Modul Pembelajaran
ERD

DATABASE SYSTEM

Kode Matakuliah: BDG23448

Bayu Nugroho, S.Kom., M.Eng

PROGRAM STUDI BISNIS DIGITAL
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
2026

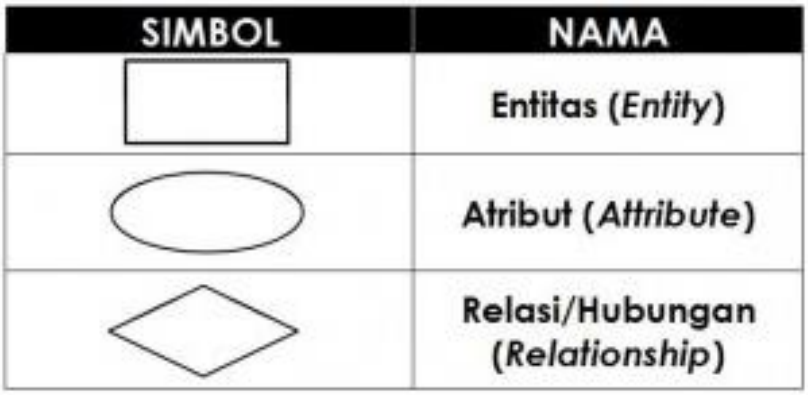
Entity Relationship Diagram

I. ERD

Entity Relationship Diagram (ERD) adalah diagram untuk menggambarkan objek/data yang perlu disimpan, atributnya, serta hubungan antar-entity. ERD membantu merancang struktur database sebelum tabel diimplementasikan

Komponen	Makna	Contoh sistem akademik
Entity	Objek yang datanya disimpan	Mahasiswa, Mata Kuliah
Attribute	Karakteristik entity atau hubungan	NIM, Nama, SKS, Semester, Nilai
Relationship	Hubungan antar-entity	Mahasiswa mengambil Mata Kuliah
Identifier/Key	Atribut pembeda record	NIM, Kode_MK, ID_Nilai

SIMBOL DASAR ERD:

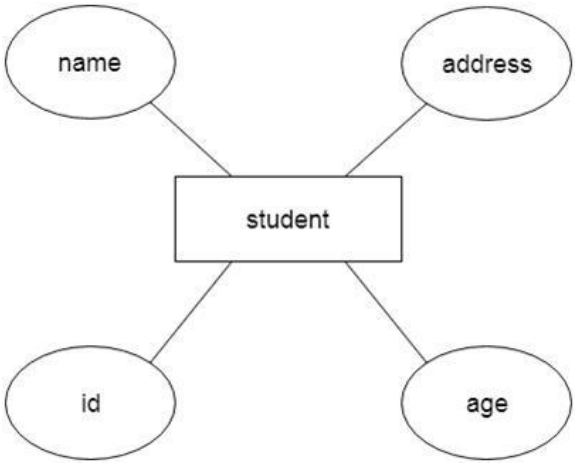


ERD merupakan alat bantu perancangan sebelum database diimplementasikan.

