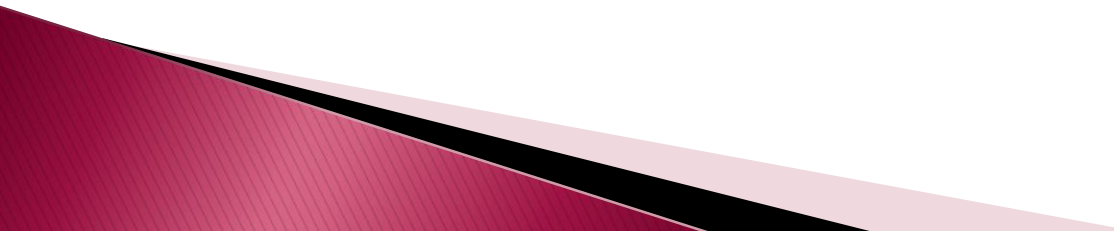


NORMALISASI DATA

Normalisasi

- ▶ Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi).
- ▶ *Normalisasi* adalah proses pembentukan struktur basis data

Tujuan Normalisasi

- ▶ Untuk menghilangkan kerangkapan data
 - ▶ Untuk mengurangi kompleksitas
 - ▶ Untuk mempermudah pemodifikasian data
- 

Proses Normalisasi

- ▶ Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- ▶ Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan Normalisasi

- ▶ Tahap Normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF)
- ▶ Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.
- ▶ Urutan: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF

Bentuk Tidak Normal



Menghilangkan perulangan group

Bentuk Normal Pertama (1 NF)



Menghilangkan Ketergantungan parsial

Bentuk Normal Kedua (2NF)



Menghilangkan Ketergantungan Transitif

Bentuk Normal Ketiga (3NF)



Menghilangkan anomali-anomali hasil dari

ketergantungan fungsional

Bentuk Normal Boyce-Codd (BCNF)



Menghilangkan ketergantungan Multivalue

Bentuk Normal Keempat (4NF)



Menghilangkan anomali-anomali yang tersisa

Bentuk Normal Kelima

Normalisasi

Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman. Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation).

Contoh 2

relasi tidak normal (Tabel Personil)

ID_Personil	Tanggal Lahir	Karakteristik
I102	17 Januari 1970	Tinggi 162
		Berat 50
		Rambut hitam
A212	12 Desember 1966	Tinggi 170
		Berat 64
		Rambut hitam

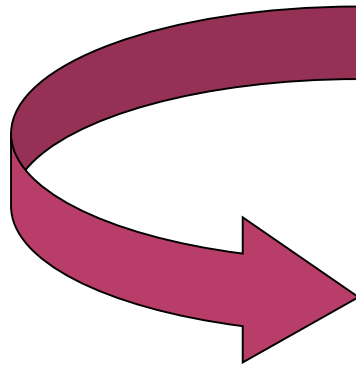
Relasi Normal (Tabel Personil)

ID Personil	Tanggal lahir	Tinggi	Berat	Warna Rambut
I102	17 Januari 1970	162	50	Hitam
A212	12 Desember 1966	170	64	Hitam

Tabel Universal

Tabel Universal (*Universal / Star Table*) → sebuah tabel yang merangkum semua kelompok data yang saling berhubungan, bukan merupakan tabel yang baik.

Misalnya:



Tabel Universal

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350 MI465	Manajemen Basis Data Analisis Prc. Sistem	B104 B317	Ati Dita	A B
5432	Bakri	AK	MI350 AKN201 MKT300	Manajemen Basis Data Akuntansi Keuangan Dasar Pemasaran	B104 D310 B212	Ati Lia Lola	C B A

Functional Dependency

- ▶ **Notasi: $A \rightarrow B$**

A dan B adalah atribut dari sebuah tabel. Berarti secara fungsional A menentukan B atau B tergantung pada A, jika dan hanya jika ada 2 baris data dengan nilai A yang sama, maka nilai B juga sama

Functional Dependency

Contoh tabel nilai

<u>Namaku</u>	<u>Nrp</u>	<u>namaMhs</u>	<u>NiHuruf</u>
<u>Struktur Data</u>	980001	<u>Ali Akbar</u>	A
<u>Struktur Data</u>	980004	<u>Indah Susanti</u>	B
<u>Basis Data</u>	980001	<u>Ali Akbar</u>	
<u>Basis Data</u>	980002	<u>Budi Haryanto</u>	
<u>Basis Data</u>	980004	<u>Indah Susanti</u>	
<u>Bahasa Indonesia</u>	980001	<u>Ali Akbar</u>	B
<u>Matematika I</u>	980002	<u>Budi Haryanto</u>	C

Functional Dependency

Functional Dependency dari tabel nilai

➤ **Nrp** → **namaMhs**

Karena untuk setiap nilai nrp yang sama, maka nilai namaMhs juga sama.

➤ **{NamaKul, nrp}** → **NiHuruf**

Karena atribut Nihuruf tergantung pada NamaKul dan nrp secara bersama-sama. Dalam arti lain untuk NamaKul dan nrp yang sama, maka NiHuruf juga sama, karena NamaKul dan nrp merupakan key (bersifat unik).

