

APA ITU ERD?

Entity-Relationship Diagram (ERD) adalah representasi visual dari struktur data dalam sebuah sistem. ERD digunakan untuk memodelkan entitas, atribut, dan hubungan di antara entitas dalam sebuah basis data.

- **Entitas:** Objek nyata atau konsep (misalnya, Film, Pelanggan).
- **Atribut:** Properti atau karakteristik dari entitas (misalnya, Judul, Nama).
- **Hubungan:** Koneksi antara entitas (misalnya, Pelanggan membeli Tiket).

ERD membantu dalam merancang dan memahami struktur basis data secara logis sebelum implementasi teknis dilakukan.

KONTEKS

"CineMath Network" adalah jaringan bioskop yang tersebar di seluruh Indonesia. Mereka ingin membangun sistem manajemen terpusat untuk mengelola film, studio, penayangan,tiket, dan pelanggan. Setiap cabang bioskop memiliki jadwal tayang film yang berbeda dan pelanggan dapat memesan tiket secara online atau langsung di lokasi.

PERSYARATAN

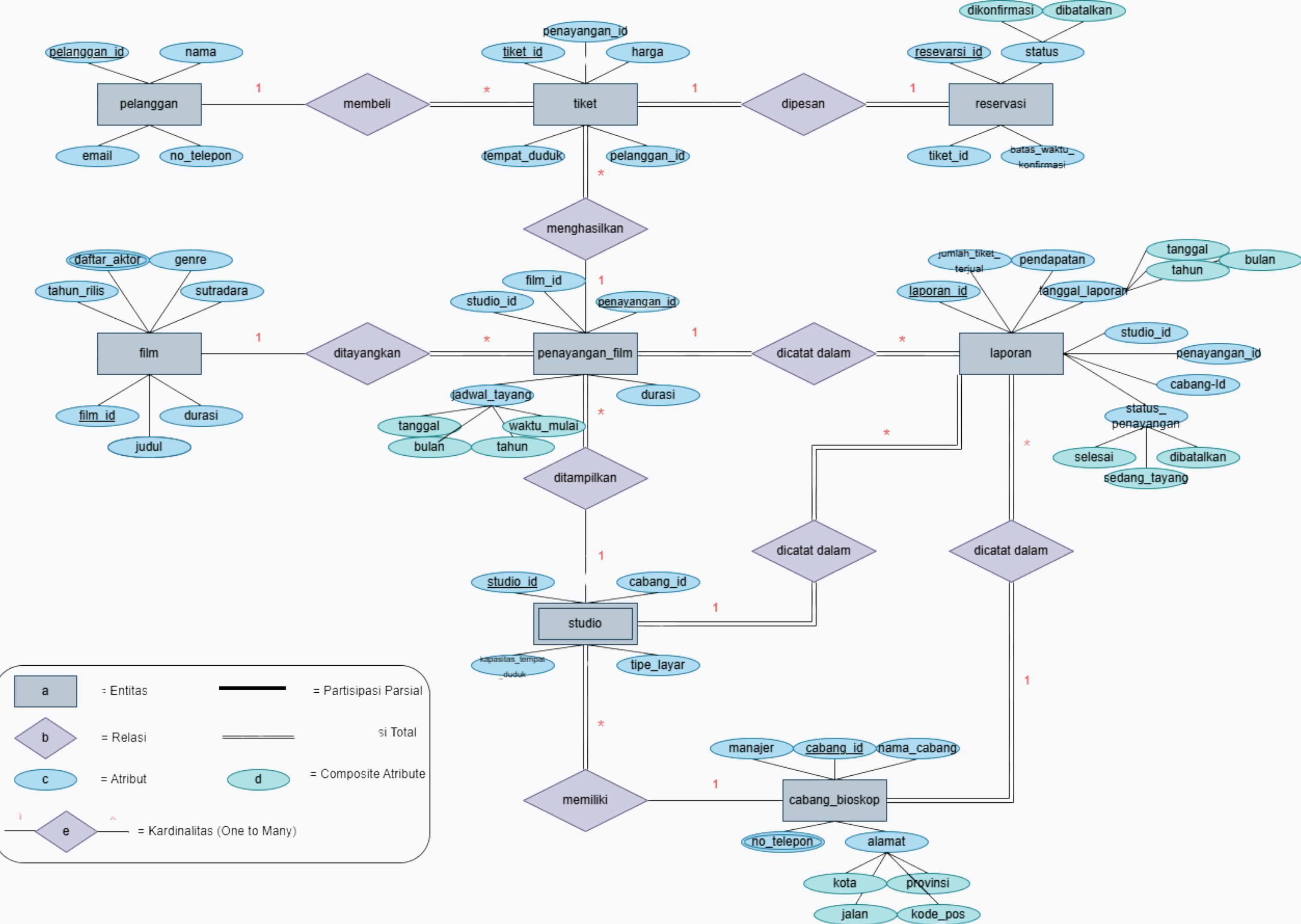
1. Setiap film memiliki ID unik, judul, genre, durasi, dan tahun rilis. Film juga memiliki daftar aktor dan sutradara.
2. Cabang bioskop memiliki nama unik, alamat, nomor telepon, dan manajer.
3. Setiap cabang memiliki beberapa studio, dan setiap studio memiliki ID unik dalam cabang, kapasitas tempat duduk, dan tipe layar (misalnya, 2D, 3D, IMAX).
4. Penayangan film di setiap studio memiliki jadwal tayang (tanggal dan waktu mulai), durasi, dan ID unik yang terkait dengan studio dan film.
5. Pelanggan dapat membeli tiket untuk penayangan tertentu. Setiap tiket memiliki ID unik, harga, dan tempat duduk yang terdaftar.
6. Pelanggan yang membeli tiket harus memiliki akun dengan ID unik, nama, alamat email, dan nomor telepon. Satu pelanggan dapat membeli lebih dari satu tiket dalam satu transaksi.
7. Sistem harus mengelola reservasi tiket untuk pelanggan. Pelanggan dapat memesan tiket terlebih dahulu, tetapi harus mengkonfirmasi atau membatalkan reservasi sebelum waktu tertentu.
8. Sistem juga mencatat laporan penjualan tiket berdasarkan cabang, studio, dan penayangan film.