II. Komponen ERD

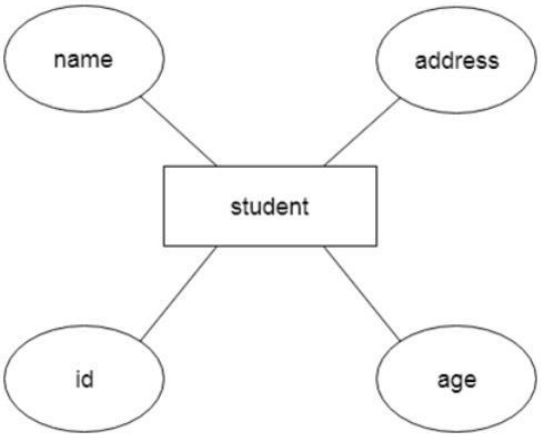
Tiga komponen ERD

A. Entity (Entitas)



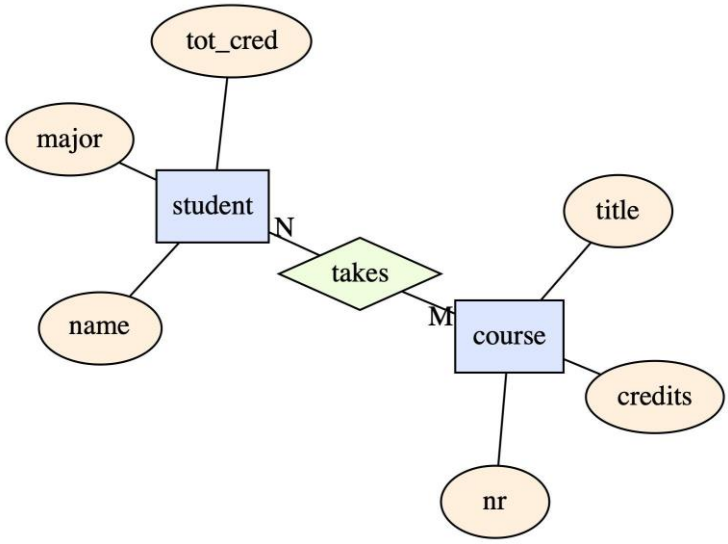
- Objek atau konsep yang datanya perlu disimpan dalam database.
- Contoh: MAHASISWA, DOSEN, MATA_KULIAH, PELANGGAN.
  Pertanyaan kunci: objek apa yang datanya perlu kita simpan?

B. Attribute (Atribut)



- Karakteristik atau informasi yang menjelaskan suatu entitas.
- Contoh atribut MAHASISWA: NIM, Nama, Prodi, Tanggal_Lahir.
  Pertanyaan kunci: informasi apa yang perlu disimpan tentang objek tersebut?

C. Relationship (Relasi)



- Hubungan yang menjelaskan keterkaitan antara entitas.
- Contoh: MAHASISWA mengambil MATA_KULIAH.
  Pertanyaan kunci: bagaimana satu objek berhubungan dengan objek lainnya?

III. Contoh ERD

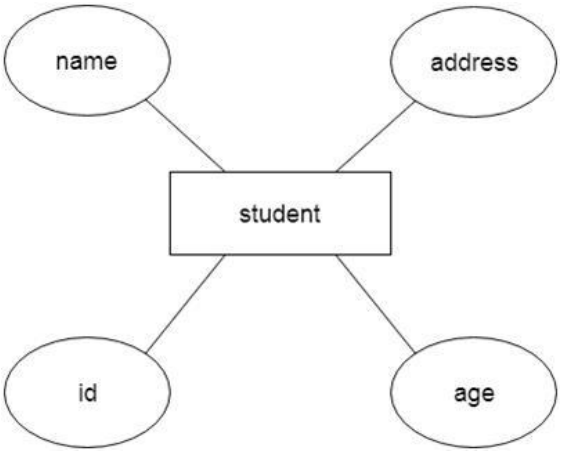
Sebuah perguruan tinggi ingin menyimpan data mahasiswa. Setiap mahasiswa memiliki NIM, nama, dan program studi.

Dari kasus tersebut, dapat kita identifikasi:

- **Entitas**: `MAHASISWA`
- **Atribut**: `NIM`, `Nama`, `Prodi`



Dalam notasi ERD yang lebih formal, entitas sering digambarkan sebagai persegi panjang, sedangkan atribut dapat digambarkan sebagai oval. Dalam notasi Crow's Foot yang banyak digunakan untuk desain database, entitas biasanya digambarkan seperti kotak berisi nama dan daftar kolom



Student
id name age address

Sebuah bengkel ingin membuat aplikasi untuk menyimpan data pelanggan. Data yang disimpan meliputi **ID pelanggan**, **nama pelanggan**, **nomor telepon**, dan **alamat**

Entitas: ?

Atribut: ?

KESIMPULAN:

1. **Entity**: objek apa yang datanya disimpan?
2. **Attribute**: informasi apa yang disimpan tentang objek itu?
3. **Relationship**: bagaimana objek tersebut berhubungan dengan objek lainnya?

Latihan 1

Mengidentifikasi Entity dan Attribute

Misalnya, kita akan membuat aplikasi perpustakaan dengan informasi berikut:

- Perpustakaan memiliki data buku.
- Setiap buku memiliki kode buku, judul, pengarang, dan tahun terbit.
- Perpustakaan juga memiliki data anggota.
- Setiap anggota memiliki nomor anggota, nama, alamat, dan nomor telepon.

