



BASIS DATA

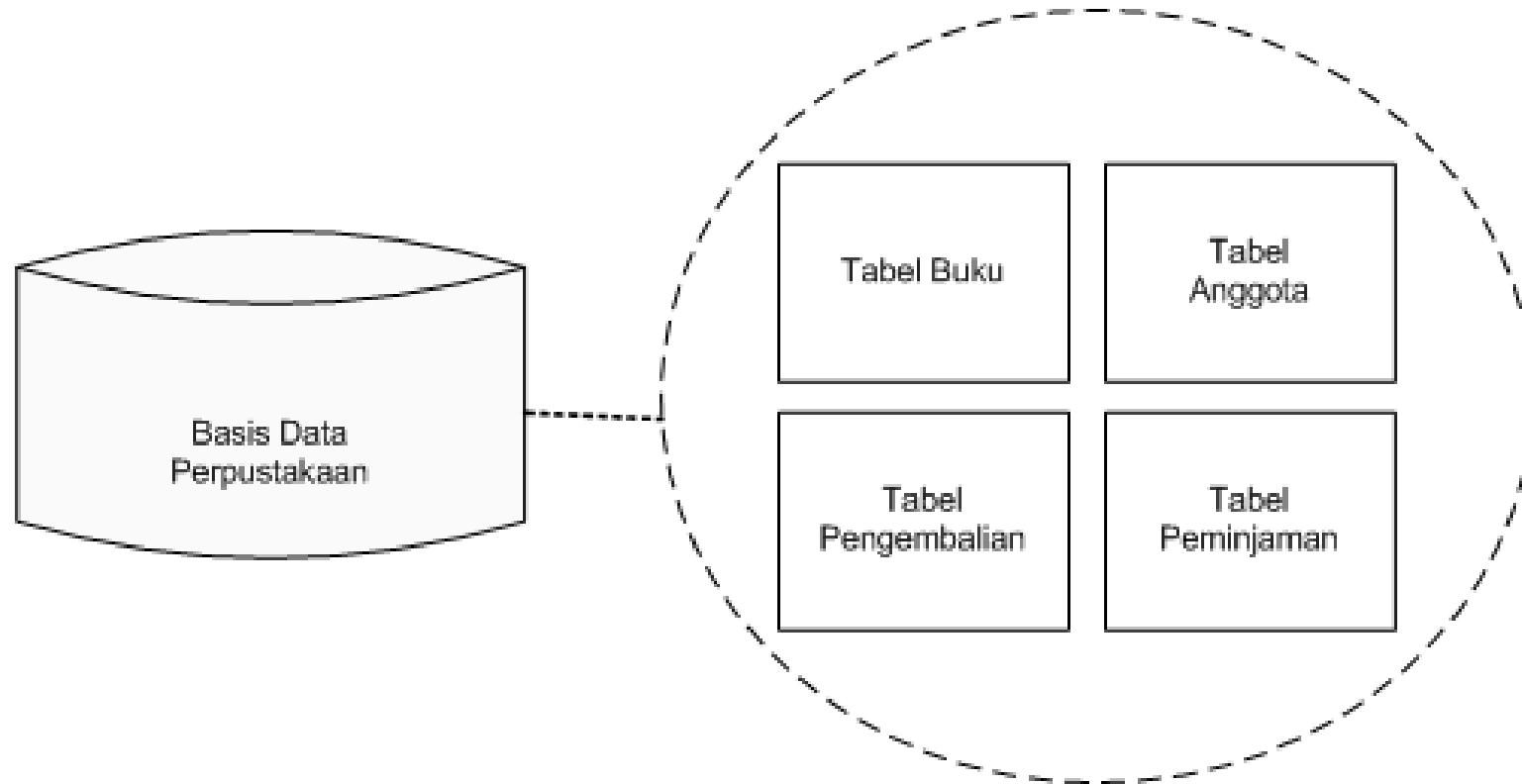
Pengenalan Basis Data



Basis Data (Database), pada saat ini sangat berdampak besar pada perkembangan ekonomi dan masyarakat. Sistem basis data berkaitan penting dalam pengembangan bidang rekayasa perangkat lunak, dan database menjadi kerangka kerja yang mendasari sistem informasi dan secara mendasar merubah cara banyak organisasi beroperasi.

Contoh Penggunaan Basis Data pada aplikasi:

aplikasi pengelolaan nomor telepon, aplikasi pembayaran gaji perusahaan, dll.



Isi dari Basis Data
Perpustakaan

Konsep Dasar Basis Data



BASIS dapat diartikan sebagai markas atau gudang, tempat bersarang (berkumpul).

DATA adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

BASIS DATA (DATABASE) adalah himpunan kelompok data/ kumpulan data yang saling berhubungan secara logis dan deskripsinya, yang disimpan secara bersama sedemikian rupa dan dirancang untuk memenuhi kebutuhan informasi organisasi.

Prinsip Dan Tujuan Basis Data



- Prinsip utamanya adalah pengaturan data/arsip.
- Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan data/arsip.

Yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokkan, pengorganisasian data yang akan kita simpan sesuai fungsi/jenisnya. Pengorganisasian data tersebut dapat dalam bentuk tabel terpisah atau dalam bentuk pendefinisian kolom (field) data dalam setiap tabel.

Operasi Dasar Basis Data



Operasi dasar yang dapat kita lakukan pada basis data , adalah :

1. Create database
2. Drop database
3. Create table
4. Drop table
5. Insert
6. Query
7. Update
8. Delete

Sistem Basis Data



Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.

Sistem Basis Data merupakan sistem yang terdiri atas kumpulan tabel data yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi tabel-tabel data tersebut.

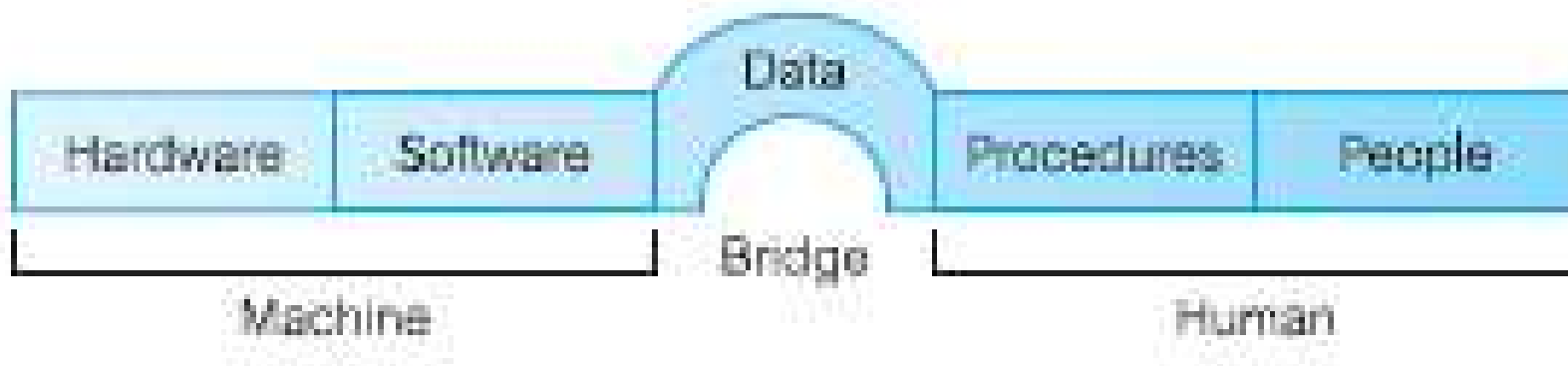
Database Management System (DBMS)



DBMS adalah perangkat lunak yang memungkinkan pemakai untuk mendefinisikan, mengelola, dan mengontrol akses ke basis data. DBMS yang mengelola basis data relational disebut dengan Relational DBMS (**RDBMS**)

Contoh perangkat lunak yang termasuk DBMS: dBase, FoxBase, Rbase, Microsoft-Access, Borland Paradox / Borland Interbase, MS-SQL Server, Oracle, Informix, Sybase, MySQL, dll.

Komponen Sistem Basis Data



Peran Dalam Lingkungan Database



Terdiri dari :

1. Data dan Database Administrator
2. Database Designer
3. Application Developers
4. End-User

Bahasa Basis Data



Cara berinteraksi antara pemakai dengan basis data diatur dalam suatu bahasa yang ditetapkan oleh pembuat DBMS.

Dua bentuk Bahasa yaitu :

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)

Keuntungan DBMS



1. Pengontrolan kerangkapan data
2. Konsistensi data
3. Lebih banyak informasi dari jumlah data yang sama
4. Sharing data
5. Peningkatan integrasi data
6. Peningkatan keamanan\
7. Penegakan standar layanan

Kekurangan DBMS



1. Kompleksitas
2. Ukuran
3. Biaya DBMS
4. Biaya Perangkat keras tambahan
5. Biaya konversi teknologi
6. Performa
7. Dampak kegagalan yang lebih besar

Basis Data Relational



Adalah basis data yang mempresentasikan data dalam bentuk tabel-tabel, dimana tabel-tabel tersebut dihubungkan oleh nilai-nilai yang sama/umum pada kolom-kolom terkait.

Komponen penyusun basis data :

1. Tabel
2. Kolom/atribut
3. Baris/tuple
4. Domain



1. Tabel

Tabel memiliki nama dan terdiri atas baris dan kolom. Tabel pada suatu basis data tidak boleh memiliki nama yang sama (unik). Tabel disebut juga dengan Relation atau File.

2. Kolom/Atribut

Kolom memiliki nama. Kolom yang terdapat dalam suatu tabel tidak boleh memiliki nama yang sama. Urutan nama boleh sembarang dan tidak mempengaruhi makna dari tabel. Nama lain kolom adalah Field atau Atribut.

3. Domain

Adalah sekumpulan nilai-nilai yang dapat disimpan pada satu atau lebih kolom. Sebuah domain bisa dimiliki oleh satu kolom atau lebih, tetapi sebuah kolom hanya memiliki satu domain. Karena domain membatasi dan mengatur nilai yang dapat disimpan maka disebut **domain constraint**.

4. Baris

Berisikan data dari sebuah objek. Baris pada sebuah tabel harus unik, dapat diletakkan dalam urutan bebas dan tidak mempengaruhi makna dari tabel. Baris disebut juga dengan Record atau tuple.

Perancangan Basis Data



Proses pembangunan basis data terdiri dari dua tahapan utama :

1. Tahapan analisis dan perancangan

Adalah tahapan pemetaan atau pembuatan model dari dunia nyata menggunakan notasi perancangan basis data tertentu serta pembuatan deskripsi implementasi basis data.

2. Tahapan implementasi

Tahapan ini mengimplementasikan rancangan basis data yang telah dibuat. Implementasi menggunakan aplikasi klien yang disediakan oleh DBMS terpilih.

Model Data



PENGERTIAN MODEL DATA :

Sekumpulan konsep-konsep untuk menerangkan data, hubungan-hubungan antara data, makna data (semantik) dan batasan data.

JENIS-JENIS MODEL DATA

- A. Model Data Berdasarkan Object
- B. Model Data Berdasarkan Record

A. Model Data Berbasis Objek



Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas.

Terdiri dari:

1. Model Keterhubungan Entitas (*Entity-Relationship Model*)
2. Model Berorientasi Object (*Object-Oriented Model*)
3. Model Data Semantik (*Semantic Data Model*)
4. Model Data Fungsional (*Functional Data Model*)

Model Keterhubungan Entitas (*Entity-Relationship Model*) merupakan model yang paling populer digunakan dalam perancangan basis data.

B. Model Data Berbasis Record



Model ini berdasarkan pada record untuk menjelaskan kepada user tentang hubungan logic antar data dalam basis data

PERBEDAAN DENGAN MODEL DATA BERBASIS OBJEK

Pada record based data model disamping digunakan untuk menguraikan struktur logika keseluruhan dari suatu database, juga digunakan untuk menguraikan implementasi dari sistem database (higher level description of implementation)

Model Relational



Terdapat 3 data model pada model data berbasis record:

1. Model Relational,

Dimana data serta hubungan antar data direpresentasikan oleh sejumlah tabel dan masing-masing tabel terdiri dari beberapa kolom yang namanya unique. Model ini berdasarkan notasi teori himpunan (set theory), yaitu relation.

2. Model Hirarki

Dimana data serta hubungan antar data direpresentasikan dengan record dan link (pointer), dimana record-record tersebut disusun dalam bentuk tree (pohon), dan masing-masing node pada tree tersebut merupakan record/grup data elemen dan memiliki hubungan cardinalitas 1:1 dan 1:M

3. Model Jaringan

Distandarisasi tahun 1971 oleh Database Task Group (DBTG) atau disebut juga model CODASYL (Conference on Data System Language), mirip dengan hirarkical model dimana data dan hubungan antar data direpresentasikan dengan record dan links. Perbedaannya terletak pada susunan record dan linknya yaitu network model menyusun record-record dalam bentuk graph dan menyatakan hubungan cardinalitas 1:1, 1:M dan N:M

Simbol-simbol dalam E-R Diagram



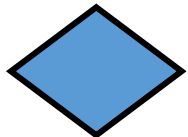
Notasi Arti



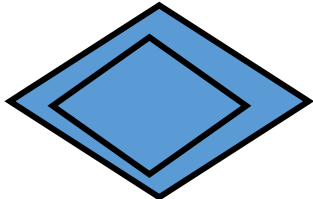
- ENTITY



- WEAK ENTITY



- RELATIONSHIP

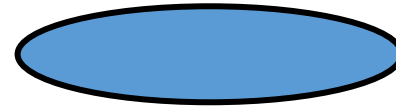


- IDENTIFYING RELATIONSHIP

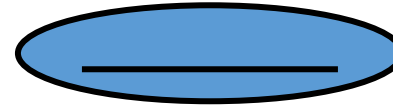


- ATRIBUT DERIVATIF

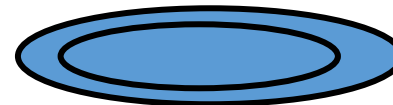
Notasi Arti



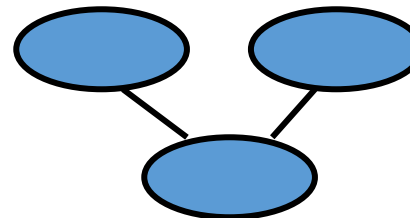
- ATRIBUT



- ATRIBUT PRIMARY KEY

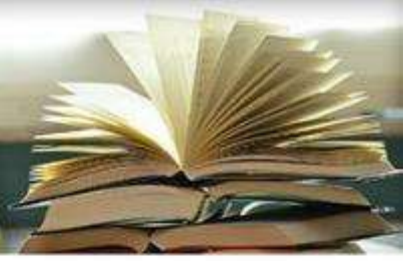


- ATRIBUT MULTI VALUE



- ATRIBUT COMPOSITE

Komponen E-R Diagram



1. Entitas yaitu suatu kumpulan object atau sesuatu yang dapat dibedakan atau dapat diidentifikasi secara unik. Dan kumpulan entitas yang sejenis disebut dengan entity set.
2. Relationship yaitu hubungan yang terjadi antara satu entitas atau lebih.
3. Atribut, kumpulan elemen data yang membentuk suatu entitas.
4. Indicator tipe terbagi 2 yaitu :
 - a. Indicator tipe asosiatif object
 - b. Indicator tipe super tipe

Entity Set



ENTITY SET TERBAGI ATAS :

1. Strong entity set yaitu entity set yang satu atau lebih atributnya digunakan oleh entity set lain sebagai key. Digambarkan dengan empat persegi panjang.

