

INTERNET PROGRAMMING

Sistem Basis Data



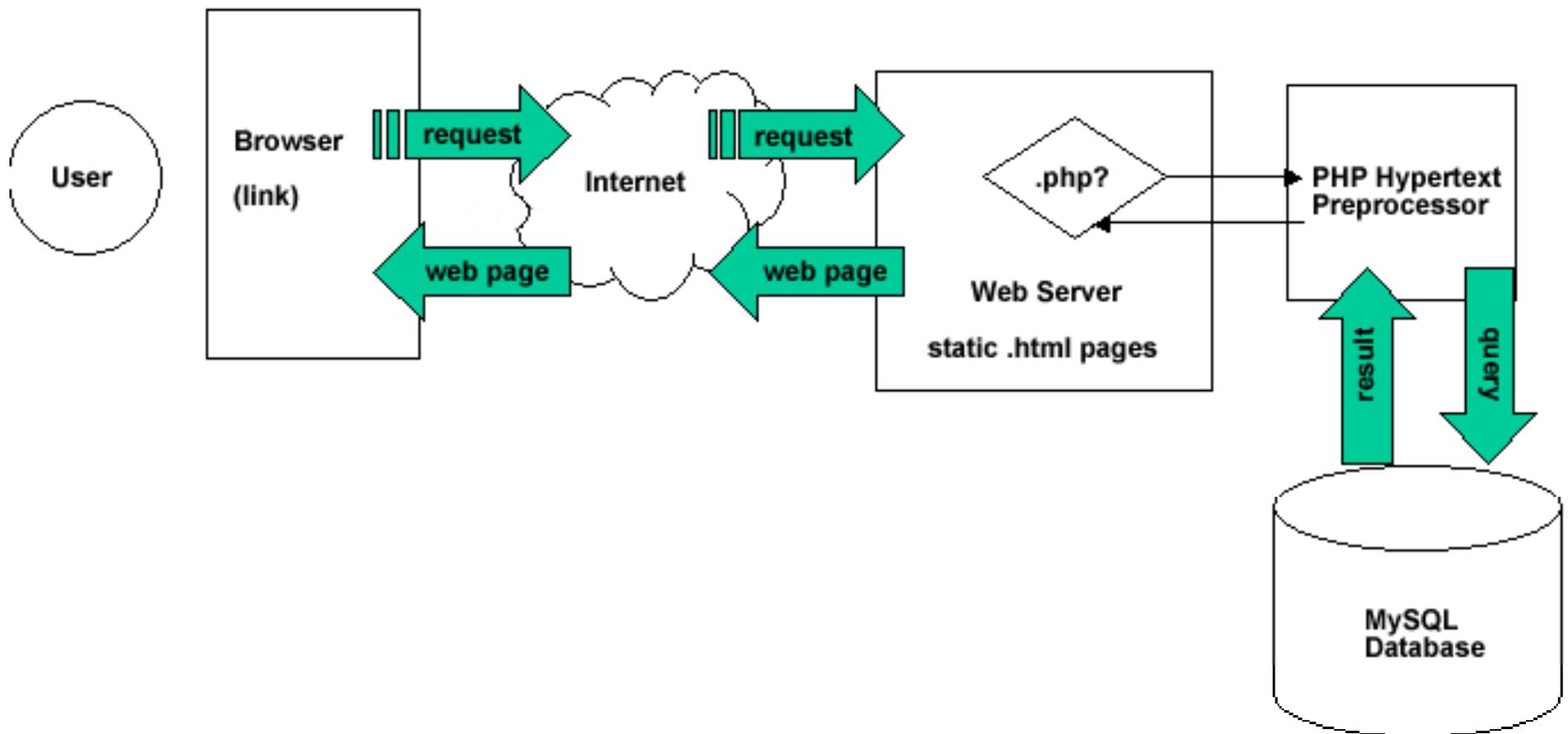
MUHAMMAD ZEN S. HADI, ST. MSC.

Bahasan



- **Sistem Database**
- **ER Diagram**
- **Database MySQL**

Internet Application



Pendahuluan



- Menyimpan data dalam file biasa memiliki banyak keterbatasan. Semakin besar ukuran file, pencarian data menjadi lebih sulit. File biasa juga tidak memiliki kemampuan untuk mengolah data, misalnya menghitung total nilai, rata-rata, dan lain sebagainya.
- Adanya keterbatasan untuk mengendalikan akses terhadap data. Kita tidak dapat menentukan siapa yang boleh dan siapa yang tidak boleh mengakses data.
- Karena itu, sekarang kita akan menggunakan media penyimpanan data yang lebih mutakhir, yaitu database.

