

MODUL PRAKTIKUM

# MANAJEMEN BASIS DATA



Ika Oktavia Suzanti S.Kom M.Cs

LABORATORIUM ENERGI TERBARUKAN

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2016

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan modul praktikum ini dengan baik.

Penulis berharap semoga modul praktikum Manajemen Basis Data ini dapat bermanfaat dan dapat dipergunakan sebagaimana mestinya. Penulis pun menyadari kalau modul praktikum ini masih belum sempurna, masih banyak terdapat kesalahan di sana sini. Oleh karena itu penulis memohon kritik dan saran yang membangun guna perbaikan modul ini ke depannya.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyelesaian modul pengajaran ini.

Penulis

# MODUL I

## NORMALISASI

### A. Tujuan Praktikum

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

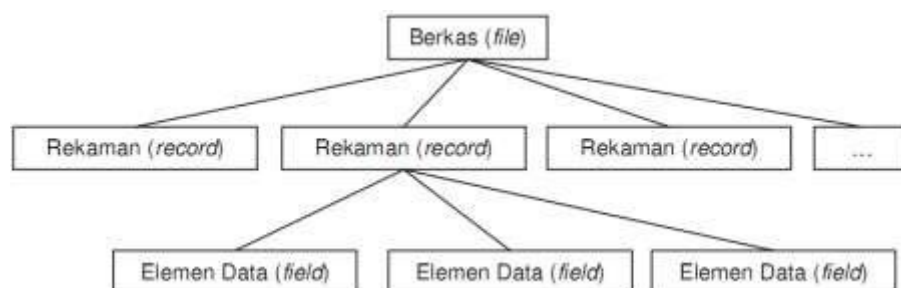
- Melakukan normalisasi untuk bukti transaksi tertentu
- Melakukan normalisasi untuk sebuah sistem hingga level ketiga

### B. Dasar Teori

#### Konsep Data, Informasi, Basis Data

Data adalah fakta berupa angka, karakter, simbol, gambar, tanda-tanda, isyarat, tulisan, suara, bunyi yang merepresentasikan keadaan sebenarnya mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya.

Apabila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*). Adapun informasi adalah hasil proses atau hasil pengolahan data meliputi hasil gabungan, hasil analisa, hasil penyimpulan, dan hasil pengolahan sistem informasi komputerisasi.



Gambar 1. Struktur Basisdata

Hubungan antar data dapat ditunjukkan dengan adanya *field*/kolom *key* dari tiap *file*/tabel yang ada. Dalam satu *file* atau *table* terdapat *record - record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari *field* yang saling

berhubungan menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*. Adapun struktur *database* ditunjukkan pada gambar 1.

Basis data mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data *oriented* dan bukan program *oriented*.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah
5. Dapat digunakan dengan cara-cara yang berbeda.

### **Istilah dalam Basis Data**

Istilah-istilah yang dipergunakan dalam sistem basis data

1. Enterprise yaitu suatu bentuk organisasi

Contoh :

Enterprise: Sekolah, Database : Nilai

Enterprise: Rumah sakit, Database : Administrasi Pasien

2. Entitas yaitu suatu obyek yang dapat dibedakan dengan objek lainnya

Contoh :

Database : Nilai, Entitas: mahasiswa, Matapelajaran

Database : Administrasi Pasien, Entitas: pasien, dokter, obat

3. Attribute/field yaitu setiap entitas mempunyai atribut atau suatu sebutan untuk mewakili suatu entitas.

Contoh :

Entity: siswa, Field = Nim, nama\_siswa, alamat, dll

Entity : nasabah, Field=Kd\_nasabah,nama\_nasabah, dll

4. Data value yaitu data aktual atau informasi yang disimpan pada tiap data elemen atau attribute.

Contoh : Atribut nama\_karyawan : sutrisno, budiman, dll

5. Record/tuple yaitu kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap.

Contoh : Record mahasiswa: nim, nm\_mhs, alamat.

6. File yaitu kumpulan record-record sejenis yang mempunyai panjang elemen sama, attribute yang sama namun berbeda-beda data valuenya

7. Kunci elemen data (key) yaitu tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas

## **Normalisasi**

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional.

### **Tujuan normalisasi :**

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data yang berulang dalam basisdata.
3. Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integrasi data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada suatu tempat.

### **Aturan Normalisasi :**

1. Hilangkan kelompok berulang, buat tabel terpisah untuk setiap himpunan atribut yang berhubungan dan tentukan kunci utama pada masing-masing tabel.
2. Hilangkan data berulang, jika sebuah atribut hanya tergantung pada sebagian kunci utama gabungan, pindahkan atribut ke tabel lain.

3. Hilangkan kolom yang tidak tergantung pada kunci, jika atribut tidak tergantung pada kunci, pindahkan atribut ke tabel lain.
4. Pisahkan relasi majemuk, tidak ada tabel yang bias mengandung dua atau lebih relasi 1:n atau n:m yang tidak berhubungan langsung.
5. Pisahkan relasi majemuk yang berhubungan secara semantik, ada batasan pada informasi yang memperbolehkan pemisahan relasi many-to-many yang berhubungan secara logis.
6. Bentuk normal optimal, sebuah model hanya dibatasi oleh fakta sederhana.
7. Bentuk normal domain-key, sebuah model harus terbebas dari semua anomaly

### **Tahapan Normalisasi :**

#### 1. Bentuk Normal Kesatu (1 NF / First Normal Form)

Bentuk Bentuk Normal Kesatu mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa "atomic value". Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (multi value). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu arti saja dan juga bukanlah pecahan kata kata sehingga artinya lain. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

#### 2. Bentuk Normal Kedua (2NF)

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya.

#### 3. Bentuk Normal Ketiga (3NF)

Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Keduadan semua atribut bukan primer tidak punya hubungan yang transitif. Artinya setiap atribut bukan kunci harus bergantung hanya pada kunci primer secara menyeluruh. Contoh pada bentuk Normal kedua di atas termasuk juga bentuk Normal Ketiga karena seluruh atribut yang ada di situ bergantung penuh pada kunci primernya.

#### 4. Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut super key.

#### **Jenis – Jenis Key Field / attribute kunci dalam database:**

##### 1. Super key

yaitu himpunan dari satu atau lebih entitas yang digunakan untuk mengidentifikasi secara unik sebuah entitas dalam entitas set.

##### 2. Candidate key

yaitu satu attribute atau satu set minimal attribute yang mengidentifikasi secara unik suatu kejadian yang spesifik dari entity.

