



**MATA KULIAH** : Basis Data Lanjut  
**KODE MATA KULIAH/SKS** : SI0054/ 3/1 sks  
**KURIKULUM** : 2017  
**VERSI** : 0.0

## DESKRIPSI MATA KULIAH

Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai basis data transaksional meliputi model, pengamanan, pengendalian, basis data berorientasi objek, basis data objek relasional, basis data terdistribusi dan aplikasi basis data, serta basis data analytical meliputi Data Warehouse, OLAP, dan Data Mining.



## CAPAIAN PEMBELAJARAN

Mahasiswa mampu membangun serta mengelola basis data transaksional dan basisdata analytical untuk menghasilkan informasi melalui sistem ataupun aplikasi yang sesuai dengan permasalahan maupun kebutuhan bisnis.



# SUMBER PUSTAKA

1. Connoly, T.M and Carolyn B. 2015. *Database Systems, A Practical Approach to Design, Implementation, and Management, Sixth Edition*. England: Pearson Education Limited. ISBN 10: 1-292-06118-9. ISBN 13:978-1-292-06118-4.
2. Date, CJ. 2000. *An Introduction to Database System Seventh Edition*. New Jersey: Pearson Addison Weesley. ISBN: 979-683-185-6.
3. Fatansyah. 2012. *Basis Data*. Bandung: Informatika. ISBN:978-602-8758-53-6
4. Nugroho, Adi. 2011. *Perancangan dan Implementasi Basis Data*. Yogyakarta: Andi. ISBN 978-979-29-2609-5.
5. Nugroho, Adi. 2014. *Sistem Basis Data ORACLE 10g*. Yogyakarta: Andi. ISBN 978-979-29-4342-9.



# Minggu 2

## Pertemuan 2



## KEMAMPUAN AKHIR YANG DIHARAPKAN

Mahasiswa mampu memahami konsep model basis data, penerapan basis data, serta denormalisasi basis data untuk basis data transaksional



# POKOK BAHASAN

- Penerapan Basis Data



# MATERI POKOK

- Transformasi diagram Entity Relationship ke Relasi
- Transformasi Model data ke Basis Data Fisik
- DBMS dan Struktur Tabel
- Indeks dan Struktur Penyimpanan





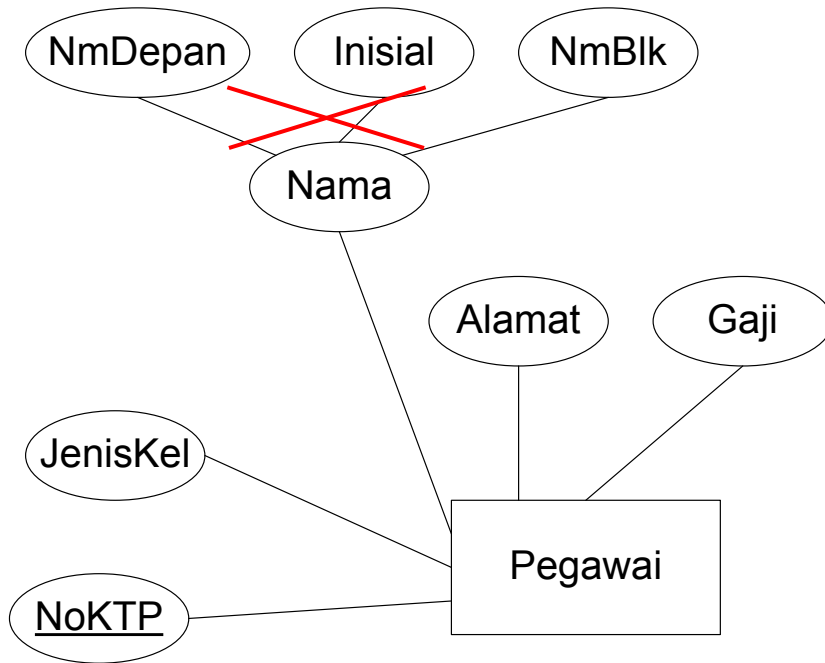
# Transformasi diagram Entity Relationship ke Relasi

## Algoritma 1

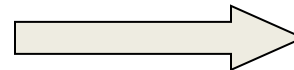
- Untuk setiap entitas kuat EK, buat tabel baru EK yang memasukkan semua atribut sederhana (simple attribut)
- Untuk atribut komposit, hanya atribut sederhananya yang disertakan
- Pilih salah satu *key attribute* dari EK sebagai primary key. Jika key yang terpilih merupakan atribut komposit, seluruh atribut sederhananya merupakan primary key



# Algoritma 1



Langkah 1



## Tabel PEGAWAI

NoKTP  
NmDepan  
Inisial  
NmBlk  
JenisKel  
Alamat  
Gaji

**Atribut komposit nama tidak menjadi field/kolom pada tabel PEGAWAI, tetapi yg dimasukkan adalah bagian simple atributnya.**

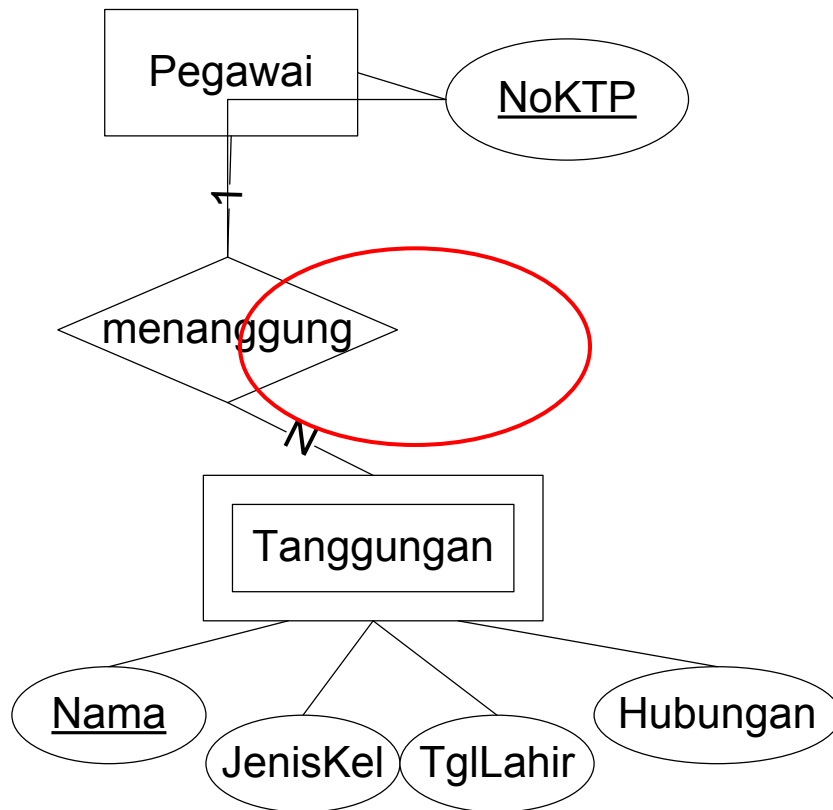


## Algoritma 2

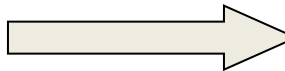
- Untuk setiap entitas lemah EL yang dimiliki oleh entitas kuat EK, buat tabel baru EL yang memasukkan semua atribut sederhana EL
- Tambahkan pada EL foreign key yang diambil dari primary key EK
- Primary key yang dibentuk merupakan gabungan primary key EK dan partial key dari EL (jika ada)