Definisi Basis Data



Prinsip kerja Basis Data:

- Pengaturan data / arsip

Tujuan Basis Data:

- Kemudahan dan kecepatan dalam pengambilan data (**speed**)
- Efisiensi ruang penyimpanan (**space**)

Mengurangi / menghilangkan redudansi 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.

Definisi Basis Data



Tujuan Basis Data: (Con't)

- Ketersediaan (**Avaibility**)
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 (**Sharebility**)
Bersifat multiuser.

Definisi Basis Data



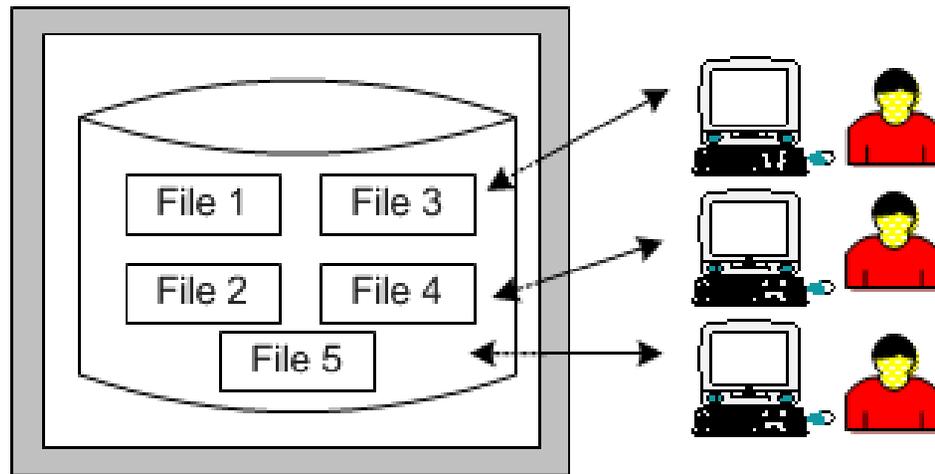
Pengguna Basis Data:

- Kepegawaian
- Pergudangan
- Akuntansi
- Bank
- Reservasi
- Customer Service, dll.

Sistem Basis Data

Sistem Basis Data:

Sistem yang terdiri atas sekumpulan **tabel yang saling berhubungan** dan **sekumpulan program (DBMS: Database Management System)** yang memungkinkan berbagai user dan/atau program lain dapat mengakses dan memanipulasi tabel-tabel tersebut.



Sistem Basis Data



Contoh program aplikasi DBMS:

- Dbase
- FoxPro
- MySQL
- MS Access
- SQL Server
- Oracle
- DB2
- dll

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 (**Insert**)
 - **Pengubahan** data pada database (**Update**)
 - **Penghapusan** data dari database (**Delete**)



Entity Relationship (ER) DIAGRAM

ER Data Model



- Pemodelan sistem dengan ER Data Model (ER Diagram) digunakan dalam pembuatan **basis data (database)**.
- **Basis data (Database)** adalah Kumpulan file atau data yang saling berhubungan.
- ER Diagram menggambarkan tipe objek mengenai **data** itu di manajemen, serta relasi antara objek tersebut.
- ER Diagram digunakan oleh **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.

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:

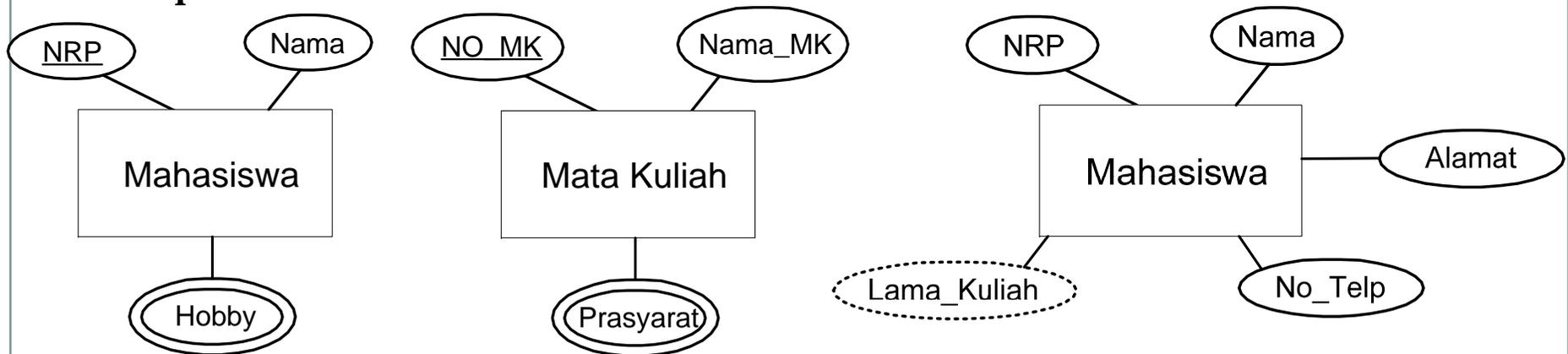
Atribut PENDIDIKAN TINGGI untuk tamatan SMP.

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 (7206...).
- Atribut INDEX_PRESTASI, dapat dihitung dari NILAI yang diperoleh MAHASISWA.

