



KONSEP BASISDATA

Kholid Fathoni, S.Kom., M.T.

Lessons

1. Pengertian Database

2. Sistem Pengorganisasian Database
3. Abstraksi Data
4. Bahasa Basis Data
5. Istilah dalam Database



Pengertian Database

- Hampir semua aplikasi komputer didukung fasilitas database.
- Database merupakan komponen terpenting dalam pembangunan SI, sebagai tempat menampung dan mengorganisasi seluruh data yang ada dalam sistem.
- Database merupakan himpunan sekelompok data yang saling berkaitan, yang diolah dan diorganisasikan sehingga menjadi sebuah informasi yang berguna.

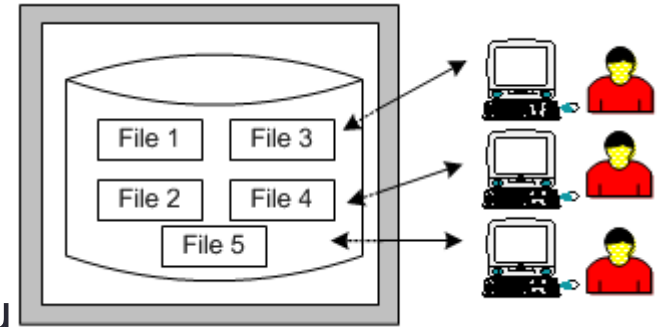
Pengertian Database

Basis Data:

- Lemari arsip
- Penyimpanan data

Basis Data:

- **Basis:** markas/gudang, tempat bersarang/berku
- **Data :** representasi fakta dunia nyata yang mewakili suatu obyek, spt: manusia (pegawai, siswa, pelanggan, dll), barang, hewan, peristiwa, konsep, dsb. Yang direkam dalam bentuk angka, huruf, teks, gambar atau suara suara.



Basis Data:

- Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama tanpa adanya pengulangan (redudansi) data.
- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Pengertian Database

Prinsip kerja Basis Data:

- Pengaturan data / arsip

Tujuan Basis Data:

- Kemudahan dan kecepatan dalam pengambilan data (speed)
- Efisiensi ruang penyimpanan (space)
Mengurangi / menghilangkan redundansi (pengulangan) data
- Keakuratan (Accuracy)
Pembentukan kode & relasi antar data berdasar aturan / batasan (constraint) tipe data, domain data, keunikan data, untuk menekan ketidakakuratan saat entry / penyimpanan data.
- Ketersediaan (Availability)
Pemilahan data yang sifatnya pasif dari database aktif.
- Kelengkapan (Completeness)
Kompleksnya data menyebabkan perubahan struktur database.
- Keamanan (Security)
Memberikan keamanan atas hak akses data.
- Kebersamaan pemakaian (Sharability)
Bersifat multiuser.



Sistem Basis Data

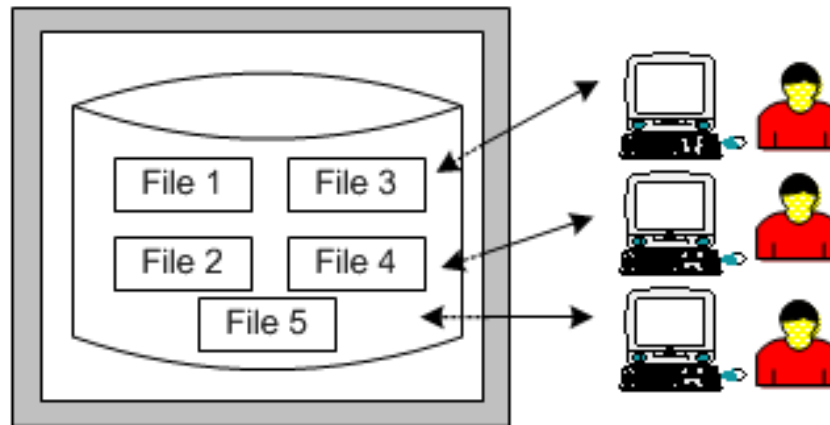
Sistem:

Tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dengan tujuan memenuhi suatu proses/pekerjaan tertentu.

Sistem Basis Data

Sistem Basis Data:

Sistem yang terdiri atas sekumpulan tabel yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan berbagai user dan/atau program lain dapat mengakses tabel tersebut.



Pengertian Database

Pengguna Basis Data:

- Kepegawaian & Administrasi
- Pergudangan
- Akuntansi
- Reservasi
- Customer Service
- Produksi
- Perhotelan dan Pariwisata
- Rumah Sakit
- Institusi Pendidikan, dll.

Komponen Sistem Basis Data

Komponen Sistem Basis Data:

- **Perangkat Keras (Hardware)**
Komputer, memori, storage (Harddisk), peripheral, dll.
- **Sistem Operasi (Operating System)**
Program yang menjalankan sistem komputer, mengendalikan resource komputer dan melakukan berbagai operasi dasar sistem komputer.
- **Basis Data (Database)**
Menyimpan berbagai obyek database (struktur tabel, indeks,dll)
- **DBMS (Database Management System)**
Perangkat lunak yang memaintain data dalam jumlah besar.
- **Pemakai (User)**
Para pemakai database.
- **Aplikasi (perangkat lunak) lain.**
Program lain dalam DBMS.