➤ **NamaKul** ↛ **nrp**

➤ **Nrp** ↛ **NiHuruf**

Contoh FD 1

- ▶ Andaikan ada tabel:
NILAI (NIM, Nm-mk, Semester, Nilai)
- ▶ Atribut kunci: NIM, Nm-mk, Semester
- ▶ Maka Functional Dependency:
NIM, Nm-mk, Semester \rightarrow Nilai

Bentuk-bentuk Normal

1. Bentuk Normal Tahap Pertama (1st Normal Form / 1NF)
2. Bentuk Normal Tahap Kedua (2nd Normal Form / 2NF)
3. Bentuk Normal Tahap (3rd Normal Form / 3NF)
4. Boyce–Code Normal Form (BCNF)
5. Bentuk Normal Tahap (4th Normal Form / 4NF)
6. Bentuk Normal Tahap (5th Normal Form / 5NF)

Normal Pertama (1st Normal Form)

- ▶ Aturan :
- ✓ Tidak adanya **atribut multi-value**, **atribut komposit** atau kombinasinya.
- ✓ Mendefinisikan atribut kunci.
- ✓ Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi)

Contoh 1 (atribut multi-value)

Misal data mahasiswa sbb:

<u>Nrp</u>	<u>nama</u>	<u>Hobi</u>
12020001	Heri Susanto	Sepakbola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, mrogram komputer
12020015	Dini Susanti	Menjahit, membuat roti

Atau:

<u>Nrp</u>	<u>nama</u>	<u>hobi1</u>	<u>hobi2</u>	<u>Hobi3</u>
12020001	Heri Susanto	Sepak Bola	Membaca komik	berenang
12020013	Siti Zulaiha	Memasak	mrogram komputer	
12020015	Dini Susanti	Menjahit	membuat kue	

Tabel-tabel di atas tidak memenuhi syarat 1NF

Contoh 1 (samb...)

Didekomposisi menjadi:

➤ Tabel Mahasiswa

<u>Nrp</u>	<u>Nama</u>
12020001	Heri Susanto
12020013	Siti Zulaiha
12020015	Dini Susanti

➤ Tabel Hobi

<u>Nrp</u>	<u>Hobi</u>
12020001	Sepakbola
12020001	membaca komik
12020001	Berenang
12020013	Memasak
12020013	mrogram komputer
12020015	Menjahit
12020015	membuat roti

Contoh 2 (composite)

JadwaKuliah

Kodekul	NamaKul	Dosen	Kelas	Jadwal
---------	---------	-------	-------	--------

- Dimana nilai pada atribut jadwal berisi gabungan antara Hari dan Jam.
- Jika asumsi hari dan jam memegang peranan penting dalam sistem basis data, maka atribut Jadwal perlu dipisah sehingga menjadi JadwalHari dan JadwalJam sbb:

JadwaKuliah

Kodekul	NamaKul	Dosen	Kelas	JadwalHari	JadwalJam
---------	---------	-------	-------	------------	-----------

Normalisasi Kedua (2nd Normal Form)

- ▶ Aturan :
- ✓ Sudah memenuhi dalam bentuk normal kesatu (1NF)
- ✓ Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci
- ✓ Jika ada **ketergantungan parsial** maka atribut tersebut harus dipisah pada tabel yang lain
- ✓ Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut–atribut yang telah dipisah tadi

Contoh

Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

Mhs_nrp	mhs_nama	mhs_alamat	mk_kode	mk_nama	mk_sks	nihuruf
---------	----------	------------	---------	---------	--------	---------

- Tidak memenuhi 2NF, karena {Mhs_nrp, mk_kode} yang dianggap sebagai primary key sedangkan:

{Mhs_nrp, mk_kode} \nrightarrow mhs_nama
{Mhs_nrp, mk_kode} \nrightarrow mhs_alamat
{Mhs_nrp, mk_kode} \nrightarrow mk_nama
{Mhs_nrp, mk_kode} \nrightarrow mk_sks
{Mhs_nrp, mk_kode} \rightarrow nihuruf

- Tabel di atas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF

Normalisasi Ketiga (3rd Normal Form)

- ▶ Aturan :
 - ✓ Sudah berada dalam bentuk normal kedua (2NF)
 - ✓ Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

Contoh

Tabel berikut memenuhi 2NF, tapi tidak memenuhi 3NF:

Mahasiswa

Nrp	Nama	Alm_Jalan	Alm_Kota	Alm_Provinsi	Alm_Kodepos
------------	-------------	------------------	-----------------	---------------------	--------------------

- karena masih terdapat atribut *non primary key* (yakni **alm_kota** dan **alm_Provinsi**) yang memiliki ketergantungan terhadap atribut *non primary key* yang lain (yakni **alm_kodepos**):

alm_kodepos → {alm_Provinsi, alm_kota}

- Sehingga tabel tersebut perlu didekomposisi menjadi:

Mahasiswa (Nrp, nama, alm_jalan, alm_kodepos)

Kodepos (alm_kodepos, alm_provinsi, alm_kota)

Tabel-tabel yang memenuhi kriteria normalisasi ketiga, sudah siap diimplementasikan. Sebenarnya masih ada lagi bentuk normalisasi yang lain; Normalisasi Boyce-Codd, 4NF, 5NF, hanya saja jarang dipakai. Pada kebanyakan kasus, normalisasi hanya sampai ketiga.

Boyce–Codd Normal Form (BCNF)

- ▶ Bentuk BCNF terpenuhi dalam sebuah tabel, jika untuk setiap *functional dependency* terhadap setiap atribut atau gabungan atribut dalam bentuk:
 $X \rightarrow Y$ maka X adalah *super key*
- ▶ tabel tersebut harus di-dekomposisi berdasarkan *functional dependency* yang ada, sehingga X menjadi *super key* dari tabel–tabel hasil dekomposisi
- ▶ Setiap tabel dalam BCNF merupakan 3NF. Akan tetapi setiap 3NF belum tentu termasuk BCNF . Perbedaannya, untuk *functional dependency* $X \rightarrow A$, BCNF tidak membolehkan A sebagai bagian dari *primary key*.

