

# MySQL – Integritas Data Dalam Basis Data

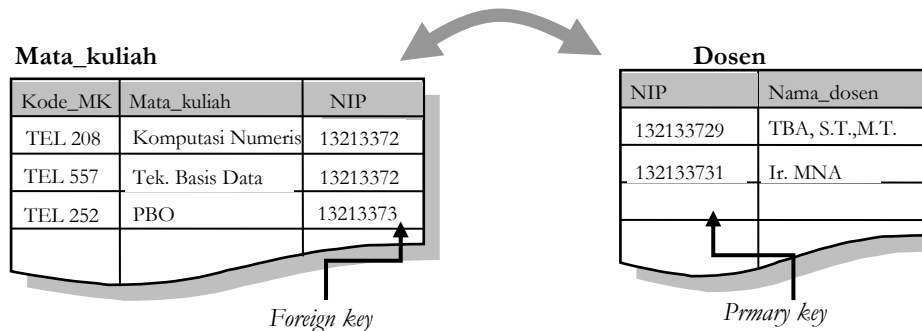
## Definisi Integritas

Secara etimologis, integritas berarti kesatuan. Dalam konteks basis data, obyek integritas tersebut adalah data. Sehingga dapat disimpulkan bahwa integritas data merupakan keutuhan dan kesatuan data dalam basis data sehingga data tersebut dapat menjadi sumber informasi yang dapat digunakan.

Munculnya istilah integritas terutama disebabkan oleh adanya konsep basis data relasional dan adanya normalisasi. Pemilahan data ke dalam tabel-tabel yang mempunyai relasi membuat integritas keseluruhan data menjadi sangat penting. Relasi antara satu tabel dengan tabel yang lain harus benar-benar terjadi sehingga keutuhan data dapat terjaga.

Pada basis data relasional dikenal istilah integrity constraint (batasan integritas). Batasan (*constraint*) integritas memastikan bahwa perubahan-perubahan yang dilakukan oleh pengguna yang berhak tidak akan menghasilkan inkonsistensi data. Batasan integritas menjaga terhadap kerusakan baik yang disengaja maupun tidak disengaja pada basis data.

Berikut ini adalah ilustrasi dari integritas data pada suatu basis data relasional. Pada gambar 1, terlihat bahwa tabel mata kuliah dan tabel dosen memiliki satu kolom yang sama yaitu kolom NIP. Relasi kedua kolom ini dapat dilihat pada **Error! Reference source not found.** Terlihat bahwa satu dosen dapat mengajar lebih dari satu mata kuliah. Dalam derajat relasi, hubungan kedua entitas ini dapat dinyatakan dengan hubungan banyak ke satu (*many to one*). Yang berarti bahwa setiap entitas pada himpunan Mata\_kuliah : NIP berhubungan dengan paling banyak 1 entitas pada himpunan Dosen : NIP.



Gambar 1. Relasi antar tabel

Adanya pemilahan tabel (yang disebabkan oleh normalisasi) menyebabkan dikenalnya kunci tamu (*foreign key*) dalam relasi antar tabel. Kunci tamu merupakan kolom yang mereferensikan pada kunci primer tabel lain, untuk menjaga relasi antar tabel. Dalam contoh di atas, yang diklasifikasikan sebagai kunci tamu (*foreign key*) adalah kolom NIP pada tabel Mata\_kuliah. Sedangkan kolom NIP pada tabel dosen merupakan kunci primer (*primary key*) dari tabel Dosen.

