

PRAKTIKUM SISTEM BASIS DATA

Nama : Afiq Tri Nugraha

NIM : L200180080

Kelas : C

MODUL 2

1. Buatlah rancangan sebuah database untuk menangani data-data kuliah. Data-data yang akan ditangani 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!

Jawab :

- 1) Menentukan entities (object-object dasar) yang perlu ada di database.

- **Mahasiswa** : menyimpan semua data pribadi mahasiswa
- **Dosen** : menyimpan semua data pribadi dosen
- **Mata_kuliah** : menyimpan semua data mata kuliah yang di ampu dosen dan yang diambil mahasiswa
- **Ruang_kelas** : menyimpan semua data ruang kelas untuk setiap mata kuliah

- 2) Menentukan attribute (sifat-sifat) masing-masing entity sesuai kebutuhan database

➤ **Mahasiswa**

- **Id_mahasiswa** : nomor id untuk mahasiswa (integer)
- **Nama_mahasiswa** : nama lengkap mahasiswa (varchar(45))
- **Alamat_mahasiswa** : alamat lengkap mahasiswa (varchar(255))

➤ **Dosen**

- **Id_dosen** : nomor id untuk dosen (integer)
- **Nama_dosen** : nama lengkap dosen (varchar(45))
- **Alamat_dosen** : alamat lengkap dosen (varchar(255))
- **Kode_dosen** : kode untuk dosen (varchar(15))

➤ **Mata_kuliah**

- **Kode_matakuliah** : kode untuk mata kuliah (varchar(15))
- **Nama_matakuliah** : nama lengkap mata kuliah (varchar(100))

➤ **Ruang_kelas**

- **Kode_ruangkelas** : kode untuk ruang kelas (varchar(15))
- **Kapasitas_ruangkelas** : kapasitas untuk ruang kelas (integer)

3) Menentukan relationship (hubungan) antar entitas

	Mahasiswa	Dosen	Mata_kuliah	Ruang_kelas
Mahasiswa	-	m : n	m : n	-
Dosen		-	m : n	m : n
Mata_kuliah			-	m : n
Ruang_kelas				-

Hubungan :

- **Mahasiswa mengambil Mata_kuliah :**
 - **Tabel utama** : Mahasiswa, Mata_kuliah
 - **Tabel kedua** : mahasiswa_take_matakuliah
 - **Relationship** : Many-to-many (m:n)
 - **Attribute penghubung** : Id_mahasiswa, Kode_matakuliah (FK Id_mahasiswa, Kode_matakuliah di **mahasiswa_take_matakuliah**)
- **Dosen mengajar Mahasiswa:**
 - **Tabel utama** : Dosen, Mahasiswa
 - **Tabel kedua** : dosen_teach_mahasiswa
 - **Relationship** : Many-to-many (m:n)
 - **Attribute penghubung** : Id_mahasiswa, Kode_dosen (**FK** Id_mahasiswa, Kode_dosen di **dosen_teach_mahasiswa**)
- **Dosen mengampu Mata_kuliah:**
 - **Tabel utama** : Dosen, Mata_kuliah
 - **Tabel kedua** : dosen_keepup_matakuliah
 - **Relationship** : many_to_many (m:n)
 - **Attribute penghubung** : Kode_dosen, Kode_matakuliah (FK Kode_dosen, Kode_matakuliah di **dosen_keepup_matakuliah**)

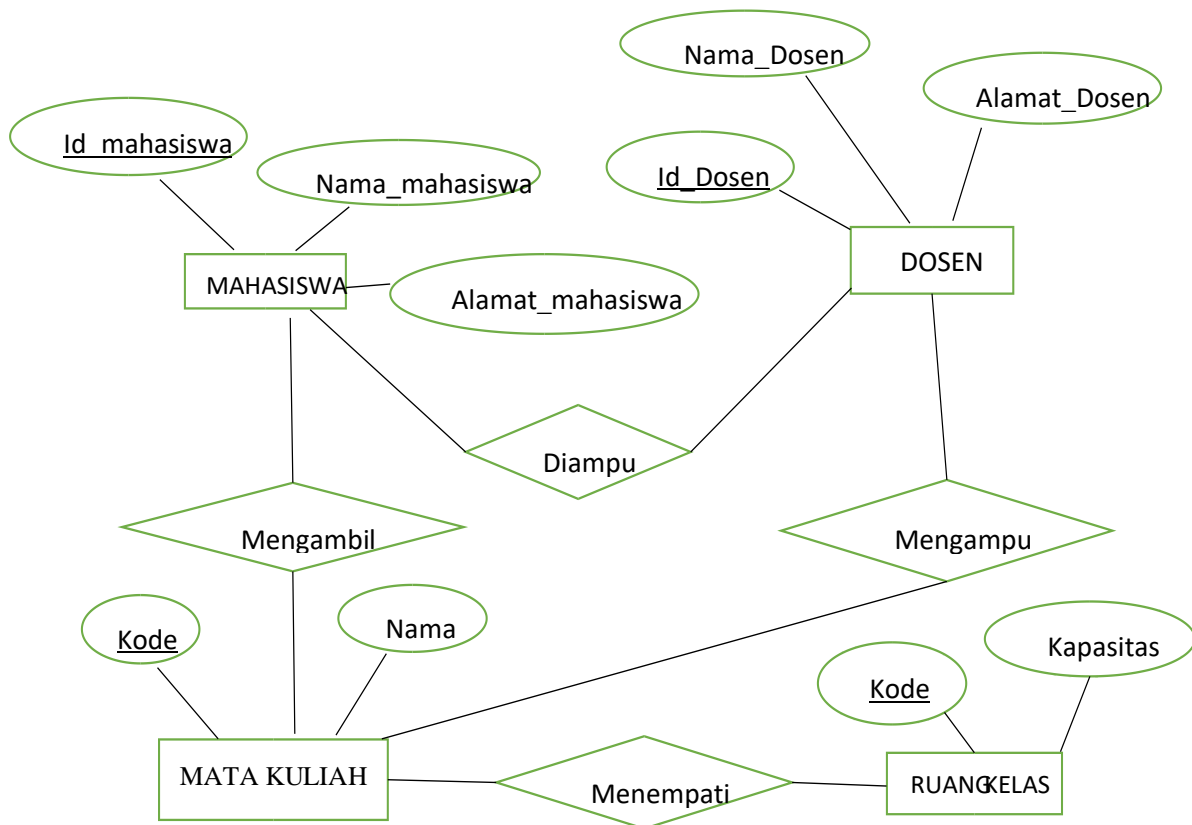
▪ **Mata_kuliah menempati Ruang_kelas :**

