Pertemuan ke-8 Design dan Manipulasi Database dengan MySQL dan PHPMyAdmin

Tutorial ini menggunakan MySQL dan PhpMyAdmin dengan platform Windows. Silakan implementasikan pada MySQL/MariaDB dan PhpMyAdmin pada WSL Ubuntu. Perintah SQL (*Structured Query Language*), baik di Windows maupun Linux tidak ada perbedaan.

Basis data (atau *database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (http://id.wikipedia.org/wiki/Database). Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

Dari sekian banyak aplikasi database, MySQL merupakan yang paling populer dan paling banyak digunakan terutama bersama PHP. Salah satu alasannya karena MySQL bersifat open source dan gratis. Namun saat ini ternyata MySQL sudah dibeli oleh Oracle. Untuk antisipasi jika suatu saat nanti MySQL sudah tidak free lagi seperti halnya Oracle, maka keluarlah MariaDB. Pada dasarnya, MariaDB adalah cloningan MySQL. Dengan demikian perintah MySQL hampir sama dengan MariaDB. Saat in jika kita install XAMPP versi terbaru, maka databasenya tidak lagi menggunakan MySQL, tetapi MariaDB.

😫 XAMPI	Control Pan	iel v3.2.2 [Co	mpiled: Nov 12th 2	2015]				- 🗆	×
ន	XAN	VPP Cont	rol Panel v3	.2.2				🥜 Config	9
Service	Module	PID(s)	Port(s)	Actions				Netsta	t
	Apache	2780 11064	80, 443	Stop	Admin	Config	Logs	Shell	
	MySQL	11108	3306	Stop	Admin	Config	Logs	Explore	er
	FileZilla			Start	Admin	Config	Logs	🦻 Service	es
	Mercury			Start	Admin	Config	Logs	😡 Help	
	Tomcat			Start	Admin	Config	Logs	Quit	
3:19:27 F 3:19:27 F 3:19:27 F 3:19:27 F 3:19:27 F 3:19:30 F 3:19:30 F 3:19:31 F 3:19:33 F	M [main] M [main] M [main] M [main] M [main] M [Apache M [Apache M [mysql]	The FileZ The Merc The Tomo Starting (Control P] Attemptin] Status ch Attemptin Status ch	illa module is dis ury module is dis cat module is dis Check-Timer anel Ready ig to start Apachi ange detected: r ig to start MySQ iange detected: r	abled sabled abled e app unning L app unning					<



1. Menjalankan MySQL Server dan MySQL Client

Sama seperti web server Apache dan web browser, aplikasi MySQL juga menggunakan arsitektur serupa, yakni ada yang bertindak sebagai MySQL Server, dan ada yang berfungsi sebagai MySQL Client. Untungnya XAMPP sudah membundle kedua aplikasi ini sehingga kita bisa langsung pakai.

MySQL Server adalah aplikasi yang akan memproses dan mengolah seluruh database. Sistem kerjanya sangat mirip seperti web server Apache yang mengolah permintaan dari web browser.

Pertama, mari kita jalankan aplikasi MySQL Server. Caranya sangat mudah, silahkan buka XAMPP Control Panel, lalu klik tombol Start pada baris MySQL, seperti gambar berikut:



merah, berarti MySQL belum berjalan atau bentrok dengan aplikasi MySQL Server lain.

Jika tidak ada masalah, saatnya kita menjalankan aplikasi MySQL Client untuk berkomunikasi dengan MySQL Server ini. Terdapat berbagai aplikasi yang bisa digunakan. Kita bisa menggunakan PHP sebagai MySQL Client. Inilah yang akan kita pelajari nantinya, yakni bagaimana cara mengakses database MySQL (yang berada di MySQL Server) dari kode program PHP.

Sebelum sampai kesana, saya ingin membahas terlebih dahulu bahasa SQL yang digunakan untuk berkomunikasi dengan MySQL Server. Ini akan lebih mudah jika menggunakan MySQL Client yang dijalankan dari cmd Windows atau Command Line.

Silakan klik tombol **search** windows, kemudian ketik **'cmd'**. Setelah itu akan muncul icon command promp, selanjutnya silakan klik command promp.

Setelah muncul command prompt, silakan arahkan ke folder instalasi XAMPP->MySQL dengan perintah cd C:\xampp\mysql\bin. Selanjutnya masuk ke MySQL Server dengan perintah mysql –u root. seperti gambar di bawah ini:



Di dalam folder c:\xampp\mysql\bin terdapat file **mysql.exe**, file inilah yang kita panggil dengan perintah mysql -u root. Tambahan perintah -u root berarti saya ingin masuk sebagai user root. User root adalah user yang memiliki hak akses tertinggi di dalam MySQL, dimana ia dapat mengakses seluruh database yang ada (sering juga disebut sebagai superadmin).

Secara default, XAMPP membuat user root MySQL tanpa password. Oleh karena itulah kita bisa masuk hanya dengan perintah mysql -u root (tidak perlu menulis password). Dari sisi keamanan, sebuah user tanpa pasword sangat tidak disarankan. Tapi karena disini kita hanya belajar cara penggunaan MySQL secara offline, jadi tidak terlalu masalah. Nantinya kita bisa memberikan password kepada user root maupun user lain.

