

Aplikasi Relasi dan Fungsi dalam Perangkat Lunak Relational Database Management System (RDBMS)

Steven Nataniel Kodyat - 13519002¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13519002@std.stei.itb.ac.id

Abstract — Pada zaman ini, perkembangan dan keperluan akan informasi sangatlah cepat. Tidak hanya itu, manusia modern sangat membutuhkan informasi dalam jumlah besar dan juga cepat. Tanpa pengelolaan data yang baik, hal tersebut tidak dapat dicapai. Teknologi pengelolaan data yang terkanl akan hal tersebut adalah Relational Database Management System. Lantas, bagaimanakah sistem dan teknologi tersebut bekerja? Makalah ini akan membahas tentang sistem pengelolaan basis data relasional sebagai solusi mengelola data yang cepat dan banyak.

Keywords — Fungsi, Relasi, SQL, Relational Database.



Gambar 0. RDBMS

Sumber: [Geeks for Geeks](#)

I. PENDAHULUAN

Ketergantungan akan teknologi dan informasi pada zaman ini menuntut manusia untuk menciptakan sebuah sistem yang efisien untuk mengelola itu semua. Apabila terdapat inefisiensi, kehidupan manusia akan terganggu oleh hal itu. Contoh, kebanyakan manusia modern sangat bergantung pada teknologi informasi sederhana berupa alarm dan kalender. Tidak hanya itu, sebagian manusia modern juga sangat bergantung pada sistem teknologi informasi terintegrasi seperti ojek *online*, pesan makanan *online*, belanja *online*, dan sebagainya. Sadar tidak sadar, hal tersebut semua bergantung pada pengolahan data.

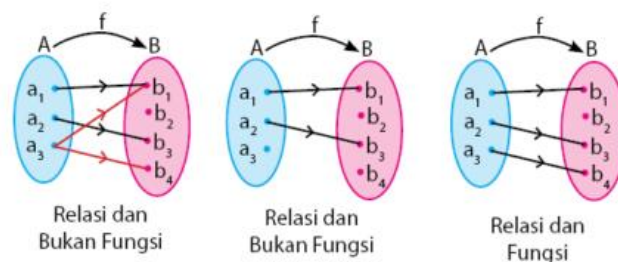
Dibalik semua yang kita gunakan, terdapat transaksi antar data. Bahkan, tingkah laku kita bisa direkam dan dijadikan sebuah data penelitian. Data-data yang didapatkan dari manusia ini diproses sehingga bisa dijadikan informasi yang berguna. Contoh, pada *maps online*, kita bisa mengetahui pola macet dan tidak, dan rute tercepat pada saat-saat tertentu. Hal tersebut juga sangat bergantung pada pemrosesan data.

Lantas, bagaimana sebuah mesin dapat menyimpan itu semua? Hampir semua data yang dimiliki oleh manusia memiliki pola relasi satu sama lain atau bahkan merupakan sebuah pemetaan setiap elemen pada elemen lainnya, dikenal sebagai fungsi. Oleh karena itu, terciptalah suatu teknologi yang

disebut *Relational Database Management System* atau RDBMS.

Relational Database Management System atau RDBMS adalah teknologi yang didesain secara spesifik untuk menyimpan data-data yang memiliki relasi. RDBMS ini juga merupakan subset dari *Database Management System*, mengingat ada cara-cara lain menyimpan data seperti, menyimpan sebagai graf, dokumen, *key-value*, dan sebagainya. Keuntungan dari RDBMS ini adalah, data yang saling berhubungan direduksi agar hanya disimpan sebagai sebuah relasi saja. Sehingga, saat ditanya dari berbagai pendekatan, data bisa didapat secara cepat. Contoh, apabila kita memiliki data berupa mata kuliah, mahasiswa yang mengikuti sebuah mata kuliah, dan nilai setiap mahasiswa pada setiap mata kuliah. Dapat dicari pendekatan seperti, ada berapa mahasiswa yang mengikuti sebuah mata kuliah, atau nilai mata kuliah tersebut, atau mungkin mahasiswa yang mendapat nilai bagus pada setiap mata kuliah. Disinilah relasi dan fungsi tersebut dipakai dalam teknologi ini.