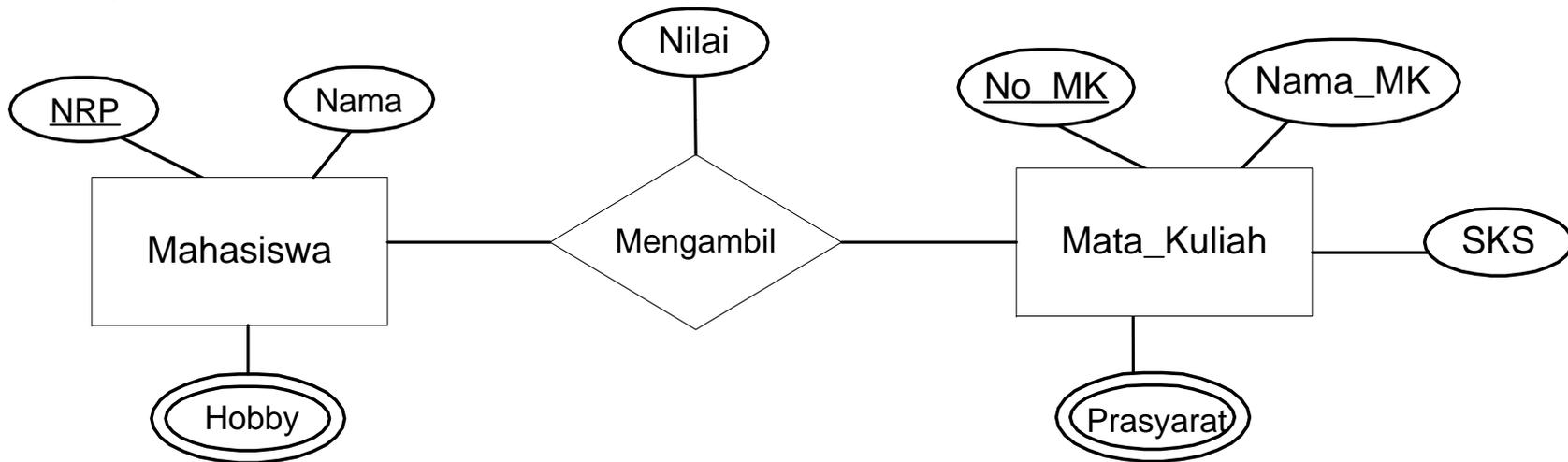


Multi-Valued Attribute

Derived Attribute

Relasi dan Rasio Kardinalitas

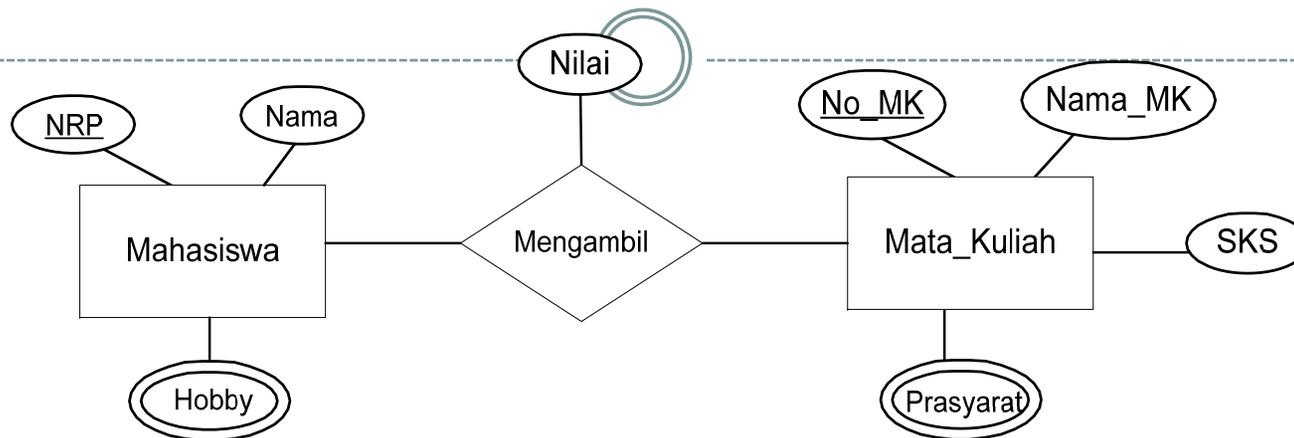
- **Relasi** adalah hubungan antar entitas.
- **Relasi** dapat memiliki **atribut**, dimana terjadi adanya transaksi yang menghasilkan suatu nilai tertentu.



Penjelasan:

- Bentuk ER diatas antara Mahasiswa Mengambil Mata_Kuliah, tentunya ada Nilai yang dihasilkan.
- Dimana atribut nilai ditempatkan?

Relasi dan Rasio Kardinalitas



Penjelasan:

- 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**).
- 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)

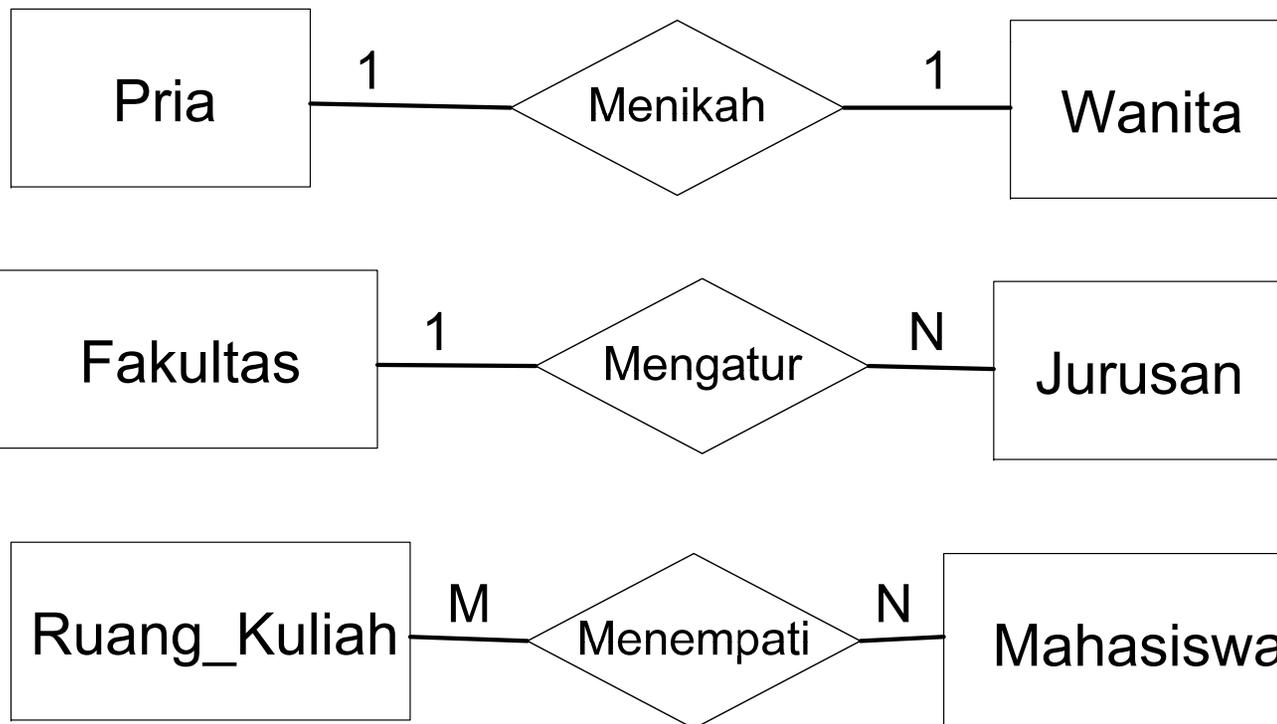
Relasi dan Rasio Kardinalitas



Binary Relationship (Relasi Berderajat 2)

