

# Aplikasi Relasi dalam Pengelolaan Basis Data

Calvin Aditya Jonathan 13513077  
Program Sarjana Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
deus.caj@gmail.com

**Abstract** — Di zaman yang sudah modern ini, keperluan akan informasi sudah semakin tinggi. Manusia modern butuh informasi dalam jumlah besar dengan cepat. Tanpa sistem pengelolaan data yang baik, kebutuhan tersebut tidak akan terpenuhi. Sistem pengelolaan basis data relasional adalah salah satu solusi untuk masalah tersebut. Makalah ini membahas tentang sistem pengelolaan basis data yang didasarkan pada konsep relasi tersebut berikut dengan keunggulannya dibandingkan sistem pengelolaan basis data lain.

**Keywords** — Basis Data, Data, Pengelolaan, Relasi.

## I. PENDAHULUAN

Dewasa ini, seiring dengan perkembangan teknologi, kehidupan manusia sudah sangat bergantung kepada informasi. Jika informasi yang kita terima tidak cukup maka aktivitas yang akan kita lakukan menjadi terganggu. Contohnya jika kita tidak tahu informasi tentang jadwal kuliah hari ini.

Informasi tersebut didapat dengan mengolah data. Karena pola kehidupan manusia saat ini sangat bergantung terhadap informasi, tingkat kebutuhan manusia terhadap sistem pengelolaan data yang baik dan mangkus juga sangat tinggi. Tanpa sistem pengelolaan data yang baik dan efisien, data yang begitu banyak tersebar di sekeliling kita akan sukar diolah menjadi informasi berguna. Tidak hanya itu, data yang diolah juga seringkali terdapat dalam jumlah yang sangat banyak sehingga kita memerlukan sistem yang tepat untuk menghindari kekacauan dalam pengolahan data.

Hal ini mungkin tidak terlihat pada kehidupan sehari-hari, namun pada skala yang lebih besar sistem pengelolaan data sangat penting. Misalnya pada rumah sakit yang membutuhkan informasi secara cepat. Jika sistem pengelolaan data di rumah sakit tersebut buruk dan tidak mangkus, keberlangsungan aktivitas di rumah sakit akan sangat terpengaruh. Mungkin saja ada pasien yang namanya tercatat dua kali atau nomor urut pasien tidak teratur.

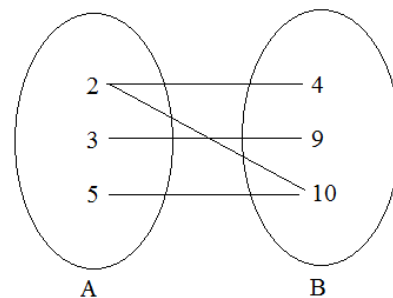
Oleh karena itu, diciptakanlah basis data relasional. Metode tersebut menghubungkan data dengan data lain seperti pada konsep relasi n-ary. Sistem pengelolaan basis data secara relasional dapat mengolah data dalam jumlah

besar dengan cepat dan tepat sehingga cocok untuk berbagai bidang kehidupan manusia.

## II. LANDASAN TEORI

### 1. Relasi

Pengertian relasi adalah suatu hubungan atau perkawanan antara elemen-elemen himpunan A dengan elemen-elemen himpunan B. Relasi biner adalah hubungan antara 2 elemen himpunan. Berikut adalah contoh relasi “faktor dari”:



**Gambar 1. Relasi R**

Himpunan A disebut daerah asal (*domain*) dari R dan himpunan B disebut daerah hasil (*range*) dari R. Pada Gambar 1, 2 dihubungkan dengan 4 sehingga dapat ditulis  $2 R 4$ , sedangkan 3 tidak terhubung dengan 4 sehingga dapat ditulis  $3 \not R 4$ . Relasi di atas juga dapat ditulis sebagai  $R = \{ (2, 4), (2, 10), (3, 9), (5, 10) \}$ .

Relasi dapat direpresentasikan oleh berbagai metode, beberapa di antaranya:

#### a. Diagram Panah

Representasi relasi menggunakan diagram panah adalah seperti Gambar 1.