II. RELASI DAN FUNGSI



Gambar 1. Relasi dan Fungsi

Sumber: [Relasi dan Fungsi oleh Dian Prima Trendi Siburian](#)

Relasi dapat diartikan sebagai hubungan antara sebuah elemen dengan elemen lainnya. Pada dunia matematika, hubungan ini dimaksud sebagai hubungan dari daerah asal (domain) ke daerah kawan (kodomain). Sedangkan, untuk fungsi, dapat diartikan sebagai relasi tetapi memiliki aturan khusus, yaitu semua domain harus memiliki tepat sepasangan kodomain.

Gambar 1 ter kiri bukan merupakan fungsi oleh karena elemen a_3 yang memiliki dua buah hubungan dengan kodomain. Pada gambar 1 yang tengah juga, bukan fungsi oleh sebab, a_3 yang tidak memiliki hubungan apapun pada kodomain. Terlihat dari

ketiga gambar tersebut, apabila terdapat satu saja domain berhubungan dengan kodomain, dapat disimpulkan dua himpunan tersebut berrelasi tak peduli jumlah hubungannya, namun untuk dikatakan sebuah fungsi, setiap elemen pada domain harus memiliki tepat satu pasang hubungan terhadap kodomain.

Relasi dapat direpresentasikan dalam berbagai hal, diantaranya.

1. Diagram Panah

Gambar 1 menggunakan representasi relasi dalam diagram panah.

2. Tabel

A	B
Amir	IF251
Amir	IF323
Budi	IF221
Budi	IF251
Cecep	IF323

P	Q
2	2
2	4
4	4
2	8
4	8
3	9
3	15

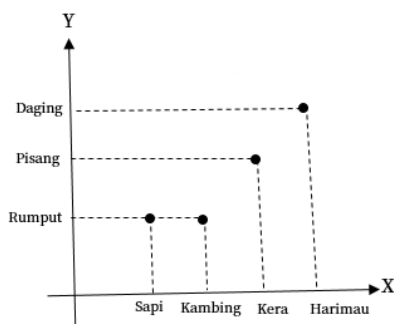
A	A
2	2
2	4
2	8
3	3
3	3

Gambar 2. Representasi Relasi dengan Tabel
 Sumber: [IF2120 Matematika Diskrit – Relasi dan Fungsi Bagian 1](#)

Kolom pertama pada tabel-tabel pada gambar 2 menyatakan daerah asal atau domain, sedangkan kolom kedua menyatakan daerah kodomain.

3. Koordinat Kartesius

Koordinat kartesius dapat dijadikan sebagai representasi relasi dengan cara meletakkan elemen domain pada sumbu X dan kodomain pada sumbu Y. Contoh dapat dilihat pada gambar 3 di bawah ini.



Gambar 3. Representasi Relasi dengan Kartesius
 Sumber: [Relasi dan Fungsi oleh Dian Prima Trendi Siburian](#)

4. Matriks

$$\begin{matrix}
 & A & B & C & D \\
 \text{Daging} & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{Pisang} & \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\
 \text{Rumput} & \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}
 \end{matrix}$$

Misalkan, A, B, C, dan D adalah Sapi, Kambing, Kera, dan Harimau secara berurut. Maka kartesius pada gambar 3 dapat diubah menjadi seperti matriks di atas. Matriks

hanya merupakan matriks biner, yang berisi 1 dan 0, 1 merepresentasikan ada relasi dan 0 sebaliknya.

5. Himpunan Pasangan Berurut

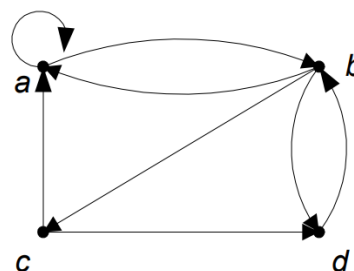
Representasi relasi juga bisa dalam bentuk himpunan, yaitu dengan membuat sepasang elemen yang saling berrelasi menjadi sebuah elemen himpunan. Contoh, sebagai berikut. Relasi pada tabel 3 pada gambar 2 dapat dinyatakan sebagai berikut.

$$R = \{(2,2), (2,4), (2,8), (3,3)\}$$

6. Graf Berarah

Terakhir, representasi relasi juga dapat dalam bentuk graf. Misalkan terdapat sebuah relasi sebagai berikut.