##### 3. Primary key

yaitu satu attribute atau satu set minimal attribute yang tidak hanya mengidentifikasi secara unik suatu kejadian yang spesifik tapi juga dapat mewakili setiap kejadian dari suatu entity

##### 4. Alternate key

yaitu kunci kandidat yang tidak dipakai sebagai primary key

##### 5. Foreign key

yaitu satu attribute (atau satu set attribute) yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya

#### **C. Tugas Pendahuluan**

1. Jelaskan yang dimaksud dengan basis data dan manfaatnya !
2. Jelaskan tahapan normalisasi dengan contoh !

#### **D. Alat dan bahan**

1. Komputer
2. Modul Praktikum Manajemen Basis Data

## E. Langkah Praktikum

Normalisasi dari salah faktor sistem berikut :

### 1. Perpustakaan

**Perpustakaan SMA Indonesia Unggul**  
Jl Karya Wisata No 76  
Medan Telp (061) 379512

**Peminjaman Buku**

No. Transaksi : 000000005      Note :  
Tanggal : 14 Apr 2004  
Anggota : 00390100265-MOHAMMAD DITYA PRAKOSA

No. Kode	Judul Buku	Tanggal Pengembalian
1 037.006 01	CHIP KOMPUTER MEDIA: ATASI ERROR PADA PC AND	21 Apr 2004
2 013.067 02	BYTE : MAC ILFOX	21 Apr 2004
3 SP004-001	Bahasa dan Kekuasaan	21 Apr 2004
4 E00001-001	Manajemen Pemasaran Global	21 Apr 2004
5 621.381 5	301 RANGKAIAN ELEKTRONIKA : RANGKAIAN-RANGK	21 Apr 2004

### 2. Akademik



**Universitas Trunojoyo Madura**  
FAKULTAS TEKNIK

### KARTU HASIL STUDI

Semester: Gasal 2014 / 2015

Nama Mahasiswa : FIRDAUS AL FIDAI  
Nomor Induk Mahasiswa : 130431100057  
Angkatan : 2013  
Program Studi : TEKNIK ELEKTRO - S1 Reguler  
Pembimbing Akademik : KOKO JONI, ST., M. Eng.

No	Matakuliah		SKS	KE	Nilai	BOBOT
	Kode	Nama				
1	TKE 208	Praktikum Pengukuran Besaran Listrik	1	1	B	3.00
2	TKE 302	Matematika Teknik	3	1	B	3.00
3	TKE 113	Mikrokontroler	3	1	B+	3.50
4	TKE 301	Medan Elektomagnetik	3	1	B	3.00
5	TKE 305	Praktikum Rangkaian Listrik ( RL I dan II)	1	1	A	4.00
6	TKE 111	Jaringan dan Komunikasi Data	3	1	A	4.00
7	TKE 304	Elektronika II	3	1	D+	1.50
8	UNG 108	Pendidikan Pancasila dan Kewarganegaraan	3	1	B	3.00
9	TKE 116	Teknik Tenaga Listrik	3	1	D	1.00
<b>Total SKS</b>			<b>23</b>			





## 5. Rental

### PROGRAM RENTAL VCD "LANCAR JAYA"

No Nota PJ00021

Tanggal Peminjaman 22 - 08 - 2015  
Tanggal Pengembalian 25 - 08 - 2015

<u>Kode Barang</u>	<u>Nama Barang</u>	<u>Harga (Rp)</u>
B0002.10	Ice Age II	5.000
B0007.15	Kungfu Panda 2	7.500
B0003.7	Sherlock Holmes "Game of Shadow"	6.000
B0005.5	Ride Along 2	5.500
B0006.9	London Fall	6.000
<b>Total Sewa (Rp)</b>		30.000

Kasir KL1005

Sukirman

## F. Hasil Praktikum

Lakukan normalisasi terhadap sistem yang ada pada langkah praktikum hingga level normalisasi ketiga

## G. Pertanyaan dan Tugas

1. Lakukan normalisasi dari salah faktor sistem berikut

- Toko Pulsa
- Akademik
- Perpustakaan
- Rental
- Toko Elektronik

## MODUL II

### PERANCANGAN BASIS DATA dan PENGENALAN My SQL

#### A. Tujuan Praktikum

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

- a. Mahasiswa memahami cara instalasi MySQL.
- b. Mahasiswa memahami cara konfigurasi MySQL

#### B. Dasar Teori

##### Metodologi Perancangan Basis Data

Perancangan merupakan suatu hal yang sangat penting dalam pembuatan basis data. Permasalahan yang dihadapi pada waktu perancangan yaitu bagaimana basis data yang akan dibangun ini dapat memenuhi kebutuhan saat ini dan masa yang akan datang. Untuk itu diperlukan perancangan basis data baik secara fisik maupun secara logik. Metodologi perancangan basis data adalah kumpulan teknik terorganisasi untuk pembuatan rancangan basis data. Teknik terorganisasi ini merupakan kumpulan tahap-tahapan yang memiliki aturan-aturan terurut. Teknik yang digunakan pada perancangan basis data dibagi menjadi dua, yaitu:

- a. Perancangan basis data tingkat logik.

Perancangan basis data secara logik dimulai dengan penciptaan model konseptual dari organisasi dan seluruhnya tak bergantung rincian implementasi seperti perangkat lunak DBMS, program aplikasi, bahasa pemrograman, platform perangkat keras, dan pertimbangan fisik lainnya. Model konsep ini kemudian dipetakan menjadi model data secara logik yang telah dipengaruhi model data target basis data seperti model relasional. Dalam perancangan basis data secara logik, kita dapat melakukannya dengan cara :

- a. Menerapkan Normalisasi terhadap struktur tabel yang telah diketahui.
- b. Langsung membuat model Entity-Relationship (ER).

Model data secara logik merupakan sumber informasi perancangan fisik. Model ini menyediakan perancang suatu kendaraan untuk pertimbangan dalam merancang basis data yang efisien.

b. Perancangan basis data tingkat fisik.

Perancangan basis data secara fisik adalah proses memproduksi deskripsi implementasi basis data pada penyimpanan sekunder, mendeskripsikan struktur-struktur penyimpanan dan metode-metode pengaksesan dalam meningkatkan efektifitas pengaksesan. Pada tahap ini, perancangan fisik telah ditujukan untuk sistem DBMS tertentu. Perancangan basis data tingkat fisik sudah dikaitkan dengan platform dan perangkat lunak sistem manajemen basis data dimana basis data diimplementasikan.

### ***Data Base Management System (DBMS)***

DBMS dapat diartikan sebagai program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memodifikasi dan memperoleh data atau informasi dengan praktis dan efisien. Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu :

- a. Model data konseptual (*high level*) menyajikan konsep tentang bagaimana user memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu : *Entity* (entitas), *Attribute* (atribut) dan *Relationship* (hubungan).
- b. Model data fisik (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pemcarian rekaman data lebih efisien.
- c. Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para user mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hirarki, jaringan dan relasional.

### **MySQL**

MySQL adalah Sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, multi *user* serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan

*Shareware*. MySQL yang biasa digunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL (*General Public License*) dengan logo ikan lumba-lumba berwarna biru seperti ditunjukkan pada gambar 2.1.

MySQL Merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang perangkat lunak dan konsultan basis data bernama MySQL AB yang berada di Swedia. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson, dan Michael Monty Widenius.



Gambar 2.1 Logo My SQL

Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan sebagai *Client* maupun *server*.

*Database* MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL.