#### b. Tabel

Berikut contoh representasi relasi dengan tabel:

**Tabel I. Representasi Relasi Menggunakan Tabel**

A	B
2	4
2	8
3	15
5	15

Kolom A adalah daerah asal dan kolom B adalah daerah hasil.

c. Matriks

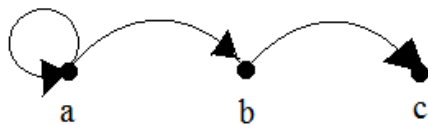
Relasi pada Tabel I dapat ditampilkan dalam bentuk matriks berikut:

$$\begin{matrix} & & 4 & 8 & 15 \\ 2 & & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \\ 3 & & & & \\ 5 & & & & \end{matrix}$$

1 menandakan ada relasi dan 0 jika tidak ada relasi.

d. Graf Berarah

Lain halnya dengan tiga cara representasi sebelumnya, graf berarah tidak didefinisikan untuk merepresentasikan relasi dari satu himpunan ke himpunan lainnya. Jika  $(a, b) \in R$  maka dibuat busur dari simpul a ke simpul b. Simpul a disebut simpul awal (*initial vertex*) dan simpul b disebut simpul tujuan (*terminal vertex*). Jika ada  $(a, a) \in R$  maka dibuat kalang atau *loop* pada simpul a. Contoh representasi relasi  $R = \{ (a, a), (a, b), (b, c) \}$  menggunakan graf berarah:



Gambar 2. Representasi Relasi dengan Graf Berarah

Relasi biner memiliki beberapa sifat:

a. Refleksif

Relasi R pada himpunan A dikatakan refleksif jika  $(a, a) \in R$  untuk setiap  $a \in A$ . Relasi tidak refleksif jika ada  $a \in A$  sedemikian sehingga  $(a, a) \notin R$ .

b. Menghantar

Relasi R pada himpunan A bersifat menghantar jika  $(a, b) \in R$  dan  $(b, c) \in R$  maka  $(a, c) \in R$  untuk  $a, b, c \in A$ .

c. Setangkup

Relasi R pada himpunan A disebut setangkup jika  $(a, b) \in R$  maka  $(b, a) \in R$  untuk  $a, b \in A$ . Relasi R tidak setangkup jika  $(a, b) \in R$  sedemikian sehingga  $(b, a) \notin R$ .

d. Tolak-setangkup

Relasi R pada himpunan A tolak-setangkup apabila  $(a, b) \in R$  dan  $(b, a) \in R$  hanya jika  $a = b$  untuk  $a, b \in A$ . Jika ada  $(a, b) \in R$  dan  $(b, a) \in R$  untuk  $a, b \in A$  yang mana a berbeda dengan b maka R tidak tolak-setangkup.

Relasi biner hanya menghubungkan dua buah himpunan, relasi yang menghubungkan lebih dari dua himpunan disebut relasi *n-ary*. Relasi *n-ary* memiliki derajat atau n. Jika  $n = 2$  maka relasi tersebut adalah relasi biner.

Relasi *n-ary* memiliki peran yang signifikan dalam pengelolaan basis data. Berikut adalah contoh relasi *n-ary*:

Tabel II. Himpunan Nama, NIM, Jenis Kelamin, dan Asal Daerah

Nama	NIM	Jenis Kelamin	Asal Daerah
Calvin	13513077	Laki-laki	Jawa Barat
Bobby	13511020	Laki-laki	Jakarta
Felicia	13510135	Perempuan	Jawa Timur

$M = \{ (Calvin, 13513077, Laki-laki, Jawa Barat), (Bobby, 13511020, Laki-laki, Jakarta), (Felicia, 13510135, Perempuan, Jawa Timur) \}$ . Relasi M adalah relasi *n-ary* berderajat 4 antara Nama, NIM, Jenis Kelamin, dan Asal Daerah.

2. Basis Data

Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah menggunakan perangkat lunak untuk menghasilkan informasi. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga pembaharuan data yang rumit.