Misal :

E adalah sebuah entity set dengan attribute-attribute a_1, a_2, \dots, a_n , maka entity set tersebut direpresentasikan dalam bentuk tabel E yang terdiri dari n kolom, dimana setiap kolom berkaitan dengan attribute-atributenya.

2. Weak Entity set, Entity set yang bergantung terhadap strong entity set. Digambarkan dengan empat persegi panjang bertumpuk.

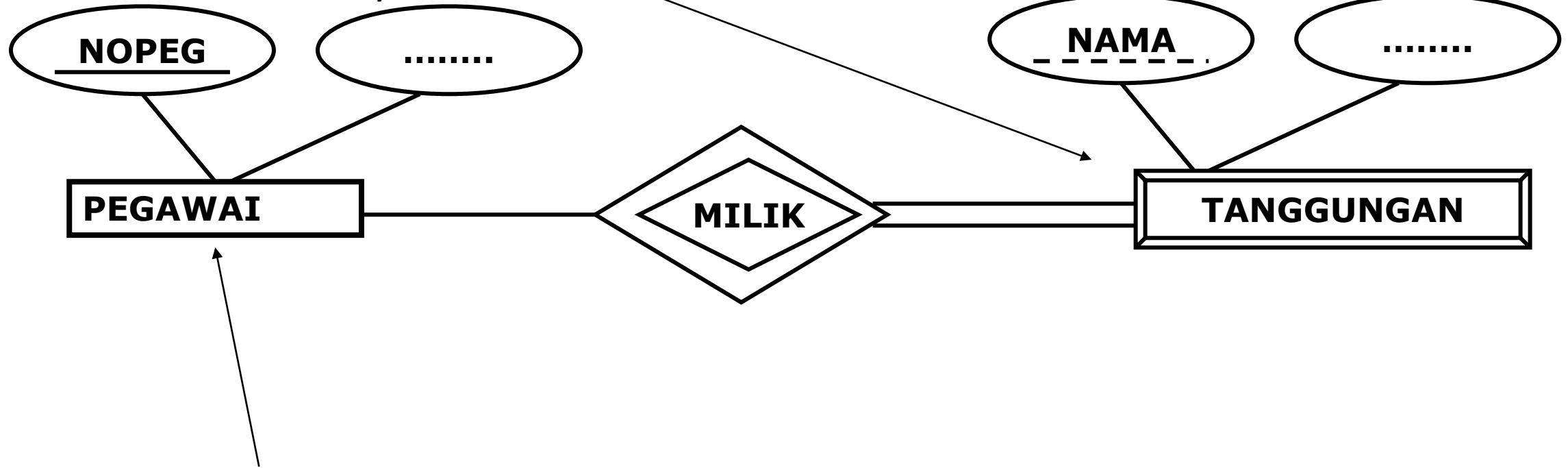
Misal :

A adalah weak entity set dari attribute-attribute a_1, a_2, \dots, a_r dan B adalah strong entity set dengan attribute-attribute b_1, b_2, \dots, b_s , dimana b_1 adalah attribute primary key, maka weak entity set direpresentasikan berupa table A, dengan attribute-attribute $\{b_1\} \cup \{a_1, a_2, \dots, a_r\}$

Entity Set lanjutan



Contoh Weak Entity



Contoh Strong Entity

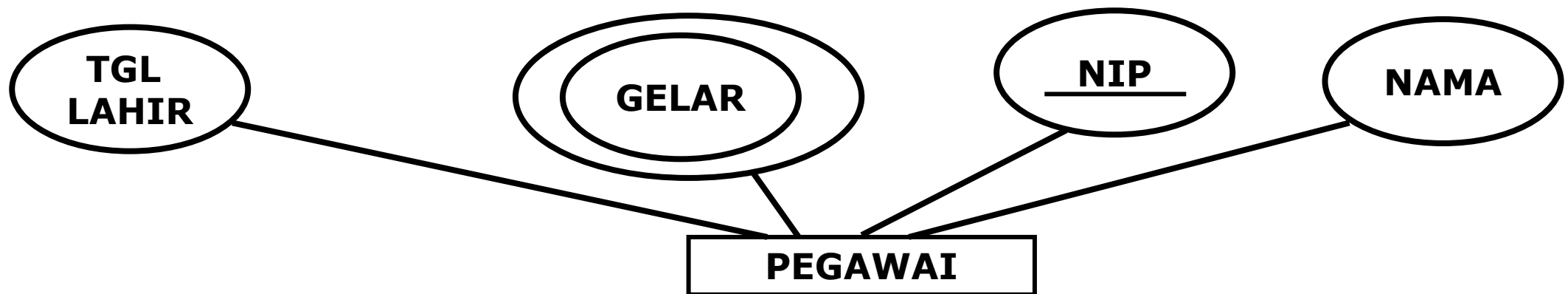
Jenis-Jenis Atribut



- KEY → atribut yang digunakan untuk menentukan suatu entity secara unik
- ATRIBUT SIMPLE → atribut yang bernilai tunggal
- ATRIBUT MULTI VALUE → atribut yang memiliki sekelompok nilai untuk setiap instan entity

Pada gambar dibawah ini, yang menjadi atribut key adalah NIP.

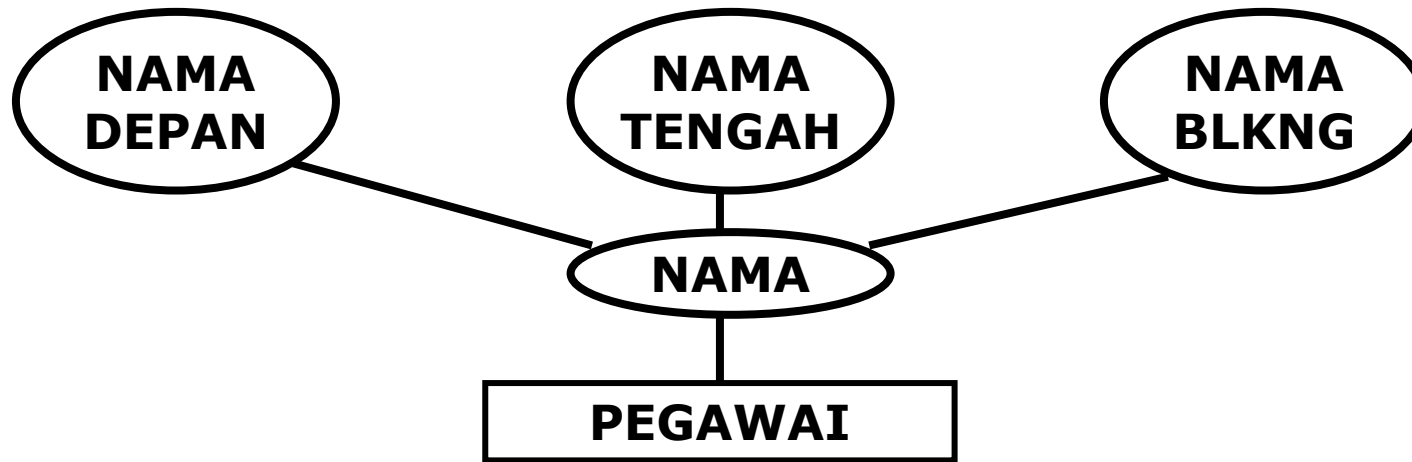
Tgl Lahir dan Nama adalah atribut simple. Sedangkan Gelar merupakan contoh atribut multivalued.





Jenis Atribut lanjutan

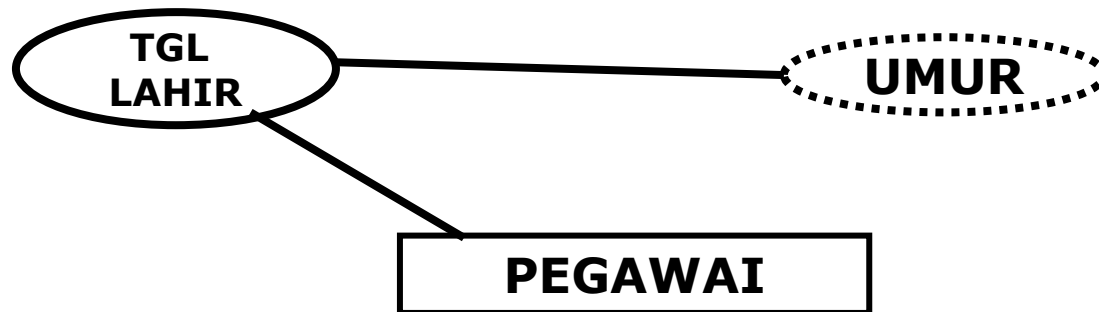
- d. **ATRIBUT COMPOSIT** → Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu contohnya adalah atribut nama pegawai yang terdiri dari nama depan, nama tengah dan nama belakang.



Jenis Atribut lanjutan



- d. **ATRIBUT DERIVATIF** → Suatu atribut yg dihasilkan dari atribut yang lain. Sehingga umur yang merupakan hasil kalkulasi antara Tgl Lahir dan tanggal hari ini. Sehingga keberadaan atribut umur bergantung pada keberadaan atribut Tgl Lahir.



Mapping Cardinality



Banyaknya entity yang bersesuaian dengan entity yang lain melalui relationship

JENIS-JENIS MAPPING :

1. One to one
2. Many to One atau One to many
3. Many to many

REPRESENTASI DARI ENTITY SET

Entity set direpresentasikan dalam bentuk tabel dan nama yang unique. Setiap tabel terdiri dari sejumlah kolom, dimana masing-masing kolom diberi nama yang unique pula



Participation Constraint

Menjelaskan apakah keberadaan suatu entity tergantung pada hubungannya dengan entity lain.

Terdapat dua macam participation constrain yaitu:

1. Total participation constrain yaitu:

Keberadaan suatu entity tergantung pada hubungannya dengan entity lain. Didalam diagram ER digambarkan dengan dua garis penghubung antar entity dan relationship.

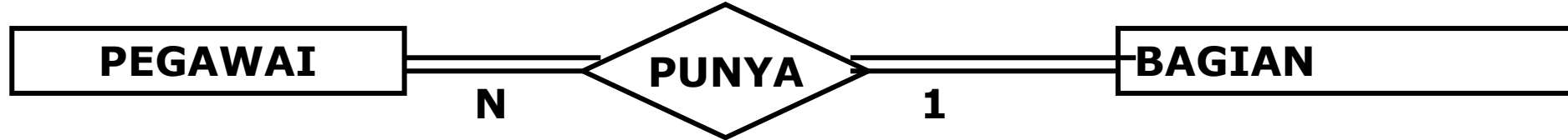
2. Partial participation, yaitu

Keberadaan suatu entity tidak tergantung pada hubungan dengan entity lain. Didalam diagram ER digambarkan dengan satu garis penghubung.

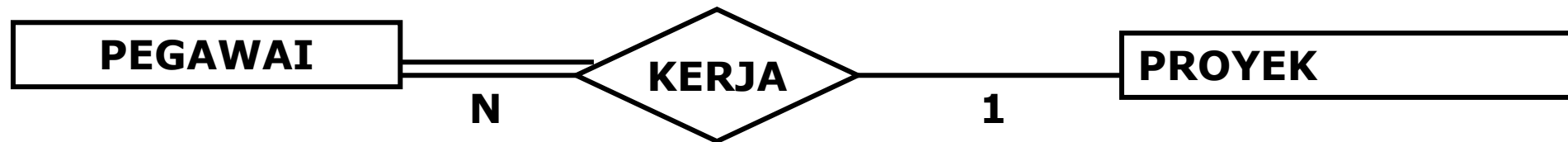
Contoh Participation Constraint



a. TOTAL PARTICIPATION



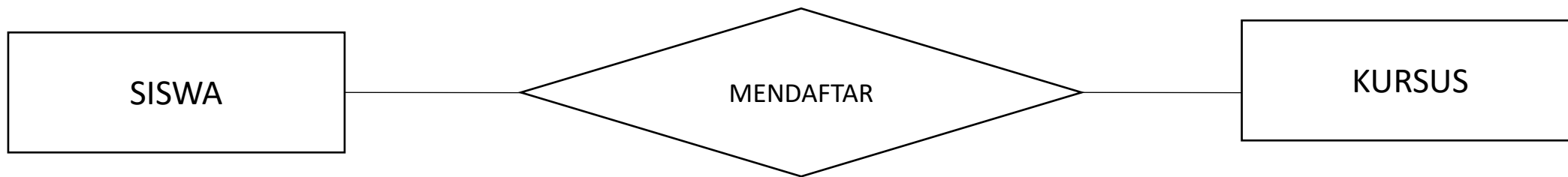
b. PARTIAL PARTICIPATION



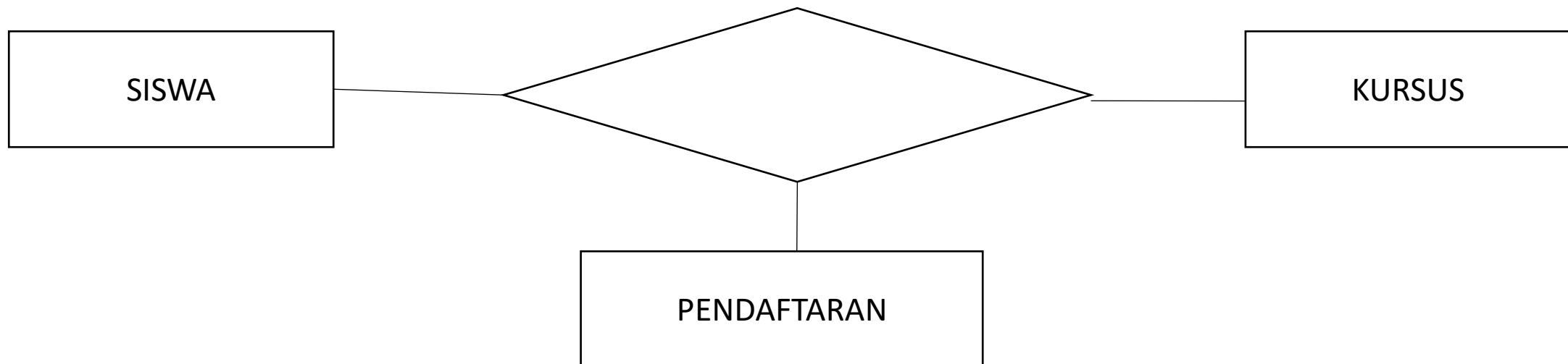
Indicator Type



Indicator tipe asosiatif object berfungsi sebagai suatu objek dan suatu relationship.