2. Dasar-dasar SQL (Structured Query Language)

Untuk bisa menggunakan MySQL, terlebih dahulu harus paham cara penggunaan bahasa SQL (Structured Query Language). Mirip seperti HTML yang digunakan untuk membuat halaman web, SQL digunakan untuk membuat database. SQL bersifat generik dan hampir semua aplikasi database menggunakannya (tidak hanya MySQL saja). Tentunya MySQL

Pemrogram Web | Design dan Manipulasi Database dengan MySQL dan

PHPMyAdmin

menambahkan beberapa perintah khusus yang hanya berlaku di MySQL dan tidak terdapat di Oracle, begitu juga sebaliknya.

Bahasa SQL mirip dengan bahasa inggris sehari-hari, seperti SHOW DATABASE, USE DATABASE, DROP TABLE, SELECT, dst. Secara umum, perintah bahasa SQL atau sering disingkat sebagai "**query**" saja, terbagi menjadi 4 jenis:

- 1. Perintah untuk membuat data (CREATE).
- 2. Perintah untuk membaca data (READ).
- 3. Perintah untuk mengubah data (UPDATE).
- 4. Perintah untuk menghapus data (DELELE).

Keempat istilah ini sering disingkat sebagai CRUD (CREATE, READ, UPDATE, dan DELETE). Oleh karena itu aplikasi yang berfungsi untuk menjalankan perintah-perintah ini disebut juga sebagai aplikasi CRUD.

MEMBUAT, MENGAKSES DAN MENGHAPUS DATABASE MySQL

Hal pertama yang harus kita lakukan adalah membuat sebuah database. Untuk membuatnya, silahkan ketik perintah berikut, lalu akhiri dengan tombol Enter:

MariaDB [(none)]> CREATE DATABASE universitas;

Jika tidak ada masalah, tampilannya adalah sebagai berikut:



Sekarang, di dalam MySQL sudah terbentuk sebuah database "universitas". Isinya tentu belum ada tabel apapun karena kita memang belum membuatnya.

Perhatikan cara penulisan perintah SQL di dalam MySQL. Sama seperti PHP, setiap perintahharus diakhiri dengan tanda titik koma (;). Seluruh perintah di dalam MySQL bersifat case insensitive, yang tidak membedakan huruf besar dan huruf kecil. Perintah di atas juga bisa ditulis dengan huruf kecil semua, seperti:

create database universitas;

Namun sudah menjadi standar tidak resmi kalau perintah SQL ditulis dengan HURUF BESAR, sedangkan untuk nama variabel seperti "universitas" tetap ditulis huruf kecil. Dengan demikian kita bisa membedakan mana perintah SQL dan mana variabel yang dibuat oleh programmer.

Khusus untuk nama database, nama tabel, nama kolom tabel, dan nama variabel lain yang bisa kita tentukan sendiri, sebaiknya tidak menukar huruf besar atau kecil. Misalkan saya

membuat database "unversitas". Di sistem windows, database ini bisa diakses dengan "Universitas" maupun "UNIVERSITAS" yang artinya tidak membedakan huruf besar dan kecil. Tapi di sistem Linux ini tidak berlaku. Jadi sebaiknya konsisten untuk penamaan variabel. Jika memutuskan untuk menggunakan huruf kecil, gunakan huruf kecil untuk semua variabel.

Setelah kita menekan tombol Enter, akan tampil keterangan: Query OK, 1 row affected (0.00 sec). Ini adalah feedback atau laporan dari perintah yang baru saja kita ketik. "Query OK" berarti query atau perintah tersebut tidak ada yang salah. "1 row affected" berarti ada 1 baris yang diubah akibat perintah ini. "(0.00 sec)" adalah waktu yang diperlukan MySQL untuk menjalankannya.

Karena proses pembuatan database cukup sederhana, MySQL Server hanya butuh sepersekian detik untuk menjalankan proses ini. Ini juga karena komputer yang saya gunakan tidak ada proses lain yang sedang berjalan.

Bagaimana jika perintah (query) yang ditulis terdapat kesalahan? Mari kita coba. Silahkan jalankan perintah berikut:

MariaDB [(none)]> CREAT DATABASE universitas;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'CREAT DATABASE universitas' at line 1

Di sini saya sengaja menghapus huruf "E" dari "CREATE" menjadi "CREAT". MySQL akan langsung komplain dan menampilkan bagian yang salah. Perintah CREATE DATABASE digunakan untuk membuat database.

Agar bisa menggunakannya, kita harus memilih database ini dengan perintah USE. Berikut querynya:

MariaDB [(none)]> USE universitas;

Database changed

Perhatikan perubahan dari **MariaDB** [(none)] menjadi **MariaDB** [universitas]. Ini artinya kita sudah menetapkan sebuah database. Setiap query yang ditulis akan berefek ke database universitas saja.

Bagaimana jika saya memutuskan untuk menukar database? Cukup jalankan perintah yang sama dengan format **USE nama_database**. Untuk melihat database apa saja yang saat ini tersedia di dalam MySQL, bisa menggunakan query SHOW DATABASES, seperti berikut:

MariaDB [universitas]> SHOW DATABASES;

++
Database
information_schema mysql performance_schema phpmyadmin test universitas
7 rows in set (0.11 sec)

Perlu diperhatikan bahwa perintahnya adalah SHOW DATABASES, bukan SHOW DATABASE (ada tambahan huruf "S" menjadi "DATABASES"). Kesalahan akibat lupa menulis "S" ini cukup sering terjadi.