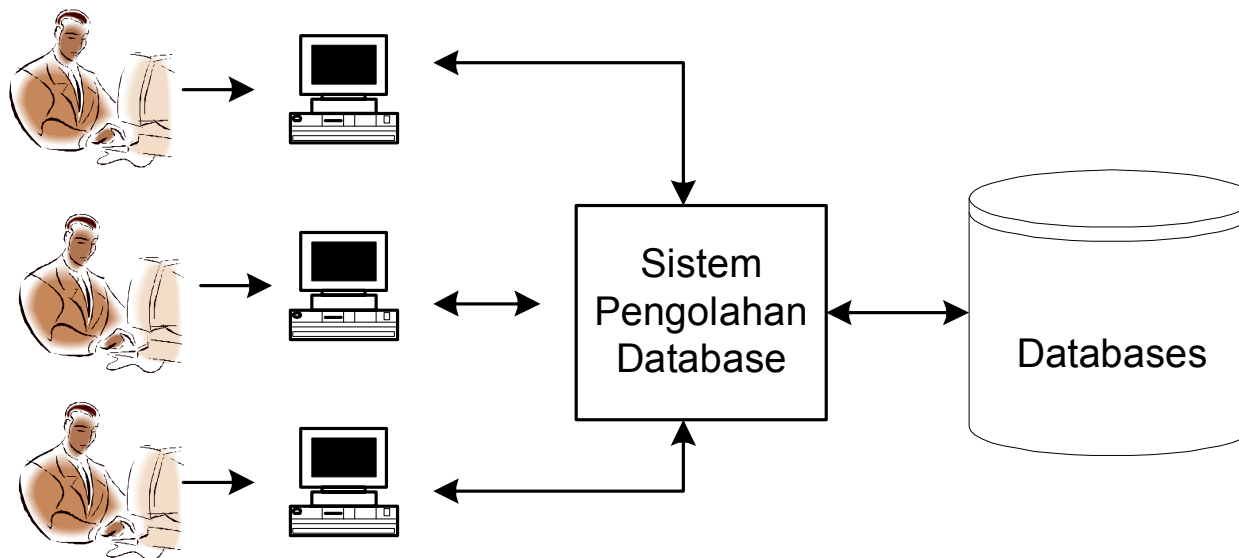
Lessons

1. Pengertian Database
- 2. Sistem Pengorganisasian Database**
3. Abstraksi Data
4. Bahasa Basis Data
5. Istilah dalam Database



Sistem Pengorganisasian Database

- Para ahli telah melakukan perbaikan dalam sistem pengorganisasian database tradisional, yaitu dengan mengembangkan sebuah sistem pengolahan database sehingga seluruh data yang terdapat di dalam SI dapat di **integrasikan**.
- Dengan pengorganisasian yang terintegrasi ini, aplikasi pe-departemental dapat melakukan akses database yang ada sesuai dengan kebutuhan masing-masing.



Sistem Pengorganisasian Database

- Fokus utama dalam sistem pengorganisasian database kontemporer yaitu pada pengaturan **hak akses**.
- Sehingga dengan sistem pengorganisasian yang baik ini dapat dilakukan akses database lintas departemental tanpa khawatir adanya kebebasan akses pada data khusus.

Lessons

1. Pengertian Database
2. Sistem Pengorganisasian Database
- 3. Abstraksi Data**
4. Bahasa Basis Data
5. Istilah dalam Database



Abstraksi Data

- Dalam database, data disimpan dan dipelihara dengan baik dan terstruktur oleh DBMS. Sistem menyembunyikan detail tentang bagaimana data disimpan dan dipelihara. Sehingga seringkali data yang terlihat oleh user, berbeda dengan data yang tersimpan secara fisik.
- Abstraksi data merupakan tingkatan/level dalam melihat bagaimana menampilkan data dalam sebuah sistem database.

Abstraksi Data

Terdapat 3 Level abstraksi data:

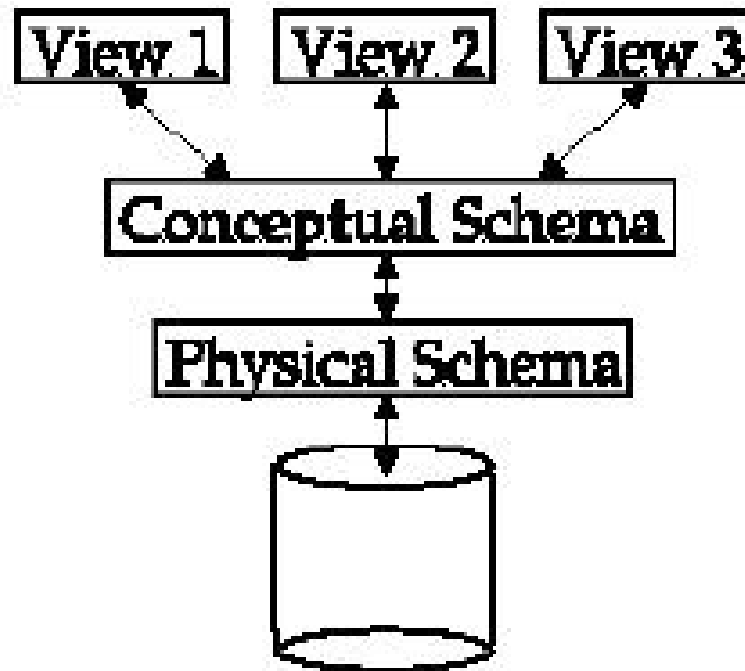
- **Level Konseptual (Conceptual Level)**
 - Level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan.
 - User melihat data sebagai gabungan dari struktur dan datanya sendiri.
 - Tingkatan ini berurusan dengan:
 - Alokasi ruang penyimpanan untuk data dan indeks
 - Deskripsi record untuk penyimpanan
 - Penempatan record data
 - Teknik kompresi dan enkripsi data

Abstraksi Data

- **Level Fisik (Physical Level)**
 - Menggambarkan data apa yang sebenarnya disimpan dalam database, serta hubungannya (relationship) dengan data lainnya.
 - Berisi struktur logik database yang hanya dapat dilihat oleh DBA.
 - Tingkat konseptual ini menyatakan:
 - Entitas, atribut dan relasinya
 - Konstrain-konstrain terhadap data
 - Informasi semantiks data
 - Informasi keamanan dan integritas data

Abstraksi Data

- **Level Pandangan (View Level)**
 - Level tertinggi dari abstraksi data, yang hanya menampilkan hanya sebagian dari database.
 - Tidak semua user membutuhkan semua data dalam database.



Lessons

1. Pengertian Database
2. Sistem Pengorganisasian Database
3. Abstraksi Data
- 4. Bahasa Basis Data**
5. Istilah dalam Database



Bahasa Basis Data

- DBMS merupakan perantara antara user dengan database.
- Cara komunikasi diatur dalam suatu bahasa khusus yang telah ditetapkan oleh DBMS.
- Contoh: SQL, dBase, QUEL, dsb.
- Bahasa database, dibagi dalam 2 bentuk:
 - Data Definition Language (**DDL**)
 - Data Manipulation Language (**DML**)

Bahasa Basis Data

- **Data Definition Language (DDL)**
 - Digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dsb.
 - Hasil dari kompilasi perintah DDL berupa kumpulan tabel yang disimpan dalam file khusus: **Kamus Data (Data Dictionary)**.
 - **Data Dictionary**: merupakan metadata (superdata), yaitu data yang mendeskripsikan data sesungguhnya. Data dictionary ini akan selalu diakses dalam suatu operasi database sebelum suatu file data yang sesungguhnya diakses.