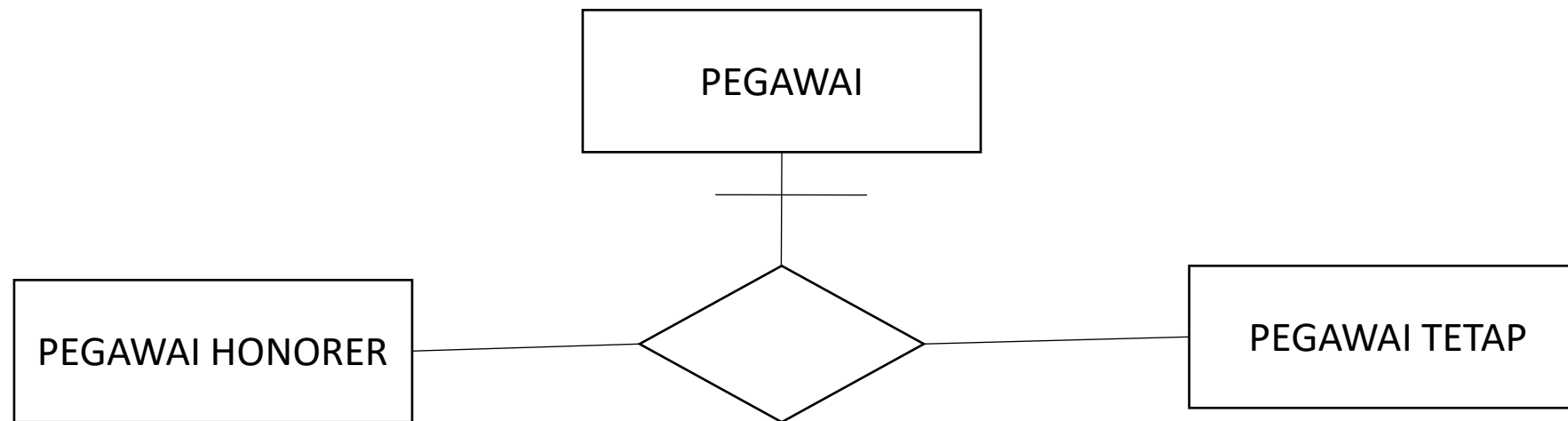
Berubah menjadi



Indicator Tipe lanjutan



Indicator tipe super tipe, terdiri dari suatu object dan satu subkategori atau lebih yang dihubungkan dengan satu relationship yang tidak bernama.



Tahapan Pembuatan ERD



1. Identifikasi dan tetapkan seluruh himpunan entitas yang akan terlibat
2. Tentukan atribut key dari masing-masing himpunan entitas
3. Identifikasi dan tetapkan seluruh himpunan relasi antar himpunan entitas yang ada beserta foreign key-nya
4. Tentukan derajat/kardinalitas relasi untuk setiap himpunan relasi
5. Lengkapi himpunan entitas dan himpunan relasi dengan atribut bukan kunci.

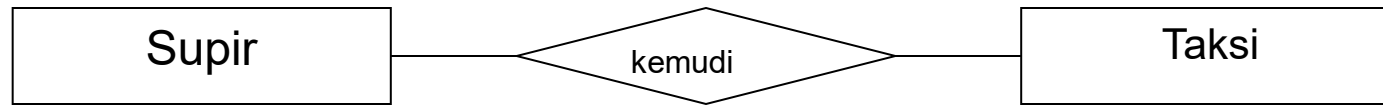
Logical Record Structured (LRS)



LRS → representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas.

Menentukan Kardinalitas, Jumlah Tabel dan Foreign Key (FK)

One to One (1-1)



Gambar di atas menunjukkan relasi dengan kardinalitas 1-1, karena: **1 supir hanya bisa mengemudikan 1 taksi, dan 1 taksi hanya bisa dikemudikan oleh 1 supir.**

Relasi 1-1 akan membentuk 2 tabel:

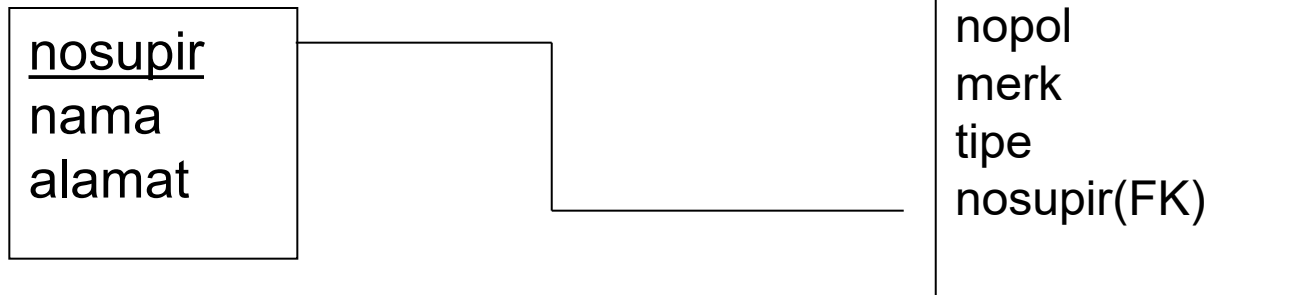
Tabel Supir (nosupir, nama, alamat)

Tabel Taksi (notaksi, nopol, merk, tipe)

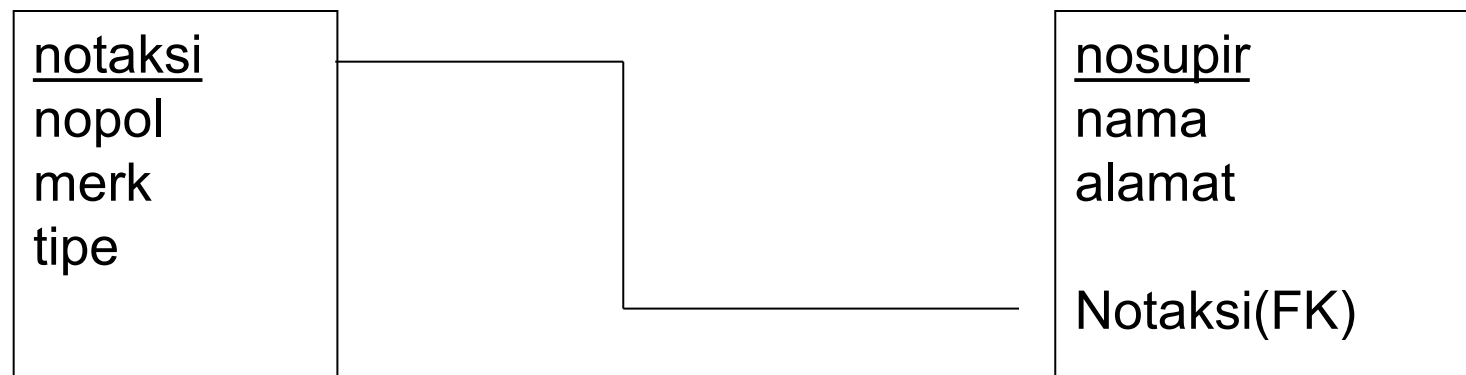
LRS lanjutan



LRS yang terbentuk sbb:



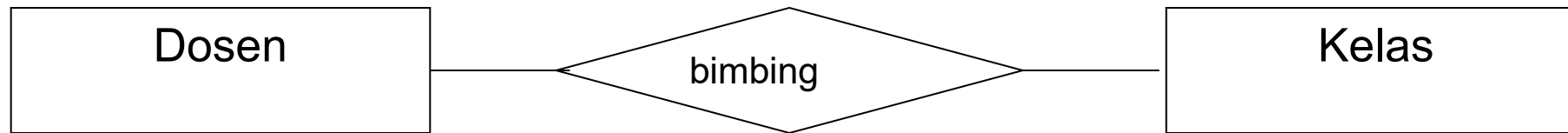
atau



LRS lanjutan



One to Many (1-M)



Gambar di atas menunjukkan relasi dengan kardinalitas 1-M, karena:
**1 Dosen bisa membimbing banyak Kelas, dan
1 Kelas hanya dibimbing oleh 1 Dosen.**

Relasi 1-M akan membentuk 2 tabel:

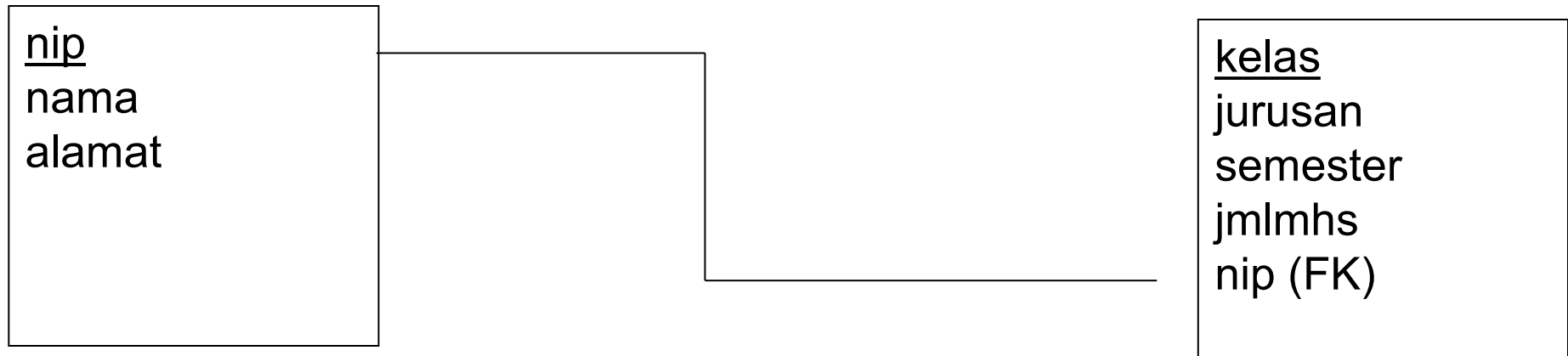
Tabel Dosen (nip, nama, alamat)

Tabel Kelas (kelas, jurusan, semester, jmlmhs)

LRS lanjutan



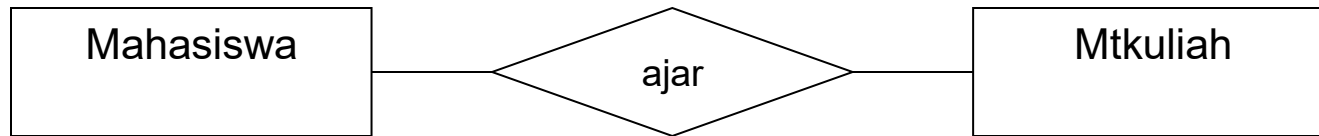
LRS yang terbentuk sbb:



LRS lanjutan



Many to Many (M-M)



Gambar di atas menunjukkan relasi dengan kardinalitas M-M, karena:

**1 Mahasiswa bisa belajar banyak Mata Kuliah, dan
1 Mata Kuliah bisa dipelajari oleh banyak Mahasiswa.**

Relasi M-M akan membentuk 3 tabel:

Tabel Mahasiswa (nim, nama, alamat)

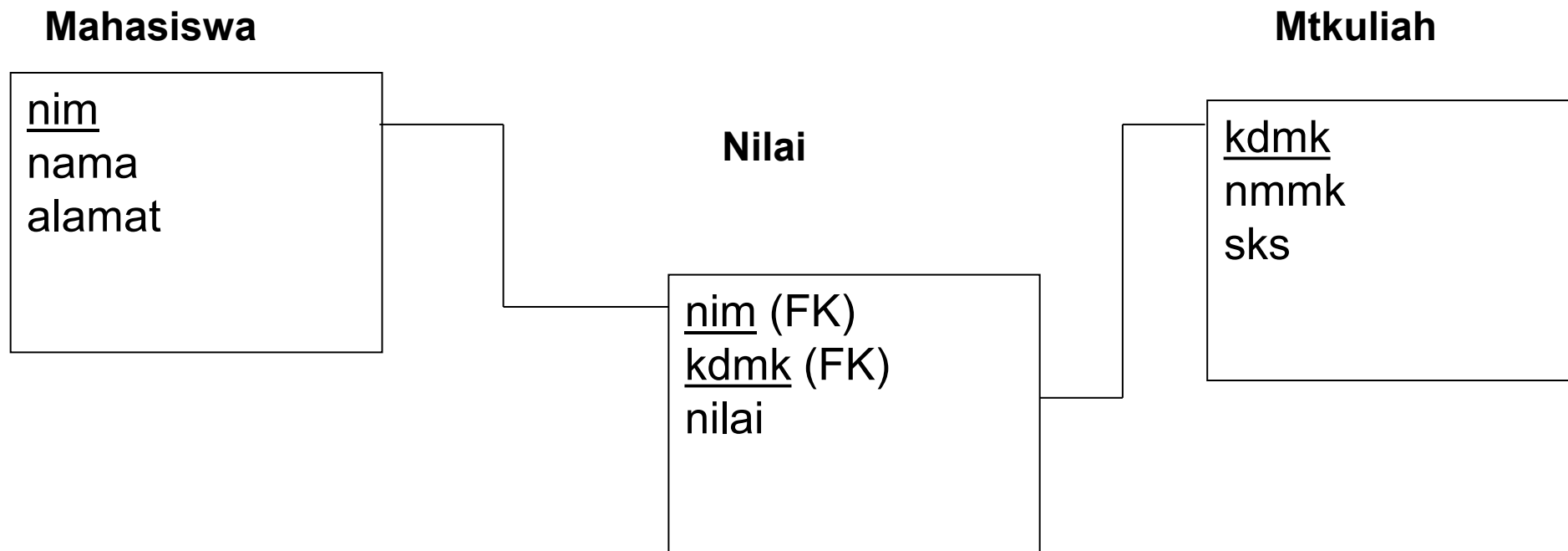
Tabel Mtkuliah (kdmk, nmmk, sks)

Tabel Nilai (nim, kdmk, nilai) → menggunakan super key/composite key

LRS lanjutan



LRS yang terbentuk sbb:



Penyelesaian



Langkah 1 : Menentukan Entitas

Entitas yang dibutuhkan adalah : Bagian, Pegawai, Pengawas, dan Proyek

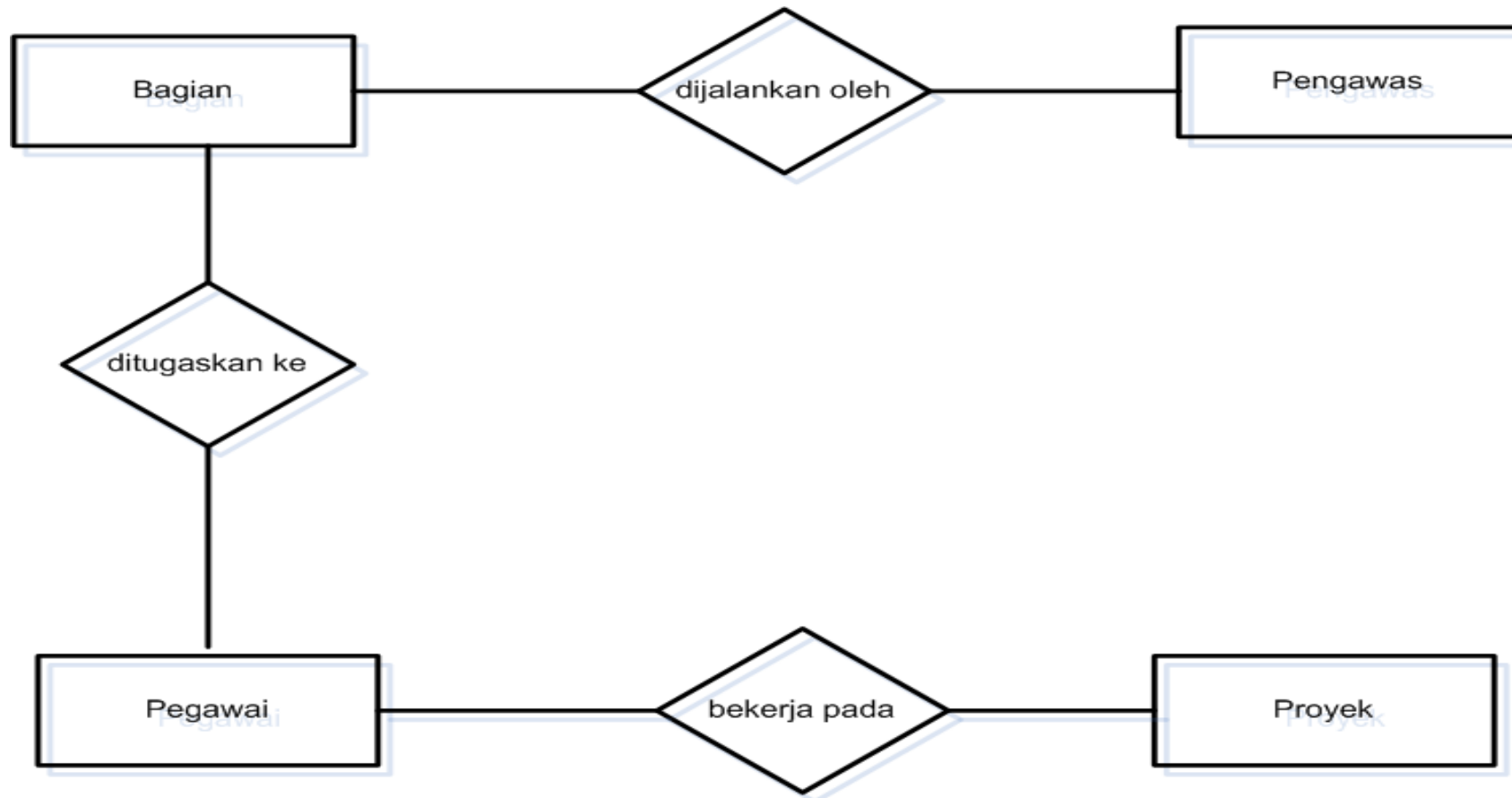
Langkah 2 :Menentukan Relasi dengan matriks relasi

	Bagian	Pegawai	Pengawas	Proyek
Bagian		ditugaskan ke	dijalankan oleh	
Pegawai	milik			bekerja pada
Pengawas	menjalankan			
Proyek		menggunakan		

Lanjutan



Langkah 3 : Menggambar ERD Sementara



Lanjutan



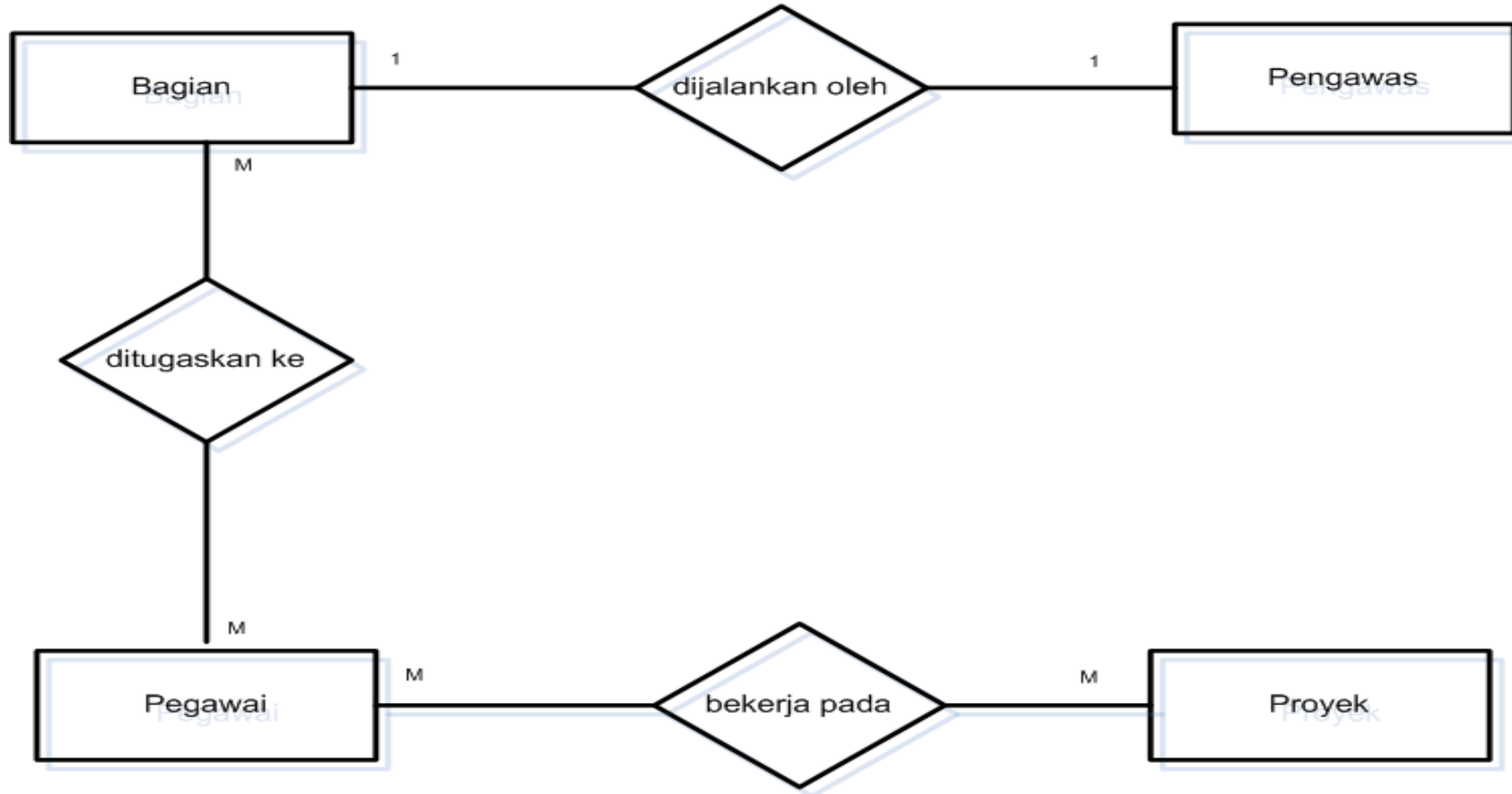
Deskripsi Permasalahan :

- Masing-masing bagian hanya mempunyai satu pengawas
- Seorang pengawas hanya bertugas pada satu bagian
- Masing-masing bagian memiliki paling tidak satu pegawai
- Masing-masing pegawai bekerja paling tidak pada satu bagian
- Masing-masing proyek dikerjakan oleh paling tidak satu pegawai
- Seorang Pengawas bisa mendapat tugas 0 atau beberapa proyek

Lanjutan



Langkah 4 : Mengisi Kardinalitas



Lanjutan



Langkah 5: Menentukan Kunci Utama

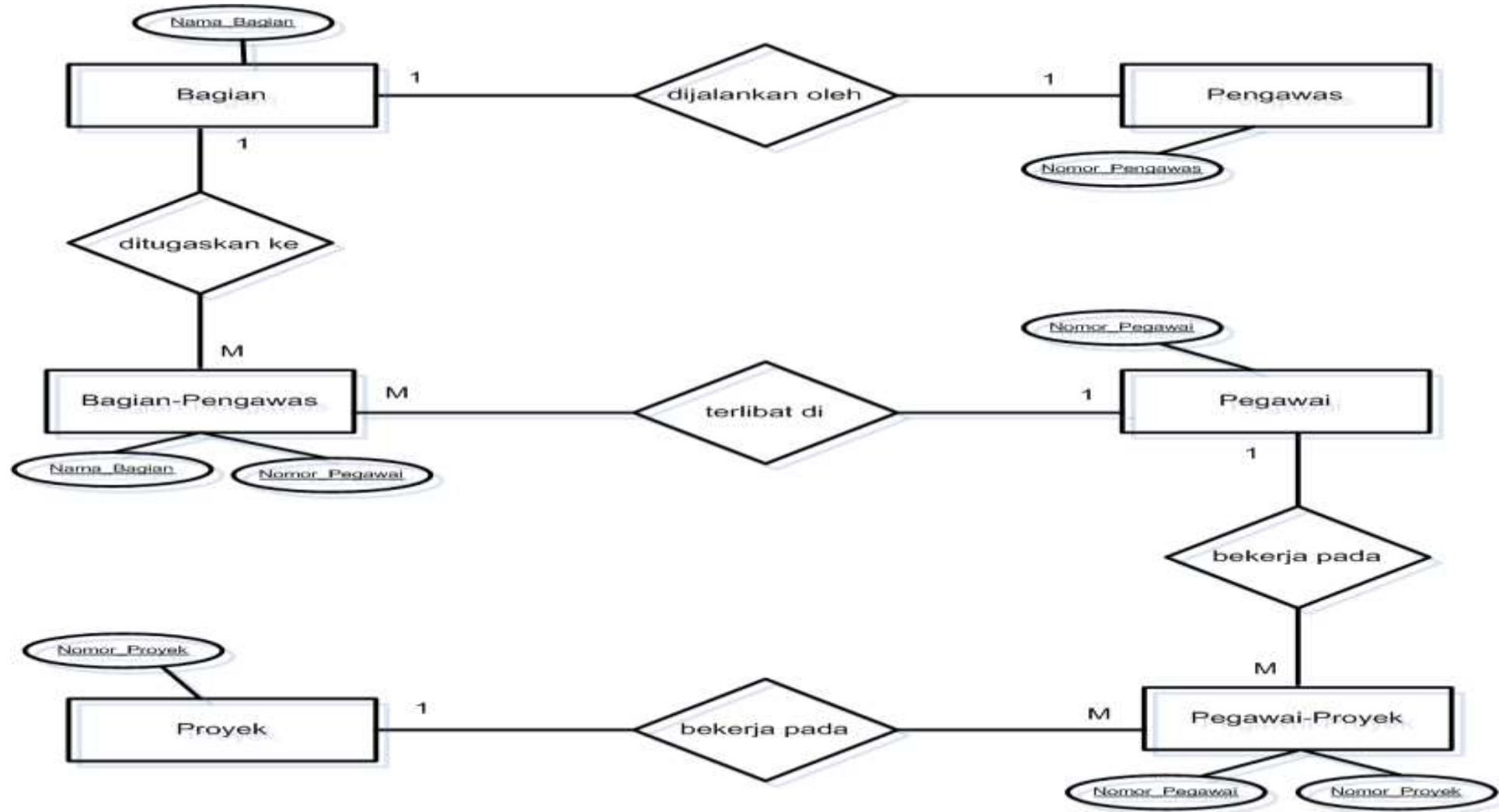
Kunci Utama : Nama Bagian, Nomor Pengawas, Nomor Pegawai, Nomor Proyek.

Langkah 6: Menggambarkan ERD berdasarkan kunci

Karena ada dua relasi many-to-many pada ERD sementara, yaitu antara Bagian dan Pegawai, serta Pegawai dan Proyek. Oleh karena itu dibuatkan entitas baru yaitu Bagian-Pegawai dan Pegawai-Proyek. Kunci utama Bagian-Pegawai adalah gabungan Nama Bagian dan Nomor Pegawai. Kunci utama Pegawai-Proyek adalah gabungan Nomor Pegawai dan Nomor Proyek

