

**MODUL PEMBELAJARAN**

**BASIS DATA I**



**Dosen Pengampu :**

**Bambang Kurniawan, A.Md, S.Kom, M.Kom**

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS HANG TUAH PEKANBARU  
2022/2023**

# BASIS DATA

## PENGENALAN BASIS DATA

### 1. Defenisi Basis Data

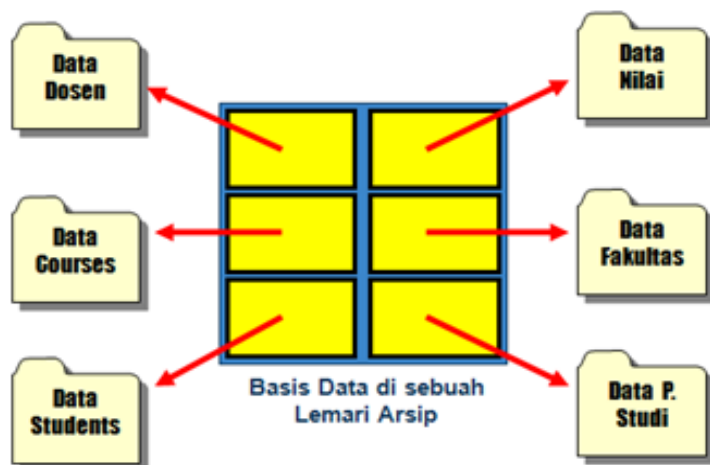
Basis data dapat dibayangkan sebuah lemari arsip dan untuk pengelolaan lemari arsip adalah seorang pegawai memberikan map untuk beberapa file yang akan dimasukkan kedalam lemari arsip tersebut, setelah itu map tersebut diberikan penomoran atau penamaan sehingga memudahkan pengguna mencari map tersebut apabila dibutuhkan suatu hari nanti.

Dengan adanya pengarsipan yang masih manual tersebut maka muncullah ide untuk membuat sebuah basis data.

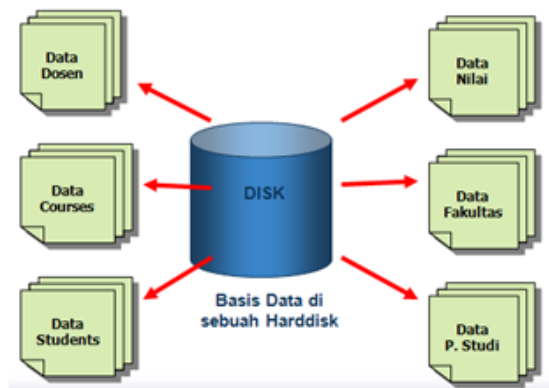
Basis data terdiri dari 2 suku kata yaitu :

- a. **basis** adalah gudang/ markas tempat berkumpul.
- b. **Data** adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, dosen, pembeli) barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi atau kombinasi ny.

**Basis data** adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.



Gambar disamping merupakan tempat penyimpanan data yang masih menggunakan lemari arsip. Yang mena lemari arsip tersebut menyimpan data-data yang dimasukkan kedalam map-map yang telah diberikan nama atau nomor.



sedangkan gambar disamping ini adalah gambar basis data yang mana penyimpanannya sudah menggunakan media elektronik seperti cakram magnetis. Dapat dilihat dari kedua penyimpanan tersebut mempunyai prinsip dan tujuan yang sama yaitu prinsip umumnya adalah pengaturan data sedangkan tujuannya adalah kemudahan dan kecepatan.

## 2. DBMS (*Database Management Systems*)

Pengelolaan basis data tidak secara langsung dilakukan oleh pengguna akan tetapi ditangani oleh sebuah software khusus. Software ini disebut dengan Database Management System (DBMS). Perangkat lunak ini lah yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali apabila diperlukan. Selain itu DBMS ini juga yang akan mengatur mekanisme pengamanan data, pemakaian data secara bersama, keakuratan/ konsistensi data dan sebagainya.

Perangkat lunak yang termasuk kedalam DBMS adalah sebagai berikut :

1. dBase
2. FoxBase
3. Microsoft access
4. MS-SQL Server
5. Oracle Database
6. MySQL dan lainnya

## 3. Penerapan Basis Data

Hampir disemua aspek pemanfaatan komputer didalam sebuah organisasi/ perusahaan akan selalu berhubungan dengan basis data. Basis data adalah salah satu komponen utama dalam setiap sistem informasi, karena tidak ada sistem informasi yang bisa dibuat atau dijalankan tanpa adanya basis data.

