

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 semua rekening 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) diantara entities tersebut.

	Nasabah	Rekening	Cabang_bank	Transaksi
Nasabah	-	m : n	-	1 : n
Rekening		-	n : 1	1 : n
Cabang_bank			-	-
Transaksi				-

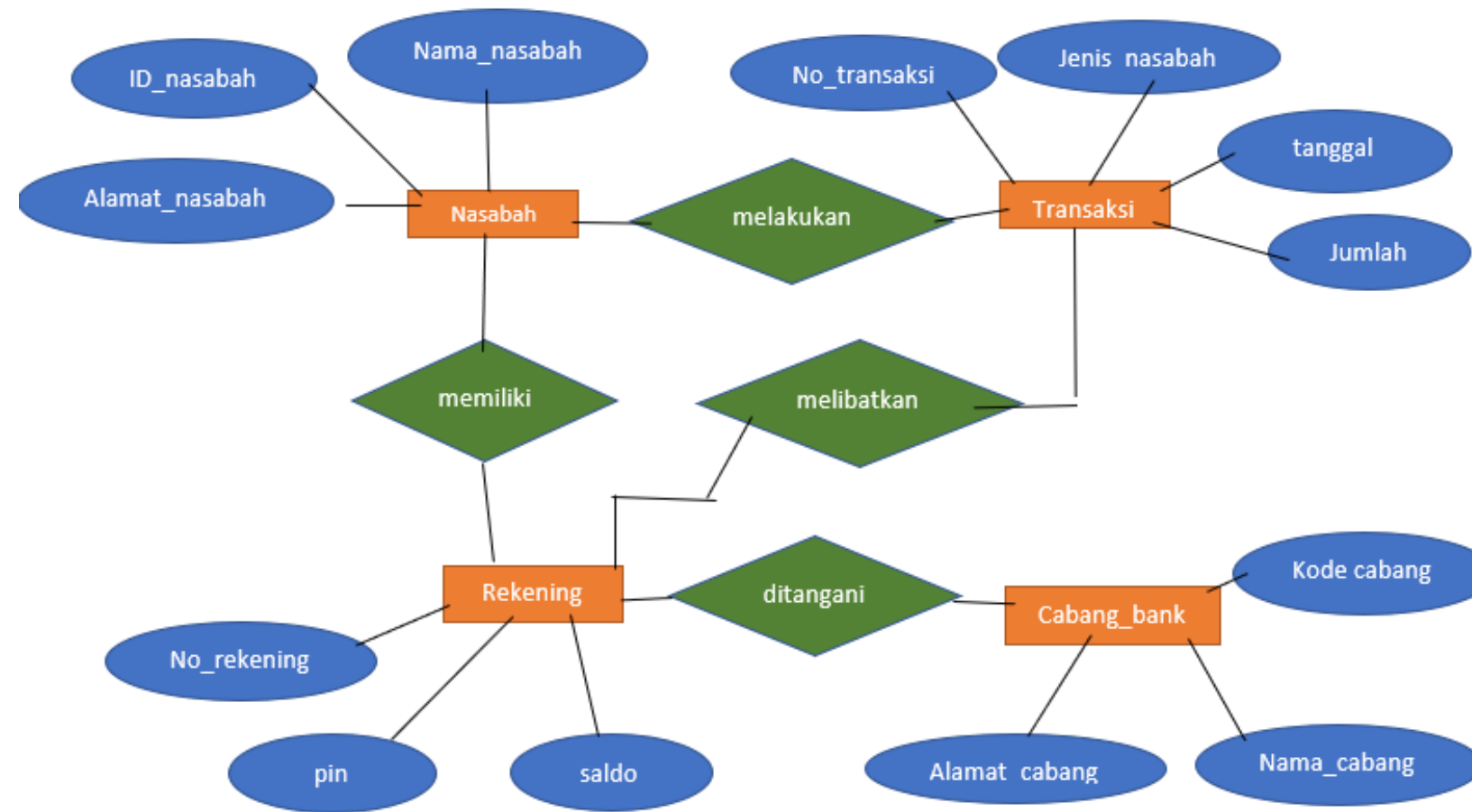
Hubungan

- a. 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, dinasabah\_has\_rekening)
- b. nasabah melakukan transaksi :
  - Tabel utama : nasabah
  - Tabel kedua : transaksi
  - Relationship : One-to-many (1 : n)
  - Attribute penghubung : id\_nasabah, (FK id\_nasabah, di\_transaksi)
- c. 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)
- d. rekening terlibat dalam transaksi :
  - Tabel utama : rekening
  - Tabel kedua : transaksi

- Relationship : One-to-many (1 : n)
- Attribute penghubung : no\_rekening (FK no\_rekening, di transaksi)

#### 4. Menggambar ERD diagram

#### 4. Menggambar ERD diagram



## A. Tugas

1. Buatlah rancangan sebuah database untuk menangani data-data kuliah. Data-data yang akan ditanganinya adalah: data pribadi mengenai mahasiswa, data pribadi mengenai dosen, data mata kuliah dan data ruang kelas. Mahasiswa boleh mengambil lebih dari satu mata kuliah, dan satu mata kuliah boleh diambil oleh lebih dari satu `mahasiswa sekaligus (joint account).