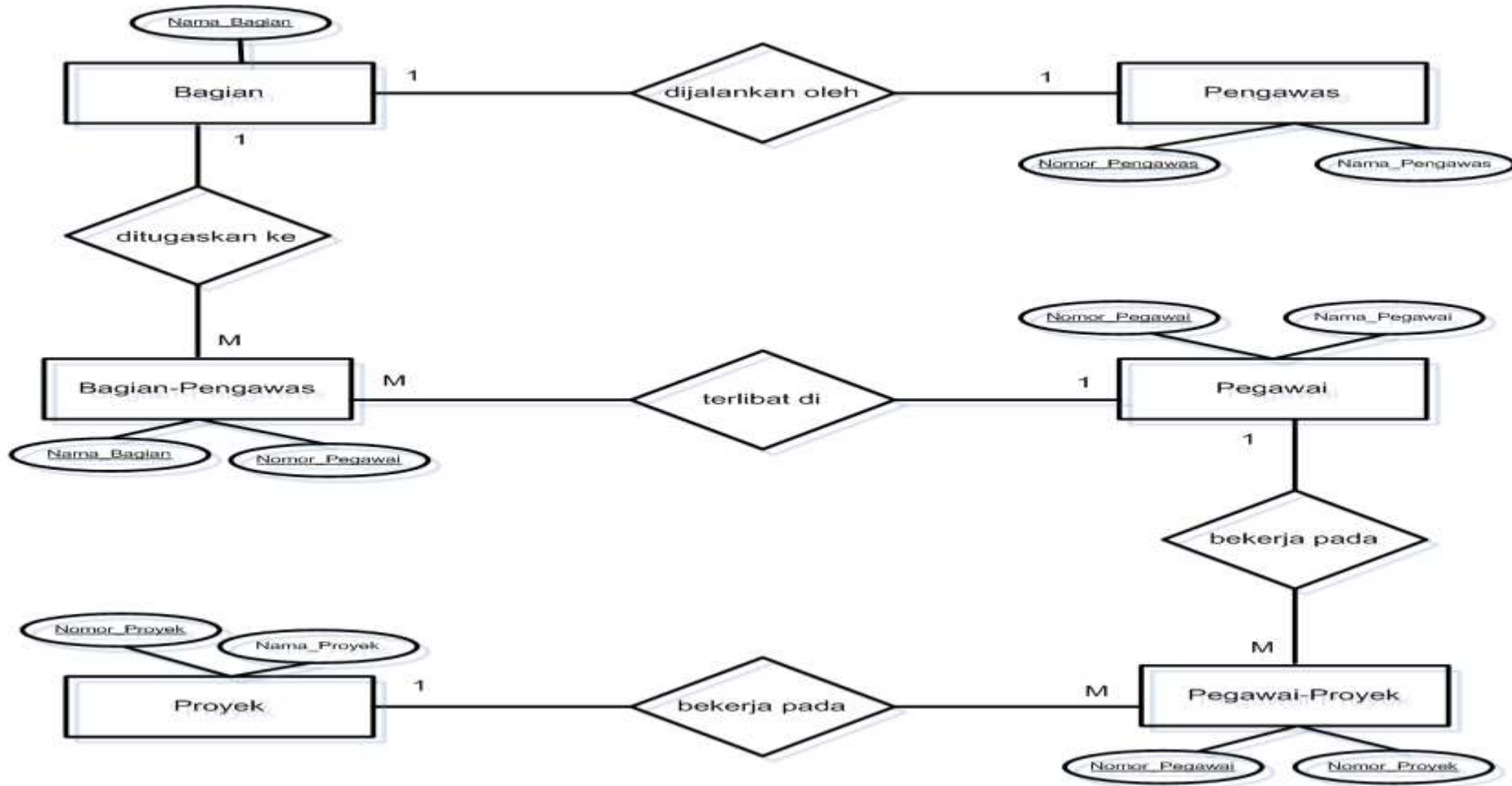
Lanjutan

Penggambaran ERD Berdasarkan Kunci

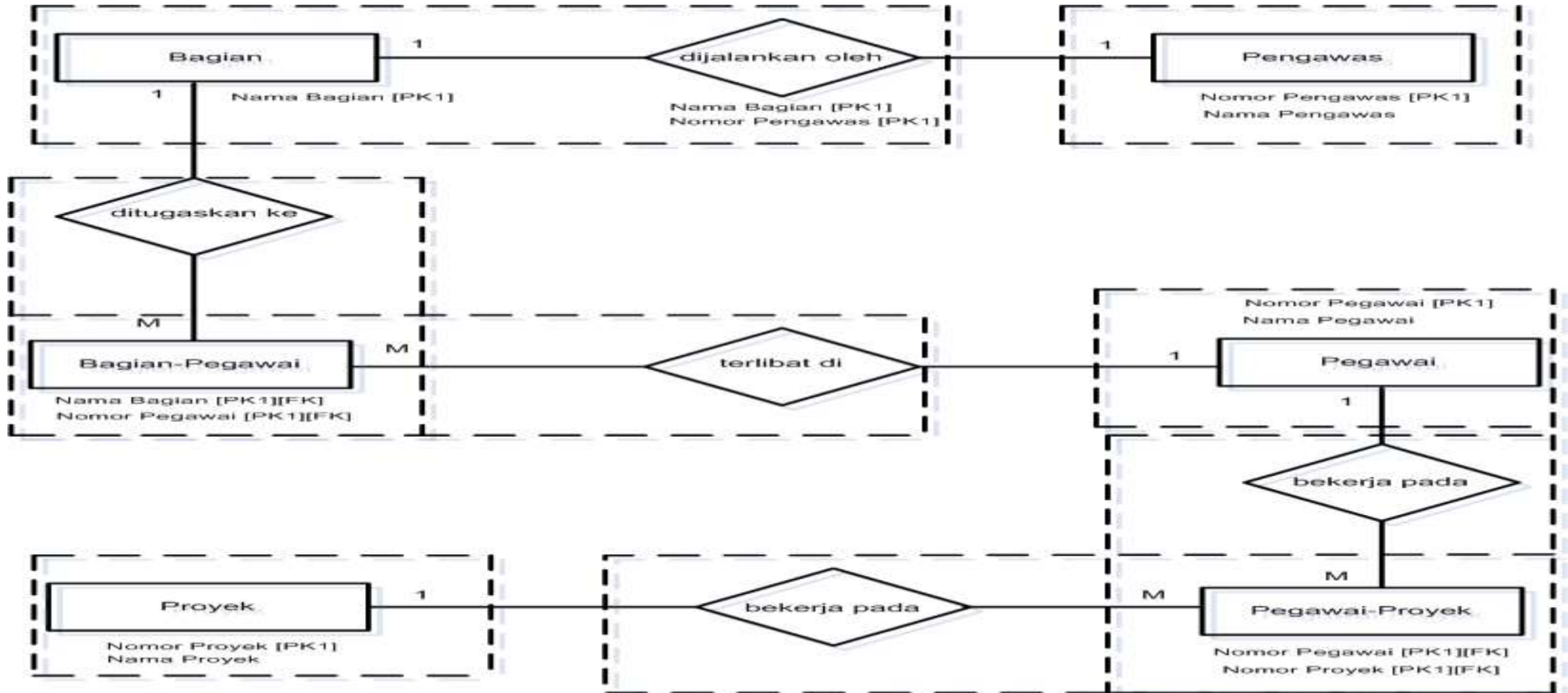


Lanjutan

Langkah 7: Menentukan Atribut yang diperlukan

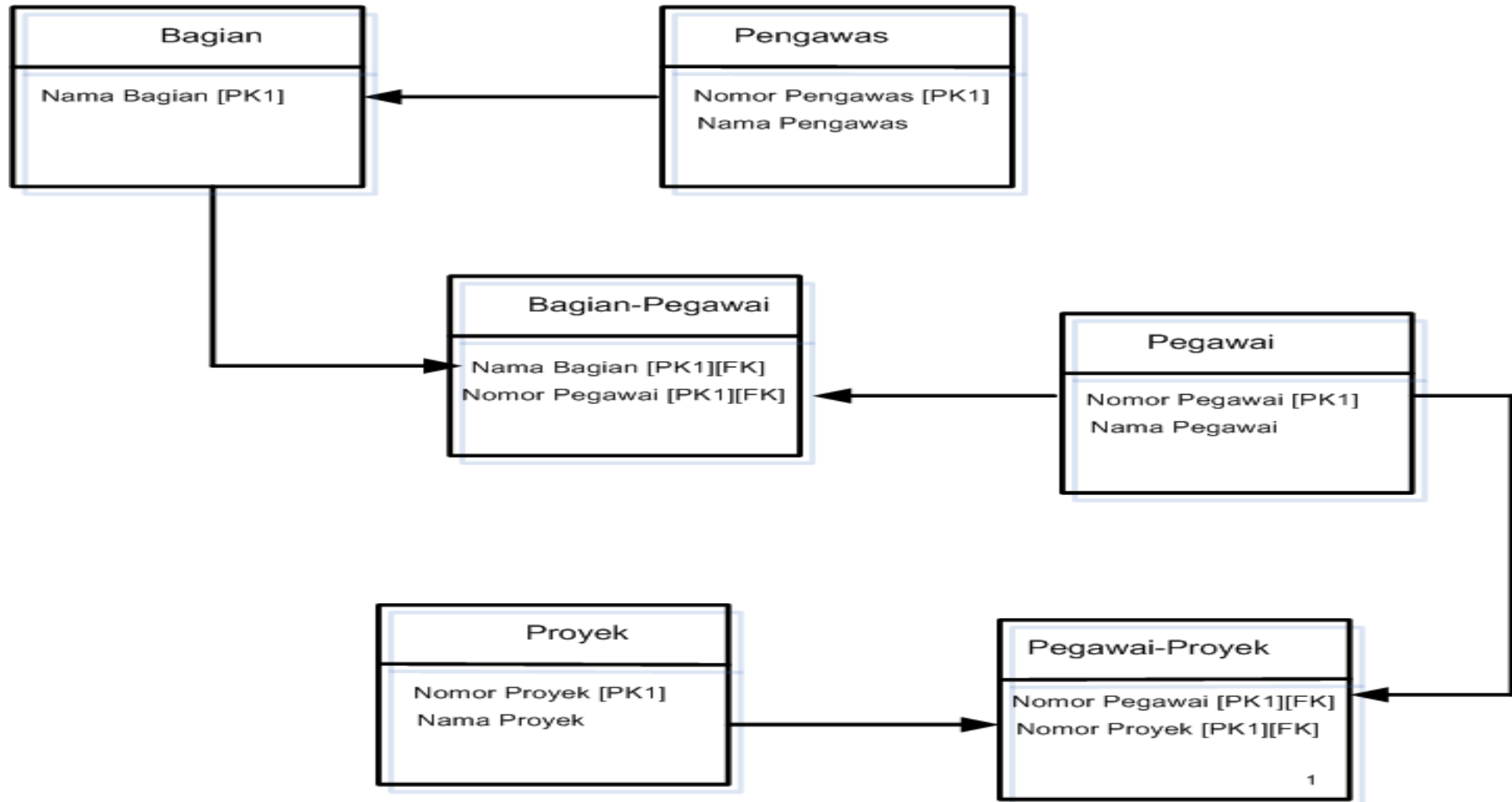


Transformasi ERD dan LRS



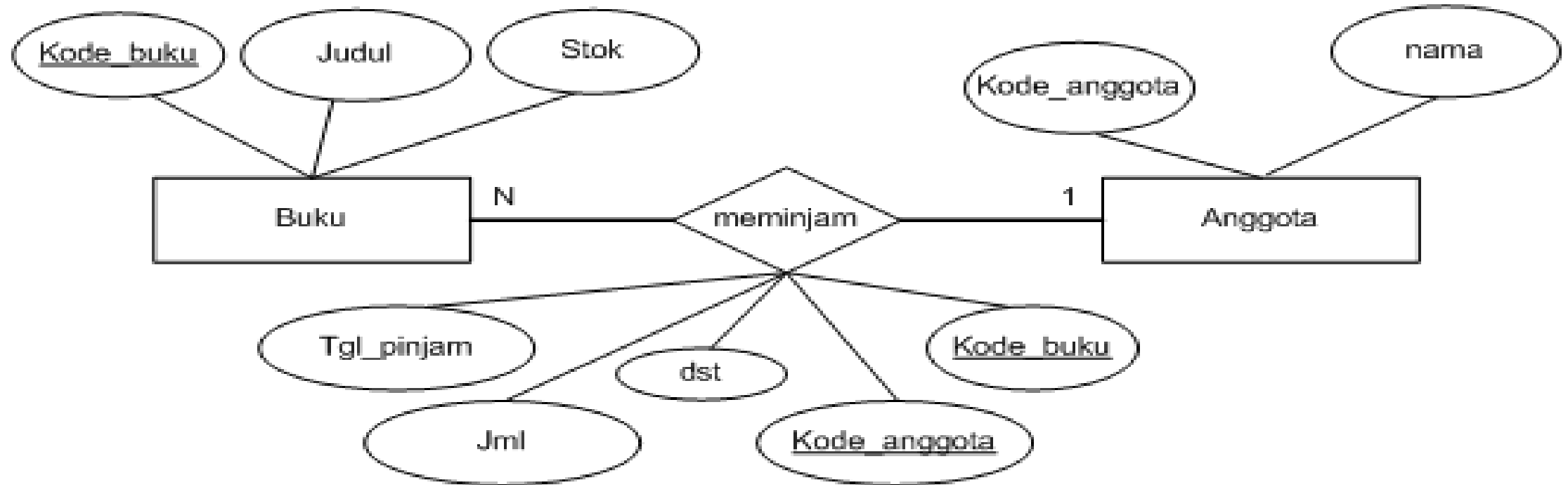
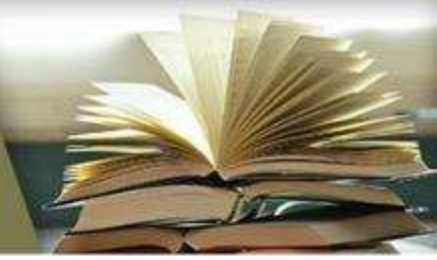
Lanjutan

LRS yang terbentuk



Contoh ERD

Perpustakaan Sederhana



Normalisasi



BEBERAPA PENGERTIAN NORMALISASI :

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel–tabel yang menunjuk-kan entity dan relasinya.

Normalisasi adalah proses pengelompokan attribute-attribute dari suatu relasi sehingga membentuk WELL STRUCTURE RELATION.

Keuntungan Normalisasi



Keuntungan dari normalisasi, yaitu :

1. Meminimalkan ukuran penyimpanan yang diperlukan untuk menyimpan data.
2. Meminimalkan resiko inkonsistensi data pada basis data
3. Meminimalkan kemungkinan anomali pembaruan
4. Memaksimalkan stabilitas struktur data

Bentuk Normal

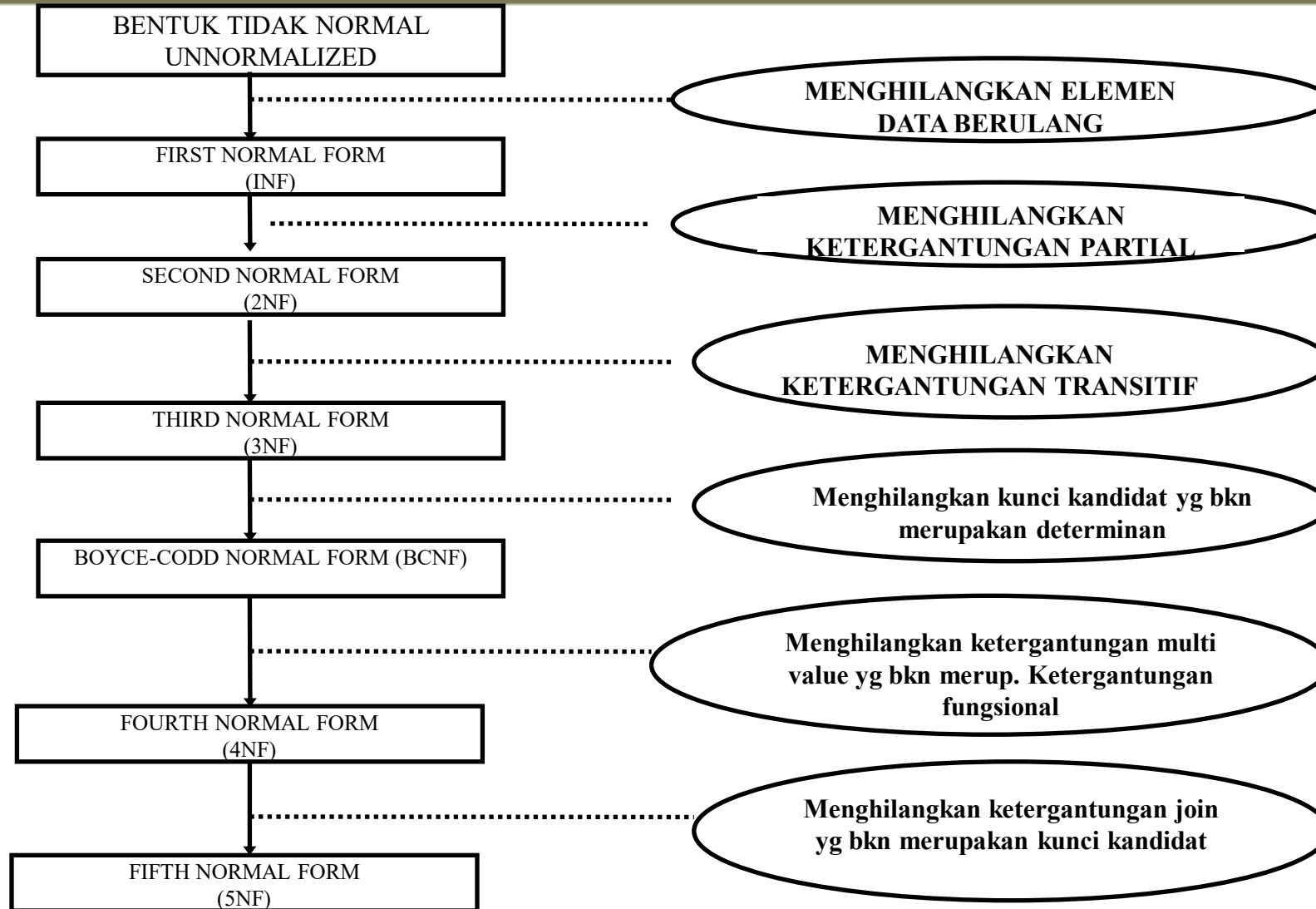


Aturan-aturan normalisasi dinyatakan dengan istilah bentuk normal. **Bentuk normal** adalah suatu aturan yang dikenakan pada relasi-relasi dalam basis data dan harus dipenuhi oleh relasi-relasi tersebut pada level-level normalisasi.

