

# Tugas Basis Data Sesi 5

Aldi Maulana Iqbal – 2021080122

# Relational Database Model

## Pengertian

Struktur data dalam model relasional menggunakan konsep tabel yang terdiri dari baris dan kolom. Kolom mewakili atribut dari data, sementara baris mewakili entitas yang mewakili data. Setiap tabel memiliki kunci primer yang digunakan untuk mengidentifikasi baris unik dalam tabel. Ini memungkinkan untuk menghubungkan tabel satu dengan yang lain melalui relasi.

Integritas data dalam model relasional dijamin melalui konsep normalisasi. Normalisasi adalah proses yang digunakan untuk mengekstrak data yang sama dari tabel dan menyimpannya dalam tabel yang berbeda. Ini memastikan bahwa data tidak ada duplikasi dan mudah untuk diperbarui.

Manipulasi data dalam model relasional dilakukan melalui perintah SQL (Structured Query Language). SQL digunakan untuk mengambil, menambahkan, memperbarui, dan menghapus data dari tabel. Ini juga digunakan untuk mengambil data dari beberapa tabel dan menyatukannya menjadi satu.

Secara keseluruhan, model relasional adalah cara yang efektif untuk menyimpan dan mengakses data dalam database. Ini memungkinkan untuk membuat koneksi antara tabel dan menjamin integritas data melalui normalisasi. SQL digunakan untuk mengakses dan memanipulasi data dengan mudah.

## Definisi Relasi

Secara sederhana, relasi adalah hubungan antara dua atau lebih himpunan. Dalam model relasional, relasi ditunjukkan sebagai tabel yang terdiri dari baris dan kolom. Setiap baris mewakili entitas yang memiliki hubungan dengan entitas lain dalam tabel.

Dalam konteks matematika, relasi dapat diartikan sebagai himpunan  $n$ -tupel terurut yang terdiri dari  $n$  elemen dari himpunan domain yang berbeda. Derajat relasi adalah jumlah himpunan domain yang digunakan dalam relasi.

Contohnya, dalam database siswa, relasi antara tabel "siswa" dan "kelas" adalah bahwa setiap siswa terdaftar dalam kelas tertentu. Domain dari relasi ini adalah himpunan "siswa" dan himpunan "kelas", dan derajat relasi adalah 2.

Secara umum, relasi membantu untuk menghubungkan dan mengkaitkan data dari berbagai tabel dalam database relasional. Ini memungkinkan untuk mengambil data dari beberapa tabel sekaligus dan menggabungkannya menjadi satu untuk analisis yang lebih baik.

## Struktur Data Database Relasional

Model relasional menggunakan tabel sederhana sebagai struktur data utama untuk menyimpan dan mengakses data dalam database. Ini berbeda dari model jaringan dan pohon yang lebih rumit dan kompleks.

Dalam model relasional, tabel adalah kumpulan data yang disimpan dalam bentuk matriks dengan baris dan kolom. Baris mewakili entitas atau data individu, sementara kolom mewakili atribut atau karakteristik dari entitas tersebut.

Berikut ini adalah struktur data dasar yang digunakan dalam basis data relasional.

1. Relation adalah kumpulan data yang disimpan dalam tabel dalam model relasional. Setiap relasi mewakili entitas atau objek yang memiliki hubungan dengan entitas lain dalam tabel. Dalam E-R model, ini disebut sebagai "entity set".
2. Tuple adalah baris dalam tabel yang mewakili entitas unik dalam relasi. Setiap tuple menyimpan data tentang entitas yang sama. Dalam E-R model, ini disebut sebagai "entity".
3. Attributes adalah kolom dalam tabel yang mewakili atribut atau karakteristik dari entitas yang diwakili oleh tuple. Setiap atribut memiliki domain yang menentukan jenis data yang diizinkan. Dalam E-R model, ini sama dengan atribut.
4. Domain adalah himpunan data yang diizinkan untuk digunakan dalam atribut. Ini dapat mencakup tipe data seperti angka, teks, atau tanggal. Dalam E-R model, ini sama dengan domain.
5. Tuple variable adalah nilai yang dapat diasosiasikan dengan atribut dalam tuple. Ini dapat digunakan dalam operasi matematika atau logika. Dalam E-R model, ini disebut sebagai "value".
6. Degree adalah jumlah atribut dalam relasi. Ini menentukan derajat dari relasi.
7. Cardinality adalah jumlah tuple dalam relasi. Ini menentukan jumlah entitas yang diwakili oleh relasi.