Buatlah ER Diagram manual untuk kasus tersebut dari tahap 1 sampai tahap 4!

Tahap 1: Menentukan entitas.

### a. Menentukan entitas

- Mahasiswa : menyimpan semua data pribadi semua mahasiswa
- Dosen : menyimpan semua data pribadi dosen
- Mata\_kuliah : menyimpan informasi tentang semua mata kuliah
- Ruang : menyimpan informasi tentang semua ruang kelas.

### b. Menentukan attributes

- Mahasiswa
  - NIM : Nomor Induk Mahasiswa (char 10)
  - Nama : nama lengkap mahasiswa (varchar 40)
  - Alamat : berisi domisili mahasiswa (varchar 255)
  - Tempat\_lahir : tempat lahir mahasiswa (varchar 25)
  - Tgl\_lahir : tanggal lahir mahasiswa (varchar 20)
- Dosen
  - NIK : Nomor Induk Kependidikan dosen (varchar 50)
  - Nama\_Dosen : nama lengkap dosen (varchar 30)
  - Alamat : berisi domisili dosen (varchar 255)
  - No\_Hp : nomor telepon dosen (integer)
- Mata\_kuliah
  - Kode\_MK : kode dari mata kuliah (varchar 15)
  - Nama\_MK : nama mata kuliah (varchar 30)
  - SKS : banyaknya sks per mata kuliah (varchar 2)
  - Jadwal : jadwal mata kuliah, yaitu hari dan jam (varchar 15)
- Ruang
  - kode\_ruang\_kelas : identitas ruang kelas (varchar 6)
  - kapasitas\_ruang\_kelas : jumlah mahasiswa yang mampu ditampung dalam ruang kelas (integer)

c. Menentukan relationship (hubungan) diantara entities tersebut.

	Mahasiswa	Dosen	Mata_kuliah	Ruang
Mahasiswa	-	n : 1	m : n	-
Dosen		-	n : 1	-
Mata_kuliah			-	1 : 1
Ruang				-

Hubungan

- Mahasiswa diampu oleh dosen :
  - Tabel utama : Dosen
  - Tabel kedua : Mahasiswa
  - Relationship : many to one (n:1)
  - Attribute : NIK (FK NIK di dosen)
- Mahasiswa mengambil mata\_kuliah :
  - Tabel utama : mahasiswa, mata\_kuliah
  - Tabel kedua : mhs\_mengambil\_mk
  - Relationship : many to many (m;n)
  - Attribute : NIM, kode\_MK (FK NIM, kode\_MK di mhs\_mengambil\_mk)
- Dosen mengampu Mata\_kuliah
  - Tabel utama : Mata\_kuliah
  - Tabe kedua : dosen
  - Relationship : many to one (n:1)
  - Attribute : kode\_MK (FK kode\_MK di mata\_kuliah)
- Mata\_kuliah menempati ruang :
  - Tabel utama : ruang
  - Tabel kedua : Mata\_kuliah
  - Relationship : one to one (1:1)
  - Attribute : kode\_ruang\_kelas (FK kode\_ruang\_kelas di Mata\_kuliah)

2. Ambil contoh sembarang database. Buatlah rancangan ER Diagram manual database tersebut dari tahap 1-4 dengan ketentuan database minimal mengandung 4 buah entitas.

Langkah-langkah perancangan database perpustakaan

1. Entitas

- a. Buku : menyimpan semua data tentang buku
- b. Peminjam : menyimpan semua data tentang peminjam
- c. Suplier : menyimpan informasi tentang suplier
- d. Staff :menyimpan semua data tentang staff

2. Atribut a. Buku

- Judul\_buku : judul untuk setiap buku(varchar(100))
- Nomer\_isbn : nomer isbn buku(varchar(100))

b. Peminjam

- Id\_peminjam : nomer id peminjam(varchar(100))
- Nama\_peminjam : nama lengkap peminjam(varchar(100))

c. Suplier

- Nama\_suplier : nama lengkap suplier (varchar(100))
- Id\_suplier : nomer id suplier (varchar(100))

d. Staff

- Nama\_staff : nama lengkap staff (varchar(100))
- Id\_staff : nomer id staff (varchar(100))

3. Relationship

	Buku	Peminj	Suplier	Staff
Buku	-	n:1	n:1	n:1
Pemin		-	-	1:1
Suplie				-
Staff				-

Hubungan

- Buku dipinjam Peminjam
  - oTabel utama: Peminjam
  - oTabel Kedua: Buku
  - oRelationship: Many-to-one(n:1)
  - oAttribute penghubung: Transaksi
- Buku disuplai Suplier
  - oTabel utama: Suplier
  - oTabel Kedua: Buku
  - oRelationship: Many-to-one(n:1)
  - oAttribute penghubung: suply
- Buku ditangani staff
  - oTabel utama: Staff
  - oTabel Kedua: Buku
  - oRelationship: Many-to-one(n:1)
  - oAttribute penghubung: Transaksi
- Peminjam ditangani staff
  - oTabel utama: Peminjam, Staff
  - oTabel Kedua: Peminjam\_has\_Staff
  - o Relationship: one-to-one(1:1)