Dari tampilan yang ada terlihat selain database universitas, sudah ada beberapa database lain. Diantaranya merupakan database bawaan MySQL (seperti information_schema dan mysql), dan database bawaan XAMPP (seperti phpmyadmin). Database ini sebaiknya dibiarkan saja, dan disarankan untuk tidak mengubah atau menghapusnya.

Untuk menghapus database, bisa menggunakan query DROP DATABASE. Misalkan saya ingin menghapus database universitas, perintahnya adalah:

MariaDB [universitas]> DROP DATABASE universitas; Query OK, 0 rows affected (0.13 sec)

Perintah ini sangat powerfull. Database akan dihapus beserta seluruh tabel di dalamnya (jika ada). Di dalam MySQL juga tidak dikenal perintah "undo". Jadi anda perlu berhatihati terhadap query DROP ini.

MEMBUAT DAN MENGHAPUS TABEL MySQL

Setelah database universitas dipilih, kita bisa mulai membuat tabel. Sebagai informasi tambahan, membuat database dan tabel tidak sering kita lakukan. Umumnya kegiatan ini hanya perlu sekali di awal, dan selesai. Oleh karena itu jangan heran jika anda sering lupa query untuk membuat tabel. Sepanjang pengembangan website kita hanya sering menggunakan perintah READ, UPDATE dan DELELE.

Untuk membuat tabel, querynya cukup sederhana. Berikut format dasar pembuatan tabel di dalam MySQL:

CREATE TABLE nama_tabel (nama_kolom1 jenis_kolom1, nama_kolom2 jenis_kolom2, nama_kolom3 jenis_kolom3,...);

Yang cukup kompleks adalah menentukan jenis kolom. Ini sama halnya dengan mempelajari tipe data di dalam PHP. Apakah itu string, integer, date, dll. Selain itu kita juga bisa memberikan syarat tambahan untuk setiap kolom, apakah tidak boleh kosong (NOT NULL), tanpa tanda (UNSIGNED), atau penomoran otomatis (AUTO_INCREMENT).

Berikut contoh query untuk membuat sebuah tabel data_mahasiswa:

```
MariaDB [universitas]> CREATE TABLE data_mahasiswa (
    nim char(9),
    nama char(50),
    umur int,
    tempat_lahir char(50),
    jurusan char (30)
);
Query OK, 0 rows affected (0.40 sec)
```

Dengan query diatas, saya membuat tabel data_mahasiswa yang terdiri dari 5 kolom: nim, nama, umur, tempat_lahir, dan jurusan. Penulisan nama kolom diikuti dengan jenis dari kolom tersebut dan dipisah dengan tanda koma. Untuk kolom yang berisi huruf (string), bisa menggunakan tipe data char. Angka di dalam kurung adalah jumlah maksimal karakter yang bisa ditampung, misalnya untuk kolom nama, saya menginstruksikan MySQL untuk menyediakan sebanyak 50 karakter. Untuk kolom yang berisi angka, MySQL

```
Pemrogram Web | Design dan Manipulasi Database dengan MySQL dan PHPMyAdmin
```

menyediakan beberapa tipe data. Dalam contoh di atas, saya memilih integer, yang bisa disingkat menjadi int.

Setelah tabel dibuat, mari kita pastikan dengan perintah SHOW TABLES:

```
MariaDB [universitas]> SHOW TABLES;
+-----+
| Tables_in_universitas |
+-----+
| data_mahasiswa |
+-----+
1 row in set (0.07 sec)
```

Untuk melihat struktur dari tabel (seperti nama kolom dan tipe datanya) bisa menggunakan query DESCRIBE atau DESC, seperti berikut ini:

MariaDB [universitas]> DESCRIBE data mahasiswa;

Field	Type	Null	Кеу	Default	Extra
nim nama umur tempat_lahir jurusan	<pre> char(9) char(50) int(11) char(50) char(30) +</pre>	YES YES YES YES YES		NULL NULL NULL NULL NULL	

⁵ rows in set (0.01 sec)

Bagaimana dengan menghapus tabel? Bisa menggunakan query DROP TABLE. Berikut contohnya:

MariaDB [universitas]> DROP TABLE data_mahasiswa; Query OK, 0 rows affected (0.34 sec)

MariaDB [universitas]> SHOW TABLES; Empty set (0.00 sec)

Sekarang tabel data_mahasiswa sudah dihapus. Sama seperti DROP DATABASE, query ini harus digunakan dengan hati-hati. Tabel yang sudah terhapus tidak dapat dikembalikan.

JENIS TIPE DATA UNTUK KOLOM TABEL MySQL

MySQL mendukung banyak tipe data yang bisa digunakan untuk membuat kolom. Mari kita bahas beberapa diantaranya.