Entitas: ?
Atribut: ?

Latihan 2

Mengidentifikasi Entity dan Attribute

Sebuah universitas menyimpan data mahasiswa, yang meliputi NIM, nama mahasiswa, program studi, dan alamat. Universitas juga menyimpan data dosen, yang meliputi NIDN, nama dosen, dan bidang keahlian

Entitas: ?
Atribut: ?

Latihan 3

Mengidentifikasi Entity dan Attribute

(STRUK / NOTA / INVOICE)

Relationship Pada ERD

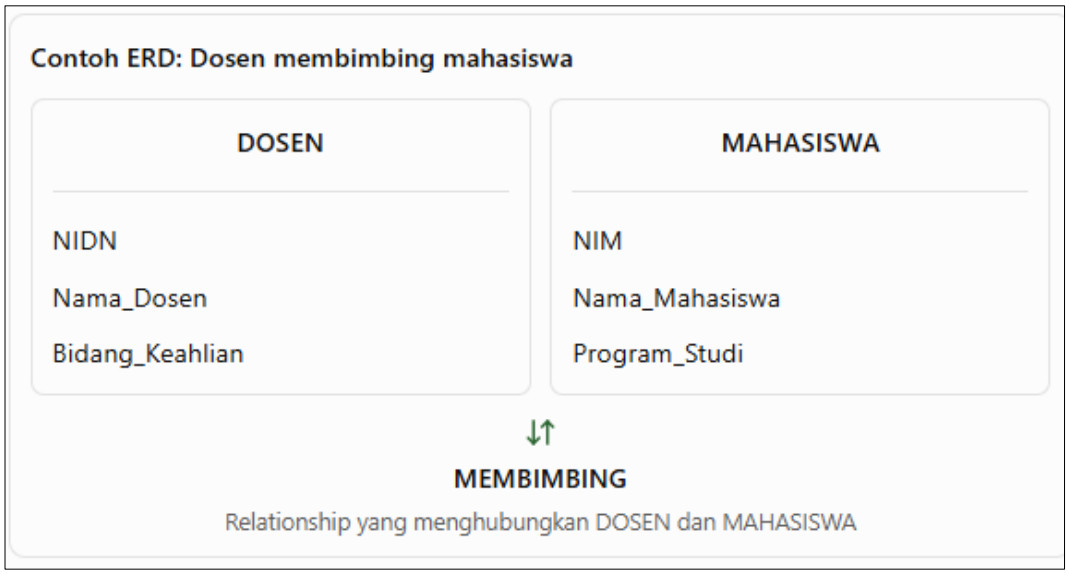
I. Relationship

Relationship adalah hubungan antara dua atau lebih entity dalam suatu sistem.

Ccontoh aplikasi akademik.

- Entity MAHASISWA menyimpan data mahasiswa.
- Entity DOSEN menyimpan data dosen.
- Seorang dosen dapat membimbing mahasiswa.

Artinya, terdapat hubungan antara entity DOSEN dan MAHASISWA



Perhatikan perbedaannya:

Komponen	Contoh	Fungsi
Entity	DOSEN	Objek yang datanya disimpan
Attribute	Nama_Dosen	Informasi tentang entity
Relationship	MEMBIMBING	Hubungan antara entity

Latihan 4

Relationship pada Entity

Sebuah aplikasi akademik memiliki entity MAHASISWA dan MATA_KULIAH.

Mahasiswa dapat mengambil mata kuliah yang ditawarkan oleh kampus

---

II. Kardinalitas

Kardinalitas dalam basis data adalah konsep yang menunjukkan jumlah maksimum entitas atau baris data yang dapat **dihubungkan** dengan entitas lain melalui suatu himpunan relasi. Konsep ini sangat penting saat merancang struktur tabel dan mengoptimalkan kinerja kueri (query)

One-to-One (1:1 / Satu ke Satu):

- Satu baris pada Tabel A hanya boleh berhubungan dengan paling banyak satu baris pada Tabel B.
- *Contoh:* Data KTP seorang Warga Negara dengan data nomor Induk Kependudukan (NIK). Satu orang punya satu NIK, dan satu NIK hanya dimiliki oleh satu orang

One-to-Many (1:N / Satu ke Banyak):

- Satu baris pada Tabel A dapat berhubungan dengan banyak baris pada Tabel B, tetapi sebaliknya baris di Tabel B hanya terhubung ke satu baris di Tabel A.
- *Contoh:* Data Pelanggan dan Pesanan. Satu pelanggan bisa melakukan banyak pesanan, namun satu pesanan tertentu hanya dilakukan oleh satu pelanggan.