$$R = \{(a, a), (a, b), (b, a), (b, c), (b, d), (c, a), (c, d), (d, b)\}$$



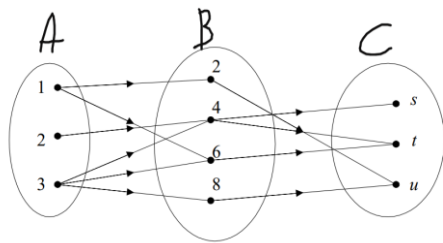
Gambar 4. Representasi Relasi dengan Graf Berarah
 Sumber: [IF2120 Matematika Diskrit – Relasi dan Fungsi Bagian 1](#)

Relasi tersebut dapat diubah menjadi graf berarah seperti pada gambar 4.

Setelah, melihat representasi relasi ada berbagai jenis, kita dapat mengambil kesimpulan bahwa berbagai data yang direpresentasikan seperti di atas dapat disebut relasi. Misalkan, kita memiliki skema pertemanan yang digambar dalam graf berarah, skema pertemanan tersebut dapat disebut sebagai data yang berrelasi. Tabel nama dan mata kuliah yang diambil seperti pada gambar 2 juga merupakan data yang memiliki relasi.

Relasi juga dapat memiliki invers, paling mudah untuk mendapatkan inversnya adalah dengan menggunakan bentuk matriks dan mentransposensya. Dengan ini, kita dapat menggunakan kodomain sebagai pusat juga. Ambil contoh tabel 1 pada gambar 2, Kita hanya bisa melihat bahwa Amir mengikuti dua mata kuliah yaitu IF251 dan IF323, dengan inversnya, dapat diambil data lain berupa jawaban dari pertanyaan “siapa saja yang mengikuti mata kuliah IF251”.

Karena relasi juga merupakan himpunan, relasi dapat melakukan gabungan, irisan, selisih, dan beda setangkup. Dengan begitu, jika kita memiliki relasi serupa, kita dapat mengolahnya untuk digabungkan, dicari perbedaannya, dan dicari persamaannya.



Gambar 5. Komposisi Relasi

Sumber: [IF2120 Matematika Diskrit – Relasi dan Fungsi Bagian 2](#)

Tidak hanya itu, karena relasi juga bersifat seperti fungsi, relasi dapat dikomposisikan. Misalkan, R adalah relasi dari himpunan A ke himpunan B , dan S adalah relasi himpunan B ke himpunan C . R dan S dapat dikomposisikan sehingga terdapat relasi A ke C . Dapat divisualisasikan dengan melihat gambar 5. Aplikasi dari komposisi fungsi ini adalah sebagai berikut, Misalkan A adalah himpunan nama mahasiswa, B adalah himpunan matakuliah, C adalah himpunan jadwal. Dengan menggunakan komposisi relasi ini, data bisa disimpan secara non-repetitif dengan cara menuliskan relasi-relasinya saja. Dan apabila ingin mencari suatu data, dengan relasi-relasi tersebut, dapat terlihat lebih jelas dan mudah untuk dicari.

Relasi-relasi yang disebutkan di atas adalah relasi biner, yaitu relasi yang hanya menghubungkan dua buah himpunan. Namun, pada kenyataannya, dibutuhkan relasi lebih dari itu. Relasi yang lebih umum menghubungkan lebih dari dua buah himpunan disebut relasi n -ary. Contoh, lihat himpunan-himpunan berikut.

$NamaDosen = \{PakRinaldi, BuHarlili, PakRila\}$
 $MataKuliah = \{Matdis, Algeo, TBFO\}$
 $HariKuliah = \{Senin, Selasa, Rabu, Kamis, Jumat\}$

Anggap ada sebuah relasi bernama JADWAL yang terdiri dari 3-tupel. Maka definisi JADWAL adalah sebagai berikut dan dapat diisi sebagai berikut.