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

Contoh:

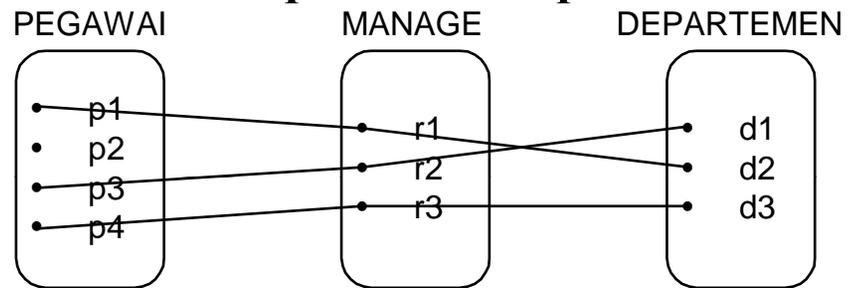


Relasi dan Rasio Kardinalitas

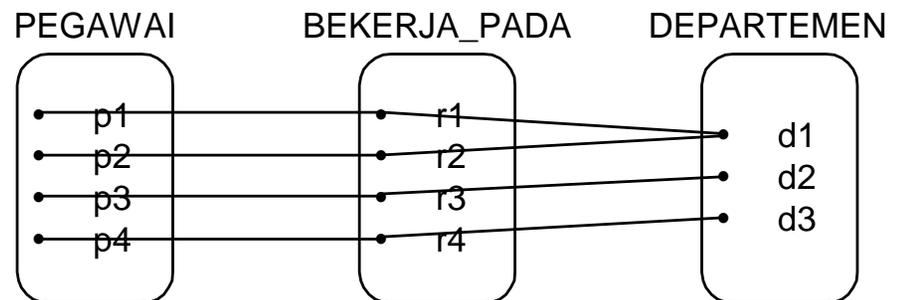
Rasio Kardinalitas

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

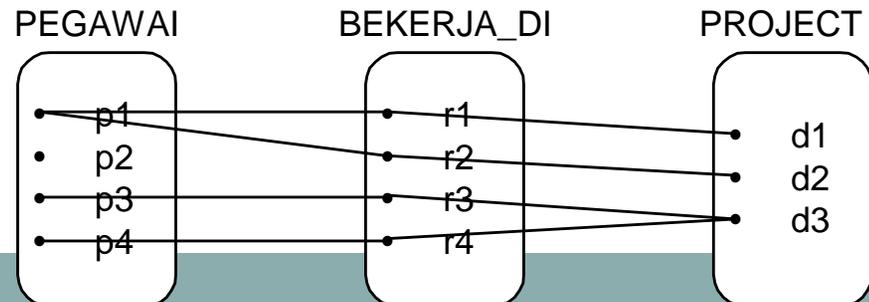
☑ **1 : 1 : One-to-One**



☑ **N : 1 : Many-to-One**



☑ **M : N : Many-to-Many**



Mapping ke Skema Relasi

Untuk melakukan **mapping (pemetaan)** dari skema ER Diagram ke skema relasi terdapat langkah-langkah yang harus diperhatikan.

Langkah-langkah mapping:

1. Untuk **setiap entitas** skema relasi R yang menyertakan seluruh Simple Attribute dan Simple Attribute dari Composite Attribute yang ada, pilih salah satu atribut kunci sebagai **Primary Key**.
2. Untuk setiap relasi binary **1:1**, tambahkan **Primary Key** dari sisi yang lebih "ringan" ke sisi (entitas) yang lebih "berat". Suatu sisi dianggap lebih "berat" timbangannya apabila mempunyai **partisipasi total**.
Tambahkan juga **Simple Attribute** yang terdapat pada relasi tersebut ke sisi yang lebih "berat".
3. Untuk setiap relasi binary **1:N**, tentukan mana sisi yang lebih "berat". Sisi dianggap lebih "berat" timbangannya adalah sisi-N. Tambahkan Primary Key dari sisi yang "ringan" ke skema relasi sisi yang lebih "berat".
4. Untuk setiap relasi binary **M:N**, **buatlah skema relasi baru R** dengan atribut **seluruh simple attribute** yang terdapat pada relasi biner tersebut. Tambahkan **primary key** yang terdapat pada kedua sisi ke skema relasi R. Kedua **Foreign Key** yang didapat dari kedua sisi tersebut **digabung** menjadi sat membentuk **Primary Key** dari skema relasi R.