Beberapa karakteristik basis data yang baik adalah sebagai berikut:

- Struktur basis data mangkus dan sistematis
- Redundansi data (data rangkap) sedikit atau tidak ada sama sekali
- Kebutuhan ruang kecil (tidak memakan memori yang banyak)
- Dapat dioperasikan dengan cepat

III. BASIS DATA RELASIONAL

1. Pengertian Basis Data Relasional

Sistem Pengelolaan Basis Data Relasional atau *Relational Database Management System* (RDMS) adalah sistem pengelolaan basis data yang menyimpan semua datanya dalam satu atau lebih tabel dua dimensi, yang mana setiap tabel merepresentasikan relasi antar data. Pengelolaan basis data secara relasional didasarkan pada konsep relasi *n-ary*. Contoh tabel pada basis data relasional:

Tabel III. Contoh Relasi pada Basis Data Relasional

Nomor KTP	Nama	Umur	Pekerjaan
111	Calvin	30	Dokter
222	Andrew	31	Guru
333	Joseph	32	Polisi

Pada tabel di atas kolom Nama, Umur, dan Pekerjaan merupakan atribut, sedangkan setiap baris merupakan data atau elemen relasi. Daerah asal dari suatu atribut adalah himpunan semua isi atribut tersebut.

## 2. Kunci Utama (*Primary Key*)

Atribut khusus yang dapat mengidentifikasi secara unik elemen dari relasi disebut dengan kunci utama. Pada Tabel II, nomor KTP adalah kunci utama karena setiap baris atau elemen relasi pasti memiliki nomor KTP yang berbeda. Setiap tabel atau relasi hanya memiliki satu kunci utama, jika ada lebih dari satu kemungkinan kunci utama maka kemungkinan-kemungkinan tersebut merupakan kunci kandidat (*candidate key*).

## 3. Kunci Komposit (*Composite Key*)

Terkadang, satu atribut tidak cukup untuk mengidentifikasi setiap elemen relasi secara unik. Dalam kasus tersebut, kunci utama dapat terdiri dari gabungan atribut. Kunci utama yang terdiri dari dua atau lebih atribut disebut kunci komposit. Berikut adalah contoh kasus yang memanfaatkan kunci komposit:

**Tabel IV. Contoh Relasi yang Menggunakan Kunci Komposit**

Nama	Mata Pelajaran	Nilai
Calvin	Fisika	90
Bobby	Fisika	95
Calvin	Matematika	100
Bobby	Matematika	85

Pada Tabel IV tidak mungkin menggunakan satu atribut saja sebagai kunci utama sehingga kunci utamanya menjadi kunci komposit berisi gabungan atribut nama dan mata pelajaran.

## 4. Kunci Asing (*Foreign Key*)

Kunci asing adalah atribut atau gabungan atribut yang sama dengan kunci utama pada tabel atau relasi lain. Tujuan dari kunci asing adalah menghubungkan data-data pada suatu relasi atau tabel dengan relasi atau tabel lain. Contoh:

**Tabel V. Relasi ID dan Nama**

ID	Nama
13	Juan
23	Alexander
47	Hans

**Tabel VI. Relasi ID, Mata Pelajaran, dan Nilai**

ID	Mata Pelajaran	Nilai
13	Bahasa Indonesia	75
13	Biologi	80
23	Fisika	85
23	Matematika	80
47	Biologi	70
47	Fisika	90

Kunci utama pada Tabel V adalah ID, kunci utama pada Tabel VI adalah gabungan ID dengan mata pelajaran, dan kunci asing Tabel VI adalah ID yang berfungsi untuk menghubungkan Tabel V dan Tabel VI.

## 5. Operasi Basis Data Relasional

Pada sistem pengelolaan basis data relasional, kita dapat melakukan beberapa operasi terhadap basis data, beberapa di antaranya adalah:

### a. Seleksi

Operasi ini digunakan untuk memilih baris yang memenuhi syarat tertentu pada basis data.

Contoh: Pada Tabel IV dilakukan seleksi untuk baris yang atribut mata pelajarannya adalah Fisika, maka akan didapat (Calvin, Fisika, 90) dan (Bobby, Fisika, 95).