Bentuk Normal Tahap Keempat (4th Normal Form /4NF)

- ▶ Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*
- ▶ Untuk setiap *multivalued dependencies* (MVD) juga harus merupakan *functional dependencies*

Contoh

Misal, tabel berikut tidak memenuhi 4NF:

Employee	Project	Skill
Jim	11	Program
Mary	5	Design
Mary	NULL	Analysis

Setiap employee dapat bekerja di lebih dari project dan dapat memiliki lebih dari satu skill. Untuk kasus seperti ini tabel tersebut harus di-dekomposisi menjadi:

(Employee, Project)

(Employee, Skill)

Bentuk Normal Tahap Keempat (5th Normal Form / 5NF)

- ▶ Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel–tabel yg lebih kecil.
- ▶ Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence*. Yakni apabila sebuah tabel telah di–dekomposisi menjadi tabel–tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula

Studi Kasus Normalisasi Data 1

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama.

Field-field tabel di atas yang merupakan group berulang : NoPegawai, NamaPegawai, Golongan, BesarGaji.

Normalisasi pertama

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial. menjadikan field-field menjadi tergantung pada satu atau beberapa field. Karena yang dapat dijadikan kunci adalah *NoProyek* dan *NoPegawai*, maka langkah kemudian dicari field-field mana yang tergantung pada *NoProyek* dan mana yang tergantung pada *NoPegawai*.

<i>Noproyek</i>	NamaProyek	<i>Nopegawai</i>	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
NP001	BRR	Peg02	Paula	B	900.000
NP001	BRR	Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
NP002	PEMDA	Peg12	Sita	B	900.000
NP002	PEMDA	Peg14	Yusni	B	900.000

Normalisasi Kedua

- ▶ Field-field yang tergantung pada satu field haruslah dipisah dengan tepat, misalnya *NoProyek* menjelaskan *NamaProyek* dan *NoPegawai* menjelaskan *NamaPegawai*, *Golongan* dan *BesarGaji*.

Normalisasi Kedua

TABEL PROYEK

<i>Noprojek</i>	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PEGAWAI

<i>Nopegawai</i>	NamaPegawai	Golongan	BesarGaji
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	<u>Koko</u>	C	750.000
Peg12	<u>Sita</u>	B	900.000
Peg14	<u>Yusni</u>	B	900.000

Untuk membuat hubungan antara dua tabel, dibuat suatu tabel yang berisi key-key dari tabel yang lain.

TABEL PROYEKPEGAWAI

<i>Noprojek</i>	<i>NoPegawai</i>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

Normalisasi Ketiga

Pada tabel diatas masih terdapat masalah, bahwa *BesarGaji* tergantung kepada *Golongan* nya. Padahal disini *Golongan* bukan merupakan field kunci.

Artinya kita harus memisahkan field non-kunci *Golongan* dan *BesarGaji* yang tadinya tergantung secara parsial kepada field kunci *NoPegawai*, untuk menghilangkan ketergantungan transitif.

TABEL PROYEK

<i>Noprojek</i>	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PROYEKPEGAWAI

<i>Noprojek</i>	<i>NoPegawai</i>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

TABEL PEGAWAI

<i>Nopegawai</i>	NamaPegawai	Golongan
Peg01	Anton	A
Peg02	Paula	B
Peg06	<u>Koko</u>	C
Peg12	<u>Sita</u>	B
Peg14	<u>Yusni</u>	B

TABEL GOLONGAN

Golongan	BesarGaji
A	1.000.000
B	900.000
C	750.000

Studi Kasus Normalisasi 2

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350 MI465	Manajemen Basis Data Analisis Prc. Sistem	B104 B317	Ati Dita	A B
5432	Bakri	AK	MI350 AKN201 MKT300	Manajemen Basis Data Akuntansi Keuangan Dasar Pemasaran	B104 D310 B212	Ati Lia Lola	C B A

TNF

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350	Manajemen Basis Data	B104	Ati	A
2683	Welli	MI	MI465	Analisis Prc. Sistem	B317	Dita	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
5432	Bakri	AK	AKN201	Akuntansi Keuangan	D310	Lia	B
5432	Bakri	AK	MKT300	Dasar Pemasaran	B212	Lola	A

2NF

<u>No-Mhs</u>	Nama-Mhs	Jurusan
2683	Welli	MI
5432	Bakri	AK

Tabel Mahasiswa

<u>Kode-MK</u>	Nama-MK	Kode-Dosen	Nama-Dosen
MI350	Manajemen Basis Data	B104	Ati
MI465	Analisis Proc. Sistem	B317	Dita
AKN201	Akuntansi Keuangan	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Tabel Kuliah

<u>No-Mhs</u>	<u>Kode MK</u>	Nilai
2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

Tabel Nilai

3NF

<u>Kode-MK</u>	Nama-MK	Kode-Dosen
MI350	Manajemen Basis Data	B104
MI465	Analisis Prc. Sistem	B317
AKN201	Akuntansi Keuangan	D310
MKT300	Dasar Pemasaran	B212