$JADWAL \subseteq NamaDosen \times MataKuliah \times HariKuliah$
 $JADWAL =$
 $\{(PakRinaldi, Matdis, Selasa),$
 $(PakRinaldi, Matdis, Kamis),$
 $(PakRinaldi, Algeo, Senin),$
 $(PakRinaldi, Algeo, Rabu),$
 $(PakRila, TBFO, Senin),$
 $(PakRila, TBFO, Rabu),$
 $(BuHarlili, TBFO, Senin),$
 $(BuHarlili, TBFO, Rabu),$
 $(BuHarlili, Matdis, Selasa),$
 $(BuHarlili, Matdis, Kamis)\}$

Seperti yang disebutkan sebelumnya, bahwa fungsi adalah relasi yang setiap domainnya berpasangan dengan tepat satu kodomain. Jika f adalah fungsi dari himpunan A ke himpunan B , dapat dikatakan f memetakan A ke B . Nama lain dari fungsi ialah pemetaan atau transformasi. A disebut juga pra-bayangan dari B , dan B disebut bayangan dari A . Fungsi juga memiliki beberapa jenis, sebagai berikut.

1. *One-to-one* atau Injektif
Jika tidak ada dua elemen domain yang memiliki bayangan sama.
2. *Onto* atau Surjektif
Jika setiap elemen kodomain merupakan bayangan dari satu atau lebih elemen domain
3. *Bijection* atau Bijeksi
Jika sebuah fungsi berkoresponden satu-ke-satu atau merupakan sebuah fungsi surjektif dan injektif.

Aplikasinya fungsi ini dapat digunakan untuk mendeskripsikan antar data, misalkan injektif dapat digunakan untuk membuat suatu kolom memiliki elemen-elemen yang unik, atau tidak ada elemen yang sama dalam satu kolom. Sedangkan Surjektif, bisa digunakan untuk mencari variasi nilai pada suatu kolom.

III. BASIS DATA RELASIONAL

A. Basis Data

Basis Data terdiri dari dua kata, yaitu basis dan data, basis sendiri dimaknai sebagai gudang atau markas, dan data adalah catatan atau kumpulan fakta yang mewakili suatu objek yang dapat diwujudkan dalam bentuk huruf, angka, gambar, simbol, teks, bunyi, atau kombinasinya. Secara matematis, basis data dapat didefinisikan sebagai himpunan kelompok data yang saling terhubung dan diorganisasi sedemikian rupa sehingga dapat dimanfaatkan kembali secara cepat dan mudah. Secara umum, teknologi basis data memiliki istilah create, read, update, dan delete. Terdapat juga istilah *database* dan *document/table*. Suatu database terdiri dari satu atau banyak *document* atau *table*, tergantung basisdata ini disimpan secara tabel atau dokumen.

B. Basis Data Relasional

Pada Makalah ini, basis data yang akan dibahas adalah basis data berkelompok *SQL* atau *structured query language*. Basis data ini memberikan gambaran atau bagan skema yang menjelaskan tentang hubungan antar tabel. Sebelum basis data jenis ini ada, terdapat pendahulunya yaitu *Hierarchical Database Model* dan *Network Database Model*. Jenis basis data ini mengubah tren pemakaian basis data di zaman ini. Kedua database pendahulu tersebut, sudah jarang digunakan sekarang karena adanya berbagai kelemahan dan fungsionalitas yang tidak memenuhi spesifikasi dan kebutuhan aplikasi modern saat ini.

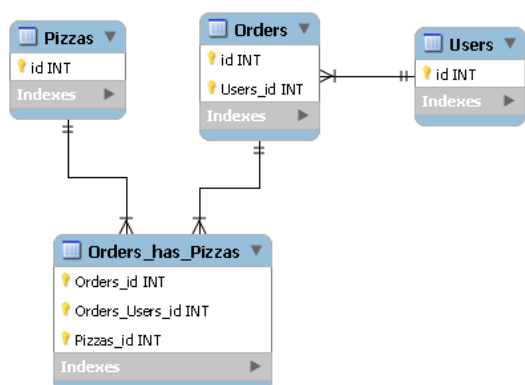
Dalam basis data relasional, setiap data memiliki sebuah kolom kunci atau biasa dikenal sebagai *primary key*. *Primary key* ini dapat diletakkan pada tabel lain agar menjadi *foreign key*. Dan dengan meletakkan sebuah *foreign key* maka relasi tersebut tercipta.

C. Relational Database Management System

Basis data atau *database* adalah sekumpulan tabel yang disimpan ke dalam bentuk elektronik dan dikelompokkan berdasarkan skema yang sudah dibuat. Untuk melakukan pembuatan struktur, pengisian, pengubahan, dan penghapusan suatu database diperlukan sebuah perangkat lunak. *Relational Database Management System* adalah perangkat lunak yang menerima *command* berupa *structured query language*.