## Pentingnya Integritas data dalam basis data

Awalnya aplikasi basis data dibuat dengan menggunakan sistem file dalam penyimpanan basis datanya. Aplikasi basis data juga belum mempunyai DBMS (*Database Management System*) sendiri. Penggunaan sistem file dalam penyimpanan data memiliki beberapa kelemahan, diantaranya adalah sebagai berikut:

1. Pengulangan dan inkonsistensi data.  
Biasanya data disimpan dalam format file yang berbeda-beda, sehingga dimungkinkan adanya duplikasi pada file yang berbeda.
2. Kesulitan dalam pengaksesan data.
3. Sulit dalam mengisolasi karena banyaknya file serta formatnya.
4. Masalah integritas  
*Constraint* (batasan) integritas menjadi bagian dari kode program. Sehingga akan cukup sulit ketika akan melakukan perubahan atau penambahan batasan tersebut.
5. Masalah keamanan
6. Akses data yang bersamaan oleh banyak pengguna

Kemudian diperkenalkan konsep DBMS, dimana data diatur dan ditangani secara terpisah. Pada sistem informasi yang biasa digunakan pada saat ini, sangat umum terlihat adanya DBMS diantara program aplikasi dan data yang diolah. Walaupun ada juga DBMS yang menyatu dengan aplikasi

program, seperti Dbase, FoxBase, MS-Access, dll. Beberapa DBMS yang terpisah dan berdiri sendiri diantaranya adalah MySQL, PostgreSQL, Oracle, IBM DB2, dll.

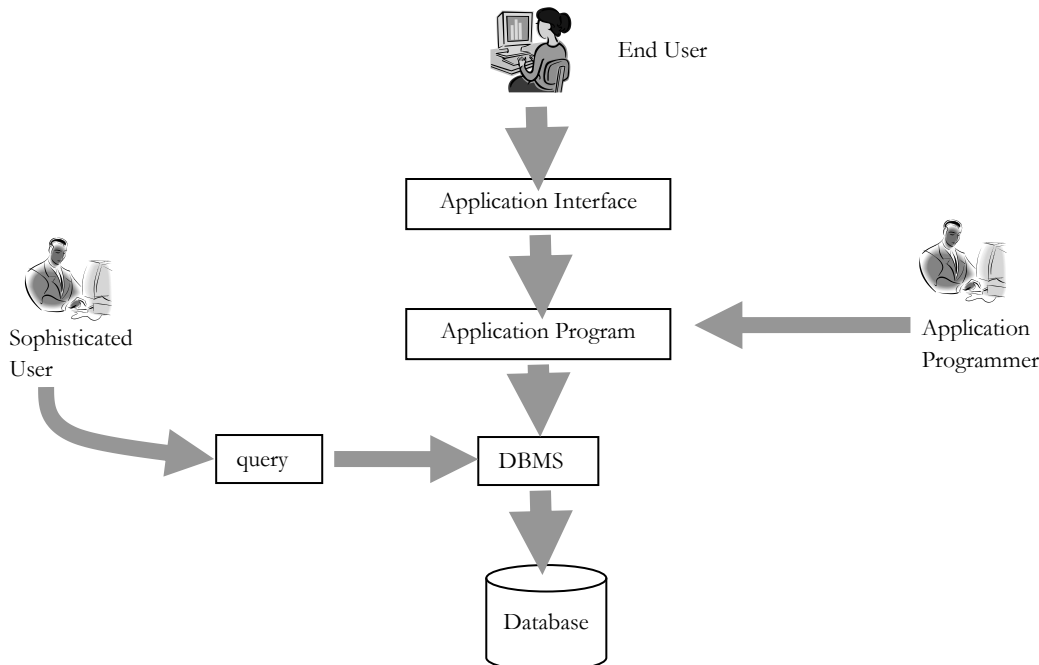
Suatu sistem informasi dengan DBMS terpisah dari aplikasi dapat dilihat pada Gambar 3.3 *Sophisticated user* merupakan pengguna yang dapat melakukan akses ke basis data dengan menggunakan DML (*Database Manipulation Language*). Dalam sistem informasi seperti ini, pemrogram aplikasi hanya melakukan pemrograman khususnya lebih kepada aplikasi yang akan digunakan oleh *end user*. Sedangkan mengenai pengorganisasian data, perancangan basis data, pembuatan query program lebih banyak dilaksanakan oleh *database administrator* dan *sophisticated user*.