Tabel Mata Kuliah

<u>Kode-Dosen</u>	Nama-Dosen
B104	Ati
B317	Dita
D310	Lia
B212	Lola

Tabel Dosen

ENTITY RELATIONSHIP DIAGRAM

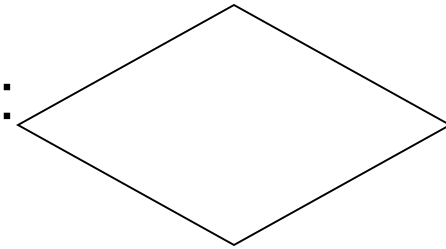
- ▶ **Entity Relationship Diagram** merupakan jaringan yang menggunakan susunan data yang disimpan dari system secara abstrak. Diagram Entity Relationship ini ditemukan oleh Chen tahun 1976.
- ▶ **Tujuan dari Entity Relationship** adalah untuk menunjukkan objek data dan relationship yang ada pada objek tersebut. Disamping itu Model ER ini merupakan salah satu alat untuk perancangan dalam basis data.

KOMPONEN (SIMBOL) ERD

RELATIONSHIP

Adalah hubungan yang terjadi antara satu entity dengan entity lainnya. Relationship tidak mempunyai keberadaan fisik atau konseptual kecuali yang sejenis dinamakan dengan Relationship Diagram.

Simbol dari Relationship adalah :



KOMPONEN (SIMBOL) ERD

RELATIONSHIP

Contoh penggunaan simbol relationship:



KETERANGAN

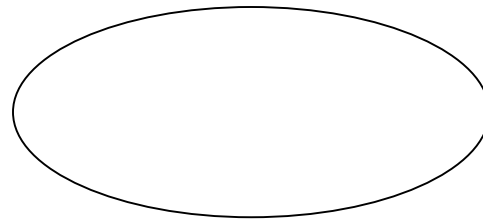
Memiliki adalah relationship set yang terbentuk antara entity pegawai dengan entity kendaraan

KOMPONEN (SIMBOL) ERD

ATRIBUT

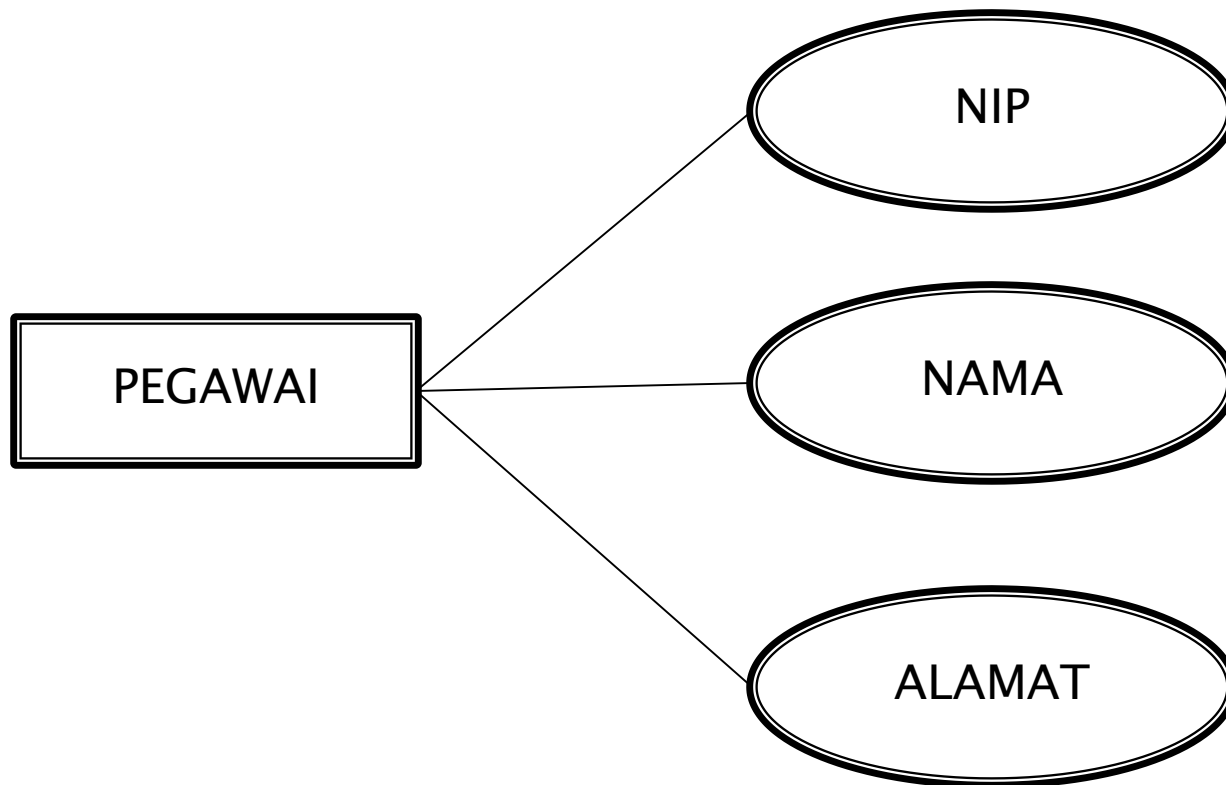
Adalah karakteristik dari entity atau relationship yang menyediakan penjelasan detail tentang entity atau relationship tersebut.

