

Virliana ar

L200180017

A

## Modul 2

### 1. Perbankan

Langkah-langkah perancangan database perbankan :

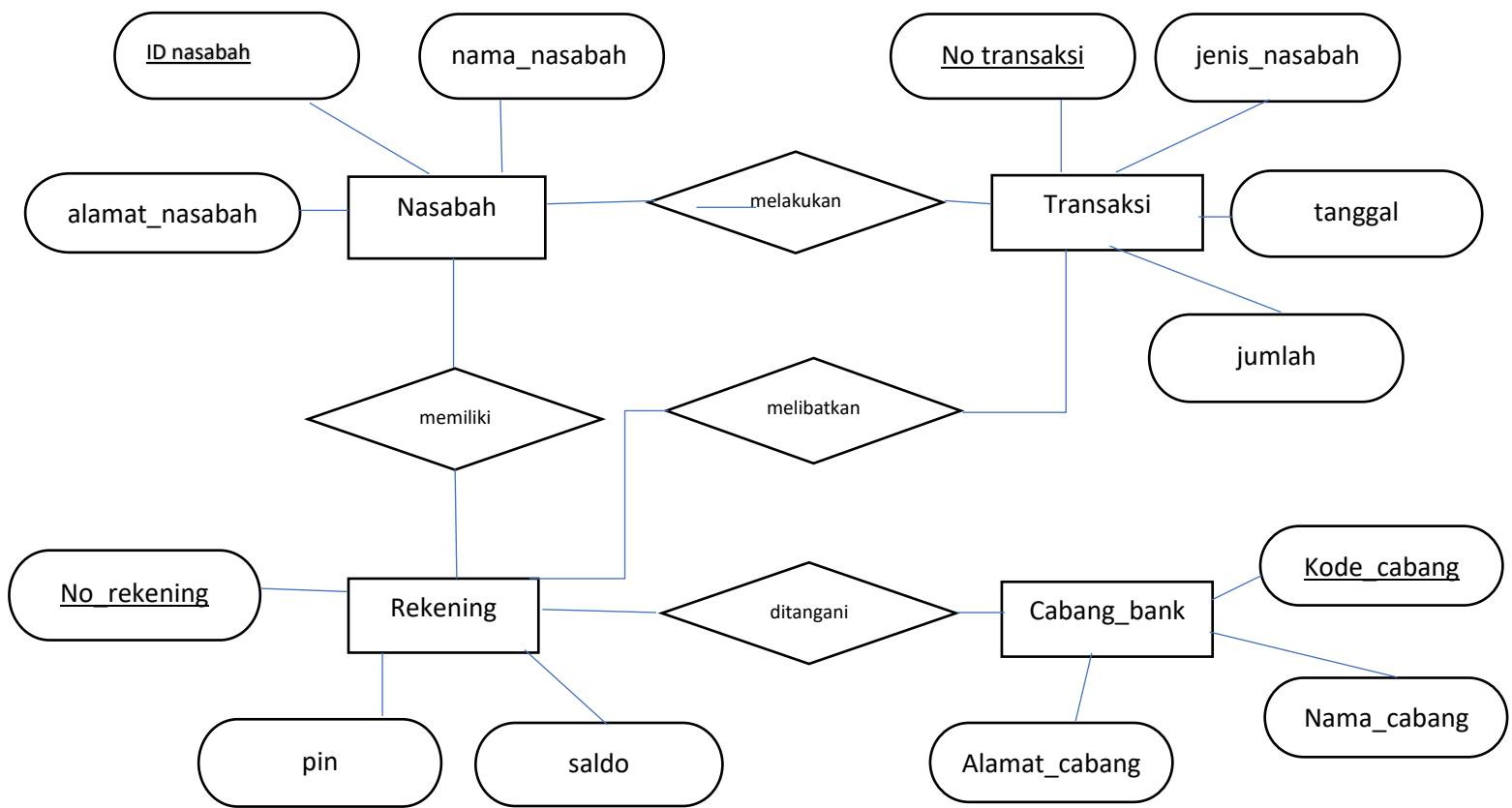
1. Menentukan entities (object-object dasar) yang perlu ada di database.
  - a. **nasabah**: menyimpan semua data pribadi semua nasabah
  - b. **rekening**: menyimpan informasi semua rekening yang telah dibuka
  - c. **cabang\_bank**: menyimpan informasi tentang semua cabang bank
  - d. **transaksi**: menyimpan informasi tentang semua transaksi yang telah terjadi
2. Menentukan attributes (sifat -sifat) masing -masing *entity* sesuai kebutuhan database
  - a. **nasabah**:
    - **id\_nasabah**: nomor id untuk nasabah (integer) pk
    - **nama\_nasabah**: nama lengkap nasabah (varchar(45))
    - **alamat\_nasabah**: alamat lengkap nasabah (varchar(255))
  - b. **rekening**:
    - **no\_rekening**: nomor rekening (integer) pk
    - **pin**: personal identification number (varchar(10))
    - **saldo**: jumlah saldo rekening dalam rp (integer)
  - c. **cabang\_bank**:
    - **kode\_cabang**: kode untuk cabang bank (varchar(10)) pk
    - **nama\_cabang**: nama lengkap cabang bank (varchar(20))
    - **alamat\_cabang**: alamat lengkap cabang bank (varchar(255))
  - d. **transaksi**:
    - **no\_transaksi**: nomor transaksi (integer) PK
    - **jenis\_transaksi**: kredit atau debit (varchar(10))
    - **tanggal**: tanggal terjadinya transaksi (date)
    - **jumlah**: besarnya transaksi dalam Rp (integer)
3. Menentukan relationship (hubungan) antar entitas:

	<b>nasabah</b>	<b>rekening</b>	<b>cabang_bank</b>	<b>transaksi</b>
<b>nasabah</b>	-	m:n	-	1:n
<b>rekening</b>		-	n:1	1:n
<b>cabang_bank</b>			-	-
<b>transaksi</b>				-

## Hubungan

- **nasabah** memiliki **rekening**:
  - ❑ Tabel utama: **nasabah**, **rekening**
  - ❑ Tabel kedua: **nasabah\_has\_rekening**
  - ❑ Relationship: many-to-many (m:n)
  - ❑ Attribute penghubung: **id\_nasabah**, **no\_rekening** (FK **id\_nasabah**, **no\_rekening** di **nasabah\_has\_rekening**)
- **nasabah** melakukan **transaksi**:
  - ❑ Tabel utama: **nasabah**
  - ❑ Tabel kedua: **transaksi**
  - ❑ Relationship: One-to-many (1:n)
  - ❑ Attribute penghubung: **id\_nasabah** (FK **id\_nasabah** di **transaksi**)
- **cabang\_bank** menangani **rekening**:
  - ❑ Tabel utama: **cabang\_bank**
  - ❑ Tabel kedua: **rekening**
  - ❑ Relationship: One-to-many (1:n)
  - ❑ Attribute penghubung: **kode\_cabang** (FK **kode\_cabang** di **rekening**)