Perangkat lunak inilah yang menggunakan konsep *basis data relasional* menjadi aplikatif.

D. Istilah pada Relational Database Management System



Gambar 6. Skema pada RDBMS
Sumber: [Stackoverflow](#)

Pertama, terdapat istilah **Entitas**, objek yang direkam sebagai data. Entitas ini merujuk kepada objek yang bersangkutan. Contoh, apabila ingin membuat *database* perpustakaan maka entitasnya adalah buku, pengarang, dan sebagainya.

Kedua, istilah **field** atau **attribut**, setiap entitas memiliki sebuah atribut yang mewakili isi datanya. Contoh, entitas buku memiliki judul dan jumlah halaman. Judul dapat diwakilkan oleh *string* dan jumlah halaman oleh bilangan bulat.

Ketiga, istilah **record** atau **baris**, kumpulan satu data konkrit, satuan informasi database. Baris ini juga dapat didefinisikan secara literal, yaitu baris, karena basis data relasional ini biasanya menggunakan *view* tabel. Baris ini terdiri dari sekumpulan field yang sudah konkrit.

Keempat, istilah **kunci utama** atau **primary keys**, setiap tabel memiliki kunci utama dan bersifat unik untuk setiap barisnya, digunakan untuk mencari suatu baris.

Kelima, istilah **kunci asing** atau **foreign keys**, digunakan untuk menghubungkan dua buah tabel atau lebih, kunci asing berisi nilai yang sama dengan kunci utama pada tabel lain, digunakan untuk merujuk ke tabel lain.

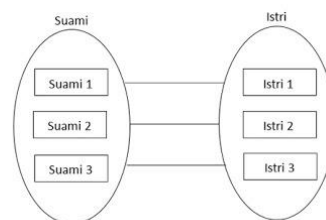
Terakhir, istilah **integritas data**, setiap data yang dibuat harus dikonfirmasi apakah sesuai atau tidak. Integritas data tersebut terbagi lagi menjadi beberapa, yaitu, *entity integrity*, *domain integrity*, dan *referential integrity*. *Entity Integrity* mengonfirmasi bahwa baris tidak ganda dalam satu tabel. *Domain Integrity* mengonfirmasi tipe data dan kondisi *blueprint database* (seperti rentang nilai atau panjang maksimum). Dan *referential integrity* memastikan bahwa setiap baris merujuk kepada tabel yang berbeda dan tidak dapat dihapus. Sehingga, setiap rujukan selalu punya nilai, tidak ada rujukan ke tempat kosong. Misalkan A merujuk kepada B, pada 3 baris pertama di tabel A merujuk pada baris pertama tabel B. Untuk menghapus baris pertama tabel B, perlu dilakukan penghapusan semua data yang merujuk pada baris pertama tabel B, pada kondisi ini, yaitu 3 baris pertama pada tabel A.

E. Jenis Relasi Entitas Basis Data Relasional

Secara umum, dengan menggunakan konsep relasi yang ada, terdapat tiga buah jenis relasi entitas basis data, yaitu:

1. One to One

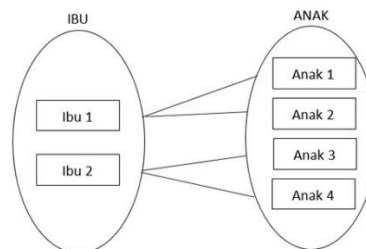
Relasi basis data ini terjadi apabila sebuah data pada dua tabel saling memiliki satu data pada masing-masing tabel. Misalkan terdapat sebuah tabel data mahasiswa yang memuat data-data pribadi mahasiswa, dan juga sebuah tabel yang memuat data nilai kuis, tugas, dan ujian mata kuliah matematika diskrit. Maka terjadilah relasi *one-to-one*, karena satu mahasiswa hanya memiliki sebuah baris pada tabel nilai matematika diskrit. Seperti gambar dibawah ini, bisa juga diibaratkan sebagai suami dan istri. Suami memiliki seorang istri, dan istri memiliki seorang suami.



