Analisis Pemodelan Data: Entity Relationship Diagram (ERD)

Muhammad Riduwan Khafidi
Program Studi Teknik Rekayasa Perangkat Lunak
Politeknik Negeri Batam
4342611034@students.polibatam.ac.id

November 2, 2026

Abstract

Perancangan basis data berskala besar tidak dapat dilakukan secara instan. Perancang seringkali dihadapkan pada kompleksitas kebutuhan pengguna yang sulit diterjemahkan langsung ke dalam tabel fisik. Untuk menjembatani celah ini, digunakanlah Entity Relationship Diagram (ERD) sebagai model konseptual. ERD memvisualisasikan entitas (objek dunia nyata), atribut (properti entitas), dan relasi (hubungan antar entitas) secara intuitif. Makalah ini membedah secara komprehensif seluruh anatomi ERD, mulai dari notasi dasar, jenis-jenis atribut, hirarki kunci (keys), derajat relasi, kardinalitas, batasan partisipasi, hingga konsep entitas lemah (weak entity). Pemahaman mendalam terhadap ERD merupakan fondasi mutlak sebelum skema konseptual ini dipetakan menjadi skema relasional yang siap diimplementasikan ke dalam DBMS.

1 Mengapa Kita Membutuhkan ERD?

Bayangkan seorang arsitek yang akan membangun gedung pencakar langit. Ia tidak akan langsung menyuruh tukang untuk menyusun batu bata. Ia harus menggambar blueprint (cetak biru) terlebih dahulu. Begitu pula dalam pembuatan basis data. Pada skala kecil, perancang mungkin bisa langsung memutuskan struktur tabelnya. Namun, pada aplikasi berskala besar, seringkali tidak ada satu orang pun yang memahami secara menyeluruh data apa saja yang dibutuhkan.

Di sinilah Entity Relationship Diagram (ERD) berperan. ERD adalah teknik pemodelan data tingkat konseptual yang mengilustrasikan secara visual entitas dan relasi di dalam sebuah sistem. ERD bertindak sebagai bahasa universal yang menjembatani komunikasi antara perancang basis data dengan *customer* (pengguna). Melalui ERD, kita dapat memetakan integritas batasan (*integrity constraints*) dan aturan bisnis (*business rules*) sebelum menuliskan satu baris kode SQL pun.

2 Anatomi Dasar: Entitas, Atribut, dan Relasi

Sebuah ERD tersusun dari tiga komponen utama. Pertama adalah **Entitas** (*Entity*), yaitu objek di dunia nyata yang dapat dibedakan dari objek lain, misalnya "Pegawai", "Departemen",

atau "Mahasiswa". Dalam diagram, entitas digambarkan dengan persegi panjang. Kumpulan entitas yang serupa disebut *Entity Set*, dan setiap entitas set wajib memiliki kunci (*key*).

Kedua adalah **Atribut**, yaitu properti yang melekat pada entitas. Misalnya, entitas "Pegawai" memiliki atribut "Nama", "Nomor Pegawai", dan "Tanggal Lahir". Sebuah atribut dilambangkan dengan elips.

Ketiga adalah **Relasi** (*Relationship*), yaitu hubungan yang terjadi antar entitas, misalnya "Pegawai" bekerja di "Departemen". Relasi digambarkan dengan belah ketupat (*diamond*). Sebuah relasi juga bisa memiliki atribut sendiri, seperti atribut "Mulai" yang menyimpan tanggal kapan seorang pegawai mulai bekerja di departemen tersebut.

2.1 Tipe-Tipe Atribut

Tidak semua atribut digambarkan dengan cara yang sama. Terdapat beberapa tipe atribut khusus yang penting untuk dipahami. Perhatikan Gambar 1 di bawah ini.

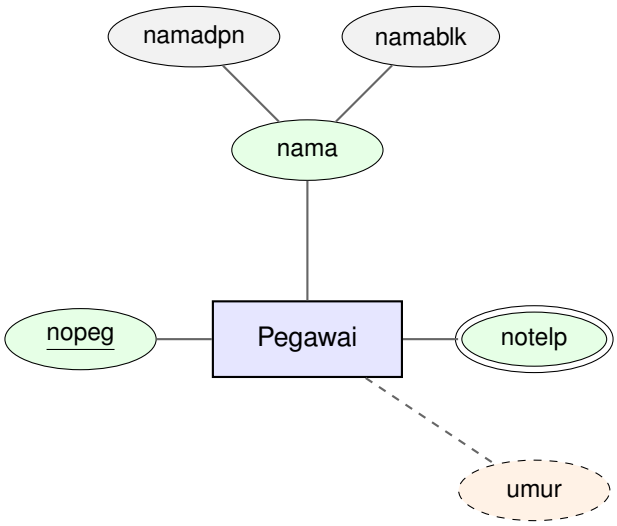


Figure 1. Ilustrasi Ragam Tipe Atribut pada Entitas Pegawai.