# Algoritma 2



Langkah 2



Tabel  
**TANGGUNGAN**

<u>Peg</u>	<u>NoKTP</u>
<u>Nama</u>	
JenisKel	
TglLahir	
Hubungan	



# Algoritma 3

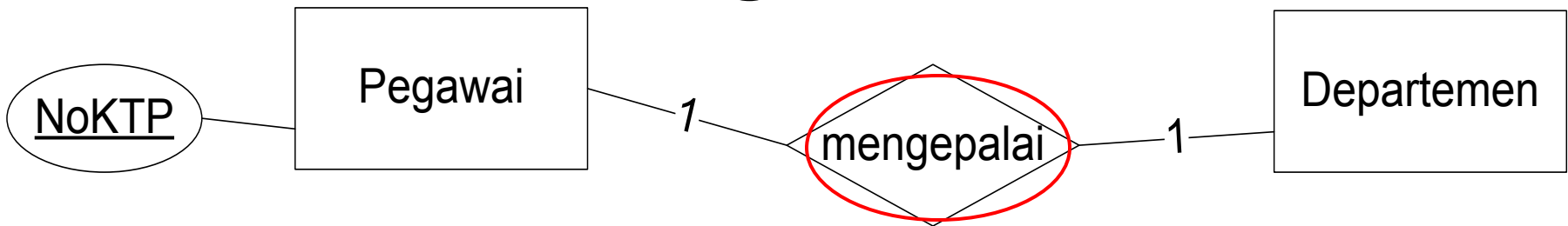
Untuk setiap relasi 1:1 antara entitas EK1 dan EK2:

- Jika sama-sama merupakan partisipasi total, pilih salah satu. Tambahkan semua simple atribut dari entitas yang lain
- Jika sama-sama merupakan partisipasi parsial, pilih salah satu. Tambahkan foreign key dari primary key entitas yang lain
- Jika salah satu merupakan partisipasi total, pilih yang berpartisipasi total. Tambahkan foreign key dari primary key entitas yang lain

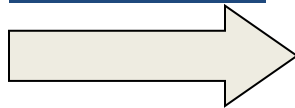
Kemudian tambahkan semua simple atribut dari relasi tersebut



# Algoritma 3



Langkah 1-3



<u>Tabel</u> <u>DEPARTEMEN</u>	
Peg_NoKTPKepala	

Kolom-kolom yang telah dibentuk  
pada langkah-langkah sebelumnya

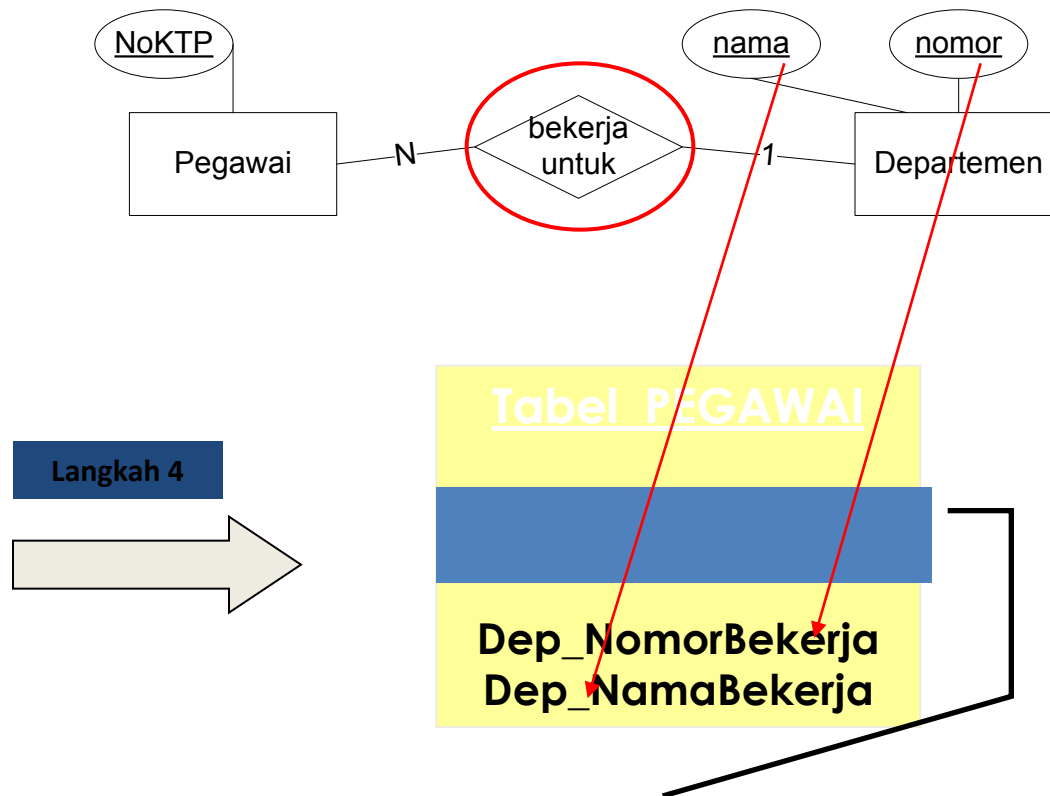


# Algoritma 4

- Untuk setiap relasi 1:N antara entitas EK1 dan EK2, pilih entitas yang memiliki derajat maksimum relasi=1
- Tambahkan sebuah foreign key dari primary key entitas lain yang memiliki derajat maksimum relasi = N
- Tambahkan pula seluruh atribut dari relasi tersebut



# Algoritma 4



Kolom-kolom yang telah dibentuk  
pada langkah-langkah sebelumnya



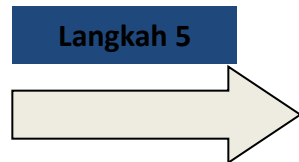
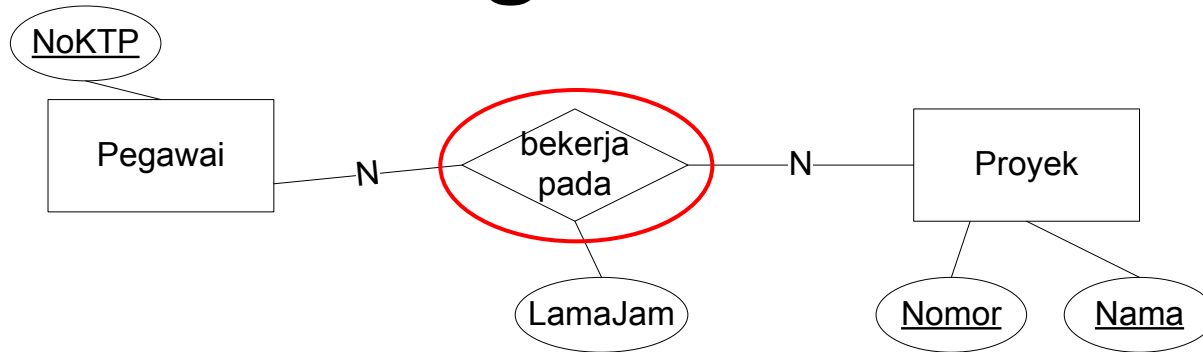


# Algoritma 5

- Untuk setiap relasi M:N antara entitas EK1 dan EK2, buat tabel baru MN
- Tambahkan seluruh simple attribut dari relasi tersebut
- Tambahkan pula foreign key yang diambil dari primary key masing-masing entitas yang direlasikan
- Primary key merupakan gabungan dari seluruh foreign key tersebut



# Algoritma 5



Tabel  
BEKERJAPADA  
Peg NoKTP  
Pro Nomor  
Pro Nama  
LamaJam

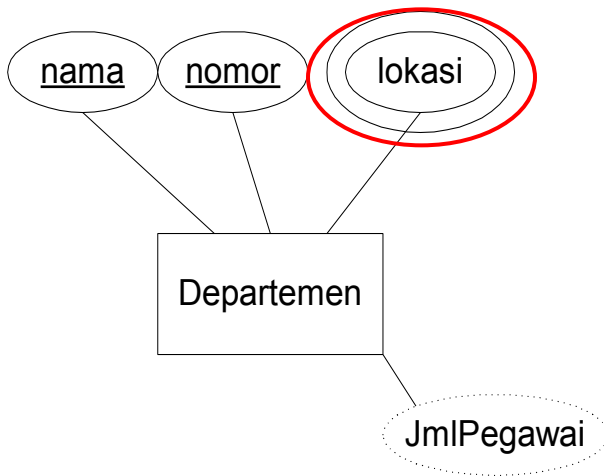


# Algoritma 6

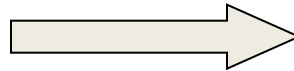
- Untuk setiap *multivalued attribute*, buat tabel baru MV
- Tambahkan seluruh simple atributnya
- Tambahkan pula sebagai foreign key, primary key dari entitas yang memiliki
- Primary key merupakan gabungan dari dari seluruh fieldnya