**Jenis Atribut:** Jenis atribut yang dibahas dalam model ER juga berlaku di sini.

1. Keys adalah atribut yang digunakan untuk mengidentifikasi tuple unik dalam relasi. Ada dua jenis kunci yaitu kunci primer dan kunci asing. Kunci primer digunakan untuk mengidentifikasi tuple unik dalam tabel, sementara kunci asing digunakan untuk menghubungkan tabel satu dengan yang lain melalui relasi.

2. Database instance adalah istilah yang digunakan untuk menggambarkan data yang disimpan dalam database pada saat ini. Ini mencakup semua data yang ada dalam tabel-tabel relasi.
3. Relation schema adalah struktur tabel yang digunakan untuk menyimpan data dalam relasi. Ini mencakup nama tabel, nama atribut, tipe data, dan kunci primer.

Relation instance adalah istilah yang digunakan untuk menggambarkan data yang disimpan dalam tabel pada saat tertentu. Ini mencakup semua data yang ada dalam tabel, termasuk tuple dan nilai atribut.

## Integrity Constraints

Integrity Constraints atau batasan integritas adalah aturan atau pengaturan yang diterapkan pada database untuk memastikan bahwa data yang disimpan dalam database tetap stabil, akurat, dan konsisten. Ini dapat berupa aturan seperti kunci primer yang digunakan untuk mengidentifikasi tuple unik dalam tabel, atau batasan referensial yang digunakan untuk menghindari data yang hilang atau tidak konsisten saat menghubungkan tabel satu dengan yang lain. Ini memastikan bahwa data yang disimpan dalam database tetap benar dan dapat diandalkan.

1. Entity Integrity Rule (Integrity Rule 1) adalah aturan yang menyatakan bahwa kunci primer atau bagian darinya dalam relasi apa pun tidak boleh kosong atau nol. Ini berarti bahwa kunci primer harus memiliki nilai yang unik dan tidak boleh kosong. Hal ini dilakukan untuk menjamin bahwa setiap tuple dalam relasi dapat diidentifikasi dengan benar dan tidak ada duplikat.
2. Referential Integrity Rule (Integrity Rule 2) adalah aturan yang digunakan untuk menjamin integritas data saat menghubungkan tabel satu dengan yang lain melalui kunci asing. Ini memastikan bahwa setiap nilai kunci asing harus memiliki nilai yang sesuai dengan kunci utama yang terkait dengannya. Misalnya, jika atribut A adalah kunci utama dalam relasi R1 dan juga kunci asing dalam relasi R2, maka nilai A dalam relasi R2 harus sama dengan nilai A dalam relasi R1 atau null. Ini memastikan bahwa tidak ada data yang hilang atau tidak konsisten saat menghubungkan tabel satu dengan yang lain.
3. Domain Constraints adalah pembatasan yang diterapkan pada setiap nilai atribut dalam database. Pembatasan ini diterapkan pada domain atribut, yang menentukan jenis data yang diizinkan dalam atribut. Pembatasan ini termasuk tipe data, ukuran variabel, dan pemeriksaan yang digunakan untuk memastikan bahwa data yang dimasukkan dalam atribut sesuai dengan kriteria yang ditentukan. Ini bertujuan untuk menjaga konsistensi dalam database dan memastikan bahwa data yang disimpan dalam database valid dan dapat diandalkan.

