

## Perancangan Basis Data: Konsep Entitas, Atribut, dan Relasi

Fathan Mujaddid Akmal<sup>1</sup>, Jiyah Suhada<sup>2\*</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspittek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: [1\\*dosen03139@unpam.ac.id](mailto:1*dosen03139@unpam.ac.id)

(\* : coressponding author)

**Abstrak**—Perancangan basis data merupakan tahap penting dalam pengembangan sistem informasi yang menentukan kualitas pengelolaan data. Salah satu pendekatan yang banyak digunakan adalah *Entity-Relationship Diagram* (ERD), yang berfungsi untuk memodelkan hubungan antar entitas, atribut, serta relasi di dalam sistem. Artikel ini membahas konsep dasar entitas, atribut, dan relasi, serta meninjau beberapa penelitian terbaru mengenai penerapannya dalam perancangan basis data. Dengan menggunakan metode *Systematic Literature Review* (SLR), studi ini menganalisis kontribusi ERD dalam meningkatkan efisiensi desain database, meminimalisir kesalahan konseptual, serta mendukung pengembangan sistem yang lebih terstruktur. Hasil kajian menunjukkan bahwa pemahaman mendalam mengenai komponen inti ERD sangat penting untuk menghasilkan rancangan basis data yang konsisten, efisien, dan mudah diimplementasikan. Penelitian ini juga memberikan rekomendasi strategi perbaikan desain, seperti identifikasi entitas yang tepat, penggunaan simbol standar, serta evaluasi iteratif desain *database*.

**Kata Kunci:** Perancangan Basis Data, *Entity-Relationship Diagram*, Entitas, Atribut, Relasi

**Abstract**—Database design is a crucial stage in information system development that determines the quality of data management. One widely used approach is the Entity-Relationship Diagram (ERD), which functions to model the relationships between entities, attributes, and relationships within a system. This article discusses the basic concepts of entities, attributes, and relationships and reviews recent research on their application in database design. Using the Systematic Literature Review (SLR) method, this study analyzes the contribution of ERDs to improving database design efficiency, minimizing conceptual errors, and supporting the development of more structured systems. The results indicate that a thorough understanding of the core components of ERDs is crucial for producing consistent, efficient, and easy-to-implement database designs. This study also provides recommendations for design improvement strategies, such as proper entity identification, the use of standard symbols, and iterative evaluation of database designs.

**Keywords:** Database Design, *Entity-Relationship Diagram*, Entities, Attributes, Relationships

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Basis data merupakan komponen fundamental dalam sistem informasi modern karena berfungsi menyimpan, mengelola, dan menyediakan data yang terstruktur. Kualitas suatu basis data sangat bergantung pada tahap perancangannya, sehingga diperlukan pendekatan yang sistematis dan tepat. Salah satu metode yang banyak digunakan adalah *Entity-Relationship Diagram* (ERD), yang menggambarkan entitas, atribut, dan relasi dalam bentuk diagram konseptual. Dengan ERD, hubungan antar data dapat dipahami secara visual, memudahkan komunikasi antara perancang sistem dan pemangku kepentingan.

Konsep entitas, atribut, dan relasi menjadi inti dari perancangan basis data. Entitas mewakili objek atau kejadian yang relevan dengan sistem, atribut mendeskripsikan karakteristik dari entitas, sedangkan relasi menunjukkan keterhubungan antar-entitas. Ketiga komponen ini harus dipahami secara mendalam agar rancangan database dapat memenuhi kebutuhan pengguna, menghindari redundansi, serta menjaga integritas data.

### 1.2 Permasalahan

Dalam praktiknya, banyak terjadi kesalahan pada tahap perancangan database, khususnya dalam penggunaan ERD. Kesalahan umum meliputi salah mengidentifikasi entitas dan atribut, penggunaan simbol yang tidak sesuai standar, hingga kesalahan dalam menentukan kardinalitas relasi. Hal ini menyebabkan rancangan basis data tidak sesuai kebutuhan, sulit diimplementasikan, bahkan menimbulkan inkonsistensi data.

### 1.3 Tujuan Penulisan

Artikel ini bertujuan untuk membahas secara konseptual perancangan basis data dengan menekankan pada peran entitas, atribut, dan relasi dalam ERD. Selain itu, artikel ini juga menguraikan kesalahan umum yang sering terjadi serta memberikan arahan sederhana untuk meminimalisasi permasalahan dalam perancangan.