Bahasa Basis Data

- **Data Manipulation Language (DML)**
 - Digunakan dalam memanipulasi dan pengambilan data pada database.
 - Manipulasi data, dapat mencakup:
 - Pemanggilan data yang tersimpan dalam database (query)
 - Penyisipan/penambahan data baru ke database
 - Penghapusan data dari database
 - Pengubahan data pada database

Bahasa Basis Data

- **Terdapat dua (2) jenis DML:**
 - **Prosedural**

Menghendaki user untuk menspesifikasikan data apa yang diperlukan dan bagaimana cara mendapatkan data itu.

Contoh: bahasa C/C++, PL/SQL, dsb.
 - **Nonprosedural**

Menghendaki user untuk menspesifikasikan data apa yang dibutuhkan, tanpa harus menspesifikasikan bagaimana cara mendapatkan data tersebut.

Contoh: SQL



ENTITY RELATIONSHIP DIAGRAM

Kholid Fathoni, S.Kom., M.T.

ER Data Model

- Pemodelan data / sistem dalam database digunakan **Model ER** (Entity Relationship) **Diagram** atau disebut **ERD**.
- ER Diagram menggambarkan tipe objek mengenai **data** itu di manajemen, serta relasi antara objek tersebut.
- Biasanya yang menggunakan ER Diagram adalah **System Analyst** dalam merancang database.
- ER Model dibuat berdasarkan persepsi atau pengamatan dunia nyata yang terdiri atas **entitas** dan **relasi** antara entitas-entitas tersebut.
- Sebuah **database** dapat dimodelkan sebagai:
 - Kumpulan **Entity/Entitas**,
 - **Relationship/Relasi** diantara entitas.

ER Data Model

- **Entitas** adalah sebuah obyek yang ada (exist) dan dapat dibedakan dengan obyek yang lain.
- Entitas ada yang bersifat **konkrit**, seperti: orang, buku, pegawai, perusahaan; dan ada yang bersifat **abstrak**, seperti: kejadian, mata kuliah, pekerjaan dan sebagainya.
- Setiap entitas memiliki **atribut** sebagai keterangan dari entitas, misal. entitas mahasiswa, yang memiliki atribut: nrp, nama dan alamat.

ER Data Model

- Setiap atribut pada entitas memiliki **kunci atribut (key atribut)** yang bersifat **unik**.

Misal.

- Entitas **Mahasiswa** dengan atribut **NRP** sebagai key atribut
- Entitas **Dosen** dengan **NIP** sebagai key atribut, dan sebagainya.
- Beberapa entitas kemungkinan tidak memiliki atribut kunci sendiri, entitas demikian disebut **Entitas Lemah (Weak Entity)**.

Jenis Atribut dan Notasi ER Diagram

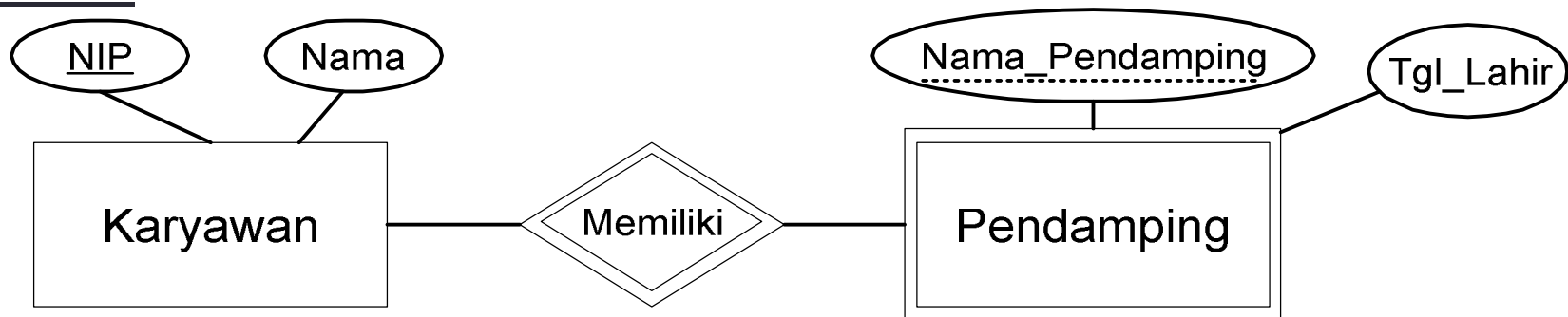
- Dalam pembuatan ERD digunakan notasi diagram. Beberapa notasi yang digunakan untuk membuat ER Diagram. Misal. notasi Chen, Martin, **EI Masri** dan Korth, akan tetapi pada umumnya adalah sama.
- Perbedaannya adalah pada pemilihan simbol-simbol yang digunakan.
- Pada materi database dan umumnya, digunakan notasi EI Masri karena lebih umum banyak digunakan dan mudah dibaca dan dimengerti.

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E ₂ IN R
	CARDINALITY RATIO 1:N FOR E ₁ :E ₂ IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

ER Data Model

- **Entitas Lemah (Weak Entity)** adalah entitas yang keberadaannya sangat bergantung dengan entitas lain.
 - Tidak memiliki **Key Attribute** sendiri.
 - Entitas tempat bergantung disebut **Identifying Owner/Owner**.
 - Entitas lemah tidak memiliki identifiernya sendiri.
 - Atribut entitas lemah berperan sebagai **Partial Identifier** (identifikasi yang berfungsi secara sebagian).

Contoh:



ER Data Model

Jenis – Jenis Atribut:

- **Simple / Atomic Attribute:** adalah atribut yang tidak dapat dibagi-bagi lagi menjadi atribut yang lebih mendasar.
- **Composite Attribute:** atribut yang terdiri dari beberapa atribut yang lebih mendasar.

Contoh:

- Atribut ALAMAT, terdiri atas atribut JALAN, KOTA, KODE_POS.
- Atribut NAME, terdiri atas atribut FNAME, MNAME dan LNAME pada suatu entitas (EMPLOYEE).
- **Single-Valued Attribute:** atribut yang hanya memiliki satu harga/nilai.

Contoh:

- Atribut UMUR pada entitas PEGAWAI
- Atribut LOCATIONS pada entitas DEPARTMENT

ER Data Model

- **Multi-Valued Attribute:** adalah atribut yang memiliki isi lebih dari satu nilai.

Contoh:

- Atribut PENDIDIKAN TINGGI pada entitas PEGAWAI, dapat berisi lebih dari satu nilai: SMP, SMU, Perguruan Tinggi (Sarjana), Doktor, dll.
 - Atribut HOBBY pada entitas MAHASISWA, dapat memiliki lebih dari satu nilai: sepak bola, menyanyi, menari, tennis, dsb.
 - Atribut PRASYARAT pada entitas MATA_KULIAH, dapat memiliki lebih dari satu nilai: Konsep Pemrograman & Algoritma Struktur Data untuk prasyarat mata kuliah Pemrograman Lanjut.
- **Null Values Attribute:** adalah atribut dari entitas yang tidak memiliki nilai.