# Algortima 6



Langkah 6



Tabel  
DEP LOKASI  
Dep\_Nomor  
Dep\_Nama  
Lokasi

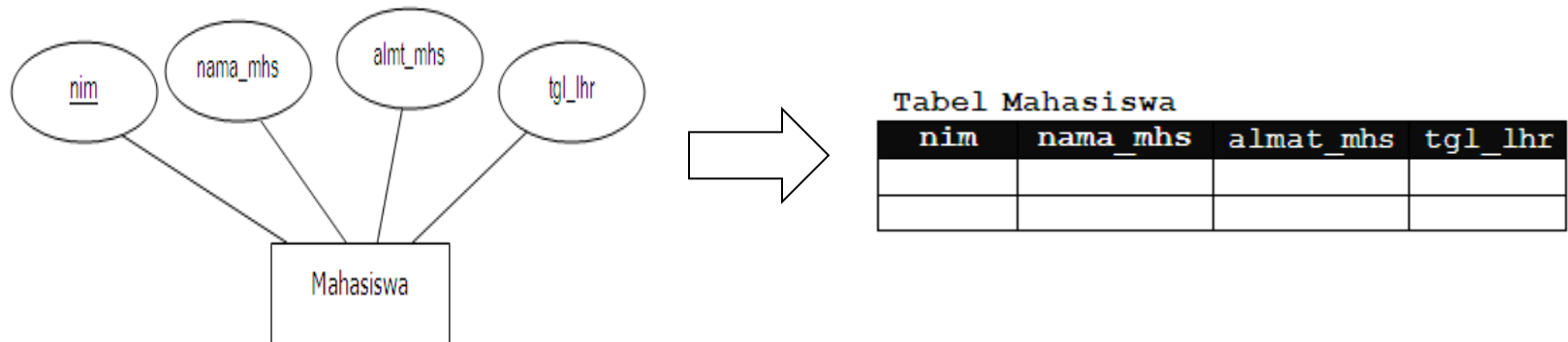


# Transformasi Model Data ke Basis Data Fisik

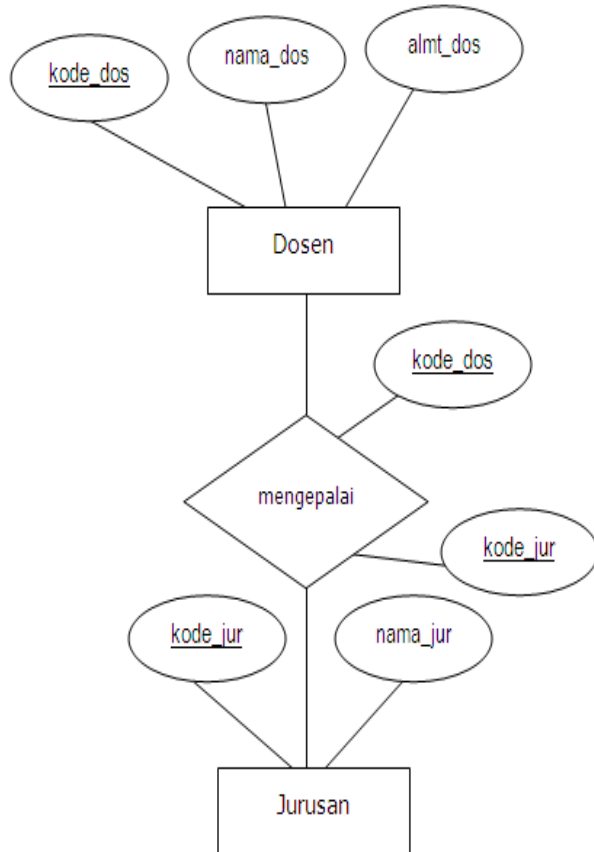
- Transformasi Umum/Dasar

Aturan umum dalam pemetaan Model Data (Level Konseptual dalam Abstraksi Data) yang digambarkan dengan Diagram E-R menjadi Basis Data Fisik (Level Fisik dalam Abstraksi Data) adalah:

1. Setiap himpunan entitas akan diimplementasikan sebagai sebuah tabel (file data)



2. Relasi dengan Derajat relasi 1:1 (satu ke satu) yang menghubungkan 2 buah himpunan entitas akan direpresentasikan dalam bentuk penambahan/penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas.



Tabel Dosen

kode_dos	nama_dos	almt_mhs



Tabel Jurusan

kode_jur	nama_jur	kode_dos

Himpunan relasi mengepalai sesungguhnya memiliki 2 atribut, yaitu kode\_dos dan kode\_jur. Kedua atribut ini seharusnya disertakan pada tabel jurusan. Tetapi karena di tabel Jurusan (yang merupakan implementasi dari himpunan entitas Jurusan) sudah ada field kode\_jur, maka cukup atribut kode\_dos yang perlu ditambahkan. Jika saja atribut relasi ini disertakan ke tabel Dosen, maka kita juga hanya tinggal menambahkan sebuah field yaitu kode\_jur.

Yang menjadi kunci pada tabel Jurusan tetap saja kode\_jur, kedudukan field kode\_dos di tabel Jurusan seperti field nama\_jur, yakni sebagai field deskriptif saja. Karena lebih bersifat deskriptif, nama field tersebut dapat saja diganti untuk lebih memperjelas fungsi/perannya di tabel Jurusan (yaitu data tentang dosen yang menjadi ketua jurusan), misalnya menjadi kode\_dos\_kajur

# Transformasi Model Data ke Basis Data Fisik

Di setiap relasi satu ke satu akan selalu ada dua pilihan peleburan relasi (dalam bentuk penyertaan atribut ke salah satu tabel).

Untuk menentukan pilihan yang tepat kita perlu melihat derajat relasi minimumnya. Setiap dosen hanya boleh mengepalai satu jurusan, tetapi tidak semua dosen dapat mengepalai suatu jurusan (jadi ada dosen-dosen yang tidak mengepalai jurusan manapun). Dengan begitu derajat relasi minimumnya adalah 0 (nol).

Sedang dari sisi jurusan : setiap jurusan hanya boleh dikepalai seorang dosen, tetapi tidak boleh ada jurusan yang tidak ada kepalanya. Dengan demikian, derajat minimumnya adalah 1 (satu).

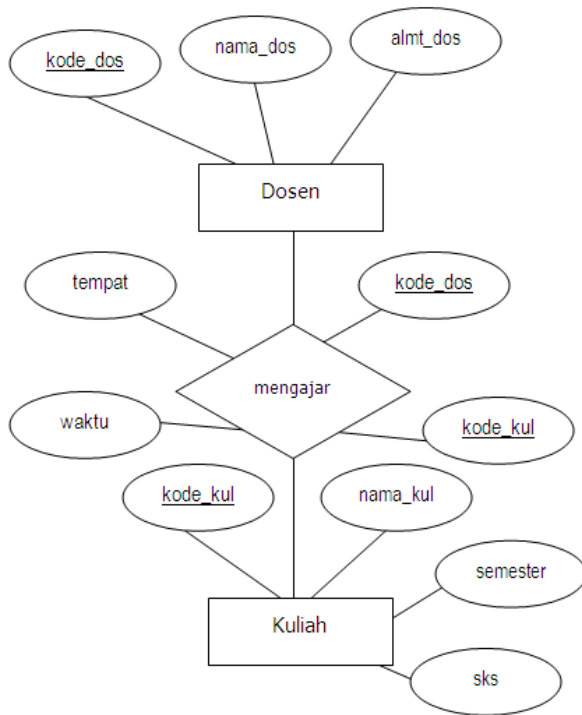


## Transformasi Model Data ke Basis Data Fisik

Jika kondisinya demikian, maka yang paling tepat relasi mengepalai itu dileburkan ke tabel yang memiliki derajat minimum yang lebih besar, yaitu tabel jurusan.