### 1.4 Manfaat Penulisan

Tulisan ini diharapkan dapat memberikan manfaat berupa:

- a. **Akademis:** Menjadi referensi tambahan bagi mahasiswa dalam memahami dasar perancangan basis data.
- b. **Praktis:** Memberikan panduan konseptual bagi praktisi atau pengembang sistem dalam membuat desain database yang lebih efektif.
- c. **Pengembangan Sistem:** Memberikan dasar teori yang dapat diterapkan dalam berbagai sistem informasi untuk menghasilkan rancangan yang terstruktur dan mudah dipelihara.

## 2. METODE PENELITIAN

### 2.1 Konsep Entitas, Atribut, dan Relasi

Entitas, atribut, dan relasi merupakan komponen inti yang membentuk dasar perancangan basis data. Entitas merepresentasikan objek nyata maupun abstrak yang relevan dengan sistem, seperti *Mahasiswa*, *Dosen*, atau *Mata Kuliah* dalam sistem akademik. Keberadaan entitas sangat penting karena menjadi titik awal dalam membangun struktur database. Tanpa identifikasi entitas yang tepat, rancangan database berpotensi mengandung redundansi dan kehilangan informasi penting.

Atribut adalah karakteristik atau detail yang melekat pada entitas dan digunakan untuk mendeskripsikan entitas tersebut. Contohnya, entitas *Mahasiswa* memiliki atribut *NIM*, *Nama*, *Alamat*, dan *Tanggal Lahir*. Beberapa atribut memiliki peran khusus, seperti *primary key* untuk memberikan identitas unik pada entitas, dan *foreign key* untuk menghubungkan entitas dengan entitas lain. Pemilihan atribut yang relevan akan menentukan sejauh mana database dapat dikelola secara efisien dan terhindar dari duplikasi data.

Relasi berfungsi untuk menghubungkan antar-entitas agar data dapat terintegrasi dengan baik. Relasi biasanya disertai dengan kardinalitas yang menggambarkan tingkat hubungan, seperti *one-to-one* (1:1), *one-to-many* (1:M), atau *many-to-many* (M:N). Misalnya, satu *Dosen* dapat mengajar banyak *Mata Kuliah* (1:M), atau satu *Mahasiswa* dapat mengambil banyak *Mata Kuliah* (M:N). Relasi yang dirancang dengan benar menjamin konsistensi data dan memungkinkan sistem basis data mencerminkan realitas dunia nyata secara akurat. Dengan pemahaman mendalam terhadap entitas, atribut, dan relasi, perancang database dapat menghasilkan struktur yang lebih terorganisir, efisien, serta mudah diimplementasikan.

## 3. ANALISA DAN PEMBAHASAN

### 3.1 Entitas

Entitas merupakan komponen dasar dalam perancangan basis data yang digunakan untuk merepresentasikan objek nyata maupun konsep abstrak yang memiliki arti penting dalam suatu sistem informasi. Objek yang dimaksud dapat berupa manusia, benda, tempat, peristiwa, atau bahkan ide yang relevan dengan domain aplikasi. Dalam konteks ERD, entitas digambarkan dalam bentuk persegi panjang yang dihubungkan dengan atribut-atributnya.

Setiap entitas memiliki karakteristik yang unik sehingga dapat dibedakan dari entitas lainnya. Oleh karena itu, pada saat perancangan, penting untuk mengidentifikasi entitas dengan benar berdasarkan kebutuhan sistem. Misalnya, dalam sistem akademik, entitas dapat berupa *Mahasiswa*, *Dosen*, atau *Mata Kuliah*. Proses identifikasi entitas sering kali dilakukan dengan menganalisis kata benda pada deskripsi kebutuhan sistem, karena kata benda biasanya menunjukkan calon entitas yang relevan.

Entitas sendiri terbagi menjadi dua jenis utama, yaitu entitas kuat dan entitas lemah. Entitas kuat adalah entitas yang dapat berdiri sendiri dan memiliki *primary key* sebagai identitas unik, sedangkan entitas lemah bergantung pada entitas lain karena tidak memiliki identitas unik tanpa keterkaitan tersebut. Misalnya, dalam sistem penjualan, entitas *Faktur* merupakan entitas kuat, sedangkan *Detail Faktur* dianggap entitas lemah karena keberadaannya tergantung pada entitas *Faktur*. Dengan memahami klasifikasi ini, perancang dapat menentukan struktur database yang lebih teratur dan konsisten.