- **Tabel utama :** Mata_kuliah, Ruang_kelas
- **Tabel kedua :** mata_kuliah_fill_ruang_kelas
- **Relationship :** many-to-many(m:n)
- **Attribute penghubung :** Kode_matakuliah, Kode_ruangkelas (FK Kode_matakuliah, Kode_ruangkelas di **mata_kuliah_fill_ruang_kelas**)

▪ **Dosen menggunakan Ruang_kelas:**

- **Tabel utama :** Dosen, Ruang_kelas
- **Tabel kedua :** dosen_use_ruang_kelas
- **Relationship :** many-to-many(m:n)
- **Attribute penghubung :** Kode_dosen, Kode_ruangkelas (FK Kode_dosen, Kode_ruangkelas di **dosen_use_ruang_kelas**)

4) Menggambar ER Diagram



2. Ambil contoh sembarang database (harus berbeda untuk setiap mahasiswa). Buatlah rancangan ER Diagram manual database tersebut dari tahap 1 sampai tahap 4, dengan ketentuan database minimal mengandung 4 buah entitas.

Jawab :

1) Menentukan entities (object-object dasar) yang perlu ada di database.

- **anggota** : menyimpan semua data pribadi semua anggota perpustakaan
- **pegawai** : menyimpan semua data pribadi semua pegawai perpustakaan
- **buku** : menyimpan semua data buku di perpustakaan
- **denda** : menyimpan informasi tentang denda

2) Menentukan attributes (sifat-sifat) masing-masing entity sesuai kebutuhan database

● **anggota :**

- no_anggota : nomor identitas untuk anggota perpustakaan (integer) PK
- nama : nama untuk semua anggota perpustakaan (varchar(45))
- alamat : alamat lengkap anggota perpustakaan (varchar(255))
- tgl_lahir : tanggal lahir semua anggota perpustakaan (varchar(45))
- jurusan : jurusan semua anggota perpustakaan (varchar(45))

● **pegawai :**

- no_pegawai : nomor identitas untuk pegawai perpustakaan (integer) PK
- nama : nama untuk semua pegawai perpustakaan (varchar(45))
- alamat : alamat lengkap pegawai perpustakaan (varchar(255))
- no_tlp : nomor telp semua pegawai perpustakaan (integer)
- jabatan : jabatan dari semua pegawai perpustakaan (varchar(45))

- **buku :**

- no_buku : nomor untuk buku perpustakaan (integer) PK
- judul : judul dari buku perpustakaan (varchar(45))
- pengarang : pengarang dari buku perpustakaan (varchar(45))
- thn_terbit : tahun terbit dari buku perpustakaan (integer)
- penerbit : penerbit dari buku perpustakaan (varchar(45))

- **denda :**

- kode_denda : kode denda dari perpustakaan (integer) PK
- tarif_denda : tarif dari denda di perpustakaan (varchar(45))
- jenis_denda : jenis denda di perpustakaan (varchar(45))
- tgl_pinjam : tanggal peminjaman buku di perpustakaan (varchar(45))
- no_anggota : nomor identitas untuk anggota perpustakaan (integer)

3) Menentukan relationship (hubungan) antar entitas

	anggota	pegawai	buku	denda
anggota	-	n:1	m:n	1:n
pegawai		-	m:n	-
buku			-	-
denda				-

Hubungan

- **anggota meminjam buku**

- Tabel utama : anggota, buku
- Tabel kedua : anggota_borrow_buku
- Relationship : many-to-many (m:n)

- Attribute penghubung : no_anggota, no_buku (FK no_anggota, no_buku di anggota_borrow_buku)

- **anggota bayar denda**

- Tabel utama : anggota
- Tabel kedua : denda
- Relationship : one-to-many (1:n)
- Attribute penghubung : no_anggota (FK no_anggota di denda)

- **pegawai melayani anggota**

- Tabel utama : pegawai
- Tabel kedua : anggota
- Relationship : one-to-many (1:n)
- Attribute penghubung : no_pegawai (FK no_pegawai di anggota)

- **pegawai mendata buku**

- Tabel utama : pegawai, buku
- Tabel kedua : pegawai_register_buku
- Relationship : many-to-many (m:n)
- Attribute penghubung : no_pegawai, no_buku (FK no_pegawai, no_buku di pegawai_register_buku)

4. Menggambar ER Diagram