Fitur-fitur MySQL antara lain :

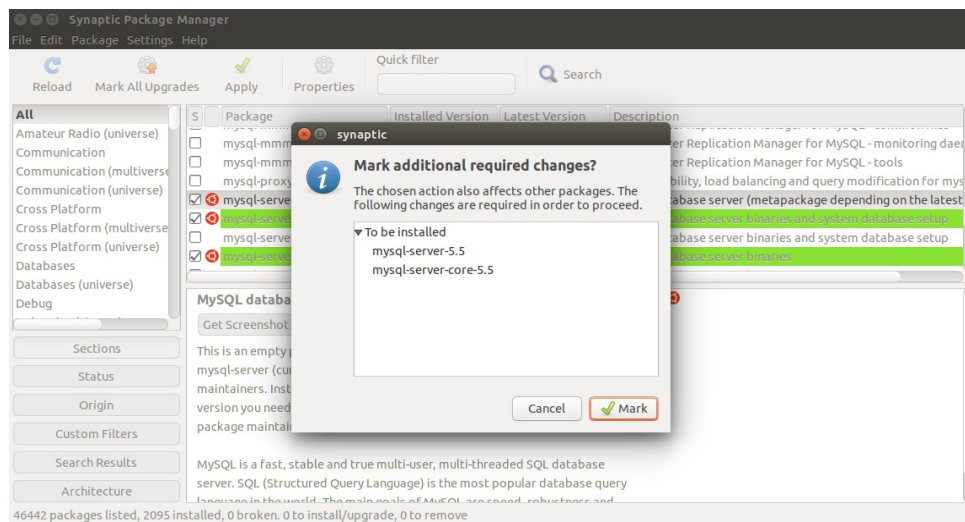
- *Relational Database System*. Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS.
- *Arsitektur Client-Server*. MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di *server*. *Client* MySQL dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.

- Mengenal perintah SQL standar. SQL merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
- Mendukung *Sub Select*. Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
- Mendukung *Views* dan *Stored Prosedured (SP)* sejak versi 5.0
- Mendukung *Triggers*. MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
- Mendukung *replication*, transaksi dan *foreign key*.

## Instalasi My SQL

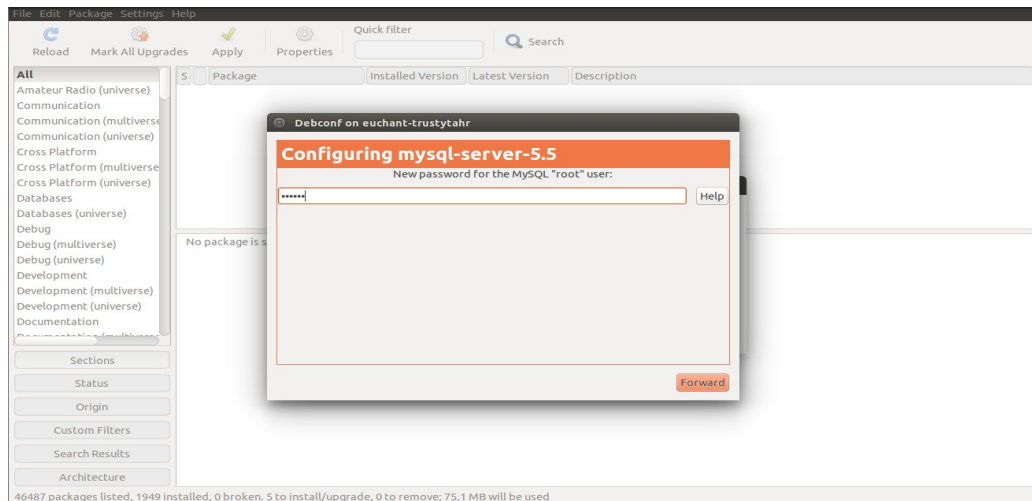
Instalasi MySQL pada ubuntu dapat menggunakan beberapa cara yaitu melalui terminal, ubuntu software center atau menggunakan synaptic package manager. Berikut panduan instalasi MySQL melalui synaptic package manager :

1. Buka aplikasi synaptic package manager lalu cari aplikasi MySQL server. Klik kanan pada kotak MySQL-server 5.xx dan pilih opsi mark for installation sehingga muncul seperti pada gambar 2.2. Pilih opsi mark, lalu klik opsi apply pada bagian menu synaptic.



Gambar 2.2 Step 1 instalasi My SQL

2. Pada saat proses instalasi berlangsung, akan muncul window untuk memberikan password untuk akun root seperti ditunjukkan pada gambar 2.3.



Gambar 2.3 Step 3 instalasi My SQL

3. Setelah proses instalasi selesai, tutup synaptic lalu buka terminal dan ketikkan perintah “**mysql -u root -p**” tekan enter dan masukkan password yang telah diberikan pada langkah 2. Apabila anda sukses menginstall MySQL maka tampilan pada terminal seperti ditunjukkan pada gambar 2.4.

```
euchant@euchant-trustyahr:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 51
Server version: 5.5.47-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> █
```

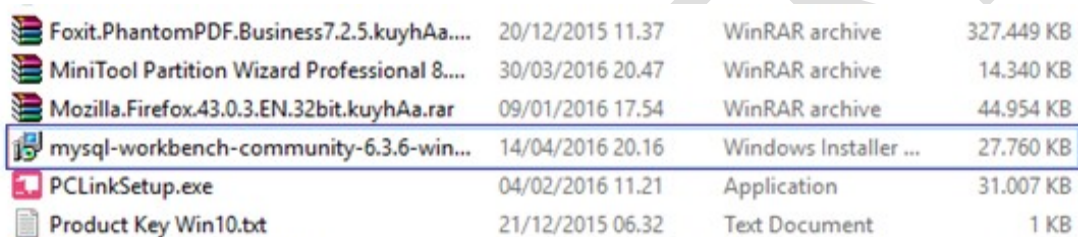
Gambar 2.3 Step 3 instalasi My SQL

## MySQL Workbench

Pada saat MySQL dibuat sebagai versi open source dari SQL sebagai database RDBMS, MySQL tidak memiliki GUI sehingga perintah untuk mendesain database didapat dari command line (perintah baris) yang harus dimasukkan satu per satu. Namun seiring dengan penggunaannya yang semakin luas MySQL banyak didukung dengan GUI dari pihak ketiga. Salah satunya adalah phpMyAdmin, yang terkenal karena kemudahannya. Saat ini MySQL merupakan produk milik Oracle, dan saat ini MySQL memiliki MySQL Workbench sebagai mesin untuk mendesain database dengan interface visual