Atribut **Simple** (seperti `nopeg`) tidak dapat dipecah lagi. Atribut yang menjadi **Primary Key** (Kunci Utama) ditandai dengan garis bawah. Atribut **Composite** (seperti `nama`) dapat dipecah menjadi bagian yang lebih kecil (`namadpn` dan `namablk`). Atribut **Multi-valued** (seperti `notelp`) memungkinkan satu entitas memiliki lebih dari satu nilai, dilambangkan dengan elips garis ganda. Terakhir, atribut **Derived** (seperti `umur`) nilainya dihasilkan (diturunkan) dari atribut lain (seperti tanggal lahir), dilambangkan dengan elips garis putus-putus.

3 Hirarki Kunci (Keys)

Dalam basis data, tidak boleh ada dua entitas yang benar-benar identik. Oleh karena itu, kita membutuhkan sebuah Kunci (*Key*) untuk membedakannya. Proses identifikasi kunci ini memiliki tahapan bertingkat.

Langkah pertama adalah **Superkey**, yaitu satu atau lebih atribut yang secara unik dapat membedakan setiap entitas. Dari superkey ini, kita memilih yang paling minimal (tanpa atribut mubazir) yang disebut **Candidate Key**. Terakhir, dari beberapa kandidat, kita memilih satu yang paling representatif sebagai **Primary Key** (Kunci Utama). Sebagai contoh, atribut `NIK` atau `NIM` biasanya dipilih sebagai Primary Key, bukan atribut `Nama`, karena nama bisa saja sama.

4 Derajat Relasi dan Peran

Relasi antar entitas memiliki tingkat kompleksitas yang disebut Derajat Relasi (*Degree of Relationship*). Relasi paling sederhana adalah **Unary** (satu entitas berelasi dengan dirinya sendiri, misalnya Pegawai memiliki atasan sesama Pegawai). **Binary** adalah relasi antara dua entitas (Pegawai dengan Departemen). Sedangkan **Ternary** melibatkan tiga entitas sekaligus, seperti pada Gambar 2.

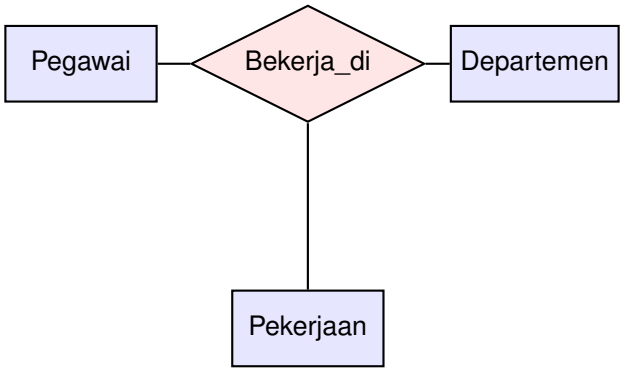


Figure 2. Relasi Ternary antara Pegawai, Departemen, dan Pekerjaan.

Selain derajat, entitas yang sama dapat memiliki peran (*role*) yang berbeda. Contoh klasiknya adalah relasi *Bertanggung_jawab_pada* antar entitas Pegawai. Di sini, Pegawai berperan sebagai "Supervisor" bagi pegawai lainnya, dan berperan sebagai "Bawahan" bagi supervisor tersebut.

5 Kardinalitas (Cardinality)

Kardinalitas mendefinisikan jumlah maksimum entitas yang dapat diasosiasikan dengan entitas lain melalui sebuah relasi. Terdapat empat jenis kardinalitas utama: One-to-One (1:1), One-to-Many (1:N), Many-to-One (N:1), dan Many-to-Many (N:M).

Sebagai contoh, relasi *Mengelola* antara Pegawai dan Departemen. Jika aturannya adalah "Setiap departemen hanya memiliki paling banyak satu manajer", dan "Seorang manajer bisa mengelola beberapa departemen", maka ini adalah relasi One-to-Many (dari sisi Pegawai) atau Many-to-One (dari sisi Departemen).

Dalam notasi ERD (seperti yang terlihat pada Gambar 3), kehadiran tanda panah (*arrow*) pada ujung garis relasi memiliki makna spesifik: **"Paling banyak satu"** (at most one). Tanda panah ini tidak menjamin keberadaan entitas, hanya membatasi jumlahnya.