Latihan



- **Buatlah design database untuk permasalahan berikut ini :**
 - 1. Peminjaman buku di perpustakaan**
 - 2. Penjualan barang di suatu toko**
 - 3. Guru mengajar di kelas**



DATABASE - MySQL

Constraints



- **NOT NULL**
Tidak boleh berisi NULL (kosong).
- **UNIQUE**
Satu data dengan data yang lainnya tidak boleh sama.
- **PRIMARY KEY**
- **FOREIGN KEY**
Sebagai relasi antara 2 tabel.

Tipe data di MySQL



- Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (*row* atau *record*) dan kolom(*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa *table*.
- Beberapa tipe data dalam MySQL yang sering dipakai:

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LOB	Teks dengan panjang maksimum 4294967295 karakter

MEMBUAT DATABASE DAN TABLE



Contoh:

- Misalkan kita ingin menyimpan data anggota yaitu: nomor, nama, email, alamat, kota. Sedangkan strukturnya seperti tabel dibawah ini:

Kolom/Field	Tipe data	Keterangan
nomor	int(6) not null primary key	angka dengan panjang maksimal 6, sebagai <i>primary key</i> , tidak boleh kosong
nama	char(40) not null	teks dengan panjang maksimal 40 karakter, tidak boleh kosong
email	char(255) not null	teks dengan panjang maksimal 255 karakter, tidak boleh kosong
alamat	char(80) not null	teks dengan panjang maksimal 80 karakter, tidak boleh kosong
kota	char(20) not null	teks dengan panjang maksimal 20 karakter, tidak boleh kosong

Data yang diinginkan



Nomor	Nama	Email	Alamat	Kota
100	Adi	adi@yahoo.com	Jl. Keputih 2A no 5	Surabaya
101	Arif	arif@hotmail.com	Jl. Gebang Lor 32D	Surabaya
102	Iqbal	iqbal@yahoo.com	Jl. Klampis 3	Surabaya
103	Setyo	setyo@mailcity.com	Jl. Madiun 5	Sidoarjo
104	Ilham	ilham@yahoo.com	Jl. Surabaya 9	Malang
105	Syamsudin	sam@yahoo.com	Jl. Pabean 11	Surabaya
106	Faruq	faruq@yahoo.com	Jl. Raya 5	Kediri
107	Hari	hari@yahoo.com	Jl. Raya 9	Banyuwangi

Menggunakan phpMyAdmin



- **software yang digunakan untuk membuat dan maintenance database**
- **Kita dapat mengakses database MySQL dengan account kita di phpMyAdmin.**



- Pastikan server MySQL dan program phpMySQL sudah berjalan.
- untuk membuka tampilan phpMyAdmin di komputer lokal melalui URL <http://localhost/phpmyadmin/>

Masukkan login sebagai root atau sesuai dengan user yang sudah dibuat

The screenshot displays the phpMyAdmin 2.6.2 interface. At the top, it says "Welcome to phpMyAdmin 2.6.2" and "MySQL 4.1.11-Debian_4-log running on localhost as root@localhost". The interface is divided into two main sections: "MySQL" and "phpMyAdmin".

MySQL Section:

- Create new database**: Includes a text input field, a "Collation" dropdown menu, and a "Create" button.
- Show MySQL runtime information**
- Show MySQL system variables**
- Show processes**
- Character Sets and Collations**
- Storage Engines**
- Reload MySQL**
- Privileges**
- Binary log**
- Databases**
- Export**
- Change password**
- Log out**

phpMyAdmin Section:

- Language**: Set to "English (en-utf-8)".
- MySQL charset**: Set to "UTF-8 Unicode (utf8)".
- MySQL connection collation**: Set to "utf8_general_ci".
- Theme / Style**: Set to "Original".
- phpMyAdmin documentation**
- Show PHP information**
- Official phpMyAdmin Homepage**
- [\[ChangeLog\]](#) [\[CVS\]](#) [\[Lists\]](#)

Left Sidebar:

- Database: (Databases) ...
- Please select a database

Membuat database dengan memasukkan nama database

The screenshot shows the phpMyAdmin 2.6.2 web interface in a Mozilla Firefox browser. The browser's address bar displays the URL: `http://localhost/phpmyadmin/index.php?lang=en-utf-8&server=1&collation_connectio`. The page title is "Welcome to phpMyAdmin 2.6.2" and it indicates the MySQL version is 4.1.11-Debian_4-log running on localhost as user udInharun@localhost.