Contoh:

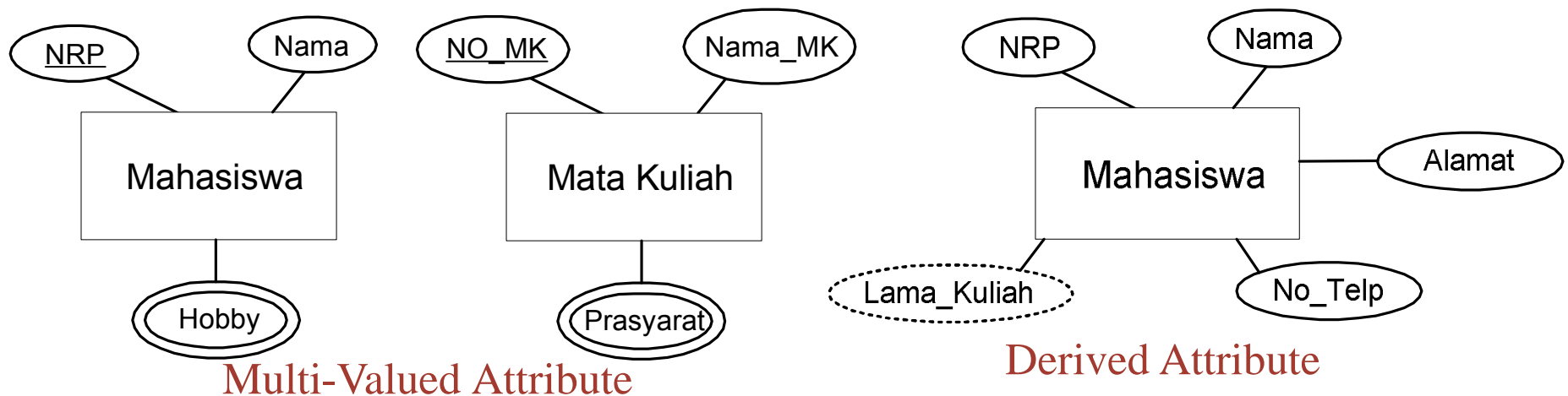
Atribut PENDIDIKAN TINGGI untuk tamatan SMP.

ER Data Model

- **Derived Attribute:** adalah atribut yang nilainya dapat diisi atau diturunkan dari perhitungan atau algoritma tertentu.

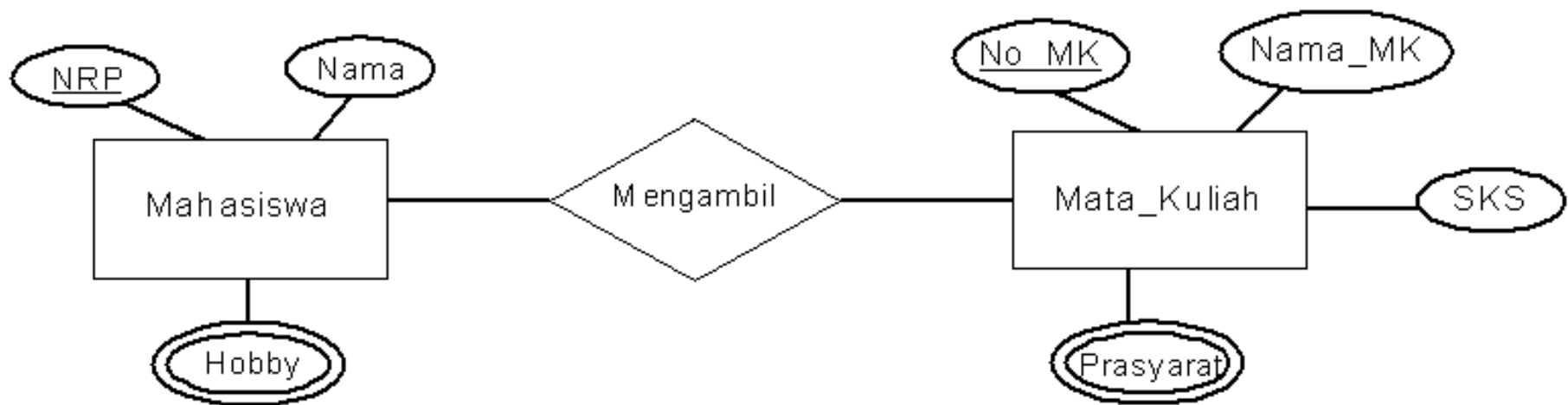
Contoh:

- Atribut UMUR, dapat dihitung dari atribut TGL_LAHIR
- Atribut LAMA_KULIAH, dapat dihitung dari NRP yang merupakan kombinasi antara digit tahun dan digit yang lain (2696 100...).
- Atribut INDEX_PRESTASI, dapat dihitung dari NILAI yang diperoleh MAHASISWA.



Relasi dan Rasio Kardinalitas

- **Relasi** adalah hubungan antar entitas.
- **Relasi** dapat memiliki **atribut**, dimana terjadi adanya transaksi yang menghasilkan suatu nilai tertentu.

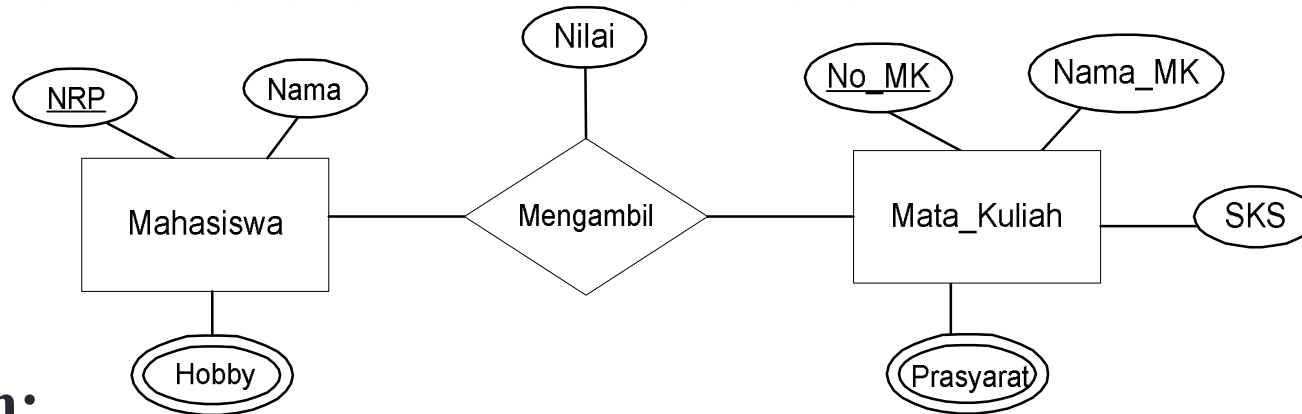


Penjelasan:

É Bentuk ER diatas antara Mahasiswa Mengambil Mata_Kuliah, tentunya ada Nilai yang dihasilkan.