4. Key Constraints adalah aturan yang diterapkan pada relasi untuk memastikan bahwa atribut primary key memiliki nilai unik. Atribut primary key digunakan untuk mengidentifikasi tuple unik dalam tabel. Oleh karena itu, tidak diperbolehkan adanya nilai duplikat dalam atribut primary key. Hal ini membuat data dalam tabel tetap konsisten dan dapat diandalkan. Ini juga membantu dalam proses pembuatan relasi antar tabel.
5. Tuple Uniqueness Constraints adalah batasan yang digunakan untuk memastikan bahwa tidak ada tuple duplikat dalam relasi. Ini berarti bahwa setiap tuple dalam relasi harus memiliki nilai yang unik dan tidak boleh ada dua tuple yang sama dalam relasi. Ini memastikan bahwa data yang disimpan dalam relasi valid dan konsisten, dan memudahkan proses analisis dan manipulasi data. Tuple Uniqueness Constraints dapat diterapkan dengan menggunakan kunci primer yang digunakan untuk mengidentifikasi tuple unik dalam relasi.

### CODD's Rules

Dr. Edgar F. Codd menyarankan seperangkat aturan yang diperlukan untuk sistem untuk memenuhi syarat sebagai Sistem Manajemen Database Relasional. Aturan Codd, atau "Codd's Rules", terdiri dari 12 aturan yang digunakan untuk menentukan apakah suatu sistem dapat dianggap sebagai sistem manajemen database relasional yang baik. Aturan Codd adalah sebagai berikut:

No	Rule Number	Rule Name	Description
1	Rule 0	Foundation rule	Sebuah sistem manajemen basis data relasional harus mengelola basis data secara keseluruhan melalui kemampuannya yang relasional.
2	Rule 1	Information rule	Semua data dalam basis data harus diwakili secara logis oleh nilai-nilai dalam tabel.
3	Rule 2	Guaranteed access rule	Semua data dalam basis data diakses secara logis menggunakan kombinasi nama tabel, nama kolom, dan nilai kunci utama tanpa ambiguitas.
4	Rule 3	Systematic treatment of missing information	Basis data harus mendukung representasi nilai NULL atau informasi yang hilang dan informasi yang tidak sesuai. Representasi nilai NULL harus berbeda dari semua nilai reguler dan independen dari tipe data.
5	Rule 4	Active online catalog rule	Pengguna harus dapat mengakses basis data berdasarkan struktur relasional menggunakan bahasa query yang sama yang digunakan model atau sistem untuk mengakses data basis data.

6	Rule 5	Comprehensive data sub-language rule	Dari semua bahasa yang didukung oleh RDBMS, harus ada setidaknya satu bahasa yang mendukung semua hal berikut: mendefinisikan tabel dan tampilan, mengambil dan memperbarui data, menetapkan kendala integritas, menetapkan otorisasi dan mendefinisikan kendala.
7	Rule 6	View updating rule	Secara teoritis, pembaruan dalam basis data harus dapat diperbarui oleh sistem.
8	Rule 7	Set level update rule	Sistem harus mendukung operasi insert, update, dan atau insert tingkat tinggi, delete pada tingkat tabel. Hal ini menangani update tabel dan delete sebagai objek tunggal. Hal ini meningkatkan performa karena perintah beraksi pada sekumpulan catatan bukan satu catatan pada satu waktu.
9	Rule 8	Physical data independence rule	Data, program aplikasi, operasi pengguna harus independen dari perubahan apa pun dalam penyimpanan fisik atau metode akses.
10	Rule 9	Logical data independence rule	Operasi pengguna dan program aplikasi harus independen dari perubahan apa pun dalam tabel dan tampilan seperti menambahkan/menghapus kolom, mengubah panjang bidang dll.
11	Rule 10	Integrity independence rule	Kendala integritas untuk basis data harus ditentukan secara terpisah dan disimpan dalam basis data. Mereka harus dapat diubah tanpa mempengaruhi aplikasi. Mereka harus diterapkan oleh sistem sendiri.
12	Rule 11	Distribution independence rule	Program aplikasi dan operasi pengguna harus independen dari lokasi data dan perubahan (hal ini berlaku untuk distribusi data fisik. Lokasi basis data yang didistribusikan harus tidak terlihat oleh pengguna sistem).
13	Rule 12	Non-subversion rule	Jika sistem menyediakan antarmuka tingkat rendah untuk mengakses data, maka sistem tidak boleh dapat merusak kendala integritas sistem. (Merusak berarti menghancurkan).