Beberapa level yang biasa digunakan pada normalisasi adalah:

- Bentuk normal pertama (1NF)
- Bentuk normal kedua (2NF)
- Bentuk normal ketiga (3NF)
- Bentuk normal Boyce-Codd (BCNF)
- Bentuk normal keempat (4NF)
- Bentuk Normal kelima (5NF)

Langkah-Langkah Pembuatan Normalisasi



Studi Kasus Perpustakaan



Daftar Anggota Perpustakaan

Kode Anggota	Nama
A01	Surya
A02	Fitri
A03	Syahrur

Daftar Buku Perpustakaan

Kode Buku	Judul	Stok Buku
B01	Pemograman C++	10
B02	Membuat Aplikasi 30 Menit	15
B03	Cooking is Easy	15

Bukti Peminjaman Buku

Tanggal Pinjam : 10 Januari 2019

No. Anggota : A01

No. Pinjam : PJ01

Kode Buku	Judul Buku	Jumlah Buku
B01	Pemograman C++	1
B02	Membuat Aplikasi 30 Menit	1
B03	Cooking is Easy	1

Tanggal Kembali : 13 Januari 2019

Dari ketiga dokumen tersebut buatlah normalisasinya.



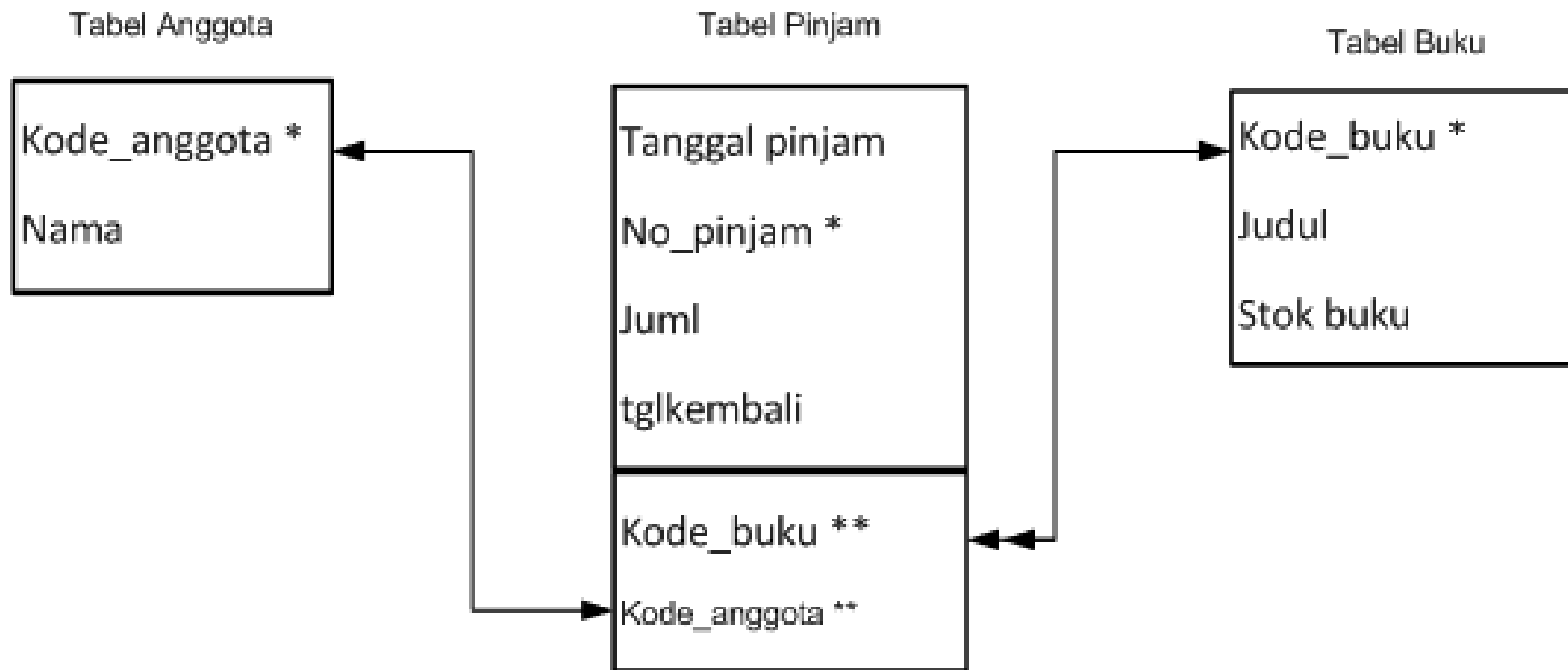
Bentuk Tidak Normal

Kode_anggota
Nama
Kode_buku
Judul
Stok buku
Tanggal pinjam
No_pinjam
kode_anggota
Kode_buku
Judul
Juml
tglkembali

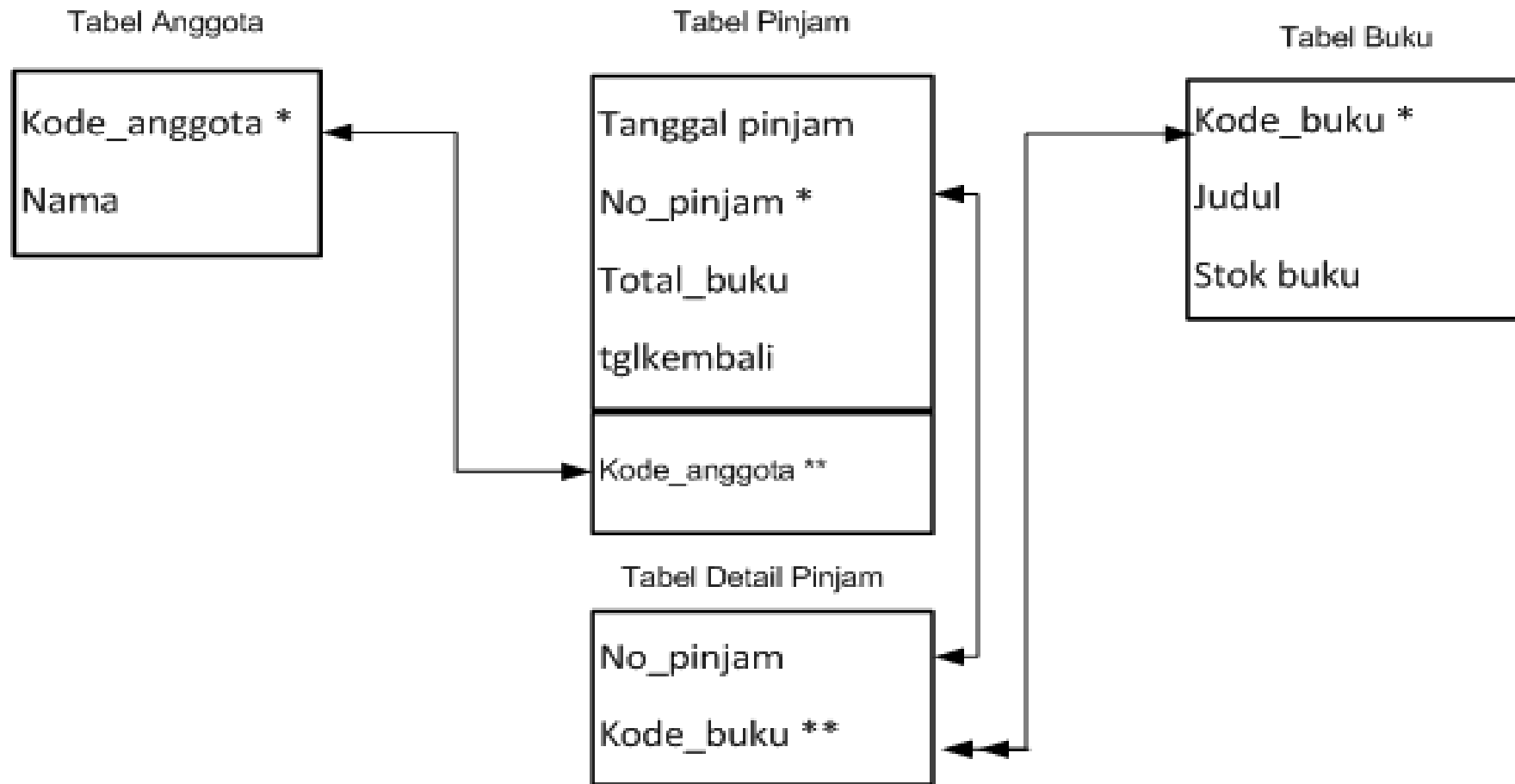
Bentuk 1NF

Kode_anggota *
Nama
Kode_buku *
Judul
Stok buku
Tanggal pinjam
No_pinjam *
kode_anggota
Kode_buku
Juml
tglkembali

2NF



3NF



Structured Query Language (SQL)



SQL merupakan bahasa query terapan yang banyak digunakan oleh berbagai DBMS, diterapkan dalam berbagai *development tools* dan program aplikasi untuk berinteraksi dengan basis data.

Subdivisi SQL:

1. Data Definition Language (DDL)

Query-query ini digunakan untuk mendefinisikan struktur atau skema basis data.

2. Data Manipulation Language (DML)

Query-query ini digunakan untuk manajemen data dalam basis data.

SQL lanjutan



PENGELOMPOKAN STATEMEN SQL

1. Data Definition Language (DDL)

CREATE DATABASE	DROP DATABASE
CREATE TABEL	DROP TABEL
CREATE INDEX	DROP INDEX
CREATE VIEW	DROP VIEW
ALTER TABLE	

2. Data Manipulation Language (DML)

INSERT, SELECT, UPDATE, DELETE

SQL lanjutan



3. Data Access

GRANT , REVOKE

4. Data Integrity

RECOVER TABLE

5. Auxiliary

SELECT INTO OUTFILE,
LOAD, RENAME TABLE

Data Definition Language (DDL)



A. CREATE

1. Pembuatan Database

Nama Database adalah yang dapat mewakili suatu kejadian dapat berupa nama organisasi atau perusahaan.

Sintaks : CREATE DATABASE nama_database

Contoh : Buat database dengan nama KAMPUS

```
CREATE DATABASE KAMPUS
```

2. Pembuatan Tabel

Sintaks : CREATE TABLE nama_table

```
( nama_kolom1 tipe_data_kolom1,  
  nama_kolom2,tipe_data_kolom2,....)
```

Contoh :

Buat struktur tabel dengan nama tabel Mahasiswa dengan data NIM char(8), NAMA char(25), ALAMAT char(30)

```
CREATE TABLE Mahasiswa (NIM char(8) not null,  
  NAMA char(25) notnull, ALAMAT char(30) notnull)
```

DDL lanjutan



3. Pembuatan Index

Sintaks : CREATE [UNIQUE] INDEX nama_index
ON nama_table (nama_kolom) ;

Contoh :

Buat index data Mahasiswa berdasarkan NIM dengan nama MHSIDX Dimana NIM tidak boleh sama

```
CREATE UNIQUE INDEX MHSIDX ON Mahasiswa(NIM)
```

4. Pembuatan View

Sintaks :

```
CREATE VIEW nama_view [ (nama_kolom1,...) ]  
AS SELECT statement  
[WITH CHECK OPTION] ;
```

Contoh :

Buat view dengan nama MHSVIEW yang berisi semua data mahasiswa

```
CREATE VIEW MHSVIEW  
AS SELECT * FROM Mahasiswa
```

DDL lanjutan



B. DROP (MENGHAPUS)

1. Menghapus Database

Sintaks : DROP DATABASE nama_db ;

2. Menghapus Tabel

Sintaks : DROP TABLE nama_table ;

3. Menghapus Index

Sintaks : DROP INDEX nama_index ;

4. Menghapus View

Sintaks : DROP VIEW nama_view ;

Contoh :

```
DROP DATABASE KAMPUS;
```

```
DROP TABLE MHS;
```

```
DROP INDEX MHSIDX;
```

```
DROP VIEW MHSVIEW;
```

DDL lanjutan



C. ALTER TABLE (MERUBAH STRUKTUR TABEL)

Sintaks: ALTER TABLE nama_tabel
 ADD nama_kolom jenis_kolom
 [FIRST | AFTER nama_kolom]
 CHANGE [COLUMN] oldnama newnama
 MODIFY nama_kolom jenis_kolom, ...
 DROP nama_kolom
 RENAME newnama_tabel

Contoh :

1. Tambahkan kolom JKEL dengan panjang 1 char pada tabel Mahasiswa
ALTER TABLE Mahasiswa ADD JKEL char(1);
2. Ubah panjang kolom JKEL menjadi 15 char
ALTER TABLE Mahasiswa MODIFY COLUMN JKEL char(15);
3. Hapus kolom JKEL dari data table MHS
ALTER TABLE Mahasiswa DROP JKEL;

Data Manipulation Language (DML)



A. INSERT

Sintaks SQL yang digunakan untuk penambahan record baru kedalam sebuah tabel.

Sintaks: `INSERT INTO Nama_tabel [(nama_kolom1,...)]
values (nilai atribut1, ...)`

Contoh:Masukan data Mahasiswa dengan Nim 10296832,
Nama Nurhayati beralamat di Jakarta

```
INSERT INTO Mahasiswa (Nim, Nama, Alamat) values  
("10296832","Nurhayati","Jakarta");
```

DML lanjutan



B. UPDATE

Sintaks SQL yang digunakan untuk mengubah nilai atribut pada suatu record dari sebuah tabel.

Sintaks : UPDATE nama_tabel
 SET nama_kolom = value_1
 WHERE kondisi ;

Contoh:

Ubah alamat menjadi “Depok” untuk mahasiswa yang memiliki NIM “10296832”

```
UPDATE Mahasiswa  
SET ALAMAT="Depok"  
WHERE NIM=" 10296832";
```

DML lanjutan



C. DELETE

Sintaks SQL yang digunakan untuk menghapus record dari sebuah tabel.