É Dimana atribut nilai ditempatkan?

Relasi dan Rasio Kardinalitas



Penjelasan:

- É Jika atribut **Nilai** ditempatkan pada entitas **Mahasiswa** (dimana **Nilai** merupakan salah satu atribut dari entitas **Mahasiswa**), maka semua mata kuliah yang diambil oleh seorang mahasiswa menghasilkan nilai yang sama (**tidak realistis**).
- É Jika atribut **Nilai** ditempatkan pada entitas **Mata_Kuliah** (dimana **Nilai** merupakan salah satu atribut dari entitas **Mata_Kuliah**), maka semua mahasiswa yang mengambil mata kuliah tertentu akan memiliki nilai yang sama (**tidak realistis**).
- ☑ Atribut **Nilai** harus ditempatkan pada relasi **Mengambil**, yang berarti seorang mahasiswa tertentu yang mengambil mata kuliah tertentu, akan mendapatkan nilai tertentu pula.

Relasi dan Rasio Kardinalitas

Derajat Relasi

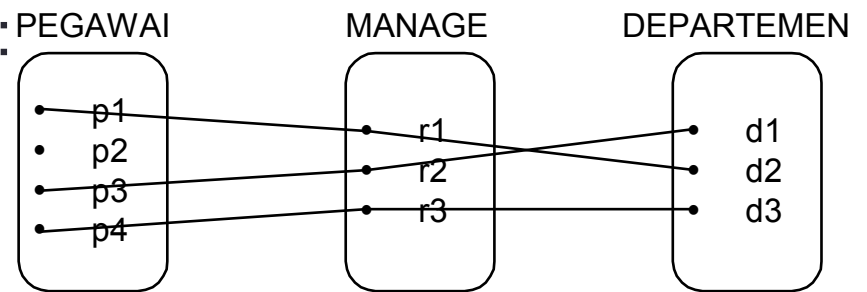
- **Derajat Relasi** adalah jumlah entitas yang berpartisipasi dalam suatu relasi.
- Derajat Relasi dapat berupa:
 - Unary Relationship (Relasi Berderajat 1)
 - Binary Relationship (Relasi Berderajat 2)
 - Ternary Relationship (Relasi Berderajat 3)

Relasi dan Rasio Kardinalitas

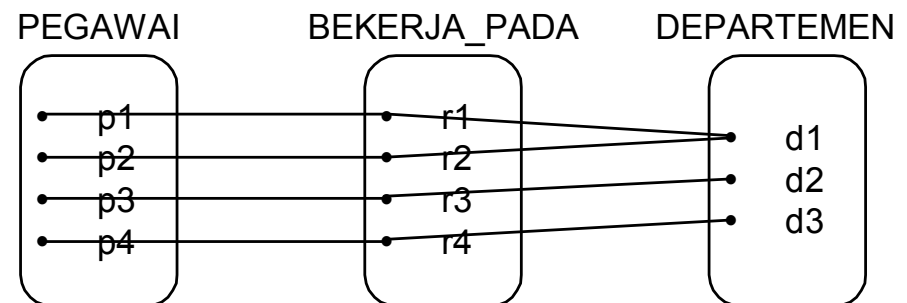
Rasio Kardinalitas

- Dalam relasi binary antar 2 entitas (relasi umumnya), terdapat beberapa kemungkinan:

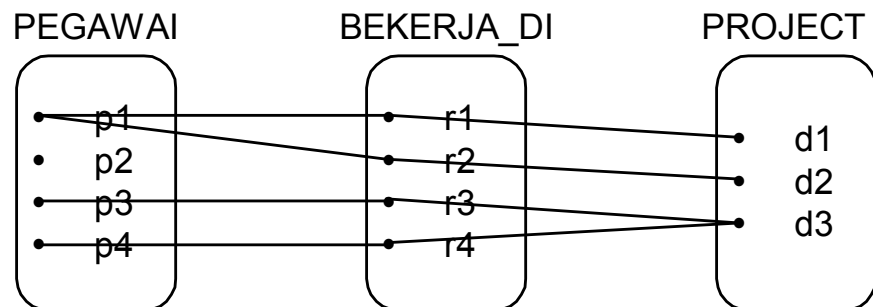
☑ **1 : 1 : One-to-One**



☑ **N : 1 : Many-to-One**



☑ **M : N : Many-to-Many**

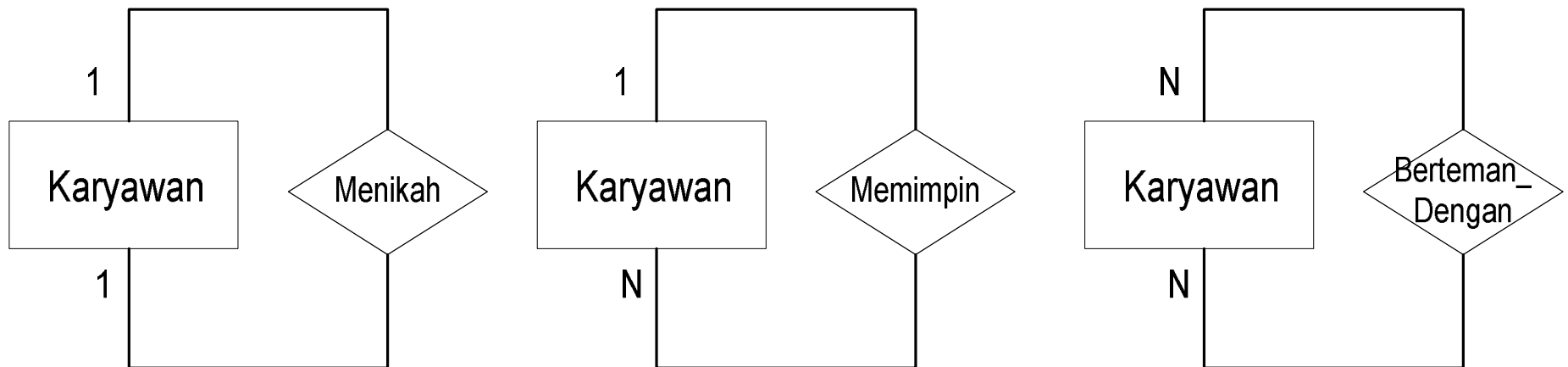


Relasi dan Rasio Kardinalitas

Unary Relationship (Relasi Berderajat 1)

- adalah relasi dimana entitas yang terlibat hanya 1.
- Sering disebut relasi rekursif (recursive relationship).