## Operasi pada Model Relasional

Operasi dasar pada model data relasional adalah insertion, deletion, updation, dan retrieval.

1. **Insertion** digunakan untuk menambahkan record baru ke dalam tabel, tetapi terdapat beberapa kondisi yang harus diperhatikan seperti:
  - a. Jika kita mencoba untuk memasukkan nilai duplikat untuk atribut yang dipilih sebagai primary key, maka akan terjadi kesalahan karena primary key harus unik.
  - b. Jika kita mencoba untuk memasukkan nilai NULL di atribut kunci utama, maka akan terjadi kesalahan karena primary key harus memiliki nilai.
  - c. Jika kita mencoba untuk memasukkan nilai data dalam atribut kunci asing yang tidak ada dalam atribut kunci primer yang sesuai, maka akan terjadi kesalahan karena referensi harus valid.
  - d. Jika atribut diberi nilai yang tidak muncul di domain yang sesuai, maka akan terjadi kesalahan karena nilai harus sesuai dengan domain yang telah ditentukan.
2. **Deletion** digunakan untuk menghapus record dari tabel.
3. **Updation** digunakan untuk memperbarui nilai data dari suatu record dalam tabel, tetapi harus memperhatikan kondisi seperti:
  - a. Nilai yang dimodifikasi tidak memiliki nilai kunci asing yang sesuai, sehingga harus diperiksa apakah nilai yang diubah tersebut sesuai dengan referensi yang ada.
  - b. Nilai baru seharusnya belum ada di tabel, sehingga harus diperiksa apakah nilai yang diubah tersebut unik.
4. **Retrieval** digunakan untuk mencari dan mengambil record dari tabel.

## Keunggulan Relational Model

Model Relasional memiliki beberapa keuntungan utama, di antaranya:

1. **Simplicity**: Model basis data relasional sangat mudah dan sederhana untuk dirancang dan diimplementasikan pada tingkat logika. Tabel-tabel yang berbeda dalam basis data dapat dengan mudah dirancang menggunakan atribut dan nilai data yang sesuai. Semua hubungan dirancang dalam bentuk tabel, yang membantu pengguna untuk fokus pada tampilan logika basis data daripada detail internal yang rumit tentang bagaimana data disimpan.
2. **Flexible**: Basis data relasional memberikan fleksibilitas yang memungkinkan perubahan dalam struktur basis data dengan mudah diterima.
3. **Data Independence**: Karena data berada dalam tabel, struktur basis data dapat diubah tanpa harus mengubah aplikasi yang didasarkan pada struktur tersebut. Jika menggunakan basis data non relasional, mungkin harus memodifikasi aplikasi yang akan mengakses informasi ini dengan

menyertakan pointer ke data baru. Namun dengan basis data relasional, informasi segera dapat diakses karena secara otomatis terkait dengan data lain karena posisinya dalam tabel.

4. **Structural Independence:** Basis data relasional hanya mengenai data dan bukan struktur, yang meningkatkan performa. Oleh karena itu waktu pemrosesan dan ruang penyimpanan lebih besar dalam basis data relasional tetapi pengguna tidak perlu tahu detail desain struktur. Fleksibilitas struktur basis data relasional memungkinkan kombinasi data untuk diambil yang tidak pernah diantisipasi pada saat basis data awal dibuat.
5. **Query Capability:** Ini memungkinkan bahasa query tingkat tinggi yaitu SQL (Structure Query Language) yang menghindari navigasi basis data yang rumit. Dalam model ini, query didasarkan pada hubungan logika dan pemrosesan query tersebut tidak memerlukan jalur akses yang ditentukan sebelumnya di antara data yaitu pointer.
6. **Matured Technology:** Model relasional berguna untuk mewakili sebagian besar objek dunia nyata dan hubungan antar objek. Implementasi hubungan sangat mudah melalui penggunaan kunci.
7. Kemampuan untuk dengan mudah memanfaatkan teknologi perangkat keras baru yang membuat hal-hal lebih mudah bagi pengguna.