### 3.2 Atribut

Atribut merupakan karakteristik atau properti yang dimiliki oleh entitas maupun relasi dalam basis data. Dalam ERD, atribut digambarkan dengan elips yang dihubungkan ke entitas atau relasi terkait. Atribut berfungsi untuk memberikan informasi detail sehingga setiap entitas dapat dikenali dengan jelas. Misalnya, entitas *Mahasiswa* memiliki atribut *NIM*, *Nama*, *Alamat*, dan *Tanggal Lahir*.

Atribut dalam perancangan basis data dapat dibedakan ke dalam beberapa jenis sebagai berikut:

#### 3.2.1 Atribut Sederhana (*Simple Attribute*)

Atribut sederhana adalah atribut yang tidak dapat diuraikan lagi menjadi bagian yang lebih kecil. Contohnya adalah *Nama* atau *Umur*. Atribut sederhana biasanya bersifat langsung dan tunggal, sehingga mudah diimplementasikan ke dalam skema tabel.

#### 3.2.2 Atribut Komposit (*Composite Attribute*)

Atribut komposit adalah atribut yang masih bisa dipecah menjadi subatribut. Misalnya, atribut *Alamat* dapat diuraikan menjadi *Jalan*, *Kota*, *Provinsi*, dan *Kode Pos*. Pemodelan atribut komposit memudahkan perancang dalam menyimpan data yang lebih detail dan terstruktur.

#### 3.2.3 Atribut Multivalued (*Multivalued Attribute*)

Atribut multivalued adalah atribut yang dapat memiliki lebih dari satu nilai untuk satu entitas. Contoh paling umum adalah *Nomor Telepon*, karena satu orang bisa memiliki beberapa nomor telepon. Dalam ERD, atribut multivalued biasanya digambarkan dengan elips ganda.

#### 3.2.4 Atribut Turunan (*Derived Attribute*)

Atribut turunan adalah atribut yang nilainya diperoleh dari atribut lain. Misalnya, *Umur* dapat dihitung dari atribut *Tanggal Lahir*. Atribut turunan tidak selalu disimpan langsung dalam database, tetapi dapat dihitung ketika dibutuhkan agar tidak terjadi redundansi data.

#### 3.2.5 Atribut Kunci (*Key Attribute*)

Atribut kunci adalah atribut yang berfungsi sebagai identitas unik untuk membedakan antar instance entitas. *Primary key* merupakan atribut kunci utama yang harus ada pada setiap entitas. Selain itu, terdapat pula *foreign key* yang digunakan untuk menghubungkan antar-entitas melalui relasi. Pemilihan atribut kunci yang tepat sangat penting untuk menjaga integritas referensial dalam basis data.

### 3.3 Relasi

Relasi adalah komponen yang digunakan untuk menggambarkan hubungan antar-entitas dalam basis data. Relasi menunjukkan bagaimana entitas saling berhubungan sehingga data dapat diintegrasikan dalam satu sistem. Dalam ERD, relasi digambarkan dengan simbol belah ketupat yang dihubungkan dengan entitas yang terkait. Relasi sangat penting karena melalui relasi, perancang dapat memastikan bahwa struktur data benar-benar mencerminkan kondisi dunia nyata.

#### 3.3.1 Kardinalitas Relasi

Kardinalitas adalah aturan yang menunjukkan jumlah minimum dan maksimum keterhubungan antar-entitas. Kardinalitas terdiri atas:

- One-to-One (1:1)** → satu entitas hanya dapat berhubungan dengan satu entitas lain, misalnya seorang *Warga Negara* hanya memiliki satu *Paspor*.

- b. **One-to-Many (1:M)** → satu entitas dapat berhubungan dengan banyak entitas lain, misalnya satu *Dosen* dapat mengajar banyak *Mata Kuliah*.
- c. **Many-to-Many (M:N)** → banyak entitas dapat berhubungan dengan banyak entitas lainnya, misalnya *Mahasiswa* dapat mengambil banyak *Mata Kuliah* dan sebaliknya satu *Mata Kuliah* dapat diambil oleh banyak *Mahasiswa*.