Contoh:

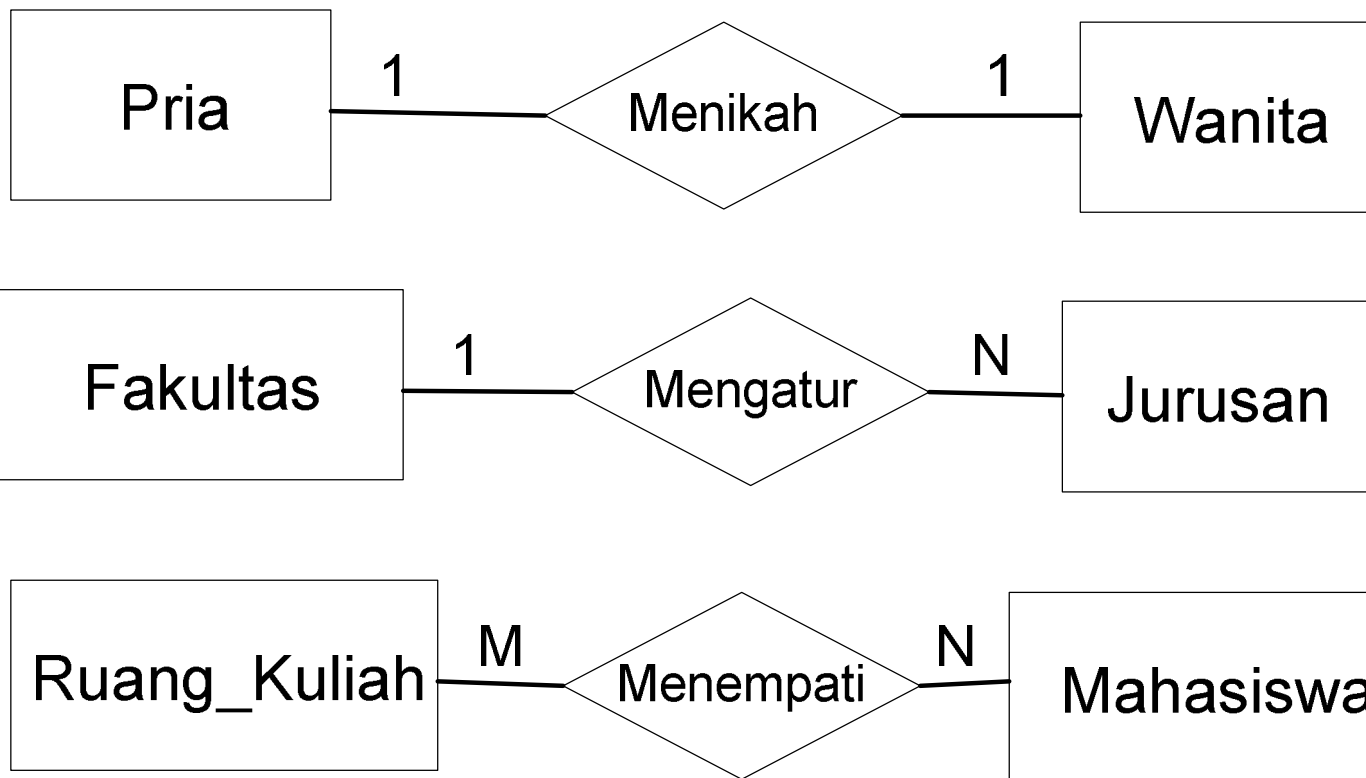


Relasi dan Rasio Kardinalitas

Binary Relationship (Relasi Berderajad 2)

- Atau relasi **Biner** adalah relasi yang melibatkan 2 entitas.

Contoh:

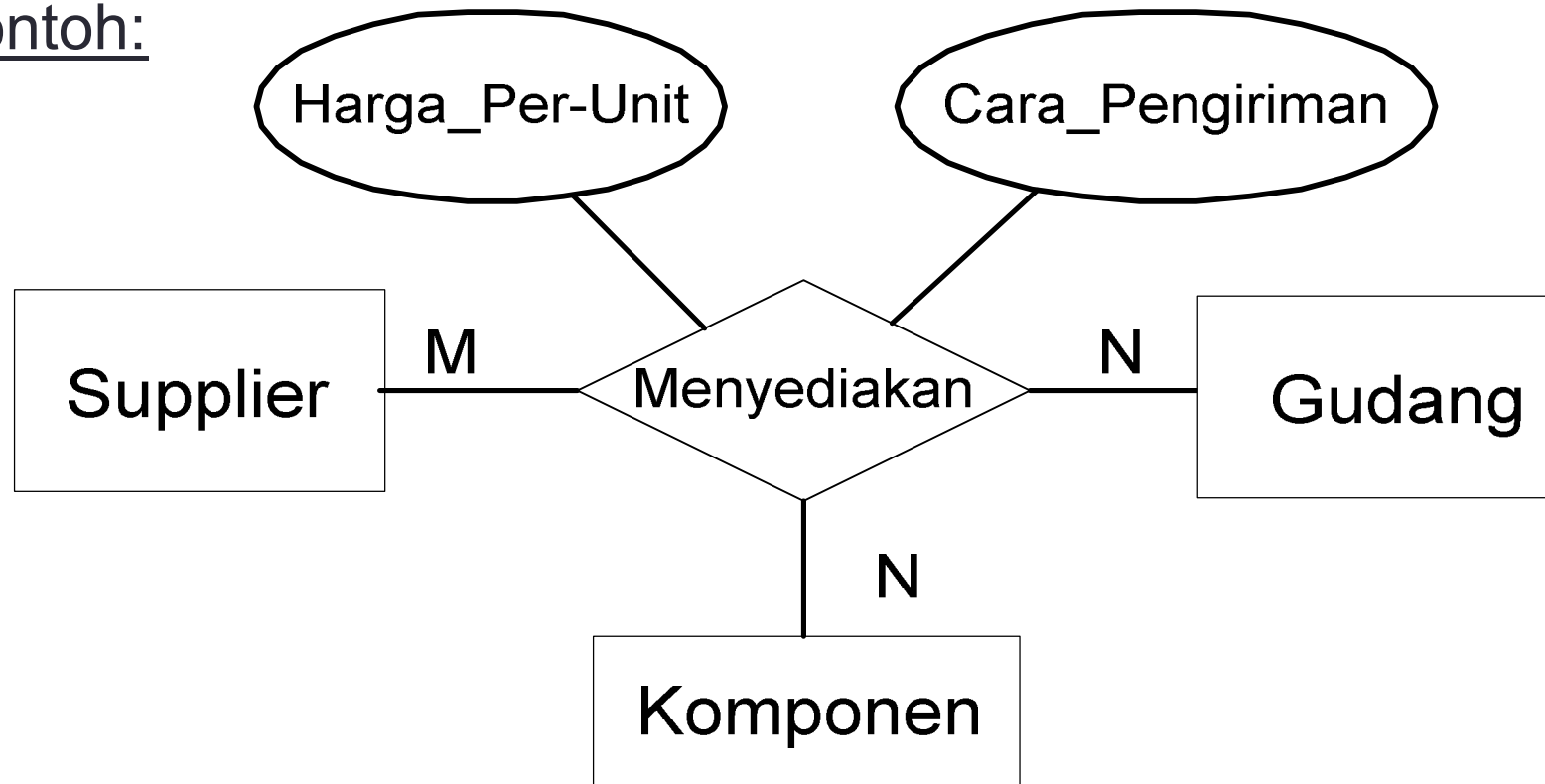


Relasi dan Rasio Kardinalitas

Ternary Relationship (Relasi Berderajat 3)

- adalah relasi tunggal yang menghubungkan 3 entitas yang berbeda.

Contoh:



Catatan ERD:

- Penentuan derajat relasi tergantung dari aturan bisnis (business rule) yang ada di perusahaan/instansi.
- Walau notasi derajat relasi tidak sedominan notasi yang lain, tapi derajat relasi sangat berperan dalam tahap implementasi.

Tahapan Pembuatan E-R

- Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
- Menentukan atribut-atribut key dari masing-masing himpunan entitas
- Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta foreign key-nya
- Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi
- Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (non key)

Tahap membangun ERD

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat

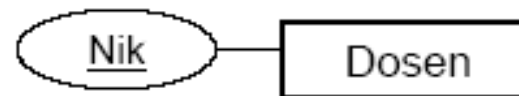
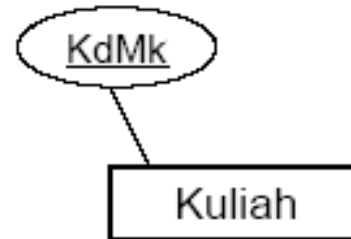
Mahasiswa

Kuliah

Dosen

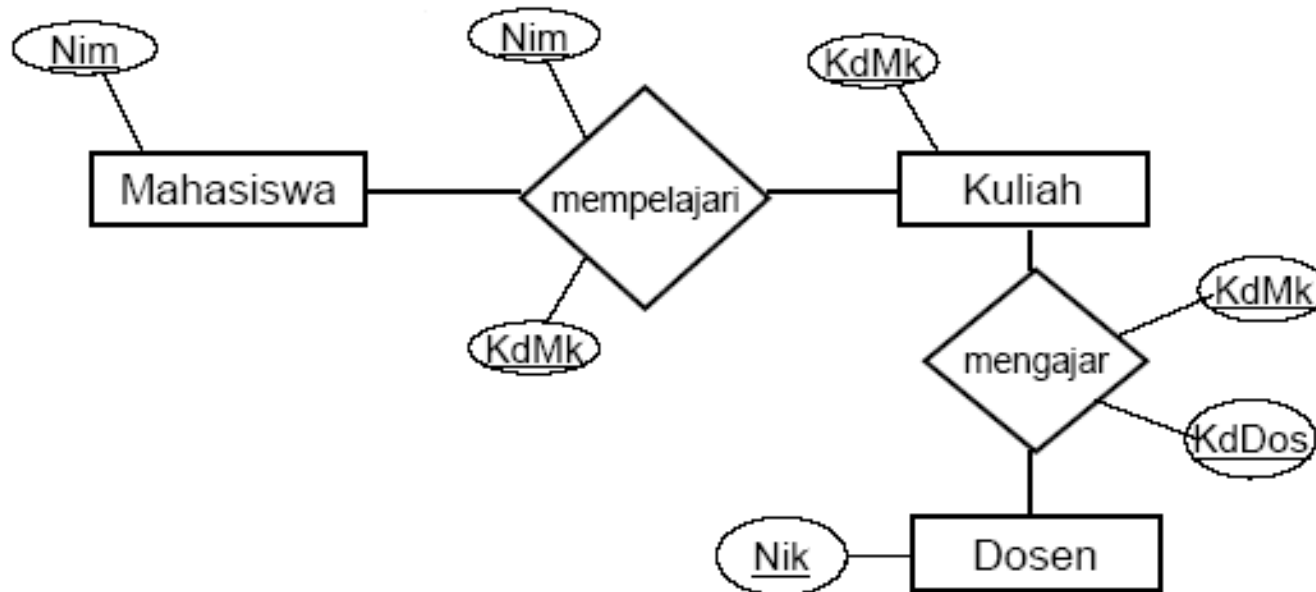
Tahap membangun ERD....