Jika pada sebuah relasi satu-ke-satu, derajat relasi minimumnya sama (sama-sama berderajat minimum 0 (nol) atau sama-sama berderajat minimum 1 (satu)), maka relasi lebih baik dileburkan ke tabel yang jumlah row-nya lebih sedikit atau yang ukuran tabelnya diperkirakan lebih kecil.

3. Relasi dengan Derajat relasi 1-N (satu ke banyak) yang menghubungkan 2 buah himpunan entitas, juga akan direpresentasikan dalam bentuk pemberian/pencantuman atribut key dari himpunan entitas pertama (yang berderajat 1) ke tabel yang mewakili himpunan entitas kedua (yang berderajat N). Atribut key dari himpunan entitas pertama ini menjadi atribut tambahan bagi himpunan entitas kedua



Tabel Dosen

kode_dos	nama_dos	almt_mhs

Tabel Kuliah

kode_kul	nama_kul	sks	semester	kode_dos	tempat	waktu

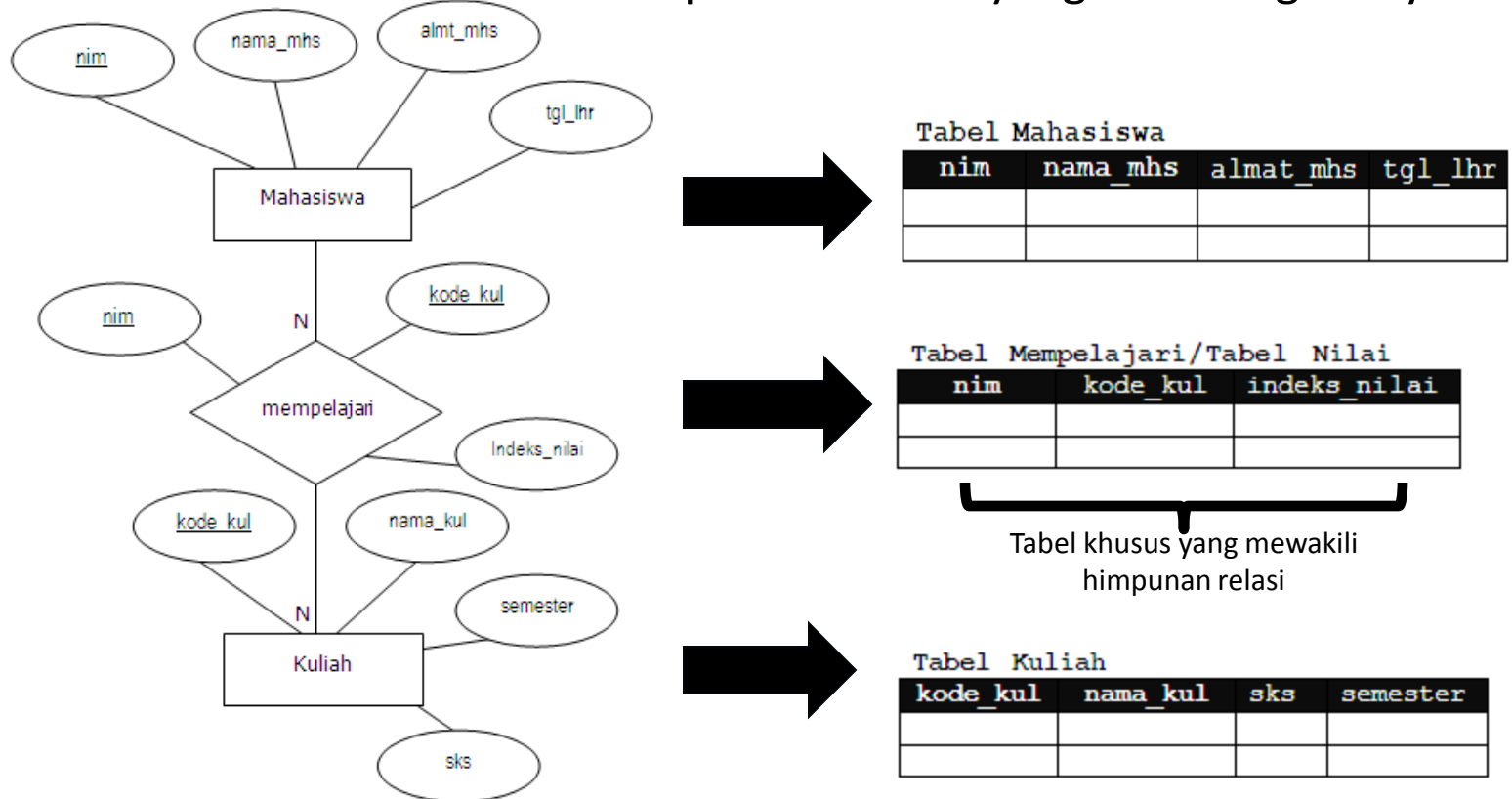
field-field yang ditumbuhkan  
dari relasi Mengajar

## Transformasi Model Data ke Basis Data Fisik

Jika kemudian kita melihat bahwa atribut bernilai banyak (*multivalued attribute*), maka tabel tersebut tidak memenuhi Bentuk Normal tahap Pertama (1<sup>st</sup> NF). Dengan menerapkan aturan Normalisasi tersebut, maka sesungguhnya tabel ini dapat didekomposisi lagi menjadi tabel Kuliah (*kode\_kul, nama\_kul, sks, semester, kode\_dos*), dan tabel Jadwal (*kode\_kul, tempat, waktu*)



4. Relasi dengan Derajat relasi N-N (banyak ke banyak) yang menghubungkan 2 buah himpunan entitas, akan diwujudkan dalam bentuk tabel (file data) khusus, yang memiliki field (tepatnya foreign key) yang berasal dari key-key dari himpunan entitas yang dihubungkannya.