### 3.3.2 Derajat Relasi

Derajat relasi menunjukkan jumlah entitas yang terlibat dalam sebuah hubungan. Terdapat beberapa derajat relasi yang umum:

- a. **Relasi Binary** → melibatkan dua entitas, misalnya relasi *Mengajar* antara *Dosen* dan *Mata Kuliah*.
- b. **Relasi Ternary** → melibatkan tiga entitas, misalnya relasi *Penjualan* yang menghubungkan *Pelanggan*, *Barang*, dan *Penjual*.
- c. **Relasi N-ary** → melibatkan lebih dari tiga entitas. Meskipun jarang digunakan, relasi jenis ini bisa muncul pada sistem yang kompleks.

### 3.3.3 Relasi Asosiasi (*Associative Entity*)

Relasi Many-to-Many (M:N) pada tahap konseptual biasanya sulit diimplementasikan langsung pada model relasional. Oleh karena itu, relasi ini diubah menjadi entitas baru yang disebut **entitas asosiasi** atau **associative entity**. Misalnya, relasi *Mengambil* antara entitas *Mahasiswa* dan *Mata Kuliah* dapat diubah menjadi entitas *KRS* (Kartu Rencana Studi). Dengan demikian, entitas asosiasi ini memungkinkan penyimpanan informasi tambahan seperti *Semester* atau *Nilai*.

## 3.4 Kesalahan Umum dalam Mendesain ERD

Meskipun ERD sudah menjadi alat standar dalam perancangan basis data, kesalahan masih sering terjadi baik pada tingkat pemula maupun praktisi berpengalaman. Kesalahan tersebut umumnya terbagi dalam tiga kategori: kesalahan konseptual, prosedural, dan teknis. Setiap kategori kesalahan memiliki dampak yang signifikan terhadap kualitas desain, sehingga perlu dipahami dengan baik agar dapat diminimalkan sejak awal.

### 3.4.1 Kesalahan Konseptual

Kesalahan konseptual muncul akibat kurangnya pemahaman tentang elemen dasar ERD. Contohnya adalah menempatkan atribut sebagai entitas, atau menjadikan entitas sebagai atribut. Misalnya, *Alamat* yang seharusnya menjadi atribut dari entitas *Mahasiswa*, justru dijadikan entitas terpisah. Kesalahan ini membuat model tidak sesuai dengan realitas yang hendak direpresentasikan, sehingga database yang dihasilkan berpotensi tidak konsisten dengan kebutuhan sistem.

### 3.4.2 Kesalahan Prosedural

Kesalahan prosedural terjadi ketika perancang tidak mengikuti aturan standar simbolisasi dalam ERD. Contoh yang sering dijumpai adalah menggambarkan relasi dengan persegi panjang alih-alih belah ketupat, tidak mencantumkan kardinalitas pada relasi, atau tidak menandai atribut kunci dengan tepat. Akibatnya, ERD sulit dipahami oleh pihak lain dan dapat menyebabkan salah tafsir dalam tahap konversi ke model logis.

### 3.4.3 Kesalahan Teknis

Kesalahan teknis terlihat sepele, tetapi berdampak pada keterbacaan diagram. Contoh kesalahan teknis adalah penggunaan ukuran font yang terlalu kecil, tata letak yang berantakan, atau warna yang tidak kontras sehingga diagram sulit dipahami. Walaupun tidak memengaruhi isi secara langsung, kesalahan teknis membuat komunikasi antaranggota tim menjadi tidak efektif. Diagram yang sulit dibaca berisiko ditafsirkan salah dan dapat berakibat fatal pada implementasi database.

## 3.5 Strategi Pemecahan Masalah

Untuk mengatasi kesalahan dalam perancangan ERD, diperlukan strategi yang sistematis dan terarah. Strategi ini mencakup tahapan sejak identifikasi awal hingga evaluasi akhir agar ERD yang

dihasilkan sesuai standar dan kebutuhan sistem. Tiga aspek utama yang perlu diperhatikan adalah identifikasi entitas dan atribut, penerapan notasi standar, serta perbaikan teknis dan estetika diagram.

### **3.5.1 Identifikasi Entitas dan Atribut dengan Tepat**

Langkah pertama adalah memastikan entitas dan atribut diidentifikasi secara benar. Analisis kata benda dalam dokumen kebutuhan sering digunakan untuk menentukan calon entitas, sementara kata sifat atau keterangan biasanya menunjuk pada atribut. Penentuan *primary key* dan *foreign key* juga harus dilakukan sejak awal untuk menjaga integritas data. Identifikasi yang cermat akan meminimalisasi risiko redundansi maupun kehilangan informasi penting.