Many-to-Many (M:N / Banyak ke Banyak):

- Banyak baris di Tabel A dapat terhubung ke banyak baris di Tabel B.
- *Contoh:* Data Mahasiswa dan Mata Kuliah. Satu mahasiswa bisa mengambil banyak mata kuliah, dan satu mata kuliah bisa diambil oleh banyak mahasiswa. (Biasanya dipecah menjadi tabel perantara / *junction table* saat diimplementasikan ke database fisik)

Ringkasan tiga jenis kardinalitas

Kardinalitas	Makna	Contoh
1:1	Satu berhubungan dengan satu	Satu orang memiliki satu nomor identitas utama
1:N	Satu berhubungan dengan banyak	Satu dosen mengajar banyak mata kuliah
M:N	Banyak berhubungan dengan banyak	Banyak mahasiswa mengambil banyak mata kuliah

Kardinalitas ditentukan berdasarkan aturan bisnis sistem yang dirancang, bukan sekadar nama entity-nya. Jika aturan universitas memungkinkan satu mata kuliah diajar beberapa dosen, kardinalitasnya bisa berbeda.

Latihan 5

Kardinalitas

Sebutkan Contoh Lainnya (Kardinalitas)

- **(1:1)**
- **(1:N)**
- **(M:N)**

Tabel Penghubung

Contoh:

- Seorang mahasiswa dapat mengambil banyak mata kuliah.
- Satu mata kuliah dapat diambil oleh banyak mahasiswa

Mengubah Relationship M:N menjadi tabel penghubung

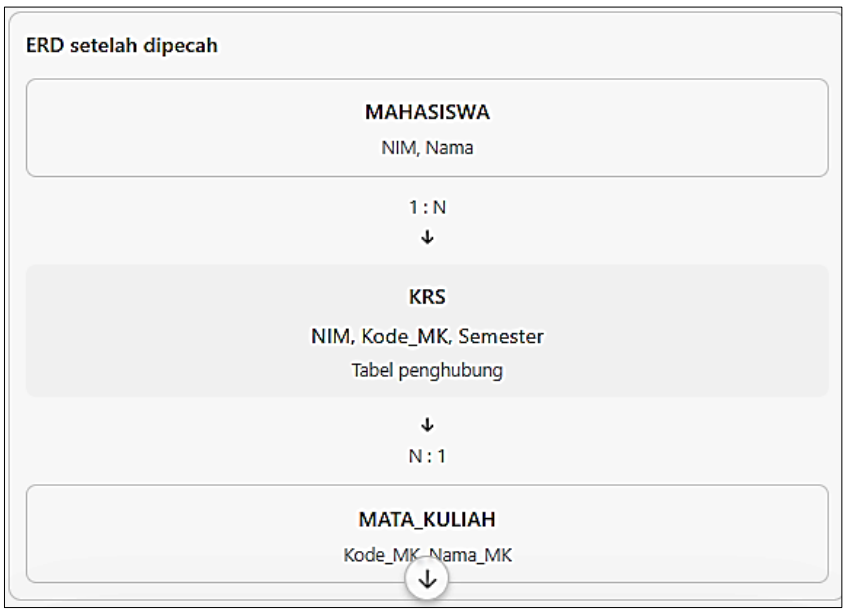
Mahasiswa	**Mata kuliah**
Andi	Basis Data
Andi	Pemrograman
Budi	Basis Data
Budi	Jaringan Komputer

Hubungan MAHASISWA dan MATA_KULIAH ini adalah Many-to-Many (M:N).

Mengapa perlu tabel penghubung?