Gambar 7. Konsep One to One
Sumber: [Model Data ERD – mfikri.com](#)

2. One to Many

Relasi basis data ini memperbolehkan salah satu tabel memiliki relasi ganda terhadap satu tabel. Contoh, sebuah tabel berisi data-data pribadi mahasiswa, dan satu lagi tabel yang berisi informasi makanan di kantin. Bisa dibuat relasi tabel mahasiswa ke tabel makanan di kantin sebagai makanan favorit. Jika mahasiswa hanya diperbolehkan memilih satu makanan favorit, maka terjadilah relasi *one-to-many* karena bisa saja terdapat mahasiswa yang memiliki makanan favorit yang sama. Disini *onenya* adalah makanan di kantin, dan *manynya* adalah mahasiswa. Seperti gambar dibawah ini, bisa juga diilustrasikan sebagai ibu dan anak, seorang ibu memiliki banyak anak, namun seorang anak hanya memiliki seorang ibu.

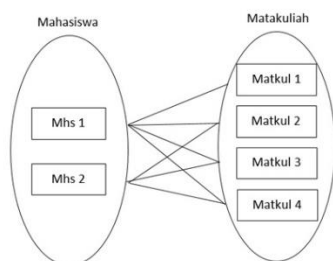


Gambar 8. Konsep One to Many
Sumber: [Model Data ERD – mfikri.com](#)

3. Many to Many

Sama seperti *one-to-many*, hanya saja, semuanya boleh berganda. Sebagai contoh, asumsikan contoh pada *one-to-many* diperbolehkan memilih lebih dari 1 makanan. Sehingga, terdapat *many* mahasiswa dan *many* makanan.

Dapat diilustrasikan seperti gambar dibawah ini, dimana seorang mahasiswa bisa memiliki banyak mata kuliah, dan mata kuliah bisa memiliki banyak mahasiswa.



Gambar 9. Konsep Many to Many
Sumber: [Model Data ERD – mfikri.com](http://Model Data ERD - mfikri.com)

F. Kelebihan dan Kekurangan RDBMS

Perangkat lunak *Relational Database Management System* menggunakan konsep relasi dan fungsi untuk mengantisipasi hal-hal seperti redundansi data. Sehingga, data-data yang berulang apalagi ukurannya yang besar dapat dicekikan menggunakan konsep normalisasi. Basis data relasional itu sendiri sudah memiliki keuntungan dengan menyimpan data yang terstruktur dalam tabel yang menggunakan baris dan kolom. Tidak hanya itu, integritas data pada RDBMS sangat dapat dipercaya konsistensinya. Sehingga, data-data yang disimpan akan sangat terstruktur, rapih, dan konsisten. Mengontrol, memperbaiki dan memelihara basis data yang menggunakan RDBMS juga relatif mudah. Secara singkat, RDBMS ini dapat diandalkan dan diuji, tentunya cocok untuk struktur basis data yang kompleks namun tetap terstruktur.

Untuk aplikasi lebih lanjut, terdapat juga keuntungan apabila memakai RDBMS. Kontrol konkurensi pada perangkat lunak RDBMS sangat bisa diandalkan. Kontrol konkurensi adalah pengaturan dari eksekusi transaksi yang berkelanjutan dalam sistem basis data *multiuser*. Misalkan, terdapat sebuah aplikasi penjual tiket pesawat atau kereta, dan ada dua pengguna yang membeli sebuah kursi di saat yang bersamaan, tanpa kontrol konkurensi yang baik, akan terjadi *error*, namun, RDBMS ini akan memberikan ke salah satu pemesan.

Adapun juga kekurangan RDBMS ini, yaitu, kurang mampunya menyimpan data besar yang tidak terstruktur seperti *big data* atau *cloud*, skema data juga bersifat statis dan perlu diperbaiki jika ingin mengubah skema, dan membutuhkan skema tertentu jika diterapkan basis data terdistribusi.

G. Teknologi yang Menggunakan RDBMS

Terdapat beberapa teknologi populer yang sering dipakai oleh aplikasi-aplikasi modern, diantaranya sebagai berikut.