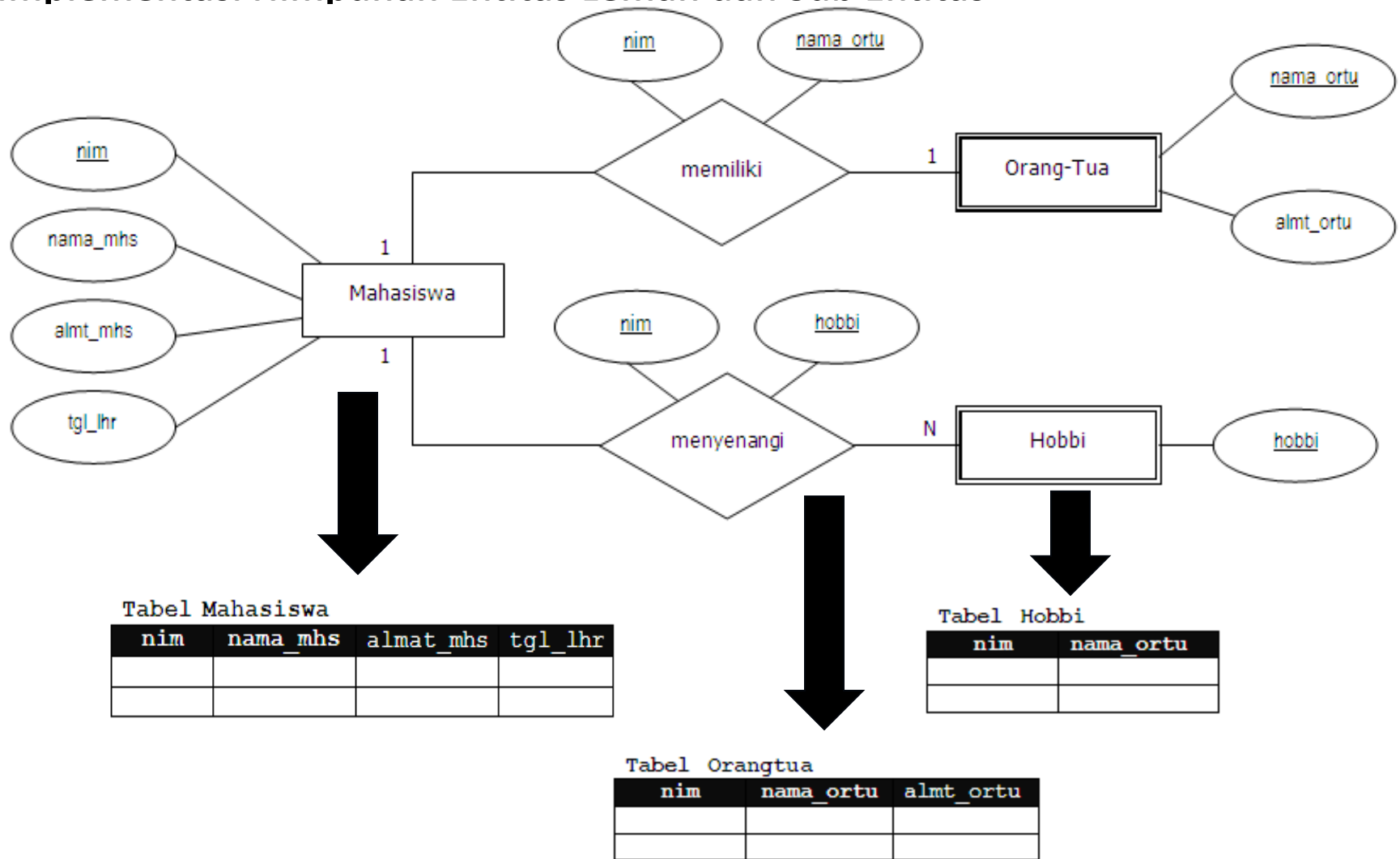


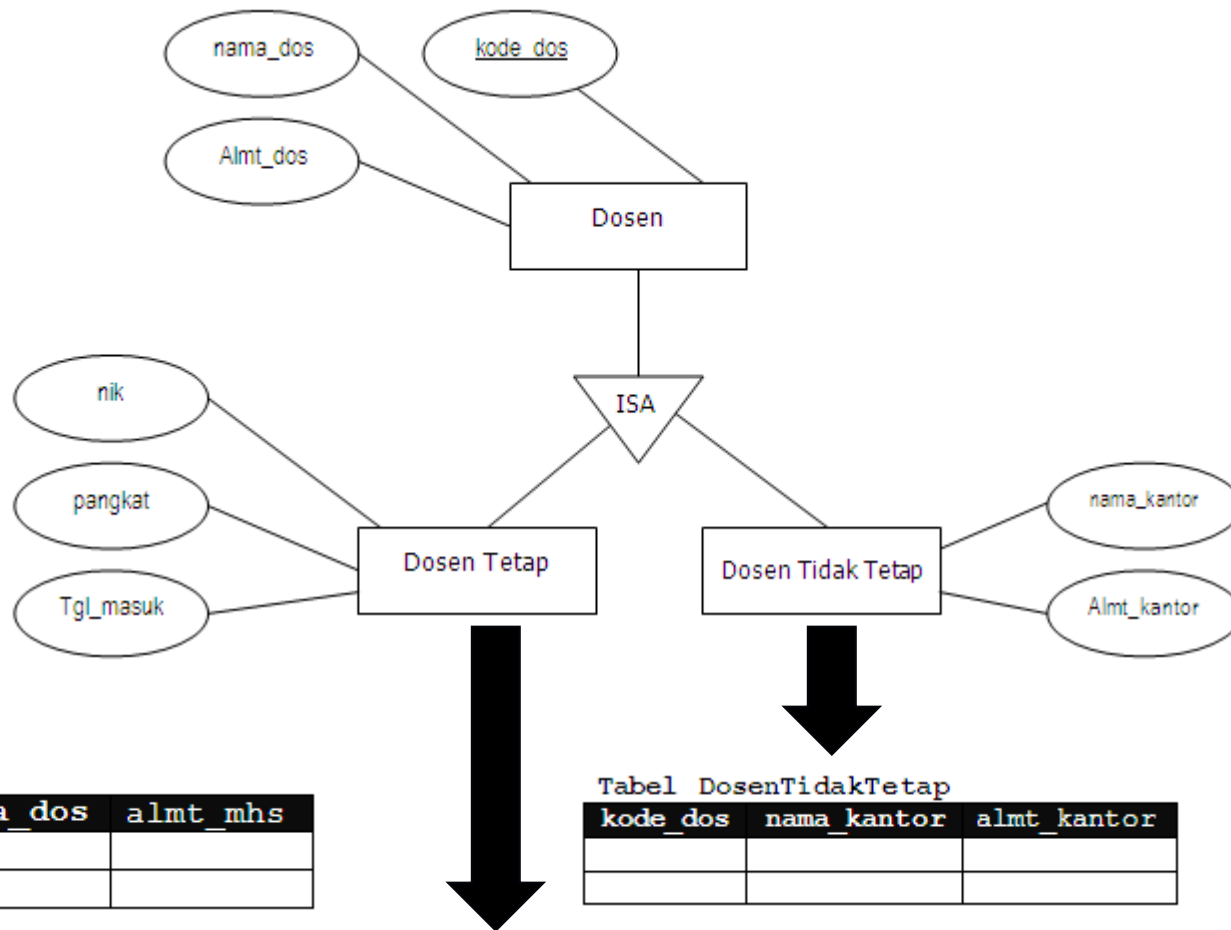
- **Implementasi Himpunan Entitas Lemah dan Sub Entitas**

Penggunaan Himpunan Entitas Lemah (*Weak Entity Sets*) dan Sub Entitas dalam diagram E-R diimplementasikan dalam bentuk tabel sebagaimana Himpunan Entitas Kuat (*Strong Entity Sets*). Bedanya jika Himpunan Entitas Kuat sudah dapat langsung menjadi sebuah tabel utuh/sepurna walaupun tanpa melihat relasinya dengan himpunan entitas yang lain, sedangkan Himpunan Entitas Lemah dan Sub Entitas hanya dapat ditransformasikan menjadi sebuah tabel dengan menyertakan pula atribut key yang ada di himpunan entitas kuat yang berelasi dengannya.



# • Implementasi Himpunan Entitas Lemah dan Sub Entitas





Tabel Dosen

kode_dos	nama_dos	almt_mhs

Tabel DosenTidakTetap

kode_dos	nama_kantor	almt_kantor

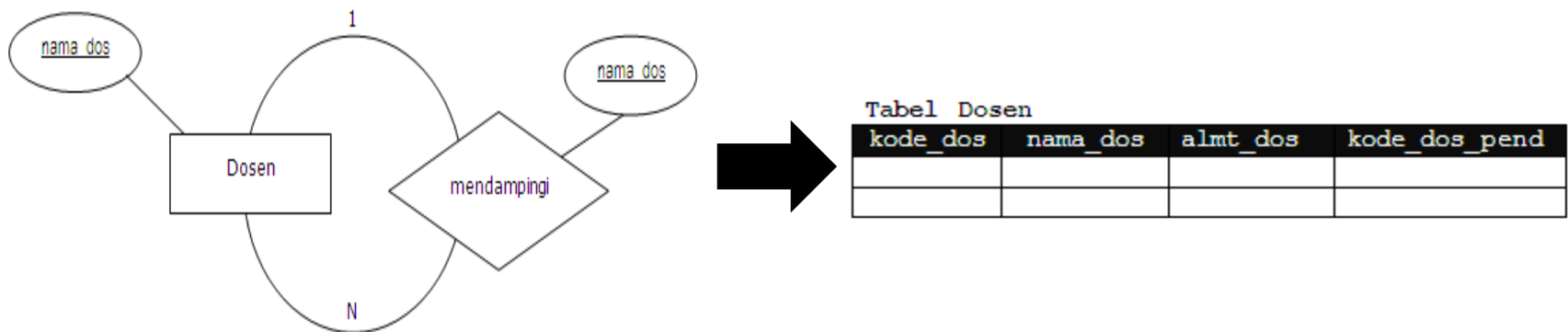
Tabel DosenTetap

kode_dos	nik	pangkat	tgl_masuk

## • Implementasi Relasi Tunggal (*Unary Relation*)

Implementasi Relasi Tunggal (*Unary Relation*). Dari/ke himpunan entitas yang sama dalam Diagram E-R tergantung pada Derajat Relasinya. Untuk relasi satu-ke-banyak dapat diimplementasikan melalui penggunaan field key dua kali tapi untuk fungsi yang berbeda.