Integritas yang baik dari suatu basis data dapat memudahkan pemrogram dalam melakukan pembuatan program. Karena dalam hal ini pemrogram tidak perlu mengkhawatirkan integritas data, dan tidak perlu terlalu memikirkan masalah basis data secara mendetil.

Sebagai ilustrasi dapat kita lihat dari tabel pada Gambar . Peremajaan yang dilakukan secara kurang tepat dapat mengakibatkan data yang tidak terjaga integritasnya. Apabila suatu ketika user yang berwenang melakukan peremajaan data, entah itu *sophisticated user* ataupun *naïve user*, dan ternyata melakukan kesalahan sehingga pada tabel Dosen tidak terdapat data NIP 132133729 seperti pada Gambar 2., tentu akan terjadi kejanggalan. Data akan menjadi rancu karena seharusnya mata kuliah komputasi numeris dan teknologi basis data diajar oleh dosen dengan NIP 132133729 yaitu TBA,S.T.,M.T. Akibatnya apabila dilakukan *query* pada *tuple* tersebut, maka DBMS akan memberikan informasi yang keliru dan tidak akurat.

Mata_kuliah			Dosen	
Kode_MK	Mata_kuliah	NIP	NIP	Nama_dosen
TEL 208	Komputasi Numeris	13213372	132133731	Ir. MNA
TEL 557	Tek. Basis Data	13213372		
TEL 252	PBO	13213373		

Gambar 2. Ambiguitas data ketika integritas data tidak terjaga



Gambar 3. Sistem informasi dengan DBMS terpisah

## Konsep Integritas Basis Data

### 1. Integrity Constraints (Batasan Integritas)

*Constraint* (batasan) merupakan aturan yang diberikan pada suatu tabel agar data yang dimasukkan terjamin validitasnya. Batasan integritas akan menjaga basis data dari kerusakan yang terjadi secara tidak sengaja dengan memastikan bahwa perubahan yang diperbolehkan tidak mengakibatkan terjadinya inkonsistensi data.

*Constraint* dapat diklasifikasikan sesuai dengan elemen dari basis data yang bersangkutan menjadi sebagai berikut.

- a) *Constraint* tersebut intra-relasional apabila batasan (*constraint*) terpenuhi dalam satu tabel. *Constraint* ini sendiri dapat dibedakan menjadi 2 yaitu: tuple constraint dan domain constraint. *Tuple constraint* merupakan *constraint* yang bisa dievaluasi secara independen pada setiap tuple-nya. *Domain constraint* atau sering disebut sebagai *value constraint* merupakan suatu *constraint* dengan referensi kepada nilai (*value*) tertentu. Implementasi dari penggunaan domain constraint pada SQL adalah penggunaan klausa check
- b) Suatu *constraint* dikatakan inter-relasional apabila melibatkan lebih dari satu relasi. Bentuk dari constraint ini adalah *referential integrity*.

Ada beberapa bentuk dari integrity constraint yaitu *domain constraint*, *referential integrity*, *assertion*, *trigger*.

## **2. Domain Constraint**

Domain constraint merupakan bentuk integrity constraint yang paling sederhana. Setiap ada pemasukan data baru, maka akan langsung diperiksa oleh sistem. Domain constraint diterapkan pada atribut basis data sehingga sangat dimungkinkan beberapa atribut memiliki domain yang sama.

Untuk menciptakan domain baru dapat digunakan perintah create domain. Sebagai contoh adalah pada perintah:

```
create domain Dollar numeric (12,2)
create domain Pound numeric (12,2)
```

Perintah ini akan mendefinisikan domain Dollar dan Pound dengan yang merupakan angka desimal sebanyak 12 digit dengan 2 digit berada di belakang koma.

Pada bahasa SQL juga terdapat klausa check yang bisa digunakan untuk memeriksa suatu kondisi nilai tertentu yang diinginkan. Berikut adalah contoh penggunaan klausa check pada perintah create domain.