Tipe Data Angka Bulat (Integer)

Untuk kolom dengan data angka bulat (integer), MySQL menyediakan 5 pilihan yang dibedakan berdasarkan jangkauannya:

- TINYINT: Mendukung angka bulat dari -128 hingga 127 (signed), atau 0 hingga 255 (unsigned).
- SMALLINT: Mendukung angka bulat dari -32768 hingga 32767 (signed), atau 0 hingga 65535 (unsigned).
- MEDIUMINT: Mendukung angka bulat dari -8388608 hingga 8388607 (signed), atau 0 hingga 16777215 (unsigned).
- INT: Mendukung angka bulat dari -2147483648 hingga 2147483647 (signed), atau 0 hingga 4294967295 (unsigned).

 BIGINT: Mendukung angka bulat dari -9223372036854775808 hingga 9223372036854775807 (signed), atau 0 hingga 18446744073709551615 (unsigned).

Pemilihan tipe data ini tergantung kebutuhan. Misalkan untuk nomor urut absensi pada suatu kelas yang memiliki maksimal 30 orang, tipe data TINYINT sudah mencukupi. Namun jika kita bermaksud membuat absensi seluruh rakyat indonesia, tipe data TINYINT sampai dengan MEDIUMINT tidak akan cukup. Kita harus memilih INT atau BIGINT. Format penulisan query untuk kolom dengan tipe data integer adalah:

INT[(M)] [UNSIGNED] [ZEROFILL]

Dalam deklarasi tipe data integer, nilai M digunakan untuk mengatur jumlah digit yang disediakan untuk menampilkan data. Misalkan saya mendefiniskan suatu kolom dengan INT(5), ketika diinput angka 102, MySQL akan menambahkan spasi sebanyak 2 buah di depan angka agar tampilan data menjadi 5 digit (istilah programmingnya: padding left). Nilai M tidak mempengaruhi jangkauan dari kolom tersebut. Jika angka yang diinput melebihi digit M, MySQL tetap menyimpan hasilnya (selama masih dalam jangkauan tipe data tersebut).

Setiap tipe data integer dapat didefinisikan sebagai UNSIGNED, dimana kita mengorbankan nilai negatif untuk mendapatkan jangkauan nilai positif yang lebih besar. Artinya kolom tersebut hanya bisa diisi dengan angka positif saja.

Selain tambahan UNSIGNED, kolom integer juga bisa ditambah atribut ZEROFILL. ZEROFILL berhubungan dengan nilai M yang akan mengganti tempat spasi menjadi angka 0. Misalkan jika kita mendefenisikan INT(5) ZEROFILL dan menginput angka 102, hasilkan akan menjadi 00102.

Tipe Data Angka Pecahan (Float)

Untuk tipe data angka pecahan (float), terdapat 3 pilihan: FLOAT, DOUBLE, dan DECIMAL. Tipe data FLOAT dan DOUBLE menggunakan pendekatan presisi dengan tingkat ketelitian tertentu. Jangkauan tipe data DOUBLE lebih besar daripada FLOAT. Untuk penggunaan sehari-hari, sebaiknya kita menggunakan tipe data DECIMAL karena terdapat perbedaan pembulatan. Tipe data FLOAT dan DOUBLE lebih cocok untuk perhitungan data-data sains.

Penulisan tipe data pecahan di MySQL sedikit berbeda dengan integer. Format penulisannya adalah sebagai berikut:

DECIMAL [(M[,D])] [UNSIGNED] [ZEROFILL]

Besar dari jangkaian tipe data float tergantung kepada angka [M,D]. M adalah total jumlah digit keseluruhan, sedangkan D adalah jumlah digit dibelakang koma (pecahan). Contohnya DECIMAL [6,2] akan mendefiniskan suatu kolom agar memuat 6 digit angka, dengan 4 angka di depan koma, dan 2 digit angka di belakang koma.

Untuk tipe data DECIMAL, maksimal nilai untuk M = 65, dan D = 30.

Tipe Data Teks (String)

Untuk menyimpan tipe data teks atau string, MySQL juga menyediakan banyak tipe data. Saya akan membahas yang sering digunakan, yakni CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, dan LONGTEXT.

Tipe data CHAR dan VARCHAR adalah tipe data karakter (string) yang akan sering kita gunakan. Format penulisannya adalah sebagai berikut:

Pemrogram Web | Design dan Manipulasi Database dengan MySQL dan

PHPMyAdmin

CHAR [(M)] VARCHAR [(M)]

Digit M akan menentukan jumlah karakter maksimum yang akan disiapkan MySQL. Sebagai contoh, nilai M=5, maka MySQL menyediakan 5 karakter untuk kolom tersebut. Nilai maksimal M adalah 255 karakter untuk CHAR, dan 65535 karakter untuk VARCHAR. Jika nilai M tidak dinyatakan, nilai defaultnya adalah M = 1.

Perbedaan antara CHAR dan VARCHAR terletak dari cara MySQL mengalokasikan ukuran memory untuk menyimpan data. Sebagai contoh, jika kita mendefinisikan kolom bertipe CHAR(5), walaupun huruf atau karakter yang kita inputkan hanya 1 karakter, MySQL tetap menyimpan kolom tersebut untuk 5 karakter.

Namun jika kita menggunakan VARCHAR(5) dan menginput data dengan jumlah karakter 2, ukuran penyimpanan hanya akan menggunakan 2 karakter. Disini terlihat VARCHAR lebih fleksibel dan efisien. Tetapi untuk fleksibilitas ini, tipe VARCHAR memerlukan proses tambahan untuk menyimpan ukuran dari masing-masing data. Baik CHAR maupun VARCHAR menyimpan data secara case insensitif, dimana huruf besar dan kecil tidak dibedakan.