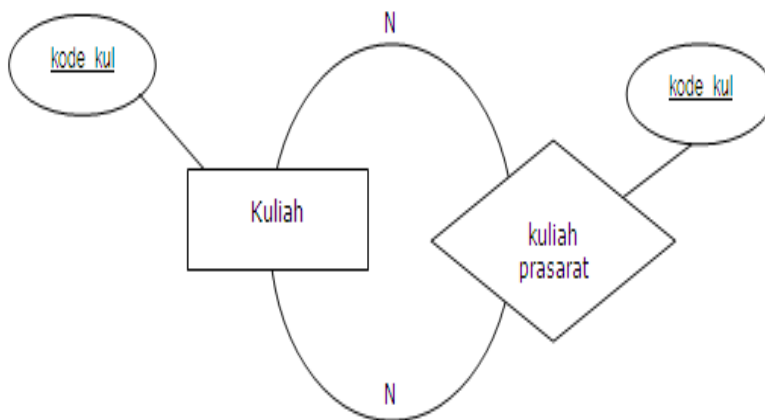
Jika kita memiliki himpunan entitas A dengan 2 atribut x dan y dengan x sebagai key, maka relasi tunggal terhadap himpunan entitas tersebut diwujudkan dengan menambahkan kembali field x ke tabel A. Karena nama field di setiap tabel harus unik, maka field x yang kedua harus diganti namanya sesuai dengan fungsinya/ relasinya.





## • Implementasi Relasi Tunggal (*Unary Relation*)

Sedang relasi yang derajatnya banyak-ke-banyak akan diimplementasikan melalui pembentukan tabel baru yang merepresentasikan relasi tersebut. Tabel baru ini mendapatkan field dari semua atribut relasi (jika ada) yang ditambah dengan atribut key dari himpunan entitasnya.



Tabel Kuliah

kode_kul	nama_kul	sks	semester

Tabel Prasyarat Kuliah

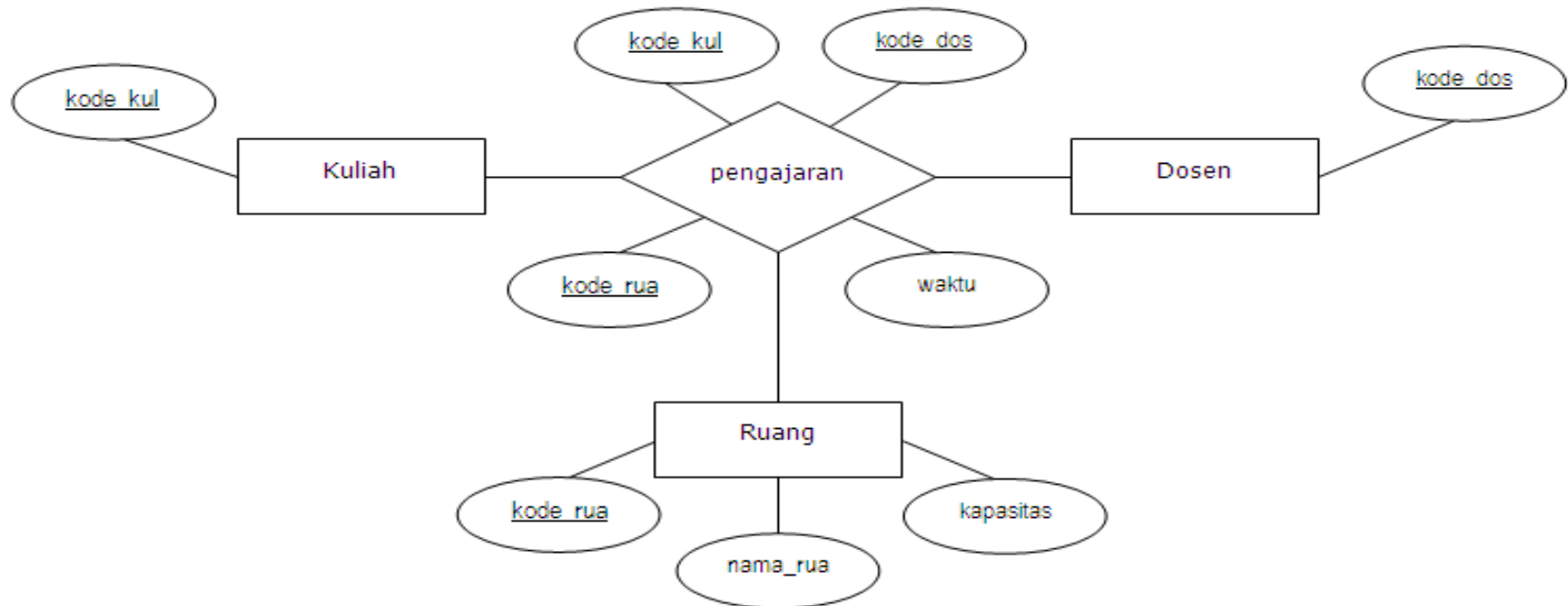
kode_kul	kode_kul_prasyarat

- **Implementasi Relasi Multi Entitas (*N-ary Relation*)**

Secara umum, relasi multi entitas yang menghubungkan lebih dari dua himpunan entitas ( $N$  himpunan entitas, di mana  $N > 2$ ) akan diimplementasikan sebagai sebuah tabel khusus (tentu saja, setiap himpunan entitas yang terlibat dalam relasi juga akan direpresentasikan dalam tabel-tabel terpisah).

Namun jika pada relasi yang menghubungkan  $N$  buah himpunan entitas kita dapat memastikan bahwa Derajat Relasi parsial di antara  $(N-1)$  buah himpunan entitas dengan suatu himpunan entitas (misalnya  $X$ ) adalah satu-ke-banyak, maka relasi tadi tidak perlu diwujudkan sebagai sebuah tabel khusus dan atribut-atributnya cukup dilekatkan pada himpunan entitas  $X$  tersebut.