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



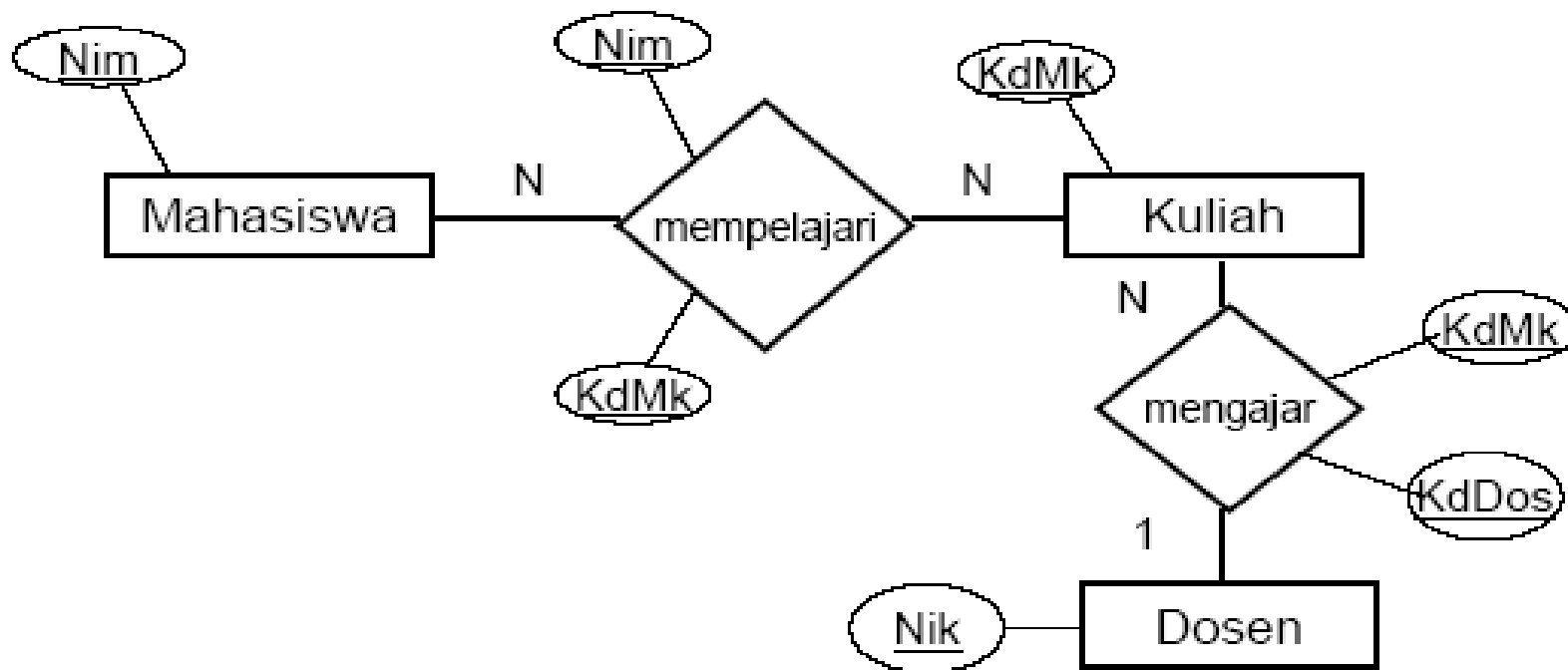
Tahap membangun ERD....

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



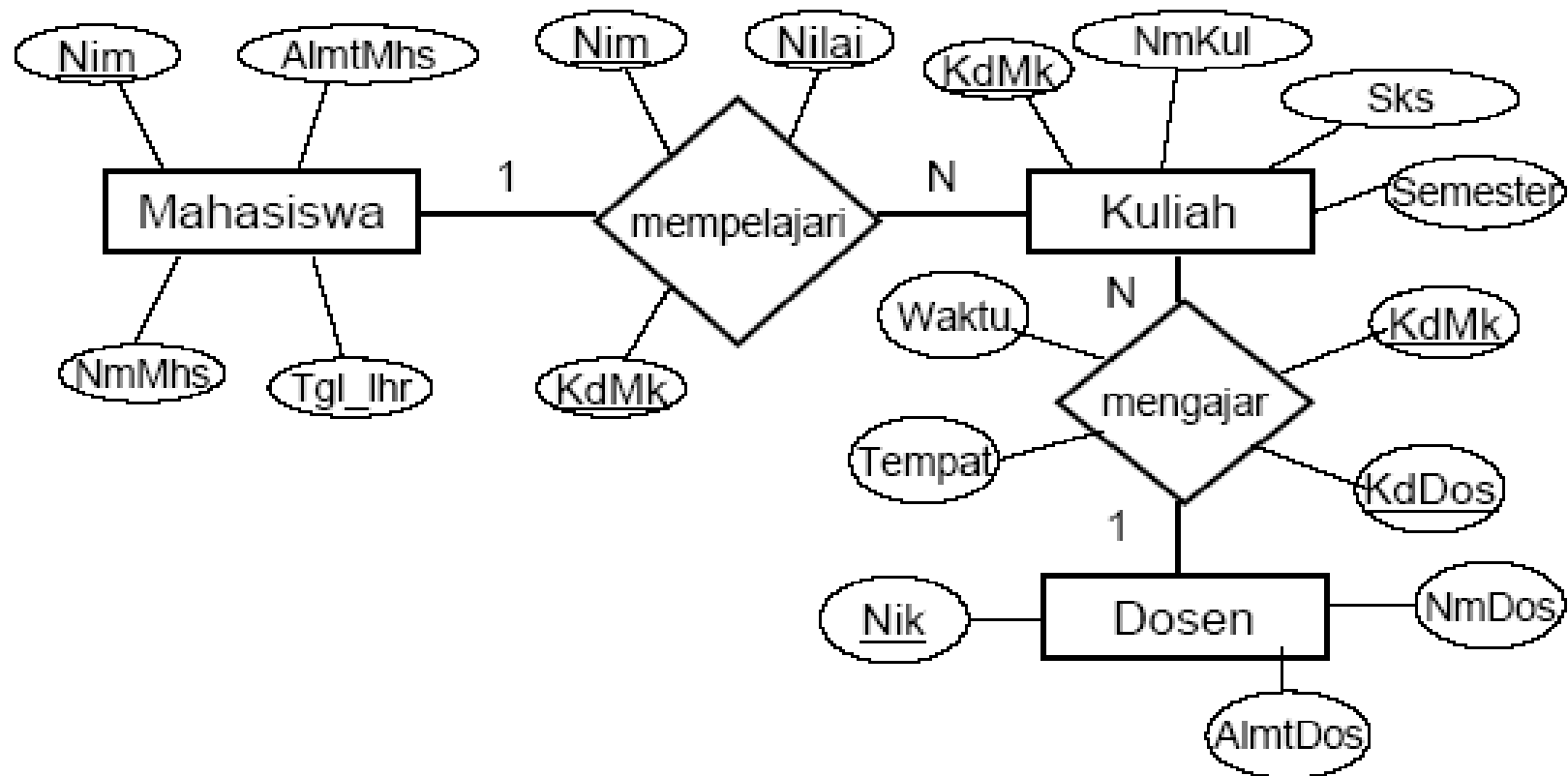
Tahap membangun ERD....

4. Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi



Tahap membangun ERD....

5. Melengkapi himpnan entitas dan himpunan relasi dengan atribut deskriptif (non key)





Question?