Bagaimana jika jumlah huruf yang diinput melebihi 65535 karakter? Kita bisa menggunakan tipe data TEXT. Tipe data TEXT hadir dengan 4 jenis:

- TINYTEXT: maksimum karakter 255
- TEXT: maksimum karakter 65.535
- MEDIUMTEXT: maksimum karakter 16.777.215
- LONGTEXT: maksimum karakter 4.294.967.295

Terlihat tipe data LONGTEXT sanggup menampung hingga 4GB teks. Sudah sangat besar untuk keperluan menyimpan sebuah teks. Setiap tipe data text disimpan berdasarkan berapa banyak data yang diinput, bukan berdasarkan jangkauan maksimalnya (ini sama seperti VARCHAR). Jika saya mendefinisikan suatu kolom sebagai LONGTEXT dan isinya hanya 100 karakter, ukuran penyimpanan yang digunakan hanya sekitar 100 byte, bukan 4GB.

Tipe Data Tanggal (DATE)

MySQL menyediakan berbagai tipe data tanggal: DATE, TIME, DATETIME, TIMESTAMP, dan YEAR. Perbedaannya terletak di bentuk format data. Berikut rinciannya:

- **DATE**: Memiliki format YYYY-MM-DD, dengan jangkauan dari 1000-01-01 hingga 9999-12-31. Tanggal 29 November 2016 akan disimpan sebagai 2016-11-29.
- **DATETIME**: Memiliki format YYYY-MM-DD HH:MM:SS, dengan jangkauan dari 1000-01-01 00:00:00 hingga 9999-12-31 23:59:59. Tanggal 29 November 2016 pukul 3 lebih 20 menit 45 detik sore akan disimpan sebagai 2016-11-29 15:20:45.
- **TIMESTAMP**: Memiliki format YYYYMMDDHHMMSS dengan jangkauan 1 Januari 1970 hingga 19 Januari 2038. Tanggal 29 November 2016 pukul 3 lebih 20 menit 45 detik sore akan disimpan sebagai 20161129152045.
- **TIME**: Memiliki format HH:MM:SS dengan jangkauan 00:00:00 hingga 23:59:59. Pukul 3 lebih 20 menit 45 detik sore akan disimpan sebagai 15:20:45 menit 45 detik sore akan disimpan sebagai 15:20:45
- **YEAR(M):** Memiliki format YYYY atau YY tergantung angka M. Jika ditulis sebagai YEAR(2), bisa menyimpan data dari 70 hingga 69 (tahun 1970 hingga 2069). Jika ditulis sebagai YEAR(4), bisa menyimpan data dari 1901 hingga 2155. Nilai M default adalah 4.

3. CREATE, READ, UPDATE DAN DELETE (CRUD) TABLE

Membuat Tabel mahasiswa

Misalkan kita ingin membuat table seperti berikut:

NIM	Nama	Tempat Lahir	Tanggal Lahir	Fakultas	Jurusan	IPK
15002032	Rina Kumala Sari	Jakarta	28/06/1997	Ekonomi	Akuntansi	3.4
13012012	James Situmorang	Medan	02/04/1995	Kedokteran	Kedokteran Gigi	2.7
14005011	Riana Putria	Padang	23/11/1996	FMIPA	Kimia	3.1
15021044	Rudi Permana	Bandung	22/08/1994	FASILKOM	Ilmu Komputer	2.9
15003036	Sari Citra Lestari	Jakarta	31/12/1997	Ekonomi	Manajemen	3.5

Langkah pertama kita membuat tabel dengan nama tabel Mahasiswa dengan membuat field sebanyak kolom table di atas.

```
MariaDB [universitas]> CREATE TABLE mahasiswa (
    nim CHAR(8),
    nama VARCHAR(100),
    tempat_lahir VARCHAR(50),
    tanggal_lahir DATE,
    jurusan VARCHAR(50),
    ipk DECIMAL(3,2),
    PRIMARY KEY (nim)
);
Query OK, 0 rows affected (0.06 sec)
```

Command Prompt	- mysql -u root					_	×
<pre>MariaDB [univers: -> nim CHAR(; -> nama VARC(-> tempat_laf -> tanggal_la -> jurusan V/ -> jpk DECIM/ -> ipk DECIM/ -> PRIMARY KI ->); Query OK, 0 rows MariaDB [univers: </pre>	itas]> CREATE T, 8), HAR(100), hir VARCHAR(50), ahir DATE, ARCHAR(50), AL(3,2), EY (nim) affected (0.20 itas]> DESC maha	ABLE mal , sec) asiswa;	hasiswa	a (^
Field	Туре	Null	Key	Default	Extra		
nim nama tempat_lahir tanggal_lahir jurusan ipk	char(8) varchar(100) varchar(50) date varchar(50) decimal(3,2)	NO YES YES YES YES YES	PRI 	NULL NULL NULL NULL NULL NULL			
6 rows in set (0	.02 sec)						~

Tabel mahasiswa sudah dibuat, kini saatnya kita input beberapa data. Kolom **nim** diset sebagai **primary key**, sehingga kolom ini tidak bisa diisi karakter yang sama (harus unik). Agar tabel tidak terlalu panjang, kolom fakultas sengaja tidak dibuat.