### Instalasi MySQL Workbench

1. Pilih *installer* MySQL Workbench dan klik 2x



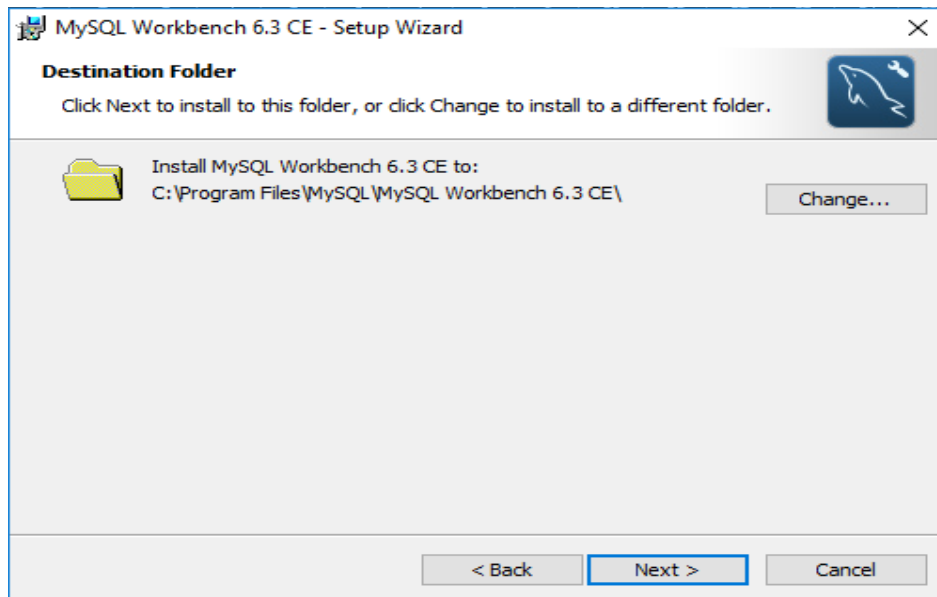
Foxit.PhantomPDF.Business7.2.5.kuyhAa....	20/12/2015 11.37	WinRAR archive	327.449 KB
MiniTool Partition Wizard Professional 8...	30/03/2016 20.47	WinRAR archive	14.340 KB
Mozilla.Firefox.43.0.3.EN.32bit.kuyhAa.rar	09/01/2016 17.54	WinRAR archive	44.954 KB
mysql-workbench-community-6.3.6-win...	14/04/2016 20.16	Windows Installer ...	27.760 KB
PCLinkSetup.exe	04/02/2016 11.21	Application	31.007 KB
Product Key Win10.txt	21/12/2015 06.32	Text Document	1 KB

2. Klik next pada *home setup wizard* MySQL Workbench

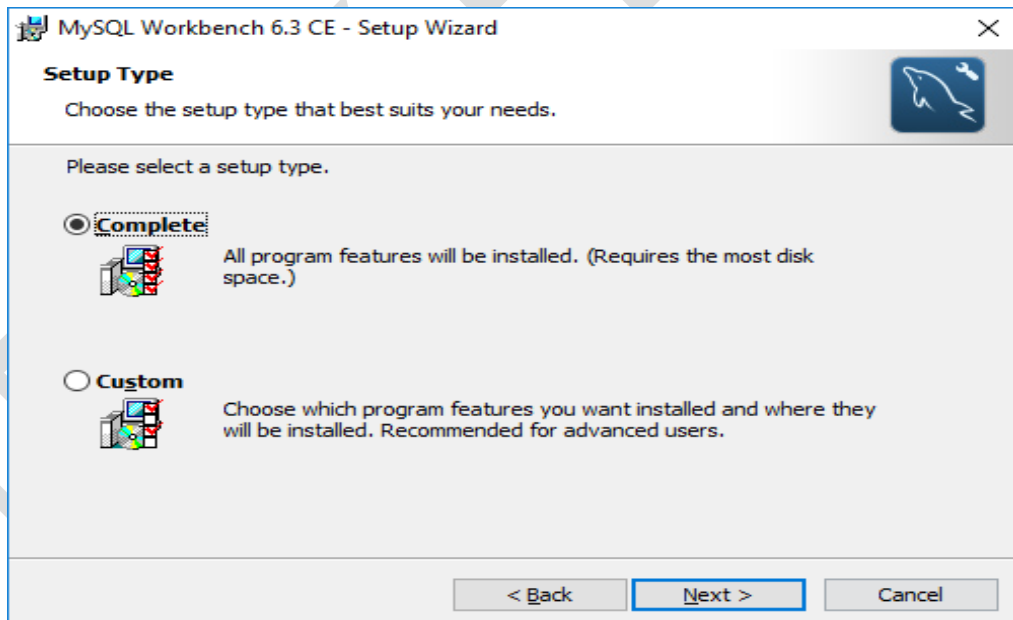




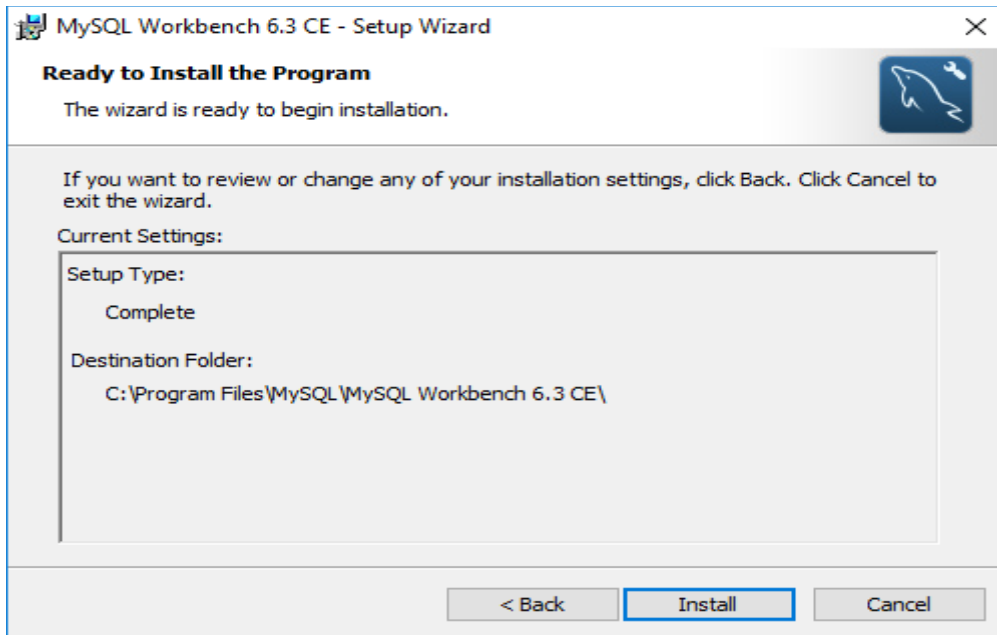
3. Pada bagian *destination folder*, klik *change* untuk merubah target folder tempat instalasi MySQL Workbench, kemudian klik *next*