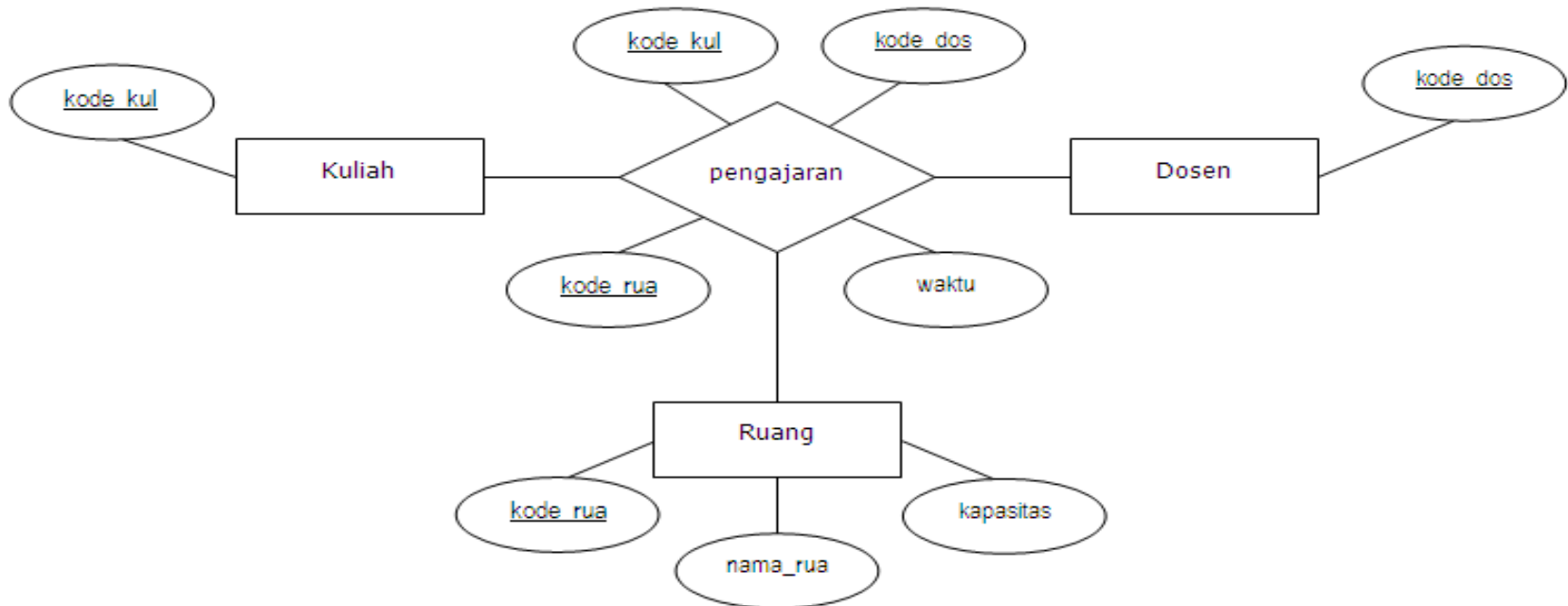
- Implementasi Relasi Multi Entitas (*N-ary Relation*)



Pada Diagram E-R diatas, Derajat Relasi parsial di antara setiap pasang himpunan entitas yang ada adalah sebagai berikut:

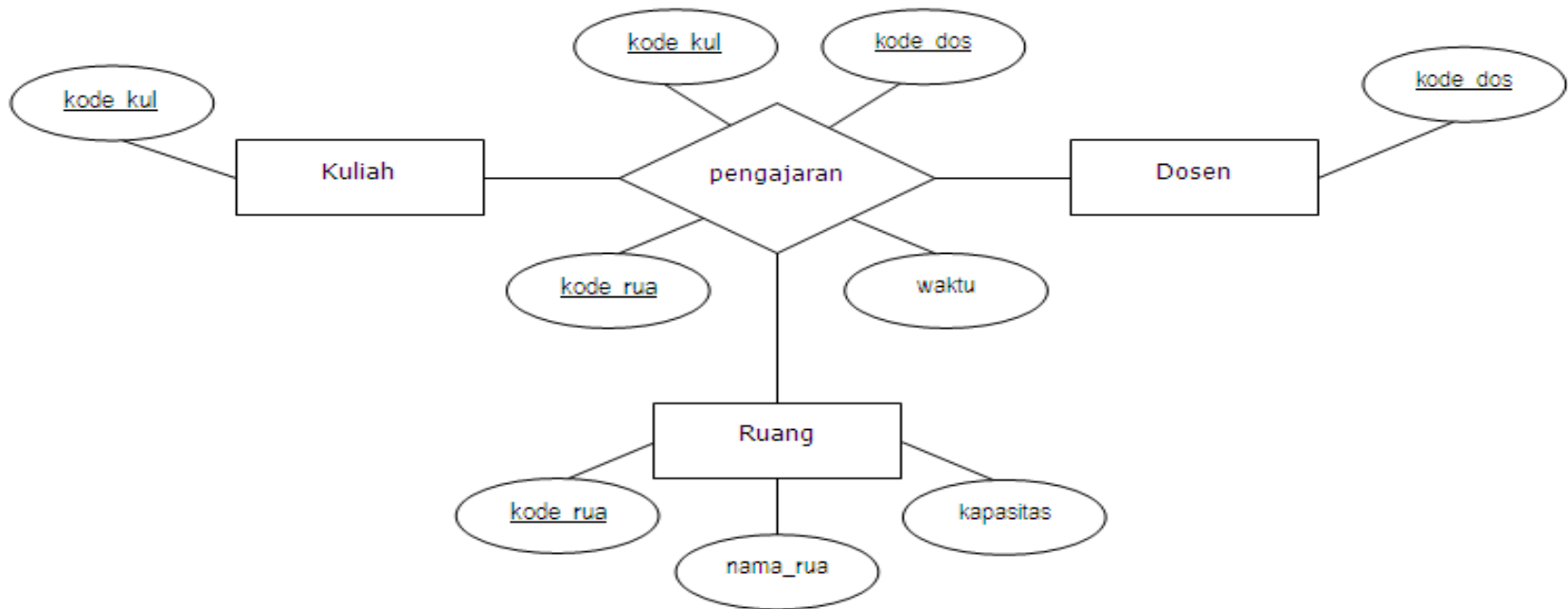
- Pada relasi pengajaran tersebut setiap mata kuliah dapat diajarkan oleh seorang dosen dan setiap dosen dapat melakukan pengajaran banyak mata kuliah, maka Derajat Relasi parsial antara himpunan entitas Dosen–Kuliah adalah 1-N (satu-ke-banyak).

- Implementasi Relasi Multi Entitas (*N-ary Relation*)



- Pada relasi pengajaran tersebut setiap mata kuliah hanya dapat diselenggarakan di sebuah ruang yang telah ditentukan dan setiap ruang pada saat yang berbeda dapat digunakan untuk pengajaran berbagai mata kuliah, maka Derajat Relasi parsial antara himpunan entitas Ruang-Kuliah adalah 1-N (satu-ke-banyak).

- Implementasi Relasi Multi Entitas (*N-ary Relation*)



- Pada relasi pengajaran tersebut setiap ruangan dapat digunakan oleh banyak dosen (untuk mengajarkan berbagai mata kuliah) dan setiap dosen dapat menggunakan berbagai ruangan karena memang mengajarkan lebih dari satu matakuliah, maka Derajat Relasi parsial antara himpunan entitas Ruang-Dosen adalah N-N (banyak-ke-banyak).

- **Implementasi Relasi Multi Entitas (*N-ary Relation*)**

Dari hasil pengamatan tersebut kita dapat menyetujui bahwa semua derajat relasi parsial antara himpunan entitas Dosen ataupun Ruang dengan himpunan entitas Kuliah selalu satu-ke-banyak. Dengan demikian, relasi Pengajaran tersebut tidak perlu diimplementasikan sebagai sebuah tabel khusus, tetapi atribut-atributnya dilekatkan pada tabel yang mewakili himpunan entitas kuliah.