Simbol dari atribut adalah:



KOMPONEN (SIMBOL) ERD

Contoh penggunaan simbol atribut:

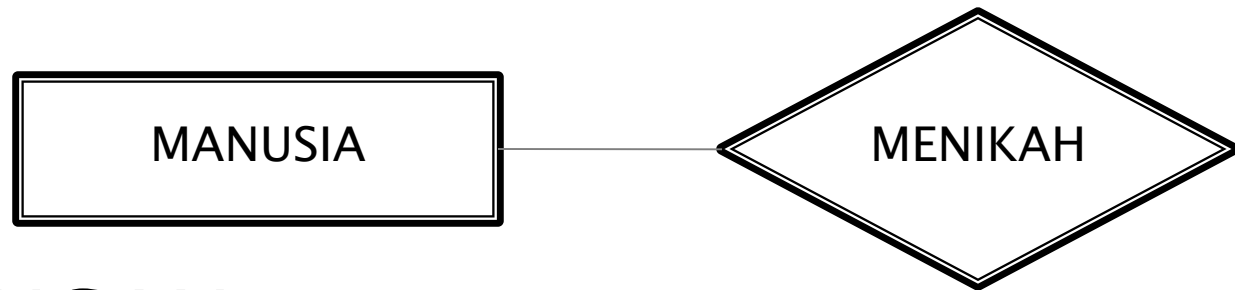


DERAJAT RELATIONSHIP

UNARY

Adalah satu buah relationship menghubungkan satu buah entity

Contoh:



KETERANGAN

Manusia menikah dengan manusia, relationship menikah hanya menghubungkan entity manusia.

DERAJAT RELATIONSHIP

BINARY (DERAJAT DUA)

Adalah satu buah relationship yang menghubungkan dua buah entity

Contoh:



KETERANGAN

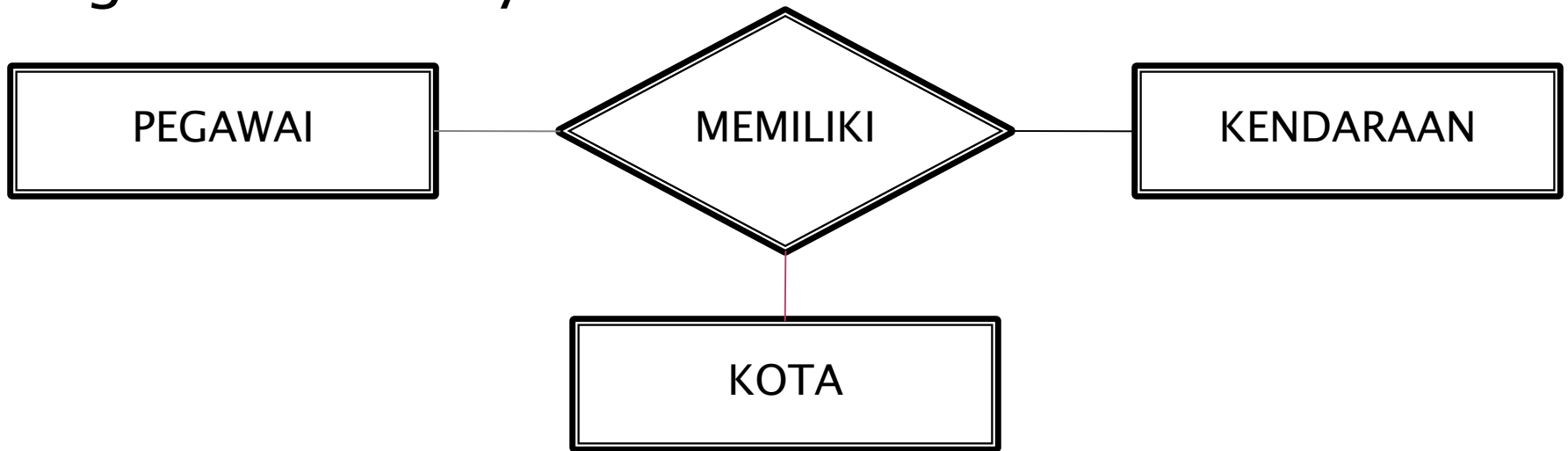
Pegawai memiliki kendaraan, sebuah relationship memiliki mengubungkan entity

Pegawai dan entity Kendaraan.

DERAJAT RELATIONSHIP

TERNARY (DERAJAT TIGA)

Adalah satu buah relationship menghubungkan tiga buah entity.



KETERANGAN

Pegawai pada kota tertentu mempunyai suatu proyek

Entity bekerja menghubungkan entity pegawai, proyek dan kota

CARDINALITY RATIO

Yaitu menjelaskan batasan pada jumlah entity yang berhubungan melalui suatu relationship.

CARDINALITY RATIO

ONE TO ONE (1:1), yaitu perbandingan antara entity pertama dengan entity kedua berbanding satu berbanding satu.



CARDINALITY RATIO

ONE TO MANY (1:M), Yaitu perbandingan antara entity pertama dengan entity kedua berbanding satu berbanding banyak



CARDINALITY RATIO

MANY TO ONE (M:1), Yaitu perbandingan antara entity pertama dengan entity kedua berbanding banyak berbanding satu.



CARDINALITY RATION

MANY TO MANY (M:M), Yaitu perbandingan antara entity pertama dengan entity kedua berbanding banyak berbanding banyak.



LANGKAH-LANGKAH MEMBUAT ERD

- ▶ Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
- ▶ Menentukan atribut-atribut key dari masing-masing himpunan entitas.
- ▶ Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key-nya.
- ▶ Menentukan derajat dan cardinality rasio relasi untuk setiap himpunan relasi
- ▶ Melengkapi himpunan relasi dengan atribut-atribut yang bukan kunci (non-key).