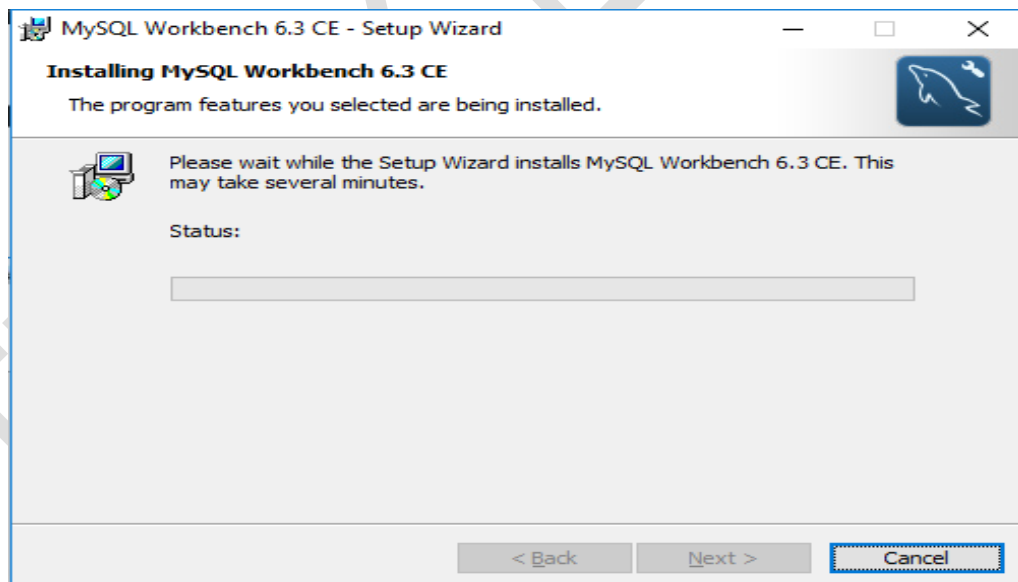
4. Pada *setup type*, pilih *complete* untuk menginstall seluruh *tools* (perangkat) pada MySQL Workbench, apabila hanya ingin menginstall beberapa *tools* saja, pilih *custom*. Setelah menentukan *setup type*, klik *next*



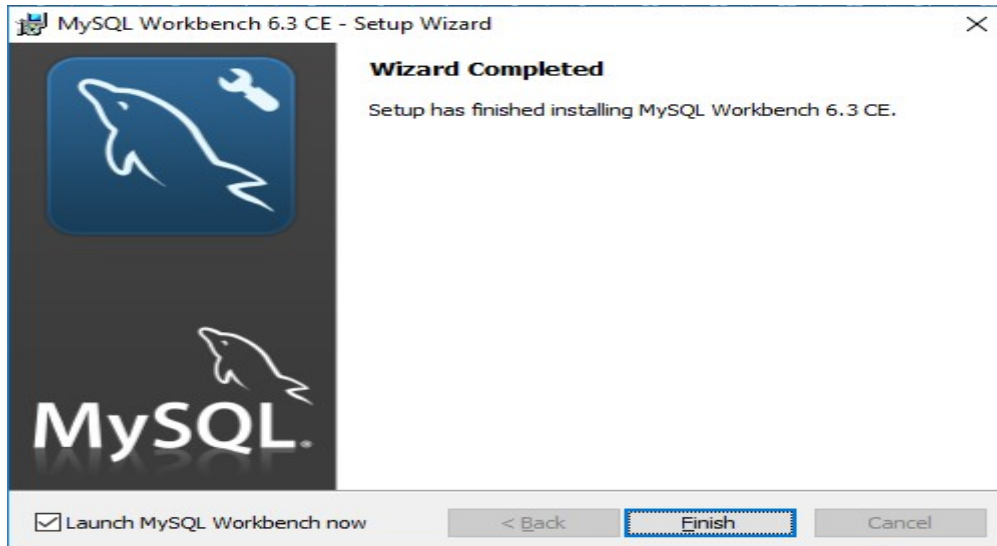
5. Klik *install* untuk melakukan instalasi MySQL Workbench pada perangkat



6. Tunggu hingga proses instalasi selesai. Apabila muncul *pop – up* persetujuan instalasi program MySQL Workbench, pilih OK untuk melanjutkan instalasi, dan *Cancel* untuk membatalkan instalasi



7. Proses instalasi sudah selesai, centang *Launch MySQL Workbench now* untuk menjalankan MySQL Workbench, dan hilangkan centang apabila tidak ingin menjalankan MySQL Workbench. Klik *finish* untuk keluar dari *setup wizard* MySQL Workbench



### C. Tugas Pendahuluan

1. Apa perbedaan instalasi MySQL pada sistem operasi Windows dan Linux?
2. Apa yang anda ketahui tentang XAMPP, Apache dan MySQL? Jelaskan!

### D. Alat dan bahan

1. Komputer yang terinstall My SQL dan My SQL Workbench
2. Modul Praktikum Manajemen Basis Data

### E. Langkah Praktikum

1. Buat CDM dari tabel hasil normalisasi dalam modul 1!
2. Lakukan instalasi MySQL menggunakan salah satu perangkat lunak yang telah dijelaskan!
3. Masuk ke terminal MySQL dan lakukan perintah berikut :

1. `SELECT VERSION (), CURRENT_DATE;`

Ketik perintah di atas dengan huruf kecil semua dan huruf besar semua

2. `SELECT VERSION (); SELECT NOW();`

3. `MySQL>SELECT`

-> USER ()

-> ,

-> CURRENT\_DATE;

4. MySQL>SELECT USER ()

Apa yang terjadi ?

Kemudian ketikkan :

-> ;

Apa yang terjadi ?

5. MySQL>SHOW DATABASES;

## **F. Hasil Praktikum**

Berikan screenshot untuk hasil dari langkah praktikum

## **G. Pertanyaan dan Tugas**

1. Perintah apa yang digunakan untuk mengaktifkan salah satu basis data maupun tabel ?
2. Apakah perintah – perintah di MySQL bersifat case sensitive?
3. Apa perbedaan pengetikan lebih dari satu perintah, yang dipisahkan dengan tanda koma (,) dan tanda baca titik koma (;)?
4. Haruskah kita mengetik perintah dalam satu baris?
5. Bagaimana untuk mengakhiri atau membatalkan suatu perintah?
6. Sebelum membuat database, langkah – langkah apa saja yang sebaiknya kita lakukan? (Berdasarkan percobaan di atas)

## MODUL III

### DATA DEFINITION LANGUAGE

#### A. Tujuan Praktikum

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

- a. Membuat database dan tabel dengan menerapkan data definition language (DDL)
- b. Mampu memodifikasi tabel

#### B. Dasar Teori

##### Data Definition Language

DDL SQL memungkinkan dilakukannya spesifikasi tidak hanya pada himpunan relasi tetapi juga informasi untuk setiap relasi, yang meliputi :

- Skema setiap relasi
- Domain nilai setiap atribut relasi
- Konstrain integritas
- Himpunan indeks untuk setiap relasi
- Sekuriti dan otorisasi setiap relasi
- Struktur penyimpanan secara fisik untuk setiap relasi dalam disk.

DDL (Data Definition Language) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut – atribut database, tabel, atribut kolom (field), maupun batasan – batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah : CREATE, ALTER, dan DROP.

CREATE merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi

(space). Nama database tidak boleh sama antar database.

ALTER merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel.

DROP merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel.

#### a. DATABASE

Syntax untuk membuat database sebagai berikut :

```
CREATE DATABASE namadatabase;
```

Berikut merupakan syntax untuk membuat database universitas :

```
CREATE DATABASE db_universitas;
```

Sedangkan syntax tambahan untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

```
SHOW DATABASES;
```

Sebelum kita membuat suatu tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan kita buat.