SOAL

1. Buatlah ERD (Entity-Relationship Diagram) yang kompleks berdasarkan skenario di atas. ERD harus mencakup entitas berikut: Film, Cabang Bioskop, Studio, Penayangan Film, Tiket, Pelanggan, dan Reservasi.
2. Tunjukkan semua atribut kunci (primary key), atribut lainnya, hubungan antar entitas, kardinalitas, dan partisipasi dari setiap hubungan.
3. Jelaskan secara singkat bagaimana Anda menentukan kardinalitas dan partisipasi dari hubungan-hubungan tersebut dalam skenario ini.
4. Berikan contoh implementasi dalam SQL untuk membuat tabel Film dan tabel Tiket berdasarkan ERD Anda. Pastikan tabel tersebut memiliki constraint seperti primary key, foreign key, dan unique constraint jika diperlukan.

ENTITAS & ATRIBUT

- 
- **film**
 - film_id (Primary Key); judul; genre; durasi; tahun_rilis; daftar_aktor (Multivalued); sutradara
 - **penayangan_film**
 - penayangan_id (Primary Key); jadwal_tayang (Composite: tanggal, bulan, tahun, waktu_mulai); durasi; film_id (Foreign Key ke film); studio_id (Foreign Key ke studio)
 - **studio**
 - studio_id (Primary Key); cabang_id (Foreign Key); kapasitas_tempat_duduk; tipe_layar
 - **cabang_bioskop**
 - cabang_id (Primary Key); nama_cabang; alamat (jalan, kota, provinsi, kode_pos); no_telepon; manajer
 - **pelanggan**
 - pelanggan_id (Primary Key); nama; email; no_telepon
 - **tiket**
 - tiket_id (Primary Key); harga; tempat_duduk; penayangan_id (Foreign Key ke penayangan_film); pelanggan_id (Foreign Key ke pelanggan)
 - **reservasi**
 - reservasi_id (Primary Key); status (Composite: dikonfirmasi & dibatalkan); batas_waktu_konfirmasi; tiket_id (Foreign Key ke tiket)
 - **laporan**
 - laporan_id (Primary Key); jumlah_tiket_terjual; pendapatan; tanggal_laporan (Composite: tanggal, bulan, tahun); studio_id (Foreign Key ke studio); penayangan_id (Foreign Key ke penayangan_film); cabang_id (Foreign Key ke cabang); status_penayangan (Composite: selesai, sedang_tayang, dibatalkan)



HUBUNGAN, KARDINALITAS, DAN PARTISIPASI

Hubungan Ditayangkan (film → penayangan_film)

- **Kardinalitas:** 1:N: (Setiap film dapat memiliki banyak jadwal penayangan (contoh: film yang ditayangkan di beberapa hari/minggu). Namun, setiap jadwal penayangan hanya terkait dengan satu film.)
- **Partisipasi:**
 - Film → Parsial: Tidak semua film aktif ditayangkan, seperti film lama yang tidak dijadwalkan lagi.
 - Penayangan Film → Total: Setiap jadwal penayangan harus terkait dengan satu film.

Hubungan Ditampilkan (penayangan_film → studio)

- **Kardinalitas:** 1:N (Satu studio dapat menampilkan banyak jadwal penayangan (misalnya, film yang diputar setiap hari). Namun, setiap satu jadwal penayangan hanya terjadi di satu studio.)
- **Partisipasi:**
 - Studio → Parsial: Tidak semua studio selalu digunakan (misalnya, studio yang sedang direnovasi).
 - Penayangan Film → Total: Semua jadwal penayangan harus terjadi di satu studio.

Hubungan Memiliki (cabang_bioskop → studio)

- **Kardinalitas:** 1:N (Setiap cabang memiliki banyak studio, tetapi setiap studio hanya terkait ke satu cabang).
- **Partisipasi:**
 - Cabang Bioskop → Parsial: Tidak semua cabang mungkin memiliki studio, seperti cabang yang baru dibuka atau cabang yang sedang dibangun.
 - Studio → Total: Setiap studio harus terkait dengan satu cabang bioskop.

HUBUNGAN, KARDINALITAS, DAN PARTISIPASI

Hubungan Menghasilkan (penayangan_film ke tiket)

- **penayangan_film ke tiket: 1:N:** (Satu penayangan film dapat menghasilkan banyak tiket, tetapi setiap satu tiket hanya berlaku untuk satu penayangan film.)
- **Partisipasi**
 - penayangan_film → Parsial: Tidak semua penayangan harus memiliki tiket terjual (misalnya, jika tidak ada pelanggan).
 - tiket → Total: Setiap tiket harus terkait dengan satu jadwal penayangan.

Hubungan Membeli (pelanggan → tiket)

- **Kardinalitas: 1:N** (Satu pelanggan dapat membeli banyak tiket, tetapi setiap tiket hanya untuk satu pelanggan).
- **Partisipasi:**
 - Pelanggan → Parsial: Tidak semua pelanggan membeli tiket (contoh: pelanggan yang mendaftar akun tetapi belum membeli).
 - Tiket → Total: Setiap tiket harus dibeli oleh satu pelanggan.

Hubungan Dipesan (tiket → reservasi)

- **Kardinalitas: 1:1** (Satu tiket hanya dapat terkait dengan satu reservasi, dan satu reservasi menghasilkan satu tiket).
- **Partisipasi:**
 - Reservasi → Total: Setiap reservasi harus terkait dengan satu tiket.
 - Tiket → Total: Setiap tiket harus terkait dengan satu reservasi.