## **3. Referential Integrity (Integritas Referensial)**

Integritas referensial adalah seperangkat aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berada dalam suatu basis data relasional untuk menjaga konsistensi data. Tujuan integritas referensial sendiri adalah untuk menjamin dan memastikan agar entitas dalam suatu tabel yang menunjuk ke suatu pengenal unik pada suatu baris di tabel lain benar-benar menunjuk pada nilai yang memang ada. Sehingga kejadian seperti pada ilustrasi Gambar 2.3 tidak akan terjadi.

Berdasarkan operasi yang dilakukan, integritas referensial dapat dibedakan sebagai berikut:

1. penambahan (insert)
2. penghapusan (delete)
3. peremajaan (update)

Integritas referensial membuat ketiga operasi di atas dapat dilaksanakan pada tabel yang memiliki relasi. Sehingga proses penghapusan ataupun peremajaan suatu kolom juga akan terjadi pada kolom tabel lain yang mempunyai referensi dengannya.

Integritas referensial merupakan cara untuk menjaga agar Kunci Asing (Foreign Key) suatu tabel dan Kunci Primer (Primary Key) milik tabel yang direferensi oleh Kunci Asing selalu konsisten.

Sekarang kita coba untuk studi kasus. Kita akan membuat database untuk keperluan pengambilan matakuliah mahasiswa seperti pada kasus di atas. Berikut ini, adalah perintah SQL untuk membuat tabel-tabelnya.

### Membuat tabel mhs

```
CREATE TABLE mhs (nim varchar(8), namaMhs varchar(20), PRIMARY KEY (nim));
```

### Membuat tabel mk

```
CREATE TABLE mk (kodeMK varchar(3), namaMK varchar(20), PRIMARY KEY (kodeMK));
```

### Membuat tabel ambilMK

```
CREATE TABLE ambilMK (nim varchar(8), kodeMK varchar(3), nilai float(3,2), PRIMARY KEY (nim, kodeMK), FOREIGN KEY (nim) REFERENCES mhs (nim) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (kodeMK) REFERENCES mk (kodeMK) ON DELETE CASCADE ON UPDATE CASCADE);
```

Tabel di atas terdapat dua primary key yaitu NIM dan KODEMK. Sedangkan field NIM ini juga merupakan foreign key yang direferensikan dari field NIM yang ada dalam tabel MHS. Oleh karena itu tambahkan perintah "FOREIGN KEY (nim) REFERENCES mhs (nim)"

SQL juga menyediakan fitur yang dapat diatur pemakai sekiranya suatu baris pada tabel induk direferensi oleh suatu kunci asing dihapus atau diubah. Tindakan-tindakan yang dapat diatur oleh pemakai ini biasa disebut tindakan referensial. Penentuak tindakan ini berbentuk :

[ON UPDATE {RESTRICT | CASCADE | SET NULL}]

[ON DELETE {RESTRICT | CASCADE | SET NULL}]

Yang terdapat dalam tanda [] berarti opsional, sedangkan yang terletak dalam tanda {} berarti harus dipilih salah satu di antara pilihan-pilihan yang dipisahkan oleh tanda |.

Penjelasan :

- UPDATE : menyatakan tindakan kalau pada tabel induk terjadi perubahan nilai.
- DELETE : menyatakan tindakan kalau pada tabel induk terjadi penghapusan baris.

Adapun tindakan yang dapat didefinisikan pada tabel ON UPDATE maupun ON DELETE beserta penjelasannya sebagai berikut :

- RESTRICT : menyatakan bahwa pengubahan atau penghapusan ditolak.
- CASCADE : jika nilai kunci primer pada tabel induk berubah (UPDATE) maka kunci asing pada tabel yang mereferensi akan disesuaikan dengan nilai pada kunci primer tabel induk, sedangkan apabila terjadi proses DELETE semua kunci asing yang cocok dengan kunci primer pada tabel induk milik record yang dihapus akan ikut dihapus.
- SET NULL : Menyatakan kunci asing akan diisi dengan NULL kalau kunci primer pada tabel induk yang nilainya sama dengan nilai pada kunci asing tersebut diubah atau dihapus.