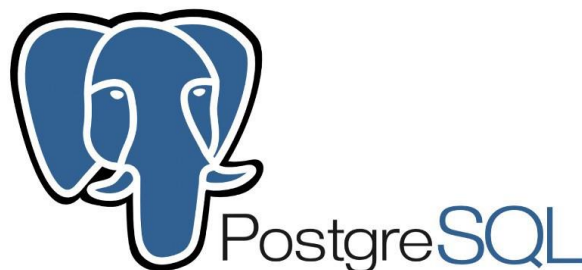
Gambar 10. SQLite
Sumber Logo: Wikipedia

Pertama, **SQLite**, basis data *SQL* open source yang populer. Teknologi ini dapat menyimpan seluruh basis data dalam sebuah file. Teknologi ini sering digunakan untuk penyimpanan basis data lokal, tanpa terhubung dengan sebuah server.



Gambar 11. MySQL
Sumber: dnhwebdesign.com

Kedua, **MySQL**, sama seperti SQLite, namun lebih sering digunakan untuk pengembangan aplikasi web. Teknologi ini sering dipertimbangkan karena kemudahannya, cukup terkenal dari tahun 1995, dan memiliki komunitas yang besar. Sayangnya, teknologi ini pada akhir-akhir ini memiliki kinerja yang buruk saat melakukan *scaling up*.



Gambar 12. PostgreSQL
Sumber: Medium

Ketiga, **PostgreSQL**, merupakan teknologi yang tidak jauh beda dengan MySQL, dan unggul dalam *query language* yang kompleks. PostgreSQL ini merupakan database objek relasional dan sangat bagus dalam hal konkurensi. Sayangnya, teknologi ini dibangun dengan mengorbankan kecepatan untuk ekstensibilitas, kepatuhan standar, skalabilitas dan integritas data yang bagus.

H. Operasi Basis Data Relasional

Adapun cara mengontrol basis data dengan menggunakan operasi-operasinya, diantaranya sebagai berikut.

- Pembuatan basis data baru,
- Penghapusan basis data,
- Pembuatan tabel baru,
- Penghapusan tabel,
- Penambahan data baru,
- Pengambilan data menggunakan *query*,
- Perubahan data, dan
- Penghapusan data.

Pada query, konsep matematika diskrit dipakai. Yaitu dengan menggunakan seleksi, proyeksi dan *join* atau penggabungan. Seleksi akan dipakai saat ingin memilih baris yang memenuhi syarat tertentu pada basis data. Proyeksi digunakan untuk

mengambil satu atau lebih kolom tertentu dari suatu tabel. Dan terakhir, *Join* dipakai untuk menggabungkan dua buah tabel atau lebih saat mereka memiliki relasi.

1. Contoh Basis Data Relasional Sederhana serta Hubungannya dengan Relasi dan Fungsi

Misalkan disediakan sebuah basis data yang terdiri dari tiga buah tabel sebagai berikut.

ID	NAMA	NIM	KELOMPOK
1	Viel	13519002	1
2	Kinan	13519044	1
3	Jason	13519019	2
4	Acin	13519118	3
5	Ryan	13519191	1
6	Aldo	13519015	2
7	Wilbert	13519025	3
8	Bije	13519094	4
9	Top	13519074	4
10	Karina	13519166	4

Tabel 4. Tabel Mahasiswa

ID	NAMA KELOMPOK	MATKUL
1	Izo osl orz	1
2	Search Engine Wibu	1
3	manasempatkeburutelat	2
4	apaya	3

Tabel 5. Tabel Kelompok

ID	MATKUL	SKS	DEADLINE TUBES
1	Algeo	3	Besok
2	Logkom	2	Lusa
3	Alstrukdat	4	Minggu Depan
4	Orkom	3	Tidak ada tubes
5	Matdis	3	Tidak ada tubes
6	TBFO	3	Tidak ada tubes

Tabel 6. Tabel Matkul Semester 3

Tinjau tabel 4, 5, dan 6. Diketahui kolom kelompok dan matkul adalah *foreign key* ke tabel sesuai namanya. Dengan menggunakan relasi, dapat diketahui bahwa mahasiswa ber-id 1 memiliki hubungan dengan kelompok ber-id 1. Dan kelompok ber-id 1 memiliki hubungan dengan matkul ber-id 1. Dengan menggunakan basis data relasional ini, data duplikat hilang karena diganti representasinya menggunakan *foreign key*. Tidak hanya itu, memroses data juga jadi lebih mudah dan cepat, sebab, jika semua dijadikan satu tabel, sulit untuk melihat semuanya secara mudah. Misalkan, ingin mencari nama kelompok pada matkul algeo, bisa dapat dicari dengan cepat dengan langsung mencari matkul ber-id 1 pada tabel kelompok. Pemrosesan data yang tidak berhubungan dengan apa yang dicari tidak menjadi masalah karena antar tabel memiliki fungsinya masing-masing.