Untuk memilih/mengaktifkan salah satu database menggunakan syntax : USE namadatabase;

Contohnya kita akan mengaktifkan database db\_universitas : USE db\_universitas;

Selain perintah - perintah di atas, terdapat perintah yang berfungsi untuk menghapus database maupun tabel. Perintah tersebut adalah sebagai berikut :

```
DROP DATABASE namadatabase;
```

Contohnya kita akan menghapus database coba:

```
DROP DATABASE coba;
```

#### b. TABEL

Membuat Tabel

Nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel ada beberapa yang harus dideklarasikan dalam pembuatannya, yaitu antara lain meliputi : nama tabel, nama Field

(Kolom), type data dari field dan panjang data. Adapun syntax yang digunakan untuk membuat tabel secara umum adalah sebagai berikut:

```
CREATE TABLE namatabel (Field1, TypeData1, Field2, TypeData2);
```

Contoh berikut syntax untuk membuat Tabel mahasiswa:

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20),  
pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));
```

### Menampilkan Tabel

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah : SHOW TABLES;

### Menampilkan deskripsi atribut tabel

Untuk menampilkan deskripsi atribut – atribut yang terdapat pada suatu tabel dengan menggunakan perintah: DESC namatabel; misalkan DESC mahasiswa;

### Menghapus Tabel

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Syntax yang digunakan adalah:

```
DROP TABLE namaTabel;
```

Tabel yang akan dihapus harus sesuai dengan nama tabel. Misal kita akan menghapus tabel mahasiswa, maka syntax nya adalah: DROP TABLE mahasiswa;

### Mendefinisikan Null/Not Null

Null ataupun Not Null merupakan pernyataan yang digunakan untuk membuat kolom yang kita buat boleh kosong (Null) atau tidak boleh kosong (Not Null). Ketika pada kolom tabel tidak diset, maka secara default akan bernilai Null (boleh kosong). Untuk mendefinisikannya maka perintah yang digunakan adalah:

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL, nama_mhs CHAR (50) NOT NULL, login CHAR(20) NOT NULL, pass CHAR(20) NOT NULL, umur INT, ipk real, PRIMARY KEY(nim));
```

#### Mendefinisikan Nilai Default

Nilai default merupakan nilai yang diberikan secara otomatis oleh sistem untuk suatu kolom ketika terjadi penambahan baris baru, sementara nilai pada kolom tersebut tidak diisi oleh pengguna. Contohnya adalah :

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT DEFAULT 0, ipk real, PRIMARY KEY(nim));
```

#### Mendefinisikan PRIMARY KEY pada Tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan hubungan/relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut:

- Satu tabel hanya diperbolehkan memiliki satu kolom kunci.
- Nama kolom kunci tidak digunakan pada kolom lain dalam satu tabel
- Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
- Bentuk kolom kunci harus diset NOT NULL.

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini syntax yang digunakan:

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));
```

Atau

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL PRIMARY KEY, nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real);
```



atau

```
ALTER TABLE mahasiswa ADD CONSTRAINT namaconstraint PRIMARY  
KEY(namakolom);
```

Coba buat perintah untuk membuat/menambahkan kolom pada tabel mahasiswa dengan kolom alamat type datanya char(100), sex type datanya enum(„P“,„W“) dengan mendefinisikan nilai Not Null dan Primary key untuk kolom nim serta nilai default untuk kolom alamat.

Menghapus PRIMARY KEY pada Tabel

Cara 1: Jika primary key dibuat menggunakan alter table:

```
ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;
```

Cara 2: jika primary key dibuat melalui create table:

```
ALTER TABLE namatabel DROP PRIMARY KEY;
```

Menambah kolom baru pada Tabel Pada saat kita membuat tabel terkadang kita ingin menambahkan kolom lagi pada tabel yang sudah kita buat. Dalam database, hal ini dapat dilakukan dengan menggunakan perintah sebagai berikut:

```
ALTER TABLE namatabel ADD fieldbaru typedata(lebar);
```

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya.

Field baru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan.

Misal kita akan menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

```
ALTER TABLE mahasiswa ADD COLUMN telepon CHAR(15) AFTER umur;
```

Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah:

```
ALTER TABLE namatabel MODIFY COLUMN field type(lebar);
```

Contoh :

```
ALTER TABLE mahasiswa MODIFY COLUMN CHANGE COLUMN telepon(12);
```

Mengubah Nama Kolom (Field)

Perintah yang digunakan adalah:

```
ALTER TABLE namatabel namakolomlama namakolombaru typedatabaru(lebarbaru);
```

Contoh:

```
ALTER TABLE mahasiswa CHANGE COLUMN telepon phone CHAR(25);
```

Menghapus Kolom pada Tabel

Perintah yang digunakan:

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

Contoh:

```
ALTER TABLE mahasiswa DROP COLUMN phone;
```

Membuat dan Menghapus Index

Index berfungsi untuk mempercepat proses pencarian data dalam suatu tabel. Dengan adanya index pada suatu field tabel menyebabkan proses pencarian otomatis akan dilakukan terlebih dahulu ke dalam index, apabila ditemukan baru akan diambilkan data sesungguhnya dari tabel, apabila tidak ditemukan dalam index, sudah dapat dipastikan bahwa data tersebut tidak ada dalam tabel. Terdapat perintah untuk membuat dan menghapus index, tapi tidak ada perintah untuk merubah index. Perhatikan contoh berikut:

```
CREATE INDEX IDXNIM ON mahasiswa(nim);
```

Atau

```
ALTER TABLE mahasiswa ADD INDEX IDXNIM(nim);
```

Sedangkan untuk menghapus:

DROP INDEX IDXNIM ON mahasiswa;

Atau

ALTER TABLE mahasiswa DROP INDEX IDXNIM;

### Penjelasan Tipe Data

Karakter :

- CHAR: Teks dengan maksimal 255 karakter
- VARCHAR: Teks maksimal 255 karakter dan bersifat variabel
- TEXT: Teks dengan panjang maksimal 65535

Bilangan :

- TINYINT: Bilangan 1 byte
- SMALLINT: Bilangan 2 byte
- INT atau INTEGER Bilangan 4 byte
- BIGINT: Bilangan 8 byte
- FLOAT: Bilangan pecahan (4 byte)
- DOUBLE atau REAL: Bilangan pecahan (8 byte)
- DECIMAL(M, D) atau NUMERIC(M, D): Bilangan pecahan

Lain-Lain :

- DATE: Tanggal
- DATETIME: Waktu (tanggal dan jam)
- TIME : Jam
- ENUM('nilai1', 'nilai2', ...): Nilai enumerasi
- BOOLEAN: tipe benar atau salah

Kata Tambahan :

- NOT NULL : Tidak boleh kosong

- PRIMARY KEY: Kunci primer
- AUTO\_INCREMENT: Nilai naik secara otomatis tanpa diisi

### C. Tugas Pendahuluan

1. Jelaskan yang dimaksud structured query language !
2. Sebutkan perbedaan antara DDL dan DML !
3. Jelaskan yang dimaksud dengan data definition language berikut kegunaannya !
4. Sebutkan dan tuliskan sintaks dari perintah yang mencakup data definition language !

### D. Alat dan bahan

1. Komputer yang terinstall My SQL
2. Modul Praktikum Manajemen Basis Data

### E. Langkah Praktikum

Jalankan MySql Query Browser anda kemudian sebagai latihan praktikum buatlah :

Database : pegawai dengan table berikut :

Table pribadi : (NIP,Nama,Tgl\_Lahir,Sex,Alamat,Kota)

Table pekerjaan : (NIP,Tgl Masuk, Kode\_Bag, Gaji)

Tabel Bagian : (Kode\_Bag, Nama\_Bag)

Sebagai contoh :