Sintaks: DELETE FROM nama_table
WHERE kondisi

Contoh:

Hapus data Mahasiswa yang mempunyai NIM "21198002"

```
DELETE FROM Mahasiswa  
WHERE NIM=" 21198002"
```

DML lanjutan



Tabel dibawah ini untuk mengerjakan perintah **SELECT**

Tabel Mahasiswa

NIM	NAMA	ALAMAT
10296832	Nurhayati	Jakarta
10296126	Astuti	Jakarta
31296500	Budi	Depok
41296525	Prananigrum	Bogor
50096487	Pipit	Bekasi
21196353	Quraish	Bogor
10296001	Fintri	Depok
21198002	Julizar	Jakarta

Tabel Nilai

NIM	KD_MK	MID	FINAL
10296832	KK021	60	75
10296126	KD132	70	90
31296500	KK021	55	40
41296525	KU122	90	80
21196353	KU122	75	75
50095487	KD132	80	0

Tabel MataKuliah

KD_MK	NAMA_MK	SKS
KK021	Sistem Basis Data	2
KD132	Sistem Informasi Manajemen	3
KU122	Pancasila	2

DML lanjutan



D. SELECT

Sintaks : SELECT [DISTINCT | ALL] nama_kolom
FROM nama_tabel
[WHERE condition]
[GROUP BY column_list]
[HAVING condition]
[ORDER BY column_list [ASC | DESC]]

DML lanjutan



Contoh :

- a. Tampilkan semua data Mahasiswa

```
SELECT NIM,NAMA,ALAMAT FROM Mahasiswa;
```

Atau

```
SELECT * FROM Mahasiswa;
```

Maka hasilnya adalah :

NIM	NAMA	ALAMAT
10296832	Nurhayati	Jakarta
10296126	Astuti	Jakarta
31296500	Budi	Depok
41296525	Prananingrum	Bogor

DML lanjutan



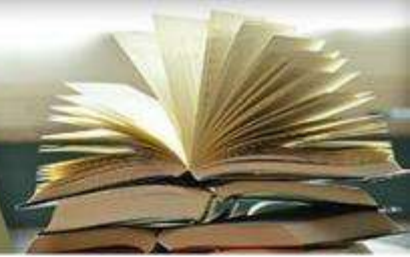
b. Tampilkan Mata Kuliah yang SKS nya 2

```
SELECT NAMA_MK FROM MataKuliah WHERE SKS=2
```

Maka Hasilnya:

NAMA_MK
Sistem Basis Data Pancasila

DML lanjutan



c. Tampilkan semua data nilai dimana nilai MID lebih besar sama dengan 60 atau nilai akhirnya lebih besar 75.

maka penulisannya :

```
SELECT * FROM Nilai WHERE MID >= 60 OR FINAL > 75
```

Hasilnya:

NIM	KD_MK	MID	FINAL
10296832	KK021	60	75
10296126	KD132	70	90
41296525	KU122	90	80
21196353	KU122	75	75

JOIN



JOIN digunakan untuk memilih data dari dua tabel atau lebih.

1. **INNER JOIN**

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian.

2. **LEFT JOIN** atau **LEFT OUTER JOIN**

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kiri.

3. **RIGHT JOIN** atau **RIGHT OUTER JOIN**

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kanan.

Contoh INNER JOIN

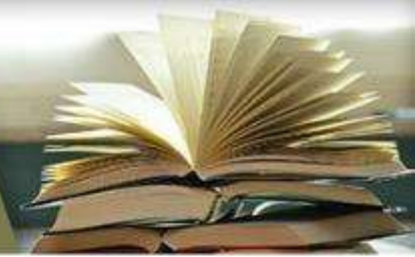


```
SELECT Nilai.NIM, Mahasiswa.NAMA, Nilai.KD_MK, Nilai.MID
FROM Nilai INNER JOIN Mahasiswa
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
21196353	Quraish	KU122	75
50095487	Pipit	KD132	80

Contoh LEFT JOIN



```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK,  
Nilai.MID
```

```
FROM Mahasiswa LEFT OUTER JOIN Nilai  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil:

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
21196353	Quraish	KU122	75
50095487	Pipit	KD132	80
10296001	Fintri	-	-
21198002	Julizar	-	-

Contoh RIGHT JOIN



```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK,  
Nilai.MID
```

```
FROM Nilai RIGHT OUTER JOIN Mahasiswa  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
21196353	Quraish	KU122	75
50095487	Pipit	KD132	80
10296001	Fintri	-	-
21198002	Julizar	-	-

Data Access



1. GRANT

Sintaks : GRANT hak_akses ON nama_db
 TO nama_pemakai
 [IDENTIFIED BY] [PASSWORD] 'Password'
 [WITH GRANT OPTION] ;

GRANT hak_akses ON [nama_db]nama_tabel
TO nama_pemakai
[IDENTIFIED BY] [PASSWORD] 'Password'
[WITH GRANT OPTION];

Contoh :

Berikan hak akses kepada Adi untuk menampilkan nilai final test pada tabel Nilai.

```
GRANT SELECT (FINAL) ON NILAI TO ADI
```

Data Access lanjutan



2. REVOKE

Sintaks : REVOKE hak_akses ON nama_db
FROM nama_pemakai ;

REVOKE hak_akses ON nama_tabel
FROM nama_pemakai ;

Contoh :

Tarik kembali dari Adi hak akses untuk menampilkan nilai final test

```
REVOKE SELECT (FINAL) ON NILAI FROM ADI
```

Data Integrity



RECOVER TABLE

Sintaks : RECOVER TABLE nama_tabel

Contoh :

Kembalikan keadaan data mahasiswa seperti pada saat sebelum terjadi kerusakan

```
RECOVER TABLE MHS ;
```

Auxiliary



1. **SELECT ... INTO OUTFILE** 'filename'

Sintaks ini digunakan untuk mengekspor data dari tabel ke file lain.

Sintaks : SELECT ... INTO
 OUTFILE 'Nama File'
 [FIELDS | COLUMNS]
 [TERMINATED BY '*string*']
 [[OPTIONALLY] ENCLOSED BY '*char*']
 [ESCAPED BY '*char*]]

Contoh :

Ubah semua data mahasiswa ke bentuk ASCII dan disimpan ke file teks di directory/home/adi dengan pemisah antar kolom '|'

```
SELECT * FROM MHS  
INTO OUTFILE "/home/adi/teks"  
FIELDS TERMINATED BY "|";
```

Auxiliary lanjutan



2. LOAD

Sintaks query ini digunakan untuk mengimpor data dari file lain ke tabel.

```
Sintaks :  LOAD DATA INFILE " nama_path"
           INTO TABLE nama_tabel [ nama_kolom] ;
           [FIELDS | COLUMNS]
           [TERMINATED BY 'string']
           [[OPTIONALLY] ENCLOSED BY 'char']
           [ESCAPED BY 'char' ]
```

Contoh :

Memasukkan data-data dari file teks yang berada pada direktori “/home/adi” ke dalam tabel MHS_2. Dimana pemisah antara kolom dalam file teks adalah tab (\t) :

```
LOAD FROM “/home/adi/teks”
INTO MHS_2
FILELDS TERMINATED BY ‘\t’;
```

Auxiliary lanjutan



3. RENAME TABLE

Sintaks :

```
RENAME TABLE OldnamaTabel  
TO NewNamaTabel
```

Contoh :

```
RENAME TABLE MHS  
TO MAHASISWA
```

Fungsi Aggregate



MENGGUNAKAN FUNGSI AGGREGATE :

1. COUNT digunakan untuk menghitung jumlah.
Menghitung jumlah record mahasiswa dari tabel MAHASISWA

```
SELECT COUNT(*) FROM MAHASISWA
```
2. SUM digunakan untuk menghitung total dari kolom yang mempunyai tipe data numerik.

```
SELECT SUM(SKS) AS 'TOTAL SKS' FROM MATAKULIAH
```
3. AVG digunakan untuk menghitung rata-rata dari data-data dalam sebuah kolom.

```
SELECT AVG(FINAL) AS 'FINAL' FROM Nilai
```
4. MIN digunakan untuk menghitung nilai minimal dalam sebuah kolom.

```
SELECT MIN(FINAL) FROM Nilai
```
5. MAX digunakan untuk menghitung nilai maksimum dalam sebuah kolom

```
SELECT MAX(MID) FROM Nilai
```

SUBQUERY



SUBQUERY

Adalah subselect yang dapat digunakan di klausa WHERE dan HAVING dipernyataan select luar untuk menghasilkan tabel akhir.

Aturan-aturan untuk membuat subquery, yaitu :

1. Klausa Order By tidak boleh digunakan di subquery, Order By hanya dapat digunakan di pernyataan Select luar.
2. Klausa subquery Select harus berisi satu nama kolom tunggal atau ekspresi kecuali untuk subquery-subquery menggunakan kata kunci EXIST
3. Secara default nama kolom di subquery mengacu ke nama tabel di klausa FROM dari subquery tersebut.
4. Saat subquery adalah salah satu dua operan dilibatkan di perbandingan, subquery harus muncul disisi kanan perbandingan

Subquery lanjutan



Penggunaan ANY dan ALL

Jika subquery diawali kata kunci ALL, syarat hanya akan bernilai TRUE jika dipenuhi semua nilai yang dihasilkan subquery itu.

Jika subquery diawali kata kunci ANY, syaratnya akan bernilai TRUE jika dipenuhi sedikitnya satu nilai yang dihasilkan subquery tersebut.

Penggunaan EXIST DAN NOT EXIST

EXIST akan mengirim nilai TRUE jika dan hanya jika terdapat sedikitnya satu baris di tabel hasil yang dikirim oleh subquery dan EXIST mengirim nilai FALSE jika subquery mengirim tabel kosong.

Untuk NOT EXIST kebalikan dari EXIST.

Subquery lanjutan



CONTOH SUBQUERY :

1. Ambil nilai mid dan final dari mahasiswa yang bernama Astuti.

```
SELECT MID, FINAL FROM NILAI WHERE NIM=( SELECT NIM FROM MAHASISWA WHERE NAMA='Astuti')
```

2. Ambil nilai kode matakuliah, mid dan final dari mahasiswa yang tinggal di jakarta.

```
SELECT KD_MK, MID, FINAL FROM NILAI WHERE NIM IN(SELECT NIM FROM MAHASISWA WHERE ALAMAT = 'Jakarta')
```

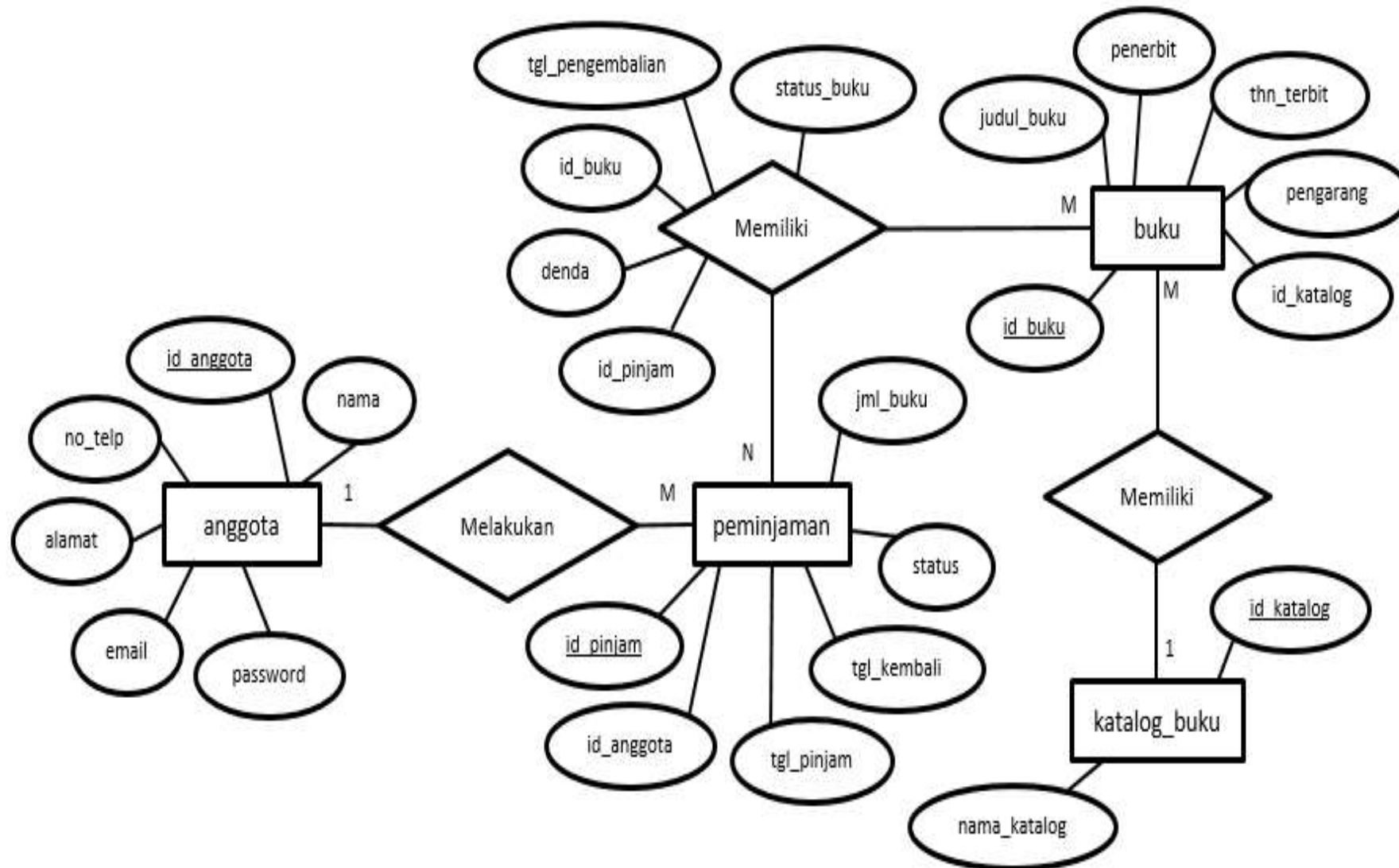
3. Ambil nama-nama mahasiswa yang mengikuti ujian.

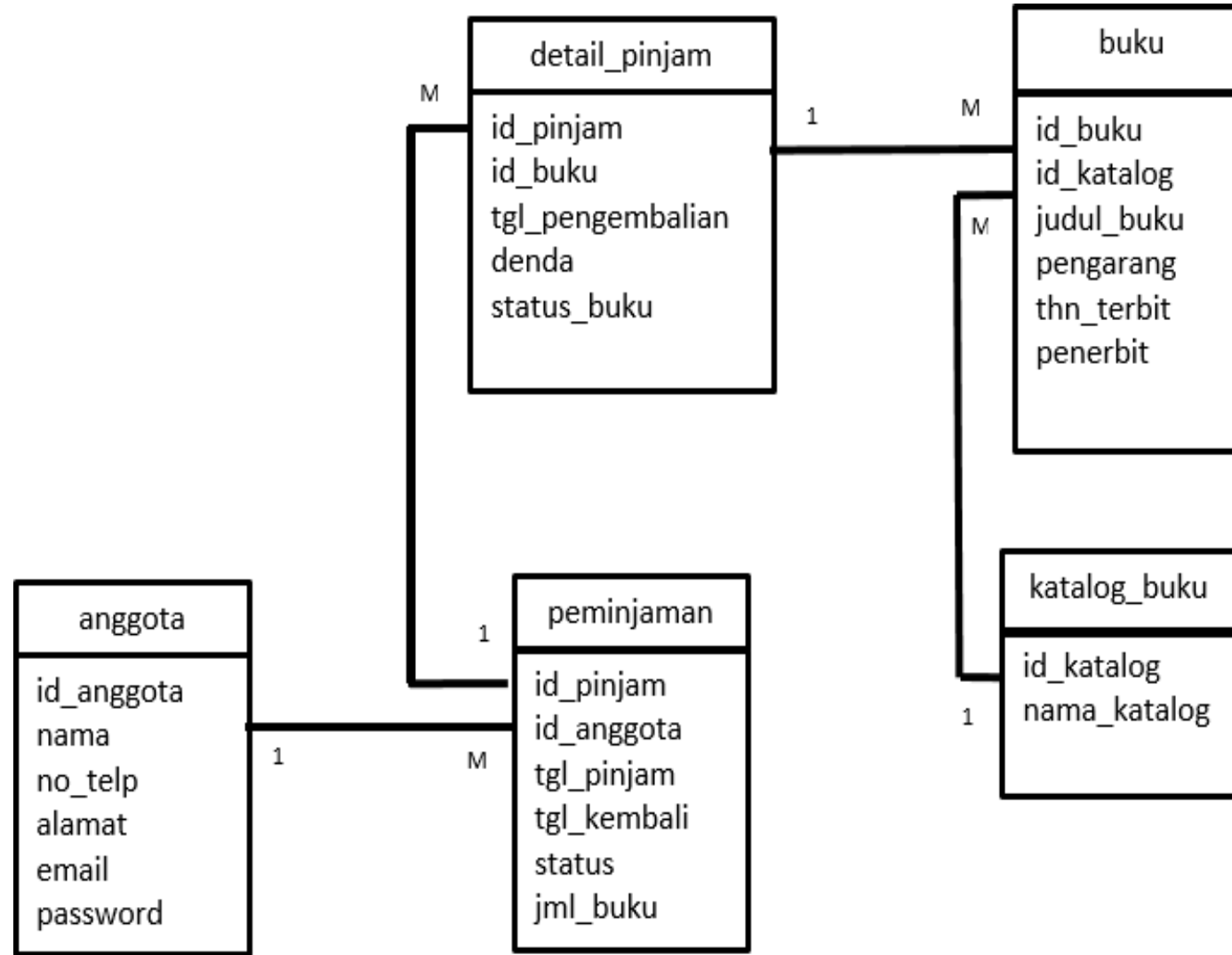
```
SELECT NAMA FROM MAHASISWA WHERE EXISTS (SELECT NIM FROM NILAI WHERE NILAI.NIM= MAHASISWA.NIM)
```

4. Ambil nama-nama mahasiswa yang tidak mengikuti ujian.