Cara Menginput Data ke Tabel MySQL

Untuk menginput data ke dalam tabel MySQL, kita menggunakan query INSERT. Terdapat beberapa alternatif penulisan. Yang pertama, tanpa menuliskan nama kolomnya. Jika ditulis seperti ini, data yang diinput harus sesuai dengan jumlah serta urutan kolom dari tabel, seperti contoh berikut:

MariaDB [universitas]> INSERT INTO mahasiswa VALUES
('15002032','Rina Kumala Sari','Jakarta','1997-06-28','Akuntansi', 3.4);
Query OK, 1 row affected (0.04 sec)

Perhatikan cara penulisan query INSERT ini. Formatnya adalah:

INSERT INTO nama_tabel VALUES (nilai_kolom1, nilai_kolom2,..dst);

Karena tabel mahasiswa memiliki 7 kolom, maka saya harus menuliskan nilai untuk setiap kolom. Selain itu urutannya juga harus sesuai dengan struktur tabel.

Untuk nilai string/char/date, harus ditulis dalam tanda kutip. Tanda kutip ini bisa tanda kutip satu (') maupun tanda kutip dua ("). Biasanya yang sering digunakan adalah tanda kutip satu agar tidak bentrok dengan tanda kutip dua yang biasa digunakan PHP.

Untuk nilai angka baik integer maupun float, tidak perlu menggunakan tanda kutip meskipun MySQL tetap akan menerima nilai ini. Namun tidak sebaliknya untuk tipa data string/char/date, jika kita lupa menulis tanda kutip, MySQL akan mengeluarkan pesan error.

Khusus untuk kolom tanggal_lahir, cara penulisan format **date** harus sesuai dengan aturan MySQL, yakni dengan format **yyyy-mm-dd**. Untuk memastikan data tersebut telah sukses diinput, kita bisa melihatnya menggunakan query SELECT:

Mari	iaDB [uni	iversitas]> SELECT [;]	* FROM mahasiswa	a;		L	
ni	im	nama	tempat_lahir	tanggal_lahir	jurusan	ipk	
15	5002032	Rina Kumala Sari	Jakarta	1997-06-28	Akuntansi	3.40	F
1 rc	ow in set	t (0.00 sec)				r	F

Terlihat data "Rina Kumala Sari" sudah sukses diinput. Sekarang, bagaimana jika saya ingin menginput sebagian kolom saja? Untuk ini nama kolom tabel harus ditulis seperti contoh query berikut:

Sebelum query VALUES, kita harus menulis apa saja nama kolom yang ingin diinput. Format dasarnya adalah sebagai berikut:

INSERT INTO nama_tabel (nama_kolom1, nama_kolom2, nama_kolom3, ..dst)
VALUES (nilai_kolom1, nilai_kolom2, nilai_kolom3,..dst);

Dari hasil query SELECT terlihat kolom yang tidak diinput datanya ditampilkan sebagai NULL.

Bagaimana cara menginput banyak data sekaligus? Tentunya kita bisa menjalankan perintah INSERT beberapa kali, atau bisa juga dipisah dengan tanda koma seperti contoh berikut ini:

```
MariaDB [universitas]> INSERT INTO mahasiswa (nim, nama, tempat_lahir,
tanggal_lahir, jurusan) VALUES
('14005011', 'Riana Putria', 'Sukabumi', '1996-11-23', 'Manajemen'),
('15021044', 'Rudi Permana', 'Bandung', '1994-08-22', 'Ilmu Komputer'),
('15003036', 'Sari Citra Lestari', 'Jakarta', '1997-12-31', 'Manajemen');
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

MariaDB [universitas]> SELECT * FROM mahasiswa;							
nim	nama	tempat_lahir	tanggal_lahir	jurusan	ipk		
13012012 14005011 15002032 15003036 15021044	James Situmorang Riana Putria Rina Kumala Sari Sari Citra Lestari Rudi Permana	Medan Sukabumi Jakarta Jakarta Bandung	1995-04-02 1996-11-23 1997-06-28 1997-12-31 1994-08-22	NULL Manajemen Akuntansi Manajemen Ilmu Komputer	NULL NULL 3.40 NULL NULL		

5 rows in set (0.00 sec)

Cara Menampilkan Data dari Tabel MySQL

Dari semua perintah query MySQL, menampilkan data tabel merupakan perintah yang paling sering dilakukan. Untuk ini kita menggunakan query SELECT. Query SELECT memiliki banyak variasi, namun secara umum format dasarnya adalah sebagai berikut:

SELECT nama kolom FROM nama tabel [WHERE kondisi];

Untuk menampilkan seluruh kolom dari sebuah tabel, nama_kolom bisa diganti dengan tanda bintang (*). Jika saya ingin menampilkan seluruh kolom dari tabel mahasiswa, querynya adalah sebagai berikut:

MariaDB [universitas]> SELECT * FROM mahasiswa;

+	+	+	+	+	++
	nama	tempat_lahir	tanggal_lahir	jurusan	ipk
	+	+	+	+	++
13012012 14005011 15002032	' James Situmorang Riana Putria Rina Kumala Sari	, Medan Sukabumi Jakarta	, 1995-04-02 1996-11-23 1997-06-28	NULL Manajemen Akuntansi	NULL NULL 3.40

15003036 Sari Citra Lestari	Jakarta	1997-12-31	Manajemen	NULL
15021044 Rudi Permana	Bandung	1994-08-22	Ilmu Komputer	NULL
+			++	+

```
5 rows in set (0.00 sec)
```

Bagaimana jika ingin menampilkan kolom nim dan nama saja? Caranya dengan menuliskan nama kolom setelah perintah SELECT.