```
CREATE TABLE Pribadi (Nip CHAR(6) NOT NULL PRIMARY KEY, Nama
VARCHAR(35) NOT NULL, Tgl_lahir DATE, Sex ENUM('P','W'), Alamat VARCHAR(35),
Kota VARCHAR(15));
```

Perintah Detail :

Buatlah tabel Bagian dengan struktur seperti berikut:

- Kode\_Bag, CHAR, 1 karakter, primary key
- Nama\_Bag, VARCHAR, 20 karakter, harus diisi

Buatlah tabel Pekerjaan dengan struktur seperti berikut:

- NIP, CHAR, 6 karakter, primary key
- Tgl\_Masuk, DATE
- Kode\_Bag, 1 karakter, harus diisi
- Gaji, bilangan

Mengganti nama field :

- Perintah yang digunakan adalah ALTER TABLE

Contoh: ALTER TABLE Pribadi CHANGE sex kelamin ENUM('P','W');

Mengganti Ukuran dan tipe data :

- Perintah yang digunakan adalah ALTER TABLE

Contoh: ALTER TABLE Pribadi CHANGE kota kota VARCHAR(20);

Cek struktur tabel setelah Anda melakukan perintah di atas

Menambahkan default :

- DEFAULT pada struktur tabel digunakan untuk memberikan nilai bawaan pada suatu field kalau nilai bersangkutan tidak dimasukkan

Contoh:

ALTER TABLE Pribadi CHANGE kelamin kelamin ENUM('P','W') DEFAULT 'P';

## **F. Hasil Praktikum**

Berikan screenshot untuk hasil praktikum diatas :

1. Create masing-masing table yang ada pada soal diatas
2. Gunakan perintah data definition language yang lain dalam MySql untuk latihan studi kasus di modul 2

## **G. Pertanyaan dan Tugas**

1. Dari rancangan database dan tabel pada modul 2 yang telah dibuat, lakukan modifikasi dengan perintah data definition languagenya dengan menggunakan MySql !

## MODUL IV

### DATA MANIPULATION LANGUAGE

#### A. Tujuan Praktikum

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

- a. Menerapkan perintah – perintah SQL seperti INSERT, DELETE, UPDATE pada database
- b. Menerapkan SELECT untuk memanipulasi data dalam database untuk satu tabel

#### B. Dasar Teori

##### Data Manipulation Language

Merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi data dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa :

1. penyisipan/penambahan data baru ke suatu basis data
2. penghapusan data dari suatu basis data
3. pengubahan data di suatu basis data

Pada level fisik, kita harus mendefinisikan algoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi, yang dipentingkan bukan hanya efisiensi akses, tetapi juga efisiensi interaksi manusia (pemakai) dengan sistem (kemudahan permintaan akses).

*Data Manipulation Language* (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 jenis DML, yaitu :

1. Prosedural, yang mensyaratkan agar pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.
2. Nonprosedural, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

Perintah yang termasuk dalam DML adalah: INSERT, DELETE, UPDATE, dan SELECT

#### a. INSERT

Perintah INSERT bertujuan untuk menambahkan record data pada suatu tabel. Terdapat beberapa cara untuk menambahkan record, yaitu:

Cara 1: Menambahkan record dengan mengisi data pada setiap kolom:

```
INSERT INTO namatabel VALUES (nilai1, nilai2, nilai-n);
```

Cara 2: menambahkan baris dengan hanya mengisi pada kolom tertentu:

```
INSERT INTO namatabel (field1, field2, field-n) VALUES (nilai1, nilai2, nilai-n);
```

Ket : Jika data bertipe string, date, atau time (contoh : didi, basis data, 1984-03-18) maka pemberian nilainya diapit menggunakan tanda petik tunggal ('Didi') atau petik ganda ("Basis Data"). Jika data bertipe numerik (29, 4) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

#### b. DELETE

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu maupun seluruh baris. Syntax yang digunakan:

```
DELETE FROM namatabel WHERE [kondisi];
```

Perintah dalam tanda [] bersifat pilihan/opsional untuk menghapus suatu baris dengan kondisi tertentu yang dipersyaratkan. Contoh perintah untuk menghapus suatu baris dalam tabel dengan kondisi persyaratan tertentu :

```
DELETE FROM mahasiswa WHERE nim 13120070;
```

#### c. UPDATE

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu tabel. Syntax yang digunakan secara umum adalah sebagai berikut:

```
UPDATE namatabel SET field1=nilai1, field2=nilai2 [WHERE kondisi];
```

Perintah dalam tanda [] bersifat pilihan/opsional untuk mengubah suatu baris dengan kondisi tertentu yang dipersyaratkan.

#### d. SELECT

Perintah SELECT digunakan untuk menampilkan isi dari suatu tabel yang dapat dihubungkan dengan beberapa tabel lainnya.

Menampilkan data semua kolom dengan menggunakan asterisk (\*):

```
SELECT * FROM namatabel;
```

Menampilkan data untuk field/kolom tertentu:

```
SELECT filed1, filed2, filed-n FROM namatabel;
```

Menampilkan data dengan kondisi tertentu menggunakan klausa WHERE:

```
SELECT * FROM namatabel WHERE kondisi;
```

### C. Tugas Pendahuluan

1. Sebutkan dan jelaskan jenis *data manipulation language* (DML) !
2. Sebutkan perintah-perintah dalam DML, jelaskan kegunaannya beserta sintaksnya!

### D. Alat dan bahan

1. Komputer yang terinstall My SQL
2. Modul Praktikum Manajemen Basis Data

### E. Langkah Praktikum

#### Penambahan data dengan MySql

Penambahan data dilakukan dengan menggunakan pernyataan INSERT

Bentuk dasar :

```
INSERT INTO nama_tabel (nama_field, nama_field, ...) VALUES (nilai, nilai, ...);
```



Contoh:

```
INSERT INTO Pribadi(Nip, Nama, Tgl_lahir, Sex, Alamat, Kota) VALUES ('12345', 'A. Hamzah Sianturi', '1965/12/23', 'P', 'Jl. Kudus 1', 'Yogya');
```

Penambahan data dengan field tertentu :

Contoh tanpa mengisi Tanggal lahir, alamat dan kota :

```
INSERT INTO Pribadi(Nip, Nama, Sex) VALUES ('12355', 'Andini', 'W');
```

### **Melihat Isi Tabel**

Perintah SELECT

Bentuk Umum:

```
SELECT * FROM Nama_Tabel
```

Contoh:

```
SELECT * FROM Pribadi
```

### **Pengubahan Data**

- Perintah yang digunakan adalah UPDATE
- Bentuk dasar:

```
UPDATE nama_tabel SET nama_field = nilai, nama_field = nilai, ... WHERE nama_field = nilai
```

Contoh mengubah Udin menjadi Udinsah:

```
UPDATE Pribadi SET Nama= 'Udinsah' WHERE NIP = '12346';
```

Ujilah dengan SELECT untuk melihat hasil perubahan.

## Penghapusan Data

Perintah DELETE

Bentuk dasar:

```
DELETE FROM nama_tabel WHERE nama_field = nilai
```

Contoh:

```
DELETE FROM Pribadi WHERE Nip = '12355';
```

Ujilah dengan SELECT untuk melihat efek perintah di atas

Dari kasus basis data pegawai pada modul 2 :

1. Untuk tabel bagian isikan data sebagai berikut:

- 1 untuk EDP
- 2 untuk Pemasaran
- 3 untuk Produksi
- 4 untuk SDM
- 5 untuk Akunting

2. Masukkan data berikut ke Tabel Pribadi:

- NIP: 430511
- Nama: Asti Damayanti
- Tanggal Lahir: 2 Februari 1983
- Jenis kelamin: Wanita

3. Ubahlah Tanggal lahir Asti Damayanti menjadi 23 Pebruari 1993 dan alamat menjadi Jl Salak 1 No 46 Kota Bangkalan

4. Hapuslah record yang berisi data Asti Damayanti dengan menyebutkan namanya (bukan NIP)

5. Menambahkan kolom Bisa\_Bhs\_Asing BOOLEAN DEFAULT FALSE pada tabel pribadi

6. Hapus kolom Bisa\_Bhs\_Asing pada tabel Pribadi
7. Masukkan 5 baris data untuk tabel pribadi dan pekerjaan (data harus variatif)
8. Tampilkan data semua pegawai dan tunjukkan umur masing-masing!

### **F. Hasil Praktikum**

Berikan hasil screenshot untuk langkah praktikum diatas

### **G. Pertanyaan dan Tugas**

1. Dari topik pembuatan basis data yang telah anda pilih, buatlah masing-masing perintah DML berikut :

Masukkan min 10 baris data

2. Tambahkan operasi update dan delete sesuai dengan perintah asisten praktikum

## MODUL V

### SELECT

#### A. Tujuan Praktikum

Setelah menyelesaikan modul ini, diharapkan mahasiswa dapat:

- Mencari dan menampilkan data dengan perintah SELECT
- Mengkombinasikan perintah SELECT dengan perintah lainnya menggunakan klausa WHERE untuk memanipulasi data dengan syarat atau kondisi tertentu, SORT BY, DISTINCT, BETWEEN, AND & OR

#### B. Dasar Teori

##### Struktur Dasar Select

*Select A1,A2,.....An*

*From r1,r2,....rn*

*Where P*

Keterangan :

- *A* is represent attributes
  - *r* is represent relations
  - *P* is a predicate.
- Klausa **Select** berhubungan dengan operasi proyeksi dari aljabar relasional.
  - Operasi ini digunakan untuk mendaftar semua atribut yang diinginkan sebagai hasil suatu *query*.
  - Klausa **From** berhubungan dengan operasi *Cartesian-product* dari aljabar relasional.
  - Operasi ini mencatat semua relasi yang di"scan" dalam evaluasi suatu *query*.
  - Klausa **Where** berhubungan dengan operasi predikat seleksi dari aljabar relasional.
  - Operasi ini terdiri dari sebuah predikat yang menyangkut atribut-atribut dari relasi yang muncul dalam klausa *from*.

Contoh :

SELECT Nama FROM Pribadi;

SELECT Nip, Nama FROM Pribadi;

SELECT Nama, Alamat, Kota FROM Pribadi;

SELECT Nama FROM Pribadi WHERE Nip = 12345 ;

SELECT Nama FROM Pribadi WHERE Kelamin = W ;

SELECT Nama FROM Pribadi WHERE Bisa\_Bhs\_asing = TRUE;

### Operator Relasional

= Sama dengan

> Lebih dari

< Kurang dari

>= Lebih dari atau sama dengan

<= Kurang dari atau sama dengan

<> Tidak sama dengan

AND digunakan sebagai kriteria DAN

OR digunakan sebagai kriteria ATAU

NOT menyatakan TIDAK atau BUKAN

BETWEEN untuk menyatakan suatu jangkauan

IN digunakan untuk melakukan pencocokan terhadap suatu daftar nilai

LIKE digunakan untuk mencari data menurut awalan, akhiran, atau penggalan suatu kata atau suku-kata

Tanda seperti % dinamakan wildcard . Wildcard % Cocok dengan nol, satu, atau sejumlah karakter apa saja

### **C. Tugas Pendahuluan**

1. Jelaskan dengan contoh 10 fungsi pada SQL !

### **D. Alat dan bahan**

1. Komputer yang terinstall My SQL
2. Modul Praktikum Manajemen Basis Data

### **E. Langkah Praktikum**

Dengan menggunakan tool My SQL lakukan perintah berikut :

1. Tampilkan semua field yang ada pada tabel Pekerja
2. Siapa saja pegawai pria yang tidak tinggal di Surabaya
3. Siapa saja pegawai wanita yang NIP-nya di bawah 430721
4. Siapa saja yang tidak lahir pada tahun 1995
5. Sebutkan NIP, Nama dan Kota Asal pegawai yang tinggal di Surabaya, Jakarta, atau Bangkalan
6. Sebutkan NIP, Nama dan Kota Asal pegawai yang tidak tinggal di Surabaya, Jakarta, atau Bangkalan
7. Siapa saja yang namanya berawalan A, B, C, D, E, F, G, H, I, atau J
8. Tampilkan Nama pegawai yang tinggal di Surabaya dan lahir sesudah tahun 1995
9. Tampilkan NIP, Nama dan jumlah gaji dari semua pegawai pada tabel pribadi
10. Tampilkan NIP, Nama dan jumlah gaji dari semua pegawai pada tabel pribadi dari jumlah gaji terbesar ke terkecil
11. Hitunglah jumlah semua pegawai
12. Tampilkan NIP, Nama Pegawai, Nama Bagian dan Kota Asal Pegawai yang ada di bagian Produksi
13. Tampilkan NIP, Nama Pegawai, Tahun Masuk dan Kota Asal pegawai yang masuk setelah tahun 2015
14. Tampilkan NIP, Nama Pegawai, Nama Bagian, Tahun Masuk dan Kota Asal Pegawai yang ada di bagian Produksi dan masuk setelah tahun 2015
15. Tampilkan NIP, Nama Pegawai, Nama Bagian, Tahun Masuk dan Umur Pegawai yang tidak di bagian Produksi dan masuk sebelum tahun 2015

## **F. Hasil Praktikum**

Berikan screenshot untuk hasil dari langkah praktikum diatas

## **G. Pertanyaan dan Tugas**

1. Lakukan query yang diberikan oleh asisten praktikum sesuai dengan database pada modul 4

TEKNIK ELEKTRO