CONTOH STUDY KASUS

Suatu perguruan tinggi mempunyai banyak mahasiswa. Setiap mahasiswa tidak harus mengikuti suatu mata kuliah. Setiap biasanya mengikuti beberapa mata kuliah. Suatu mata kuliah diajarkan oleh seorang Dosen dan seorang Dosen bisa mengajar beberapa mata kuliah. Dan seorang Dosen harus mengajarkan suatu mata kuliah. Pada Entitas Mahasiswa diperlukan informasi tentang NIM, Nama_Mhs, Alamat_Mhs dan Jurusan, sedangkan Mata Kuliah diperlukan informasi tentang Kd_MK, Nm_Mk, SKS, Semester, sedangkan Dosen diperlukan juga informasi tentang Kd_Dosen, Nama_Dosen.

Pertanyaan :

Buatlah ERD-nya !

JAWABAN STUDY KASUS

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.

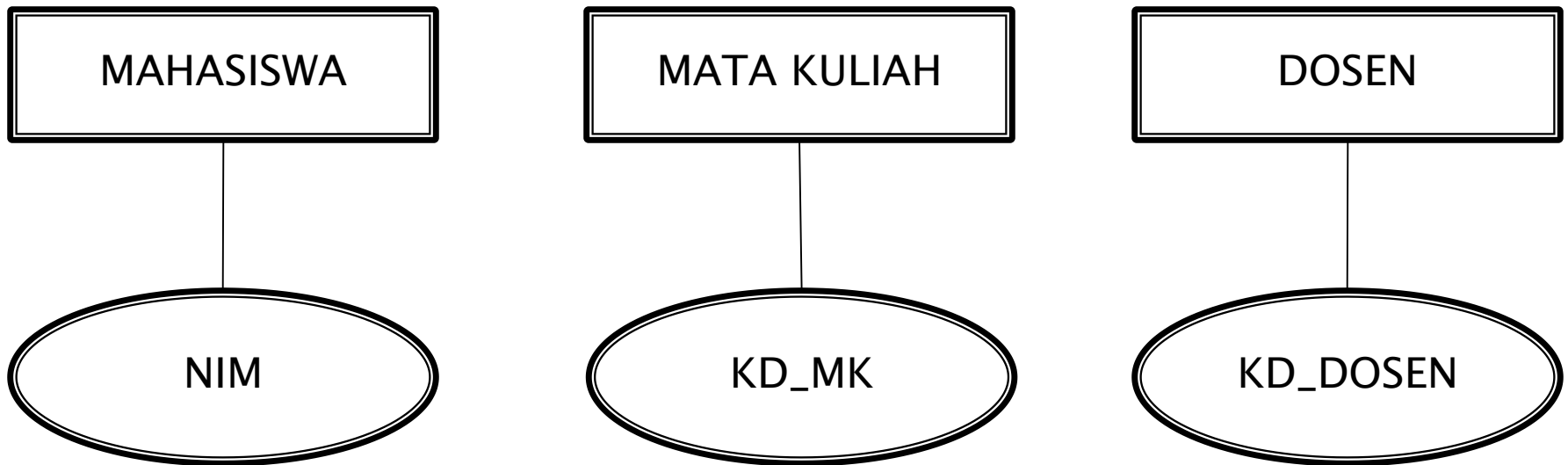
MAHASISWA

MATA KULIAH

DOSEN

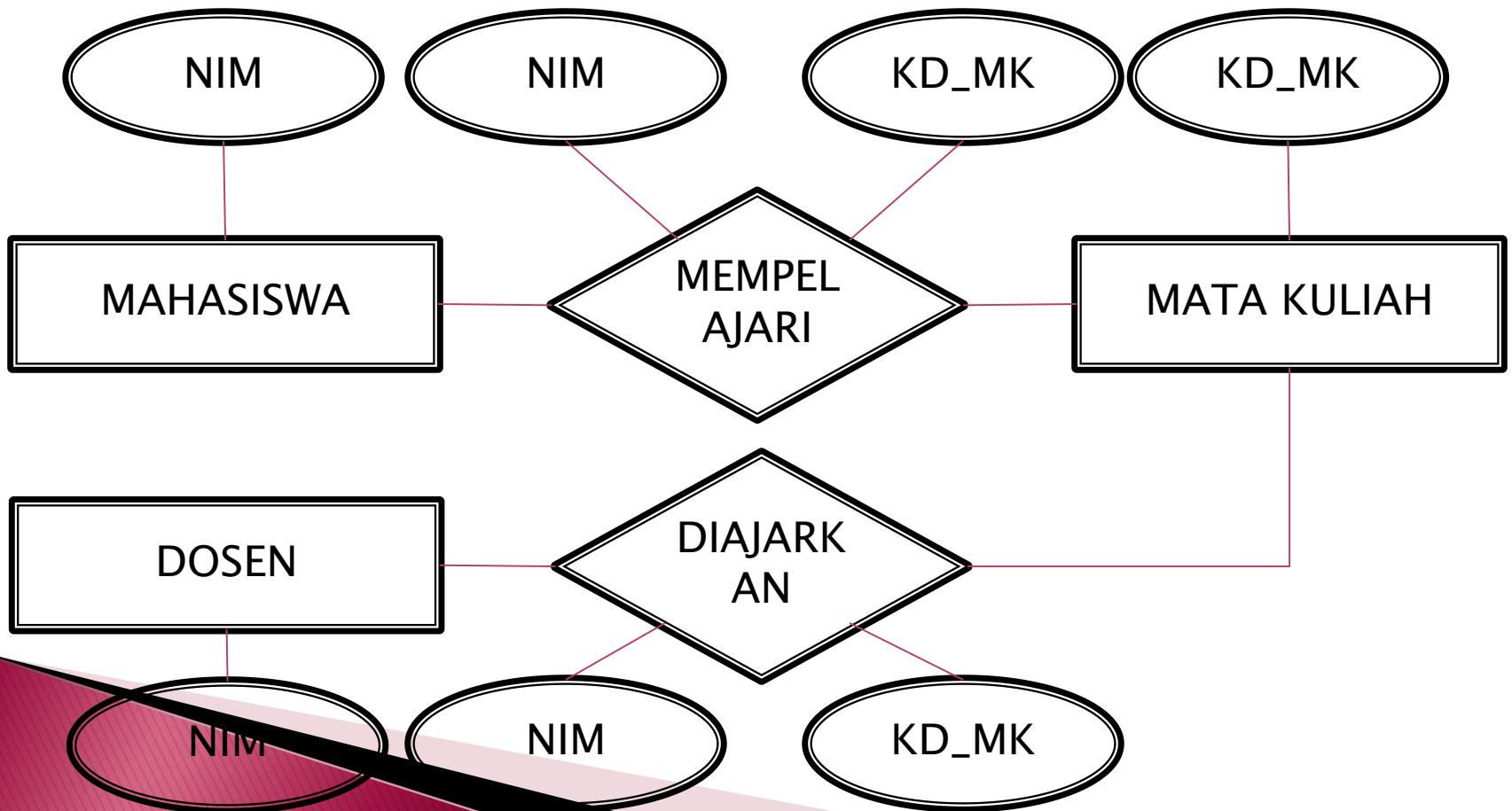
JAWABAN STUDY KASUS

2. Menentukan atribut-atribut key dari masing-masing himpunan entitas



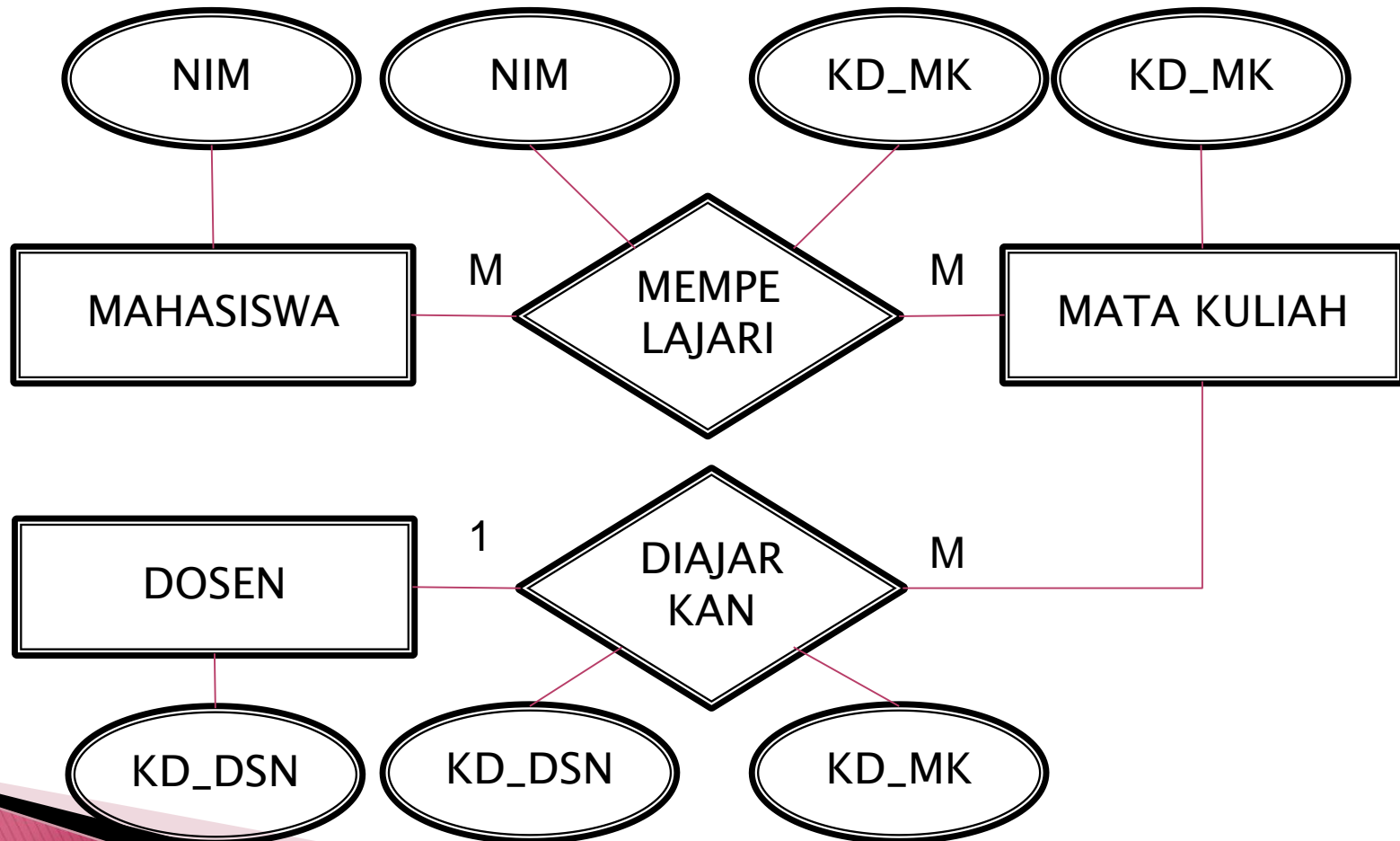
JAWABAN STUDY KASUS

3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key-nya.