```
SELECT NAMA FROM MAHASISWA WHERE NOT EXISTS (SELECT NIM FROM NILAI WHERE NILAI.NIM= MAHASISWA.NIM)
```

STUDI KASUS PERPUSTAKAAN







```
Administrator: Command Prompt - "xampp/mysql/bin/...  
MariaDB [(none)]> create database db_perpusweb;  
Query OK, 1 row affected (0.01 sec)  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| bayan    |  
| bayan2  |  
| db_bayan |  
| db_distribusi |  
| db_dml_lanjutan |  
| db_latihan1 |  
| db_latihan_dml |  
| db_les_private |  
| db_perpusweb |  
| farazbookstore |  
| information_schema |  
| komik_coba1 |  
| mysql    |  
| performance_schema |  
| perpus   |  
| perpus1  |  
| perpusmvc |  
| phpmyadmin |  
| test     |  
| xx       |  
+-----+  
20 rows in set (0.00 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/...  
MariaDB [(none)]> use db_perpusweb;  
Database changed  
MariaDB [db_perpusweb]>
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" - ...
MariaDB [ldb_perpusweb]> create table admin
  -> (id_admin int(5) not null auto_increment,
  -> username varchar(25) not null,
  -> password varchar(35) not null,
  -> primary key (id_admin));
Query OK, 0 rows affected (0.55 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" - ...
MariaDB [ldb_perpusweb]> create table katalog_buku
  -> (id_katalog int(5) not null auto_increment,
  -> nama_katalog varchar(45) not null,
  -> primary key(id_katalog));
Query OK, 0 rows affected (0.35 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mys...
MariaDB [ldb_perpusweb]> create table buku
-> (id_buku int(5) not null auto_increment,
-> id_katalog int(5) not null,
-> judul_buku varchar(50) not null,
-> pengarang varchar(35) not null,
-> thn_terbit date not null,
-> penerbit varchar(50) not null,
-> primary key(id_buku));
Query OK, 0 rows affected (0.27 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mys...
MariaDB [ldb_perpusweb]> create table anggota
-> (id_anggota int(5) not null auto_increment,
-> nama varchar(45) not null,
-> no_telp varchar(15) not null,
-> alamat varchar(50) not null,
-> email varchar(30) not null,
-> password varchar(35) not null,
-> primary key (id_anggota));
Query OK, 0 rows affected (0.28 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mys...  
MariaDB [ldb_perpusweb]> create table peminjaman  
-> (id_pinjam int(5) not null auto_increment,  
-> id_anggota int(5) not null,  
-> tgl_pinjam date not null,  
-> tgl_kembali date not null,  
-> status ENUM('Selesai','Belum Selesai'),  
-> jml_buku int(2) not null,  
-> primary key (id_pinjam));  
Query OK, 0 rows affected (0.56 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" ...  
MariaDB [ldb_perpusweb]> create table detail_pinjam  
-> (id_pinjam int(5) not null,  
-> id_buku int(5) not null,  
-> tgl_pengembalian date not null,  
-> denda double not null,  
-> status_buku ENUM('Kembali','Belum Kembali'));  
Query OK, 0 rows affected (0.30 sec)
```



```
Administrator: Command Prompt - "xampp/mys...  
MariaDB [db_perpusweb]> show tables;  
+-----+  
| Tables_in_db_perpusweb |  
+-----+  
| admin                   |  
| anggota                 |  
| buku                    |  
| detail_pinjam           |  
| katalog_buku            |  
| peminjaman              |  
+-----+  
6 rows in set (0.00 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root  
MariaDB [db_perpusweb]> desc admin;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id_admin  | int(5)        | NO   | PRI | NULL    | auto_increment |  
| username  | varchar(25)   | NO   |     | NULL    |                |  
| password  | varchar(35)   | NO   |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.04 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [ldb_perpusweb] > desc katalog_buku;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_katalog | int(5)    | NO   | PRI | NULL    | auto_increment |
| nama_katalog | varchar(45) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [ldb_perpusweb] > desc buku;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_buku    | int(5)    | NO   | PRI | NULL    | auto_increment |
| id_katalog | int(5)    | NO   |     | NULL    |              |
| judul_buku | varchar(50) | NO   |     | NULL    |              |
| pengarang  | varchar(35) | NO   |     | NULL    |              |
| thn_terbit | date      | NO   |     | NULL    |              |
| penerbit   | varchar(50) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.08 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [ldb_perpusweb] > desc anggota;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_anggota | int(5)    | NO   | PRI | NULL    | auto_increment |
| nama       | varchar(45) | NO   |     | NULL    |              |
| no_telp    | varchar(15) | NO   |     | NULL    |              |
| alamat     | varchar(50) | NO   |     | NULL    |              |
| email      | varchar(30) | NO   |     | NULL    |              |
| password   | varchar(35) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.09 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root

MariaDB [db_perpusweb]> desc peminjaman;
+-----+-----+-----+-----+-----+
| Field          | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id_pinjam      | int(5)              | NO   | PRI | NULL    | auto_in
crement |
| id_anggota     | int(5)              | NO   |     | NULL    |       |
| tgl_pinjam     | date                | NO   |     | NULL    |       |
| tgl_kembali   | date                | NO   |     | NULL    |       |
| status         | enum('Selesai','Belum Selesai') | YES  |     | NULL    |       |
| jml_buku       | int(2)              | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.10 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [db_perpusweb]> desc detail_pinjam;
+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id_pinjam      | int(5)        | NO   |     | NULL    |       |
| id_buku        | int(5)        | NO   |     | NULL    |       |
| tgl_pengembalian | date         | NO   |     | NULL    |       |
| denda          | double        | NO   |     | NULL    |       |
| status_buku    | enum('Kembali','Belum Kembali') | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.06 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root  
  
MariaDB [db_perpusweb]> insert into admin(username, password)  
-> values  
-> ('admin','admin'),  
-> ('xxx','xxx');  
Query OK, 2 rows affected (0.09 sec)
```

```
Administrator: Command Prompt - "xampp/mysql/bin/...  
  
MariaDB [db_perpusweb]> select * from admin;  
+-----+-----+-----+  
| id_admin | username | password |  
+-----+-----+-----+  
|          | 1 | admin   | admin |  
|          | 2 | xxx     | xxx   |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [db_perpusweb]> insert into katalog_buku(nama_katalog)
-> values
-> ('Sains'),
-> ('Hobby'),
-> ('Komputer'),
-> ('Komunikasi'),
-> ('Hukum'),
-> ('Agama'),
-> ('Populer'),
-> ('Bahasa');
Query OK, 8 rows affected (0.08 sec)
Records: 8 Duplicates: 0 Warnings: 0
```

```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe..."
MariaDB [db_perpusweb]> select * from katalog_buku;
+-----+-----+
| id_katalog | nama_katalog |
+-----+-----+
| 1 | Sains |
| 2 | Hobby |
| 3 | Komputer |
| 4 | Komunikasi |
| 5 | Hukum |
| 6 | Agama |
| 7 | Populer |
| 8 | Bahasa |
+-----+-----+
8 rows in set (0.00 sec)
```



Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root

```
MariaDB [ldb_perpusweb] > insert into buku
```

```
-> (id_katalog, judul_buku, pengarang, thn_terbit, penerbit)
```

```
-> values
```

```
-> ('1', 'Robotika Sederhana', 'Siswoyo Utomo', '2013-01-01', 'Wacana Ria'),
```

```
-> ('3', 'Mahir denga PHP', 'Adri Kusuma', '2011-02-02', 'Pustaka Bangsa'),
```

```
-> ('2', 'Mahir Mewarnai', 'Akhmad Rahmat', '2014-03-03', 'CU.Indo Kreasi'),
```

```
-> ('1', 'Hukum Fisika', 'Kurnia Sandi', '2013-04-04', 'Wacana Ria'),
```

```
-> ('8', 'Mahir Bahasa Inggris', 'Aliuddin', '2013-05-05', 'CU.Indo Kreasi'),
```

```
-> ('4', 'Public Speaking', 'Pambudi Prasetyo', '2015-06-06', 'Aldi Pustaka'),
```

```
-> ('3', 'Trik SQL', 'Ahdim Makaren', '2014-07-07', 'Wacana Ria'),
```

```
-> ('6', 'Kemurnian Agama', 'Pambudi Prasetyo', '2014-08-08', 'Aldi Pustaka'),
```

```
-> ('1', 'Mikrokontroller', 'Ahdim Makaren', '2012-09-09', 'Wacana Ria');
```

```
Query OK, 9 rows affected (0.33 sec)
```

```
Records: 9 Duplicates: 0 Warnings: 0
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [ldb_perpusweb]> select * from buku;
+-----+-----+-----+-----+-----+
| id_buku | id_katalog | judul_buku          | pengarang          | thn_terbit |
| penerbit |            |                     |                    |            |
+-----+-----+-----+-----+-----+
| 1 | 1 | Robotika Sederhana | Siswoyo Utomo     | 2013-01-01 |
| Macana Ria |            |                     |                    |            |
| 2 | 3 | Mahir denga PHP    | Adri Kusuma       | 2011-02-02 |
| Pustaka Bangsa |            |                     |                    |            |
| 3 | 2 | Mahir Mewarnai     | Akhmad Rahmat     | 2014-03-03 |
| CU.Indo Kreasi |            |                     |                    |            |
| 4 | 1 | Hukum Fisika       | Kurnia Sandi     | 2013-04-04 |
| Macana Ria |            |                     |                    |            |
| 5 | 8 | Mahir Bahasa Inggris | Aliuddin          | 2013-05-05 |
| CU.Indo Kreasi |            |                     |                    |            |
| 6 | 4 | Public Speaking   | Pambudi Prasetyo | 2015-06-06 |
| Aldi Pustaka |            |                     |                    |            |
| 7 | 3 | Trik SQL          | Ahdim Makaren     | 2014-07-07 |
| Macana Ria |            |                     |                    |            |
| 8 | 6 | Kemurnian Agama   | Pambudi Prasetyo | 2014-08-08 |
| Aldi Pustaka |            |                     |                    |            |
| 9 | 1 | Mikrokontroller   | Ahdim Makaren     | 2012-09-09 |
| Macana Ria |            |                     |                    |            |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root

MariaDB [ldb_perpusweb]> insert into anggota
-> (nama, no_telp, alamat, email, password)
-> values
-> ('Irfan Maulana','01244445555','BSD','irfan@gmail.com','123'),
-> ('Nur Kumalasari','01233335555','Ciledug','nur@gmail.com','123'),
-> ('Sanjaya Wijaya','01211115555','Cimone','sanjaya@gmail.com','123'),
-> ('Eva Irfianingsih','01266665555','Tangerang','eva@gmail.com','123'),
-> ('Ifqoh Permatasari','01277775555','Cengkareng','ifqoh@gmail.com','123'),

-> ('Indah Riana','01288885555','Fatmawati','indah@gmail.com','123'),
-> ('Tiwie Andrawati','01299995555','Warung Jati','tiwie@gmail.com','123'),
-> ('Mus Dalifa','01200005555','Jatiwaringin','mus@gmail.com','123'),
-> ('Hisbu Utomo','01233336666','Salemba','hisbu@gmail.com','123'),
-> ('Zaenal Abidin','01233337777','Bekasi','zaenal@gmail.com','123');

Query OK, 10 rows affected (0.07 sec)
Records: 10 Duplicates: 0 Warnings: 0
```



```
Administrator: Command Prompt - "xampp/mysql/bin/mysql.exe" -u root
MariaDB [ldb_perpusweb]> select * from anggota;
+-----+-----+-----+-----+-----+
| id_anggota | nama          | no_telp      | alamat  | email      |
| password   |               |              |         |            |
+-----+-----+-----+-----+-----+
| 1          | Irfan Maulana | 01244445555 | BSD    | irfan@gmail.com |
| 123       |               |              |         |            |
| 2          | Nur Kumalasari | 01233335555 | Ciledug | nur@gmail.com   |
| 123       |               |              |         |            |
| 3          | Sanjaya Wijaya | 01211115555 | Cimone  | sanjaya@gmail.co |
| 123       |               |              |         |            |
| 4          | Eva Irfianingsih | 01266665555 | Tangerang | eva@gmail.com   |
| 123       |               |              |         |            |
| 5          | Ifqoh Permatasari | 01277775555 | Cengkareng | ifqoh@gmail.com |
| 123       |               |              |         |            |
| 6          | Indah Riana    | 01288885555 | Fatmawati | indah@gmail.com |
| 123       |               |              |         |            |
| 7          | Tiwie Andrawati | 01299995555 | Warung Jati | tiwie@gmail.com |
| 123       |               |              |         |            |
| 8          | Mus Dalifa     | 01200005555 | Jatiwaringin | mus@gmail.com   |
| 123       |               |              |         |            |
| 9          | Hisbu Utomo    | 01233336666 | Salemba  | hisbu@gmail.com |
| 123       |               |              |         |            |
| 10         | Zaenal Abidin  | 01233337777 | Bekasi   | zaenal@gmail.com |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```



THANK
YOU