Tabel Kuliah

kode_kul	nama_kul	sks	semester	kode_dos	tempat	waktu



3 buah field yang mewakili relasi  
Pengajaran

- **Implementasi Relasi Multi Entitas (*N-ary Relation*)**

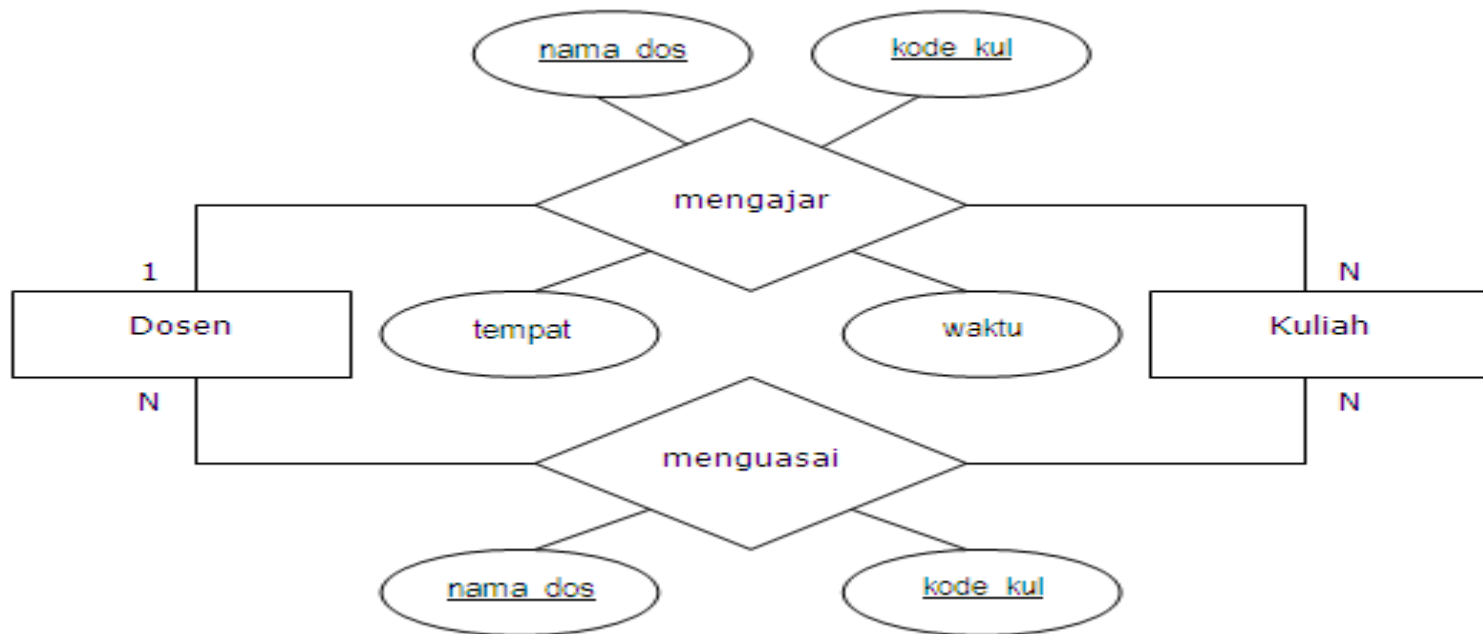
Jika ternyata di kemudian hari, suatu mata kuliah (dengan jumlah sks yang besar) dapat dilaksanakan lebih dari satu kali dalam seminggu, dan mungkin untuk diselenggarakan di ruang yang berbeda, maka Derajat Relasi parsial antara himpunan entitas Ruang-Kuliah bukan lagi satu-ke-banyak, tapi menjadi banyak-ke-banyak. Jika kenyataan ini harus diakomodasi, maka tabel Kuliah tetap sebagaimana bentuk semula (dengan 4 buah field: kode\_kul, nama\_kul, sks dan semester) dan relasi diatas harus diimplementasikan sebagai sebuah tabel khusus seperti berikut:

**Tabel Pengajaran/Jadwal**

kode_kul	kode_dos	kode_rua	waktu

- Implementasi Relasi Ganda (*Redundant Relation*)

Implementasinya ditinjau pada masing-masing relasi tanpa terikat satu sama lain berdasarkan Derajat Relasi di masing-masing relasi tersebut.



Karena derajat relasi mengajar adalah satu-ke-banyak, maka *field* *kode\_dos* yang berasal dari himpunan entitas Dosen ditambahkan ke tabel Kuliah. Sementara untuk relasi Menguasai, karena Derajat Relasinya adalah banyak-ke-banyak maka relasi akan dinyatakan dalam tabel khusus dengan 2 buah *field*: *kode\_dos* dan *kode\_kul*.





- Implementasi Relasi Ganda (*Redundant Relation*)

Hasil akhir implementasinya adalah:

Tabel Dosen

kode_dos	nama_dos	almt_mhs

Tabel Kuliah

kode_kul	nama_kul	sks	semester	kode_dos

Atribut tambahan untuk merepresentasikan relasi  
Mengajar

Tabel Menguasai

kode_dos	kode_kul

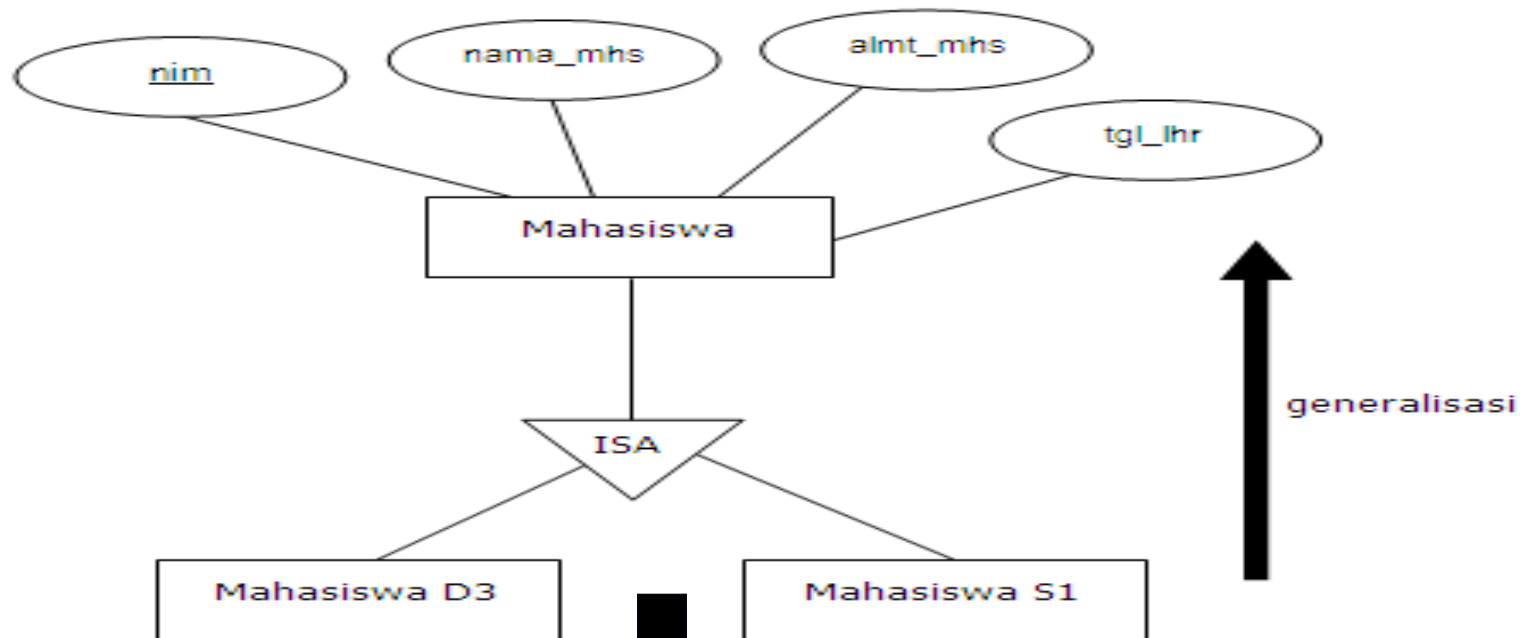
tabel khusus untuk  
merepresentasikan relasi menguasai

- **Implementasi Spesialisasi dan Generalisasi**

Spesialisasi terhadap sebuah himpunan entitas akan menghasilkan sejumlah himpunan entitas baru: satu himpunan entitas kuat/bebas yang akan menjadi acuan bagi himpunan entitas lainnya dan sisanya merupakan sub entitas.

Generalisasi dilakukan dengan ‘mengabaikan’ perbedaan beberapa himpunan entitas yang memang memiliki banyak kesamaan. Berlawanan dengan Spesialisai, pada tahap implementasi Generalisasi justru akan menyusutkan jumlah himpunan entitas menjadi hanya sebuah tabel saja. Untuk tetap mengakomodasi adanya perbedaan itu, maka di tabel tersbeut ditambahkan sebuah atribut yang nantinya akan diisi dengan kode khusus yang menyatakan perbedaan tersebut.

- Implementasi Spesialisasi dan Generalisasi



Tabel Mahasiswa

nim	nama_mhs	alamat_mhs	tgl_lhr	prog_studi

Atribut tambahan untuk mengakomodasi perbedaan kelompok entitas



Managed by BINA NUSANTARA