### b. Proyeksi

Proyeksi digunakan untuk mengambil satu atau lebih kolom tertentu dari sebuah tabel, jika ada kombinasi nilai yang sama maka hanya diambil satu kali saja. Contoh:

**Tabel VII. Relasi ID, Nama, dan Hobi**

ID	Nama	Hobi
13	Juan	Berenang
3	Jonathan	Bermain piano
7	Freddy	Membaca
13	Juan	Sepak bola
3	Jonathan	Memasak

Jika dilakukan operasi proyeksi untuk ID dan Nama terhadap Tabel VII maka hasilnya adalah:

**Tabel VIII. Hasil Operasi Proyeksi Terhadap Tabel VI**

ID	Nama
3	Jonathan
7	Freddy
13	Juan

### c. Join

Operasi ini menggabungkan dua buah relasi atau tabel jika memiliki atribut yang sama. Contohnya, jika kita melakukan operasi *join* terhadap Tabel V dan Tabel VI maka kita akan mendapatkan:

**Tabel IX. Hasil Operasi Join Terhadap Tabel V dan Tabel VI**

ID	Nama	Mata Pelajaran	Nilai
13	Juan	Bahasa Indonesia	75
13	Juan	Biologi	80
23	Alexander	Fisika	85
23	Alexander	Matematika	80
47	Hans	Biologi	70
47	Hans	Fisika	90

## 6. Aplikasi Lebih Lanjut

Pada dasarnya, aplikasi basis data relasional adalah untuk membuat struktur penyimpanan data yang mudah di tambah, baca, perbaharui, dan hapus. Di samping empat fungsi dasar tersebut, ada beberapa aplikasi lanjutan yang

lebih kompleks, beberapa di antaranya adalah:

a. Kontrol Konkurensi

Misalkan pada situs pemesanan tiket pesawat ada penerbangan dengan satu sisa kursi kosong dan ada dua orang yang ingin memesan tiket untuk penerbangan tersebut pada saat yang bersamaan. Jika kedua orang tersebut menekan tombol pesan pada saat bersamaan, sistem yang baik seharusnya memberikan kursi kosong kepada salah satu pemesan dan menampilkan pesan penerbangan sudah penuh kepada pemesan yang lain. Hal ini dimungkinkan oleh kontrol konkurensi. Tanpa kontrol konkurensi, sistem pemesanan tiket pesawat tersebut mungkin dapat menjumpai *error* yang dapat berakibat fatal.

b. Kontrol Transaksi

Sebuah transaksi adalah kumpulan aksi atau operasi terhadap basis data, contohnya menghapus dan menambah data. Dalam sebuah transaksi biasa, semua aksi yang termasuk dalam transaksi tersebut harus dijalankan dengan sukses, jika tidak maka basis data akan di *roll back* atau dikembalikan seperti keadaan semula sebelum transaksi. Lain halnya dengan *nested transaction* atau transaksi bersarang. Pada transaksi bersarang, setiap aksi merupakan sub-transaksi sehingga jika transaksi gagal maka hanya aksi yang tidak dapat dijalankan yang di *roll back*, aksi yang sudah berhasil dijalankan tidak terpengaruh.

c. Integritas Referensial

Integritas referensial adalah konsep dalam basis data yang memastikan relasi antar tabel tetap konsisten. Ketika sebuah tabel atau relasi memiliki kunci asing terhadap tabel lain, integritas referensial tidak memperbolehkan pengguna untuk memasukkan data tambahan ke tabel yang memiliki kunci asing tersebut kecuali ada data yang koresponden dalam tabel yang terhubung. Contohnya adalah sebagai berikut:

**Tabel X. Relasi Pegawai dan Manajer**

Pegawai	Manajer
Bobby	Calvin
Freddy	Juan
Steven	Felicia

**Tabel XI. Tabel Manajer**

ID	Manajer
1	Calvin
2	Juan
3	Felicia

Misalkan Tabel X memiliki kunci asing Manajer, jika pengguna ingin menambah data ke Tabel X maka sistem akan melakukan pengecekan terlebih dahulu terhadap Tabel XI untuk memastikan nilai atribut manajer yang ingin

ditambahkan ada atau tidak. Contohnya jika pengguna menambahkan (Alexander, Hans) ke Tabel X maka sistem akan mengembalikan *error* karena Hans tidak ada pada Tabel XI.