HUBUNGAN, KARDINALITAS, DAN PARTISIPASI

Hubungan Dicatat dalam (penayangan_film → laporan)

- **Kardinalitas:** 1:N (Satu jadwal penayangan dapat tercatat dalam banyak laporan, tetapi setiap satu laporan hanya mencatat satu penayangan)
- **Partisipasi:**
 - penayangan_film → Total: Semua jadwal penayangan harus tercatat dalam laporan.
 - laporan → Total: Setiap laporan mencatat setidaknya satu penayangan.

Hubungan Dicatat dalam (studio → laporan)

- **Kardinalitas:** 1:N (Satu studio dapat dicatat dalam banyak laporan, tetapi setiap satu laporan hanya mencatat satu studio.)
- **Partisipasi:**
 - studio → Total: Semua studio harus tercatat dalam laporan.
 - laporan → Total: Setiap laporan mencatat setidaknya satu studio.

Hubungan Dicatat dalam (cabang_bioskop → laporan)

- **Kardinalitas:** 1:N (Satu cabang bioskop dapat memiliki banyak laporan, tetapi setiap satu laporan hanya terkait dengan satu cabang.)
- **Partisipasi:**
 - cabang_bioskop → Total: Semua cabang bioskop harus tercatat dalam laporan.
 - laporan → Total: Setiap laporan mencatat satu cabang bioskop.

STUDIO SEBAGAI WEAK ENTITY DARI CABANG BIOSKOP

ID Tidak Unik Secara Global:

- Studio hanya memiliki ID unik dalam konteks Cabang Bioskop (StudioID). Untuk mengidentifikasi Studio secara unik, diperlukan kombinasi StudioID dan CabangID.
- **Contoh:**
 - Jika **StudioID** = S1, studio ini tidak unik tanpa mengacu ke **CabangID**.
 - Jika digabungkan dengan **CabangID** = C1, maka pasangan (C1, S1) menjadi unik.

Ketergantungan Eksistensial:

- Studio tidak dapat ada tanpa Cabang Bioskop, sehingga keberadaannya bergantung pada entitas Cabang Bioskop.
- Jika sebuah cabang bioskop dihapus, semua studio yang terkait dengan cabang tersebut juga akan dihapus (bergantung pada implementasi cascading dalam database).

Relasi Identifying:

- Weak entity seperti **Studio** harus memiliki relasi **Identifying Relationship** dengan entitas parent **Cabang Bioskop**.
- Relasi ini biasanya memiliki partisipasi **Total** dari **Studio**, artinya setiap studio harus terkait dengan satu cabang bioskop.

IMPLEMENTASI SQL MEMBUAT TABEL FILM & TIKET

```
④CREATE TABLE Film (
    film_id INT PRIMARY KEY,
    judul VARCHAR(100) NOT NULL,
    durasi INT NOT NULL, -- Durasi dalam menit
    tahun_rilis INT NOT NULL, -- Gunakan INT untuk tahun
    genre VARCHAR(50),
    sutradara VARCHAR(100),
    daftar_aktor TEXT -- Bisa digunakan untuk daftar aktor dalam format string/json
);

④CREATE TABLE Tiket (
    tiket_id INT PRIMARY KEY,
    harga DECIMAL(10, 2) NOT NULL, -- Harga tiket dengan 2 desimal
    tempat_duduk VARCHAR(10) NOT NULL, -- Misalnya A1, B2
    penayangan_id INT NOT NULL,
    pelanggan_id INT,
    FOREIGN KEY (penayangan_id) REFERENCES Penayangan_Film(penayangan_id)
        ON DELETE CASCADE,
    FOREIGN KEY (pelanggan_id) REFERENCES Pelanggan(pelanggan_id)
        ON DELETE SET NULL
);
```