Jika kita langsung menggabungkan data mahasiswa dan mata kuliah ke dalam satu tabel, nama mahasiswa, NIM, dan data mata kuliah bisa berulang.

Untuk merancang database relasional dengan baik, hubungan M:N biasanya dipecah menjadi dua hubungan 1:N, dengan menambahkan satu tabel penghubung. Tabel penghubung ini kita beri nama KRS



Sekarang hubungan M:N tersebut diimplementasikan melalui tabel KRS.

- Satu mahasiswa dapat memiliki banyak baris KRS.
- Satu mata kuliah dapat muncul pada banyak baris KRS.
- Setiap baris KRS menghubungkan satu mahasiswa dengan satu mata kuliah untuk pengambilan tertentu.

Contoh Tabel :

Tabel MAHASISWA

NIM	Nama
101	Andi
102	Budi

Tabel MATA_KULIAH

Kode_MK	Nama_MK
MK01	Basis Data
MK02	Pemrograman
MK03	Jaringan Komputer

Tabel KRS

NIM	Kode_MK	Semester
101	MK01	1
101	MK02	1
102	MK01	1
102	MK03	1

Dari tabel KRS, kita bisa mengetahui bahwa Andi mengambil Basis Data dan Pemrograman, sedangkan Budi mengambil Basis Data dan Jaringan Komputer

Pada tabel KRS:

- NIM merujuk ke mahasiswa yang mengambil mata kuliah.
- Kode_MK merujuk ke mata kuliah yang diambil.

Kedua kolom tersebut nantinya menjadi Foreign Key (FK). Untuk contoh sederhana ini, kombinasi NIM dan Kode_MK dapat dijadikan Composite Primary Key (PK), dengan asumsi mahasiswa tidak boleh memiliki lebih dari satu baris untuk mata kuliah yang sama. Jika pengambilan ulang atau beberapa kelas diperbolehkan, kunci perlu dirancang lebih lanjut.

Primary Key (PK) &
Foreign Key (FK)

Primary Key (PK) adalah kolom atau gabungan kolom yang digunakan untuk mengidentifikasi setiap baris data secara unik dalam sebuah tabel.

Contoh tabel MAHASISWA:

NIM	Nama	Program_Studi
101	Andi	Informatika
102	Budi	Sistem Informasi
103	Citra	Informatik

NIM menjadi Primary Key karena setiap mahasiswa memiliki NIM yang berbeda

NIM = Primary Key (PK) 

PK harus unik dan tidak boleh bernilai NULL. Nilainya tidak boleh sama untuk dua baris berbeda.

Foreign Key (FK) adalah kolom atau gabungan kolom yang merujuk pada Primary Key atau kunci unik yang sesuai pada tabel lain.

Contoh tabel KRS:

NIM	Kode_MK	Semester
101	MK01	1
101	MK02	1
102	MK01	1

Kolom NIM pada tabel KRS merujuk pada kolom NIM di tabel MAHASISWA. Artinya, setiap NIM dalam KRS harus mengacu pada mahasiswa yang tercatat di tabel MAHASISWA

NIM pada KRS = Foreign Key (FK)

FK menjaga hubungan antartabel agar tidak merujuk pada data induk yang tidak tersedia

Bedanya PK dan FK

Aspek	**Primary Key (PK)**	**Foreign Key (FK)**
Fungsi	Mengidentifikasi baris secara unik	Menghubungkan tabel
Keunikan	Harus unik	Boleh berulang
NULL	Tidak boleh NULL	Dapat NULL jika aturan relasi mengizinkan
Contoh	MAHASISWA.NIM	KRS.NIM

.

Hal penting: satu kolom bisa menjadi PK pada satu tabel, tetapi menjadi FK pada tabel lain.

Latihan 6

PK dan **FK**

Perhatikan tabel berikut.

Tabel DOSEN

NIDN	Nama_Dosen
D001	Pak Andi
D002	Bu Siti

Tabel MATA_KULIAH

Kode_MK	Nama_MK	NIDN
MK01	Basis Data	D001
MK02	Pemrograman	D001
MK03	Sistem Operasi	D002

Tentukan:

- **Primary Key (PK)**
- **Foreign Key (FK)**
- Tambahkan Atribut Lainnya di tiap Tabel