The interface is divided into several sections:

- Left Sidebar:** Contains the phpMyAdmin logo, navigation icons, and a "Database:" dropdown menu currently set to "(Databases) ...". Below this is the instruction "Please select a database".
- MySQL Section:** Features a "Create new database" form with a text input field containing "data", a "Collation" dropdown menu, and a "Create" button. Below the form is a list of navigation links: "Show MySQL runtime information", "Show MySQL system variables", "Show processes", "Character Sets and Collations", "Storage Engines", "Reload MySQL", "Privileges", "Binary log", "Databases", "Export", "Change password", and "Log out".
- phpMyAdmin Section:** Contains configuration options: "Language" (English (en-utf-8)), "MySQL charset" (UTF-8 Unicode (utf8)), "MySQL connection collation" (utf8_general_ci), and "Theme / Style" (Original). It also includes links for "phpMyAdmin documentation", "Show PHP information", and "Official phpMyAdmin Homepage" with sub-links for "ChangeLog", "CVS", and "Lists".

membuat tabel dan jumlah field



The screenshot shows the phpMyAdmin interface in a Mozilla Firefox browser window. The address bar shows the URL: `http://localhost/phpmyadmin/index.php?lang=en-utf-8&server=1&collation_connectio`. The interface displays the current server as 'localhost' and the database as 'data'. A navigation bar includes buttons for Structure, SQL, Export, Search, Query, Operations, and Drop. A table listing shows one table, 'kelas3', with 0 records, MyISAM type, and latin1_swedish_ci collation. Below the table listing, there are options to 'Check All / Uncheck All' and a 'With selected:' dropdown. At the bottom, there is a 'Create new table on database data:' form with 'Name: kelas3' and 'Fields: 5' entered, and a 'Go' button.

Table	Action	Records	Type	Collation	Size	Overhead
<input type="checkbox"/> kelas3		0	MyISAM	latin1_swedish_ci	1.0 KB	-
1 table(s)	Sum	0	-	latin1_swedish_ci	1.0 KB	0 Bytes

↑ [Check All](#) / [Uncheck All](#) With selected:

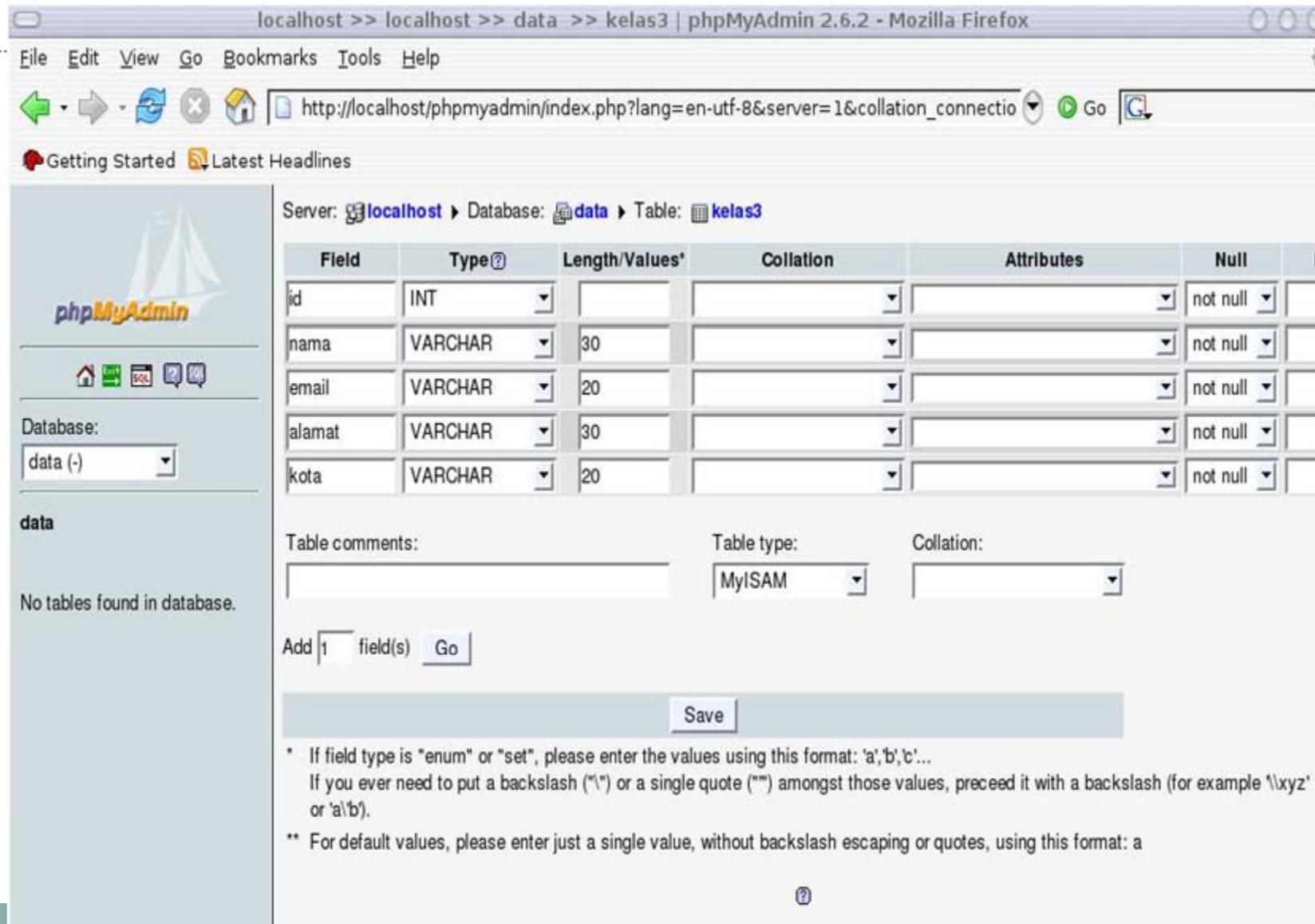
[Print view](#) [Data Dictionary](#)

Create new table on database data:

Name:

Fields:

Menentukan field dan type data



localhost >> localhost >> data >> kelas3 | phpMyAdmin 2.6.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/phpmyadmin/index.php?lang=en-utf-8&server=1&collation_connectio

Getting Started Latest Headlines

Server: localhost Database: data Table: kelas3

Field	Type	Length/Values	Collation	Attributes	Null
id	INT				not null
nama	VARCHAR	30			not null
email	VARCHAR	20			not null
alamat	VARCHAR	30			not null
kota	VARCHAR	20			not null

Database: data (-)

data

No tables found in database.

Table comments:

Table type: MyISAM

Collation:

Add 1 field(s) Go

Save

* If field type is "enum" or "set", please enter the values using this format: 'a','b','c'...
If you ever need to put a backslash ("\") or a single quote (") amongst those values, precede it with a backslash (for example '\\xyz' or 'a\\b').

** For default values, please enter just a single value, without backslash escaping or quotes, using this format: a

?

Contoh Tabel

The screenshot shows the phpMyAdmin interface for a table named 'kelas3' in a database named 'data' on a localhost server. The interface includes a navigation menu on the left, a main content area with a table structure overview, and a bottom section with indexes, space usage, and row statistics.

Server: localhost Database: data Table: kelas3

Structure Browse SQL Search Insert Export Operations Empty Drop

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	int(11)			No		auto_increment	
<input type="checkbox"/> nama	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> email	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> alamat	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> kota	varchar(20)	latin1_swedish_ci		No			

Check All / Uncheck All With selected:

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After id Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	2		id

Create an index on 1 columns Go

Space usage:

Type	Usage
Data	212 Bytes
Index	2,048 Bytes
Overhead	60 Bytes
Effective	2,200 Bytes
Total	2,260 Bytes

[Optimize table](#)

Row Statistics:

Statements	Value
Format	dynamic
Collation	latin1_swedish_ci
Rows	2
Row length o	76
Row size o	1,130 Bytes
Next Autoindex	5
Creation	Oct 14, 2005 at 05:08 PM
Last update	Oct 14, 2005 at 05:17 PM