Penggunaan fungsi disini digunakan saat ingin mencari informasi seperti “berapa jumlah SKS pada semester 3?”. Akan digunakan fungsi sum untuk menghitung sks yang ada pada

tabel tersebut. Atau bisa juga “berapa mahasiswa yang memiliki kelompok? (Asumsi tidak ada ditabel, tidak berkelompok)” maka akan digunakan fungsi count pada tabel mahasiswa.

Konsep operator juga secara tidak langsung digunakan disini, misalkan menggunakan operator *join* untuk menggabungkan tabel, bila dibutuhkan informasi seperti “nim mana saja yang berkelompok pada matkul alstrukdat?”. Harusnya tabel tersebut proyeksi dulu, ambil kolom yang diperlukan, lalu digabungkan dengan operator *join*, lalu diseleksi.

IV. KESIMPULAN

Penerapan konsep relasi dan fungsi pada pengelolaan basis data zaman modern ini sangat berperan penting untuk memberikan kecepatan akses informasi, konsistensi pengolahan data, dan kemudahan dalam pengolahan informasi. Tanpa disadari, penggunaan relasi dan fungsi ini sangat aplikatif pada kehidupan sehari-hari. Dengan menggunakan sistem pengolahan basis data relasional atau RDBMS, dapat dihindari hal-hal seperti duplikasi data, data tak konsisten dan tak *reliable*, dan data-data *redundant*. Tidak hanya itu, dengan menerapkan konsep relasi ini juga, aplikasi zaman modern ini dapat berkembang pesat karena kemudahannya dalam mengubah, membuat, menghapus, dan membaca data.

Tanpa teknologi ini, manusia zaman modern ini pastinya akan kesulitan dalam menyimpan data yang kompleks dan besar yang tentunya akan memperlambat perkembangan zaman.

V. UCAPAN TERIMA KASIH

Puji Syukur kepada Tuhan Yang Maha Esa karena berkat dan bimbingan-Nya, makalah ini dapat terselesaikan dengan tepat waktu dan tersusun dengan baik. Penulis mengucapkan terima kasih kepada teman-teman dan orang tua yang mendukung dalam studi dan proses pembelajaran. Tidak lupa rasa hormat dan terima kasih kepada empat dosen pengampu IF2120 Matematika Diskrit yaitu, Pak Rinaldi, Ibu Harlili, Ibu Fariska dan Ibu Ulfa, yang sudah memberikan ilmu selama satu semester.

VI. REFERENSI

- [1] [Munir, Rinaldi. Relasi dan Fungsi Bagian 1. Informatika, Bandung: 2020. Diakses pada 8 Desember 2020.](#)
- [2] [Munir, Rinaldi. Relasi dan Fungsi Bagian 2. Informatika, Bandung: 2020 Diakses pada 8 Desember 2020.](#)
- [3] [Munir, Rinaldi. Relasi dan Fungsi Bagian 3. Informatika, Bandung: 2020 Diakses pada 8 Desember 2020.](#)
- [4] [AWS, Amazon. Relational Database. Diakses pada 6 Desember 2020.](#)
- [5] [Dian P. T. Siburian. Relasi dan Fungsi. Diakses pada 8 Desember 2020](#)
- [6] [Tech Terms. RDBMS. Diakses pada 8 Desember 2020.](#)
- [7] [Mulyawan, Rifqi. RDBMS. Diakses pada 9 Desember 2020.](#)
- [8] [Sutiono. Database Relasional. Diakses pada 9 Desember 2020.](#)
- [9] [Saputra, Yolanda. RDBMS. Diakses pada 9 Desember 2020.](#)
- [10] [Jonathan, Calvin Aditya. 2013. Aplikasi Relasi dalam Pengelolaan Basis Data.](#)

VII. 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, 3 Desember 2020

Tertanda



Steven Nataniel
13519002