```
MariaDB [universitas]> SELECT nim,nama FROM mahasiswa;
+-----+
| nim | nama |
+----+
| 13012012 | James Situmorang |
| 14005011 | Riana Putria |
| 15002032 | Rina Kumala Sari |
| 15003036 | Sari Citra Lestari |
| 15021044 | Rudi Permana |
+----+
5 rows in set (0.00 sec)
```

Untuk menampilkan data yang lebih spesifik, kita bisa menambahkan kondisi. Misalnya kita ingin menampilkan seluruh data hanya untuk mahasiswa yang bernama Rudi Permana, berikut query yang bisa digunakan:

```
MariaDB [universitas]> SELECT * FROM mahasiswa WHERE nama='Rudi Permana';
+-----+
| nim | nama | tempat_lahir | tanggal_lahir | jurusan | ipk |
+-----+
| 15021044 | Rudi Permana | Bandung | 1994-08-22 | Ilmu Komputer | NULL |
+-----+
1 row in set (0.01 sec)
```

Atau untuk menampilkan data dimana nimnya adalah 13012012:

MariaDB [universitas]> SELECT *	FROM mahasiswa	a WHERE nim='1301	2012';	LA
nim nama	tempat_lahir	tanggal_lahir	jurusan	ipk
13012012 James Situmorang	Medan	1995-04-02	NULL	NULL
1 row in set (0.00 sec)				1

Perintah kondisi dari query SELECT ini bisa menjadi cukup rumit sekaligus powerful. Misalkan kita ingin menampilkan data mahasiswa yang lahir di Bandung atau Jakarta. Querynya adalah sebagai berikut:

	nim	nama	tempat_lahir	tanggal_lahir	jurusan	ipk
	15002032 15003036 15021044	Rina Kumala Sari Sari Citra Lestari Rudi Permana	Jakarta Jakarta Bandung	1997-06-28 1997-12-31 1994-08-22	Akuntansi Manajemen Ilmu Komputer	3.40 NULL NULL
+	rows in se	et (0.00 sec)	+	+		++

Bagaimana untuk mengurutkan data? Misalkan saya ingin menampilkan kolom nama dan tanggal_lahir dan diurutkan berdasarkan abjad dari nama. Berikut querynya:

MariaDB [universitas]> SELECT nama,tanggal lahir FROM mahasiswa ORDER BY nama ASC ; +----+ | nama | tanggal lahir | +----+ | James Situmorang | 1995-04-02 | | Riana Putria | 1996-11-23 | Rina Kumala Sari | 1997-06-28 - I | Rudi Permana | 1994-08-22 - I | Sari Citra Lestari | 1997-12-31 - I +----+ 5 rows in set (0.00 sec)

Cara Mengubah Data dari Tabel MySQL

Setelah sebuah data diinput, kadang kita butuh mengubah atau meng-update data tersebut. Untuk keperluan ini, MySQL menyedian query UPDATE. Berikut format dasarnya:

UPDATE nama tabel SET nama kolom = data baru WHERE kondisi

Misalkan saya ingin mengupate tempat lahir "Riana Putria" menjadi "Padang". Querynya adalah sebagai berikut:

```
MariaDB [universitas]> UPDATE mahasiswa SET tempat_lahir='Padang'
WHERE nama='Riana Putria';
```

Menghapus Data dari Tabel MySQL

Query terakhir dari CRUD adalah bagaimana cara menghapus sebuah data. Untuk keperluan ini kita akan menggunakan perintah DELETE. Berikut format dasarnya:

DELETE FROM nama tabel WHERE kondisi

Misalkan saya ingin menghapus data mahasiswa yang bertempat lahir di Jakarta. Quernya adalah:

M	<pre>fariaDB [universitas]> SELECT * FROM mahasiswa;</pre>							
+ + + +	nim	nama	tempat_lahir	+ tanggal_lahir	jurusan	ipk		
	13012012 14005011 15002032 15003036 15021044	James Situmorang Riana Putria Rina Kumala Sari Sari Citra Lestari Rudi Permana	Medan Padang Jakarta Jakarta Bandung	, 1995-04-02 1996-11-23 1997-06-28 1997-12-31 1994-08-22	NULL Manajemen Akuntansi Manajemen Ilmu Komputer	NULL NULL 3.40 NULL NULL		

Misalkan saya ingin menghapus data mahasiswa yang bertempat lahir di Jakarta. Querynya adalah:

MariaDB [universitas]> DELETE FROM mahasiswa WHERE tempat_lahir="Jakarta";

Query OK, 2 rows affected (0.05 sec)

Sekarang coba kita tampilkan tabel Mahasiswa. Terlihat record yang kolom tempat_lahir dari Jakarta sudah tidak ada dalam tabel Mahasiswa.

MariaDB [universitas]> select * from mahasiswa;							
nim	nama	tempat_lahir	tanggal_lahir	jurusan	ipk	ļ	
13012012 14005011 15021044	James Situmorang Riana Putria Rudi Permana	Medan Padang Bandung	1995-04-02 1996-11-23 1994-08-22	NULL Manajemen Ilmu Komputer	NULL NULL NULL		
3 rows in se	t (0.00 sec)					T	

4. PHPMyAdmin

PHPMyAdmin merupakan aplikasi untuk pengelolaan database dengan interface berbasis web (Grafis). Karena sifatnya grafis tentu saja pengelolaanya lebih mudah dan simpel daripada yang kita pelajari di atas melalui command line.

🖟 localhost / 127.0.0.1 / uni 🗙		Ahmad Suli	10 —		×	
← → C ☆ 🛈 localhost/phpm			\$ 0		:	
🏥 Apps 🤺 Berita Sepakbola Terb 🇑	BKD Kopertais Jakarta 🔲 Bootstrap 🧾 Linux 🛄 Laravel 🛄 Postgres 🛄 Ubuntu 🛄 TV Series	PHP ×	> 📙 Oth	ier bookmar	ks	
phpMyAdmin	🗕 📑 Server: 127.0.0.1 » 🍵 Database: universitas » 🔚 Table: mahasiswa			\$ ⊼	^	
<u> </u>	🖩 Browse 🥻 Structure 🗐 SQL 🔍 Search 👫 Insert 🚍 Export 🖼 In	nport 🔻 🛚	Nore		1	
Recent Favorites						
Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.) SELECT * FROM `mahasiswa`						
						+ mysql
			FITE Code J	[Reliesh]		
➡_⊜ phpmyadmin	Show all Number of rows: 25 Filter rows: Search this table S	Sort by key:	None	•	1	
+ test					1	
L New	+ Options					
🕂 📝 data_mahasiswa	$\leftarrow \overleftarrow{\top} \rightarrow \qquad \hline {\bf \nabla} nim \qquad nama \qquad tempat_lahir tanggal_lahir ji$	jurusan	ipk			
😑 🔤 mahasiswa	📄 🥜 Edit 👫 Copy 🤤 Delete 13012012 James Situmorang Medan 1995-04-02 /	NULL	NULL			
Columns	🗌 🥜 Edit 👫 Copy 🥥 Delete 14005011 Riana Putria 🛛 Padang 1996-11-23 🚺	Manajemen	NULL			
i ipk	🔲 🥜 Edit 👫 Copy 🤤 Delete 15021044 Rudi Permana 🛛 Bandung 1994-08-22 I	Ilmu Komputer	r NULL			
i jurusan						
i nama	🚹 🔲 Check all 🛛 <i>With selected: 🤌</i> Edit 👫 Copy 🤤 Delete 🚍 Export					
tempat_lahir	Show all Number of rows: 25 V Filter rows: Search this table S	ort by key:	None	•		
+ Indexes						
	Query results operations					
D Print 3: Convite cliphoard □ Event Dienlau chart □ Croate view						
	Legi tini ge cely lo exposed an export an exposit citate and					
4	Console			,		

Dari gambar di atas misalkan kita hendak mendelete database universitas, maka dengan cara mengklik menu **Home > Database** dan checklist database universitas di akhiri dengan klik **Drop**, maka terhapuslah database **universitas**.

🖟 localhost / 127.0.0.1 ph; 🗙						Ahmed Sulhi	- 0	×
← → C ☆ 🛈 localhost/phpm	nyadmin/server_databases.	php?db=				\$	0 4 🗵	:
🏥 Apps 🤺 Berita Sepakbola Terb 💧	BKD Kopertais Jakarta 📃 B	ootstrap 📙 Linux	📙 Laravel 📙 Postgr	es 📙 Ubuntu 📒	TV Series	PHP »	Other bookm	narks
phpMyAdmin	← 🗊 Server: 127.0.0.1							~ •
🏡 🗐 💿 🗊 🌼 😋	🗊 Databases 📘 S	QL 🚯 Status	User accounts	🛋 Export 🛽	Import	🌽 Settings	▼ More	
Recent Favorites	Databases							
🖶 🕞 mysql	🔒 Create database 🌘	2						
performance_schema phpmyadmin	Database name	Collatio	n	• Create)			
+ test	Database	Collation	Action					
L New	information schema	utf8_general_ci	Check privileges					
+ data_mahasiswa		latin1_swedish_ci	Check privileges					
mahasiswa Columns	performance_schema	a utf8_general_ci	Check privileges					
LIL New	phpmyadmin	utf8_bin	Check privileges					
ipk	test	latin1_swedish_ci	📺 Check privileges					
	✓ universitas	latin1_swedish_ci	Check privileges					
nim	Total: 6	latin1_swedish_ci						
tanggal_lahir	🕇 📃 Check all	With selected:	<u>Drop</u>	-				
+ Indexes	Note: Enabling the da	atabase statistics h	prop are migne cause heavy tr	affic between the w	eb server a	nd the MySQL se	rver.	
	Enable statistics							
	Console						ſ	-

Latihan:

Silakan buat database **universitas** kembali dan buat table **mahasiswa** seperti yang dilakukan melalui command line di atas.