Jika Tabel XI diperbaharui maka semua data yang berhubungan pada Tabel X harus diperbaharui juga. Hal yang sama dilakukan untuk penghapusan, jika ada data manajer yang dihapus dalam Tabel XI maka setiap pegawai dengan manajer tersebut harus dihapus dari Tabel X. Contohnya jika pada Tabel XI (2, Juan) diganti menjadi (2, Hans) maka pada Tabel X (Freddy, Juan) akan berubah menjadi (Freddy, Hans). Jika (1, Calvin) dihapus dari Tabel XI maka pada Tabel X (Bobby, Calvin) juga akan dihapus.

7. Keuntungan Pengelolaan Basis Data Relasional

Beberapa keunggulan pengelolaan basis data relasional adalah sebagai berikut:

a. Data hanya disimpan sekali saja.

Karena adanya relasi antar tabel melalui kunci asing, data tidak perlu dimasukkan berkali-kali, cukup dimasukkan ke satu tabel saja dan tabel lain yang terhubung akan secara otomatis ter-*update*. Hal ini membuat basis data relasional tidak memakan *disk storage* yang besar dan membuat perubahan terhadap basis data mudah dilakukan.

b. Keamanan yang lebih baik.

Akses masing-masing tabel dalam basis data relasional dapat diatur sesuai dengan *status* pengguna. Sebagai contoh, seorang resepsionis dapat melihat rincian profil pegawai tetapi tidak dapat melihat gaji mereka. Contoh lain adalah seorang *salesman* dapat melihat hasil penjualan timnya tetapi tidak dapat melihat hasil penjualan tim lain.

c. Mudah dimodifikasi untuk keperluan mendatang.

Karena data disimpan dalam tabel-tabel terpisah, akan sangat mudah untuk menambah data yang sekarang belum diperlukan tetapi akan diperlukan di masa depan. Contohnya adalah menambahkan manajer pada Tabel XI. Meskipun tabel lain belum menggunakan data yang baru saja ditambahkan, hal tersebut akan memudahkan penambahan data ke Tabel X. Sebuah basis data dengan *flat file* tidak bisa melakukan hal ini.

#### IV. KESIMPULAN

Pengelolaan basis data dengan konsep relasi memegang peranan penting di zaman modern ini karena dapat memenuhi kebutuhan manusia akan pengolahan data menjadi informasi secara cepat dan tepat.

Dengan menggunakan sistem pengelolaan basis data relasional, kita dapat menghindari duplikasi data, menjaga konsistensi data dan relasi antar tabel, pengamanan yang lebih baik, dan lebih mudah untuk memperbaharui basis data. Hal-hal tersebut menjadikan sistem pengelolaan basis data relasional lebih unggul dibandingkan dengan sistem pengelolaan basis data lainnya.

#### V. UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena hanya atas berkat-Nya penulis dapat menyelesaikan makalah yang berjudul “Aplikasi Relasi dalam Pengelolaan Basis Data” ini dengan tepat waktu. Tidak lupa penulis mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, MT dan Harilili S., M.Sc. atas pengajaran dan bimbingannya selama satu semester kuliah Matematika Diskrit karena ilmu yang didapat penulis sangat membantu dalam pengerjaan makalah ini. Selain itu, penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman atas dukungan, masukan, dan doa yang telah mereka berikan.

#### DAFTAR PUSTAKA

- [1] Ambler, Scott. “Relational Databases 101: Looking at the Whole Picture”. 9 Desember 2014. <http://www.agiledata.org/essays/relationalDatabases.html>.
- [2] Anonim. “A Relational Database Overview”. 9 Desember 2014. <https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html>.
- [3] Anonim. “Advantages of A Relational Database”. 9 Desember 2014. [http://www.teach-ict.com/as\\_as\\_computing/ocr/H447/F453/3\\_3\\_9/database\\_design/miniweb/pg8.htm](http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_9/database_design/miniweb/pg8.htm).
- [4] Chapple, Mike. “Referential Integrity”. 9 Desember 2014. <http://databases.about.com/cs/administration/g/refintegrity.htm>.
- [5] Lozano, Fernando. “Introduction to Relational Database Design”. 9 Desember 2014. <http://www.edm2.com/0612/msql7.html>.
- [6] Munir, Rinaldi. “Matematika Diskrit”. Informatika, Bandung: 2010

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2014



Calvin Aditya Jonathan 13513077