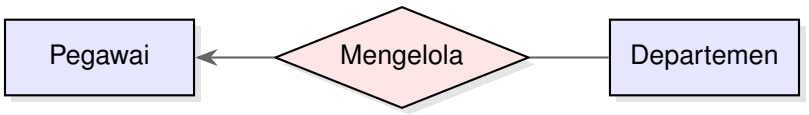


Figure 3. Relasi Many-to-One dengan Tanda Panah "Paling Banyak Satu" pada Sisi Pegawai.

6 Batasan Partisipasi (Participation Constraints)

Selain kardinalitas, kita juga harus mendefinisikan aturan partisipasi entitas di dalam sebuah relasi. Terdapat dua jenis partisipasi:

Pertama, **Partisipasi Total** (Total Participation), yang digambarkan dengan dua garis sejajar. Ini berarti setiap entitas dalam himpunan tersebut *wajib* berpartisipasi dalam relasi. Contoh: Sebuah entitas "Pinjaman" (uang) tidak mungkin ada di dunia nyata tanpa ada "Customer" yang meminjamnya. Maka, partisipasi Pinjaman dalam relasi Peminjaman adalah total.

Kedua, **Partisipasi Parsial** (Partial Participation), digambarkan dengan satu garis. Ini berarti beberapa entitas mungkin tidak berpartisipasi dalam relasi tersebut. Contoh: Tidak semua "Customer" harus memiliki "Pinjaman" (mungkin ia hanya punya tabungan).

7 Entitas Kuat vs Entitas Lemah

Sebuah entitas dikatakan **Entitas Kuat** (*Strong Entity*) jika ia memiliki cukup atribut untuk membentuk Primary Key-nya sendiri (contoh: Pegawai dengan NIP).

Sebaliknya, **Entitas Lemah** (*Weak Entity*) adalah entitas yang tidak memiliki Primary Key sendiri. Keberadaannya sangat bergantung pada entitas lain (entitas kuat). Contoh klasik adalah entitas "Pembayaran" pada sebuah "Pinjaman". Sebuah pembayaran tidak akan pernah ada jika tidak ada pinjaman.

Pada Gambar 4, entitas lemah digambarkan dengan persegi panjang bersusun dua (*double rectangle*). Hubungannya dengan entitas kuat digambarkan dengan belah ketupat bersusun dua yang disebut relasi pengidentifikasi (*Identifying Relationship*). Primary key dari entitas lemah didapat dari kombinasi Primary Key entitas kuat ditambah dengan *Discriminator* (garis bawah putus-putus) miliknya sendiri. Jadi, Primary Key untuk Pembayaran adalah (no-pinjam, no-byr).

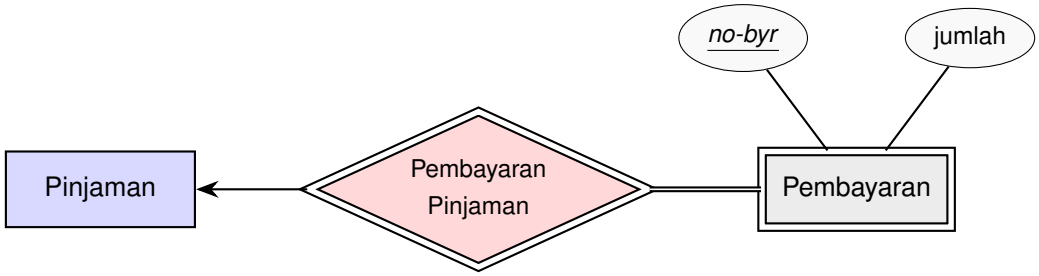


Figure 4. Representasi Entitas Lemah (Pembayaran) dan Relasi Pengidentifikasinya.

8 Kesimpulan

Entity Relationship Diagram (ERD) adalah seni menerjemahkan kebutuhan bisnis yang rumit menjadi sebuah peta visual yang terstruktur. Melalui pemahaman terhadap anatomi entitas, atribut, relasi, hirarki kunci, kardinalitas, batasan partisipasi, hingga entitas lemah, seorang perancang basis data dapat membangun fondasi yang kokoh. ERD bukanlah sekadar gambar, melainkan kontrak konseptual yang memastikan bahwa data yang akan disimpan nantinya benar-benar merepresentasikan realitas bisnis secara akurat, efisien, dan bebas dari anomali.

Referensi

References

[1] Lubis, A. I. (2026). *Entity Relationship Diagram* [Slide Presentasi Mata Kuliah Pengantar Basis Data]. Politeknik Negeri Batam.

[2] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill Education.