### **3.5.2 Penerapan Notasi Standar**

Penggunaan simbol dan aturan notasi yang benar sangat penting agar ERD mudah dipahami secara universal. Entitas harus digambarkan dengan persegi panjang, atribut dengan elips, dan relasi dengan belah ketupat. Kardinalitas juga wajib dicantumkan pada setiap relasi untuk menjelaskan jumlah keterhubungan antar-entitas. Dengan penerapan notasi standar, diagram akan lebih konsisten, mudah dipelajari, dan dapat mengurangi salah tafsir saat dikonversi ke model logis maupun fisik.

### **3.5.3 Perbaikan Teknis dan Estetika Diagram**

Selain aspek konseptual dan prosedural, perancang juga perlu memperhatikan teknis dan estetika ERD. Diagram yang dibuat harus proporsional, dengan tata letak rapi, ukuran font jelas, dan penggunaan warna yang kontras namun tidak berlebihan. Kerapihan visual mempermudah komunikasi antara pengembang dan pemangku kepentingan. Setelah itu, proses *review* iteratif bersama tim sangat disarankan untuk memastikan desain sudah sesuai kebutuhan dan bebas dari kesalahan.

## **4. KESIMPULAN**

### **4.1 Kesimpulan**

Perancangan basis data merupakan fondasi penting dalam pembangunan sistem informasi, dan pemahaman terhadap konsep entitas, atribut, serta relasi menjadi kunci utama untuk menghasilkan desain yang efektif. Entitas berfungsi sebagai objek utama, atribut memberikan detail yang relevan, sementara relasi menghubungkan antar-entitas sehingga data dapat saling terintegrasi. Kesalahan yang sering terjadi dalam pembuatan ERD, seperti kesalahan konseptual, prosedural, maupun teknis, terbukti dapat mengganggu kualitas rancangan database dan menyebabkan inkonsistensi data. Oleh karena itu, pemahaman mendalam serta penerapan prinsip dasar perancangan yang benar menjadi hal yang mutlak untuk menciptakan basis data yang efisien, konsisten, dan sesuai kebutuhan sistem.

### **4.2 Saran**

Agar perancangan basis data dapat berjalan optimal, beberapa hal perlu diperhatikan. Pertama, lakukan identifikasi entitas dan atribut secara cermat dengan melibatkan analisis kebutuhan yang mendalam. Kedua, gunakan notasi dan simbol ERD sesuai standar internasional untuk menghindari salah tafsir. Ketiga, pastikan setiap relasi disertai dengan kardinalitas yang tepat agar hubungan antar data lebih jelas. Selain itu, perlu dilakukan evaluasi secara berkala melalui *review* bersama tim pengembang maupun pengguna akhir untuk meminimalisasi kesalahan. Dengan langkah-langkah tersebut, rancangan basis data akan lebih terstruktur dan siap diimplementasikan pada sistem informasi yang kompleks sekalipun.

## **REFERENCES**

- Afiifah, K., Azzahra, Z. F., & Anggoro, A. D. (2022). Analisis teknik entity-relationship diagram dalam perancangan database: Sebuah literature review. *Jurnal Intech*, 3(1), 8–11.

- Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). Analisis teknik entity-relationship diagram dalam perancangan database. *Jurnal Ekonomi Manajemen dan Bisnis*, 2(1), 98–102.
- Alderden, J. G., Sharkey, P. D., Kennerly, S. M., Ghosh, S., Barrett, R. S., Horn, S. D., Ghosh, S., & Yap, T. L. (2023). Developing a relational database for best practice data management: The Turn Everyone and Move for Ulcer Prevention Database. *CIN: Computers, Informatics, Nursing*, 41(2), 59–65.
- Trillo-Montero, D., Cosano-Lucena, S., Gonzalez-Redondo, M., Luna-Rodriguez, J. J., & Santiago, I. (2023). Design and development of a relational database management system (RDBMS) with open source tools for the processing of data monitored in a set of photovoltaic (PV) plants. *Applied Sciences*, 13(3), 1357.
- Elmasri, R., & Navathe, S. B. (2013). *Fundamentals of Database Systems* (6th ed.). Pearson.
- Date, C. J. (2014). *An Introduction to Database Systems* (8th ed.). Pearson.