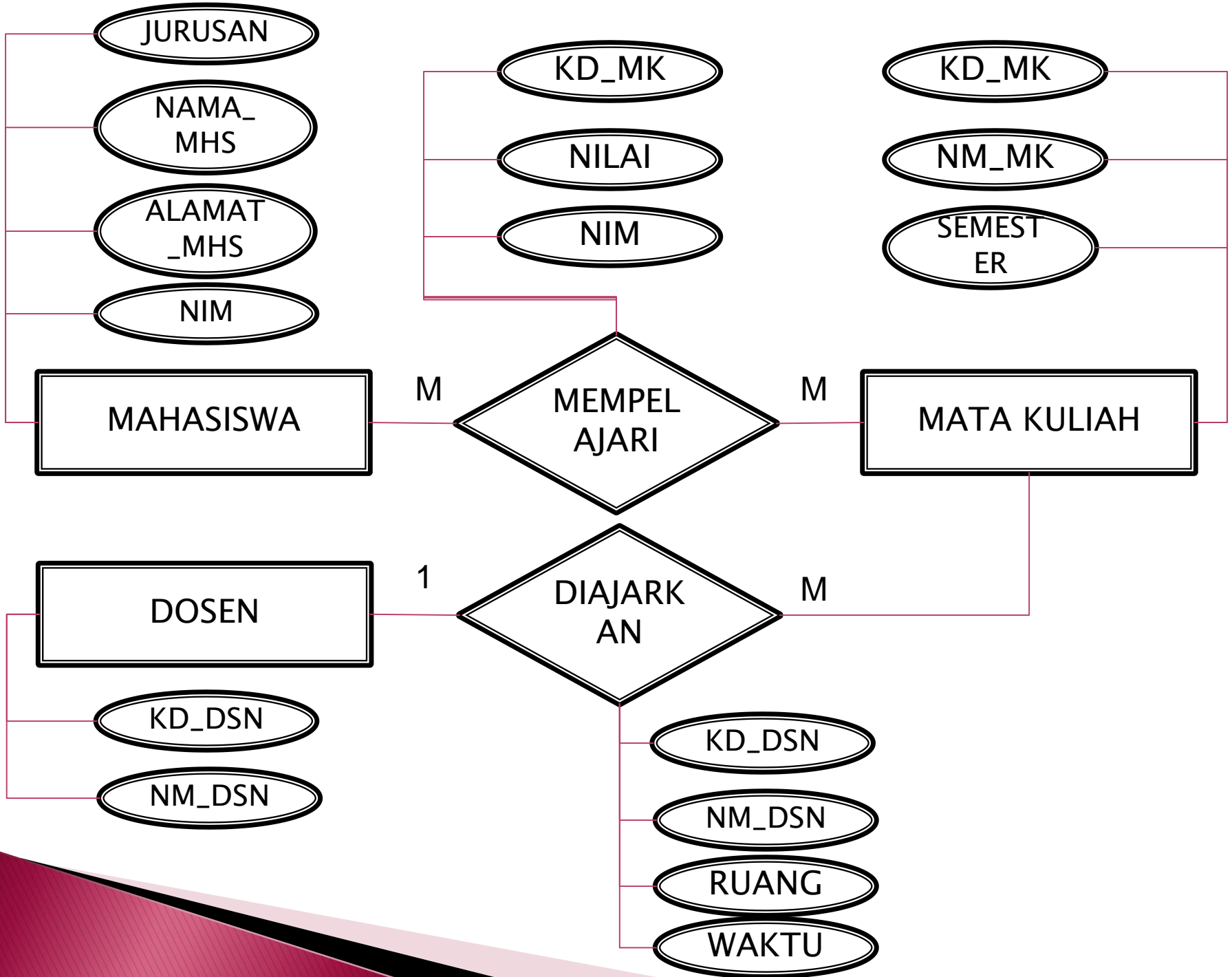
JAWABAN STUDY KASUS

4. Menentukan derajat cardinality rasio relasi untuk himpunan relasi



CONTOH STUDY KASUS

5. Melengkapi himpunan relasi dengan atribut-atribut yang bukan kunci (non-key)



TUGAS 5

Suatu klinik memiliki praktek Dokter bersama sehingga dalam klinik tersebut memiliki banyak Dokter. Seorang Pasien, apabila akan berobat harus diperiksa oleh Dokter dan sebaliknya Dokter pun harus memeriksa Pasien. Pasien yang berobat pada klinik tersebut lebih dari seorang. Setiap selesai diperiksa pasien biasanya menerima resep berupa obat dan biasanya setiap pasien menerima beberapa jenis obat.

Informasi tentang pasien adalah nomor pasien, nama pasien dan alamat. Informasi tentang obat adalah kode obat, nama obat, dan dosis.

Pertanyaan :

Buatlah ERD-nya !