## Kelemahan Relational Model

Kekurangan utama dari model basis data relasional adalah sebagai berikut:

1. **Kinerja yang terbatas:** Model basis data relasional memiliki kinerja yang terbatas ketika menangani jumlah data yang besar dan kompleks. Ini dapat menyebabkan masalah performa pada sistem yang menggunakan basis data relasional besar.
2. **Keterbatasan dalam menangani objek besar biner:** Model basis data relasional kadang-kadang kesulitan dalam menangani objek besar seperti gambar atau video, karena relasi tidak selalu cocok untuk menyimpan jenis data ini.
3. **Biaya overhead perangkat keras yang tinggi:** Model basis data relasional memerlukan perangkat keras yang kuat dan mahal untuk menyimpan dan mengelola data yang besar.
4. **Kesulitan dalam memetakan objek ke database relasional:** Model basis data relasional menggunakan skema tabel yang terstruktur, yang dapat menyulitkan dalam memetakan objek-objek dari aplikasi ke dalam tabel yang sesuai.



5. Integritas data yang sulit dipastikan: Model basis data relasional memiliki keterbatasan dalam menjamin integritas data karena ketergantungan pada relasi antar tabel.

## Sistem Basis Data Relasional yang Tersedia Secara Komersial

Beberapa contoh sistem basis data relasional yang tersedia secara komersial adalah Oracle, Sybase, MAGNUM, IBM's Query by example, dan NOMAD systems of NCSS.

**Oracle** adalah salah satu sistem basis data relasional yang paling populer dan banyak digunakan di dunia bisnis. Ini menyediakan berbagai fitur dan tools untuk membuat, mengelola, dan mengekstrak data dari basis data relasional.

**Sybase** adalah sistem basis data relasional yang dikembangkan oleh Sybase, Inc. Ini menyediakan berbagai fitur yang membantu dalam pengelolaan data, seperti pemrosesan transaksi, pembuatan laporan, dan analisis data.

**MAGNUM** adalah sistem basis data relasional yang dikembangkan oleh Magnasoft Corporation. Ini memiliki fitur yang memungkinkan untuk pengelolaan data, pembuatan laporan, dan analisis data.

**IBM's Query by Example** adalah sistem basis data relasional yang dikembangkan oleh IBM. Ini menyediakan fitur yang memungkinkan untuk pengelolaan data, pembuatan laporan, dan analisis data.

**NOMAD systems of NCSS** adalah sistem basis data relasional yang dikembangkan oleh NCSS. Ini menyediakan fitur yang memungkinkan untuk pengelolaan data, pembuatan laporan, dan analisis data.

## Perbandingan DBMS dan RDMS

DBMS	RDBMS
DBMS (Database Management System) adalah sistem software yang digunakan untuk mengelola basis data.	RDBMS (Relational Database Management System) adalah jenis DBMS yang mengelola basis data dengan menggunakan model relasional.
DBMS dapat digunakan untuk mengelola berbagai jenis basis data seperti basis data hierarki, basis data jaringan, dan basis data objek.	RDBMS hanya dapat digunakan untuk mengelola basis data relasional.
DBMS tidak memerlukan pengelompokan data dalam tabel-tabel yang saling terkait.	RDBMS memerlukan pengelompokan data dalam tabel-tabel yang saling terkait.
DBMS tidak selalu memerlukan penggunaan kunci primer dan kunci asing untuk mengatur relasi antar tabel.	RDBMS memerlukan penggunaan kunci primer dan kunci asing untuk mengatur relasi antar tabel.
DBMS dapat digunakan dalam berbagai jenis aplikasi seperti aplikasi pribadi, aplikasi bisnis, dan aplikasi sains.	RDBMS lebih cocok digunakan dalam aplikasi bisnis yang memerlukan pengelolaan data yang cepat dan efisien.