## 4. Menggambar ERD Diagram



## TUGAS

### 1. Perkuliahan

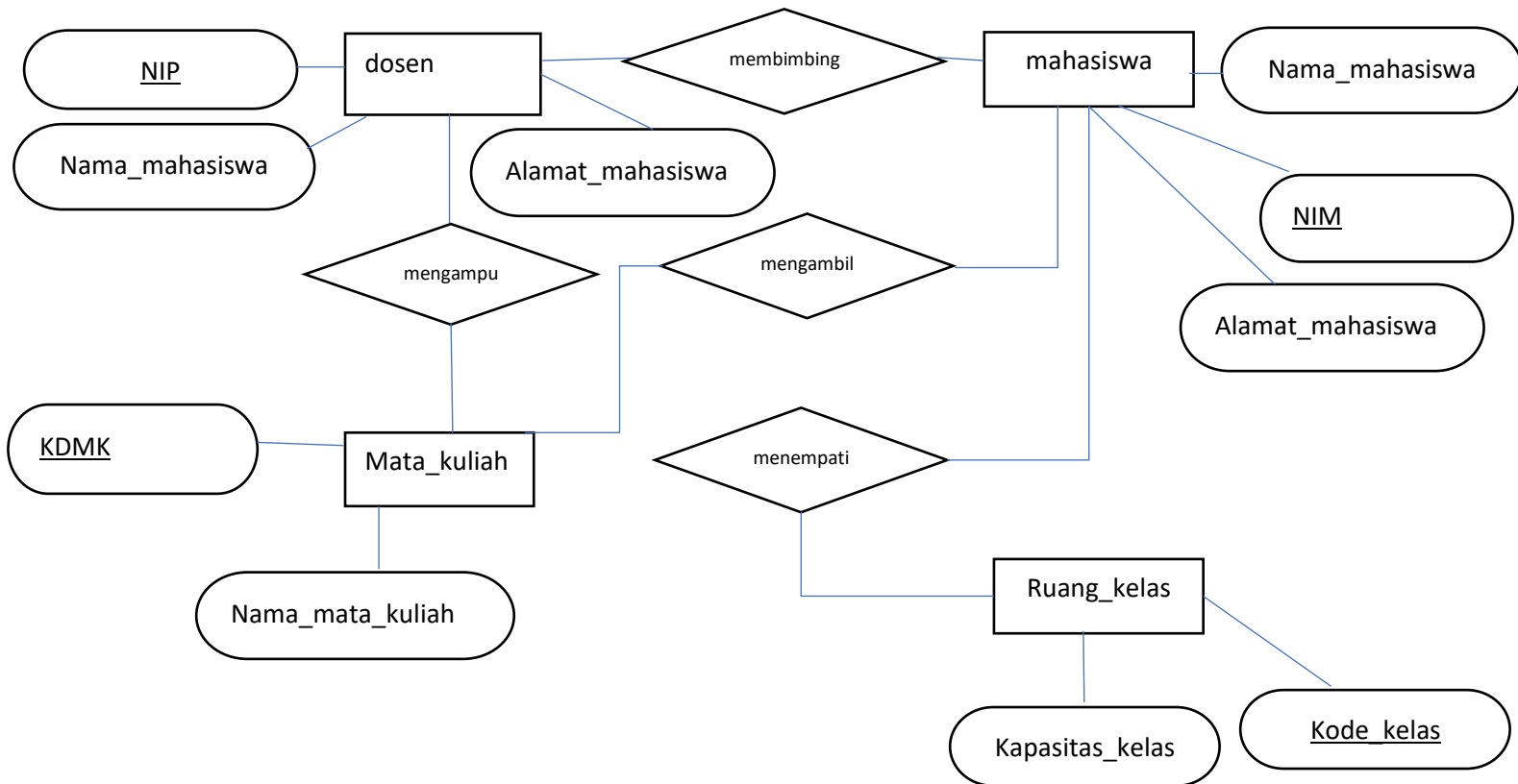
- Menentukan entities (object-object dasar ) yang perlu ada di database:
  - a. **mahasiswa** : menyimpan semua data pribadi semua mahasiswa.
  - b. **dosen** : menyimpan semua data pribadi semua dosen.
  - c. **mata\_kuliah**: menyimpan informasi tentang data mata kuliah.
  - d. **ruang\_kelas** : menyimpan informasi tentang data ruang kelas.
- Menentukan attributes (sifat-sifat) masing-masing entity sesuai dengan kebutuhan database
  - a. **mahasiswa** :
    - **id\_mahasiswa** : nomor id untuk mahasiswa (integer ) PK
    - **nama\_mahasiswa** : nama lengkap mahasiswa (varchar (45))
    - **alamat\_mahasiswa** : alamat lengkap mahasiswa (varchar(255))
    - **jurusan\_mahasiswa** : jurusan yang diambil oleh mahasiswa (varchar (20))
  - b. **dosen** :
    - **nip\_dosen** : nomor id untuk dosen (integer ) PK
    - **nama\_dosen** : nama lengkap dosen (varchar (45))
    - **gaji\_dosen** : jumlah gaji dari dosen (integer) PK
  - c. **mata\_kuliah** :
    - **kode\_matkul** : kode untuk mata kuliah (varchar(10)) PK
    - **nama\_matkul** : nama mata kuliah (varchar(20))
    - **sks** : jumlah sks dari mata kuliah (varchar(255))
  - d. **Ruang\_kelas** :
    - **kapasitas** : kapasitas dalam satu ruang kelas (integer)PK
    - **id\_kelas**: kode ruangan (varchar (10))
- Menentukan relationship (hubungan ) antar entitas:

	<b>mahasiswa</b>	<b>Dosen</b>	<b>mata_kuliah</b>	<b>ruang_kelas</b>
<b>mahasiswa</b>	-	-	m:n	
<b>dosen</b>	m:n	-	1:n	1:1
<b>mata_kuliah</b>			-	-
<b>ruang_kelas</b>				-

### Hubungan

- **mahasiswa** mengambil **mata\_kuliah**
  - Tabel utama : **mahasiswa**

- Tabel kedua : **mata\_kuliah**
- Relationship : many-to-many(m:n)
- Attribute penghubung : **id\_mahasiswa** (FK **id\_mahasiswa** di **mata\_kuliah**)
- **dosen** membimbing **mahasiswa**
  - Tabel utama : **dosen**
  - Tabel kedua : **mahasiswa**
  - Relationship : many-to-many(m:n)
  - Attribute penghubung : **nip\_dosen** (FK **nip\_dosen** di **mahasiswa**)
- **dosen** mengampu **mata\_kuliah**
  - Tabel utama : **dosen**
  - Tabel kedua : **matakuliah**
  - Relationship : one-to-many(1:n)
  - Attribute penghubung : **nip\_dosen** (FK **nip\_dosen** di **matakuliah**)
- **matakuliah** menempati **ruang\_kelas**
  - Tabel utama : **mata\_kuliah**
  - Tabel kedua : **ruang\_kelas**
  - Relationship : one-to-one(1:1)
  - Attribute penghubung : **kode\_mata\_kuliah**(FK **kode\_mata\_kuliah** di **ruang\_kelas**)



## 2. Perbengkelan

- Menentukan entities (object-object dasar ) yang perlu ada di database:
  - a. **customer** : menyimpan semua data pribadi semua customer.
  - b. **produk** : menyimpan informasi data produk.
  - c. **teknisi**: menyimpan semua data pribadi semua customer.
  - d. **petugas\_administrasi**: menyimpan informasi tentang data aktivitas customer.
- Menentukan attributes (sifat-sifat) masing-masing entity sesuai dengan kebutuhan database
  - a. **customer** :
    - **id\_customer** : nomor id untuk customer (integer ) PK
    - **nama\_customer** : nama lengkap customer (varchar (45))
    - **alamat\_customer** : alamat lengkap customer (varchar(255))
  - b. **produk** :
    - **id\_produk**: nomor id untuk produk (integer ) PK
    - **jenis\_produk** : nama jenis produk (varchar (45))
    - **harga\_produk** : harga dari produk dalam bentuk RP (integer) PK
  - c. **teknisi** :
    - **id\_teknisi** : nomor id untuk teknisi (varchar(10)) PK
    - **nama\_teknisi** : nama lengkap teknisi (varchar(20))
    - **spesialis** : spesialis bidang teknisi (varchar(255))
  - d. **petugas administrasi** :
    - **id\_petugas** : kapasitas dalam satu ruang kelas (integer)PK
    - **nama\_petugas** : kode ruangan (varchar (10))
- Menentukan relationship (hubungan ) antar entitas:

	Customer	Produk	Teknisi	Petugas administrasi
Customer		m:n	m:n	m:n
Produk				
Teknisi				
Petugas administrasi				

### Hubungan

- **customer** membeli **produk** :
  - Tabel utama : **customer**
  - Tabel kedua : **customer\_has\_produk**
  - Relationship : many-to-many (m:n)

- Attribute penghubung : **id\_customer** , **id\_produk** , **jenis\_produk** (FK **id\_customer** , **id\_produk** , **jenis\_produk** di **customer\_has\_produk**)
- **customer** mempekerjakan **teknisi** :
  - Tabel utama : **customer**
  - Tabel kedua : **teknisi**
  - Relationship : many-to-many (m:n)
  - Attribute penghubung : **id\_nasabah** (FK **id\_customer** di **teknisi**)
- **customer** dilayani petugas **administrasi**:
  - Tabel utama : **customer**
  - Tabel kedua : **petugas\_administrasi**
  - Relationship : many-to-many (m:n)
  - Attribute penghubung : **id\_customer** (FK **id\_customer** di **petugas\_administrasi**)

