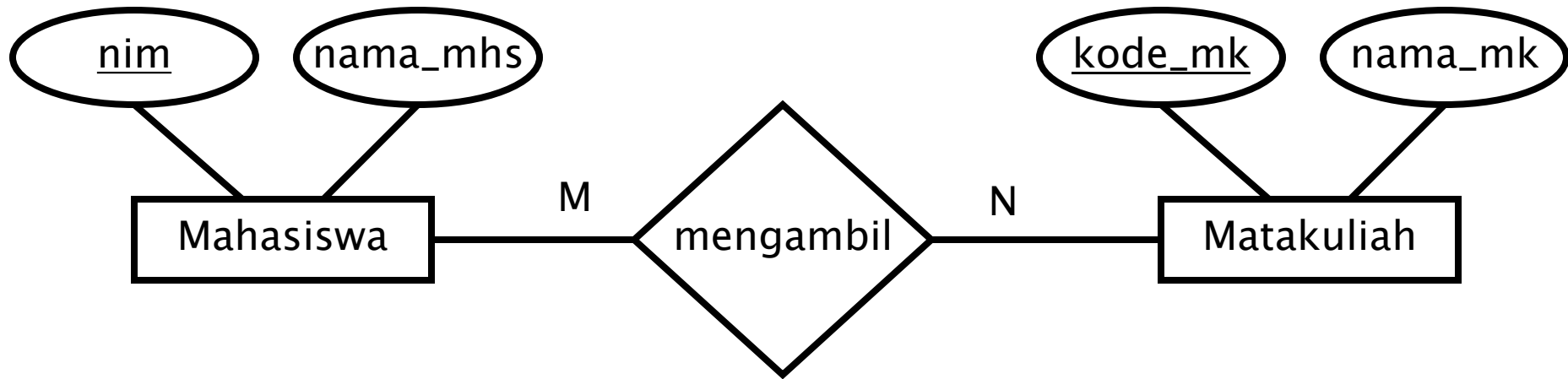


Cardinality Ratio – Many to Many

- ▶ (M : N) : setiap entitas A dapat berhubungan dengan banyak entitas B dan sebaliknya setiap entitas B juga dapat berhubungan dengan banyak entitas A.
 - Contoh : satu proyek mempunyai banyak karyawan, satu karyawan boleh bekerja di beberapa proyek.



Contoh : ERD Many to Many



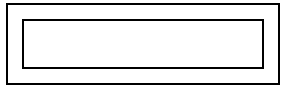
Symbol E-R Diagram

Symbol

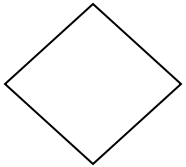
Keterangan



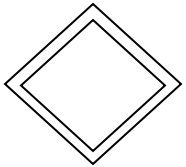
= Entity



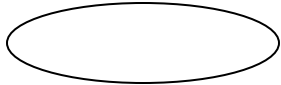
= Weak Entity



= Relationship



= Identifying Relationship



= Atribut



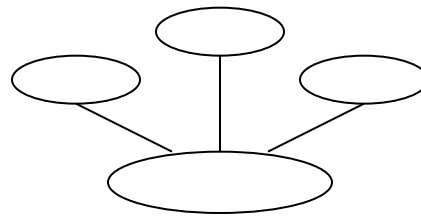
= Atribut Kunci



= Atribut Multivalue

Symbol

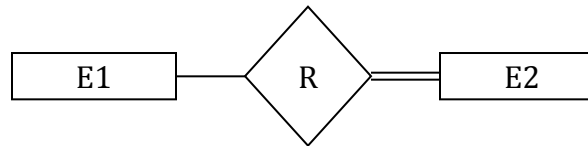
Keterangan



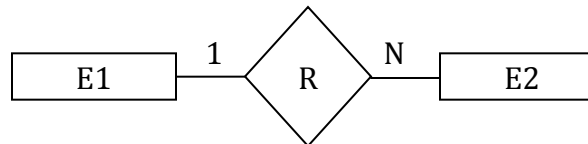
= Atribut Komposit



= Atribut Derivatif



= Total Participation Of E2 In R



= Cardinality Ratio 1:N For E1:E2 In R

Thank you