Lebih nyatanya basis data sering digunakan diantaranya :

- a. **Kepegawaian**, basis data sering digunakan oleh perusahaan yang memiliki jumlah pegawai yang banyak
- b. **Pergudangan (*inventory*)**, sering digunakan oleh pabrik, grosir, apotik dan lainnya
- c. **Akutansi**, digunakan oleh berbagai perusahaan
- d. **Reservasi**, digunakan oleh sebuah hotel, pesawat, kereta api dan lainnya
- e. **Layanan Pelanggan (*Customer care*)**, digunakan oleh perusahaan yang berhubungan dengan banyak pelanggan seperti rumah sakit, perbankan, konsultan dan lainnya.
- f. **Dan lain-lain**

#### 4. Kelebihan dan kekurangan sistem basis data

Kelebihan basis data adalah sebagai berikut :

- a. Kecepatan dan kemudahan (*Speed*)
- b. Efisiensi ruang penyimpanan (*Space*)
- c. Keakuratan data (*Accuracy*)
- d. Ketersediaan (*Availability*)
- e. Kelengkapan (*Completeness*)
- f. Keamanan (*Security*)
- g. Kebersamaan Pemakai (*shareability*)

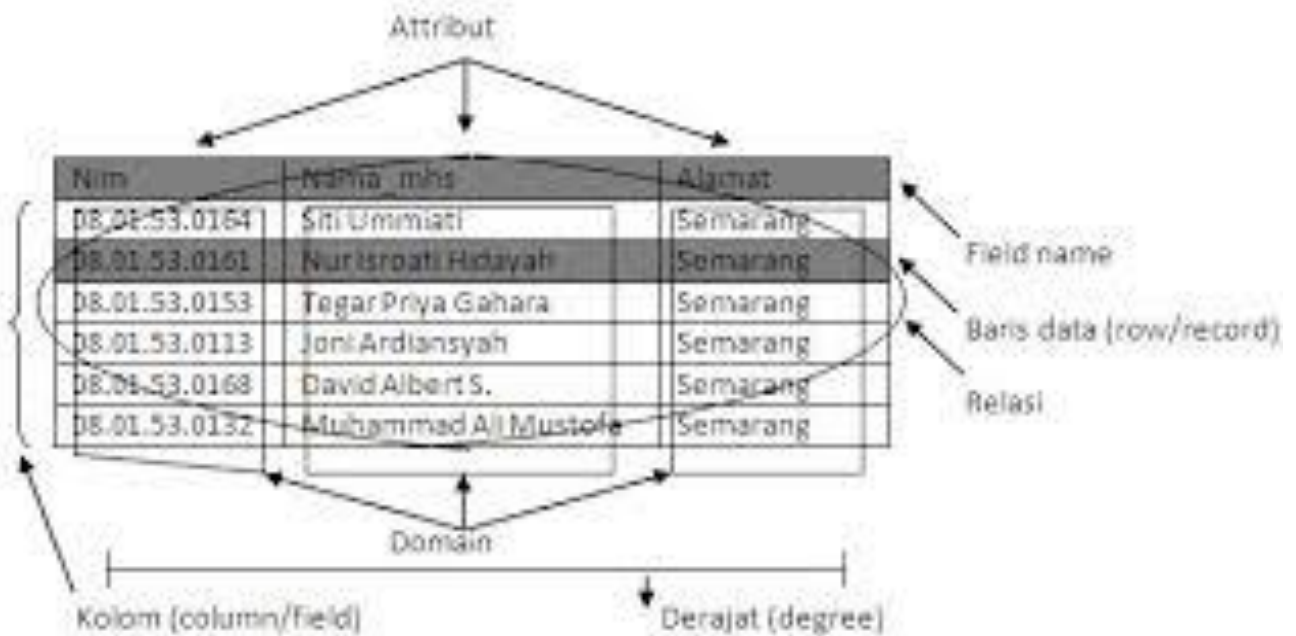
Kekurangan sistem basis data :

1. Dapat *crash* atau mati sehingga tidak menyimpan pekerjaan terakhir kalian.
2. Sistem dapat disalahgunakan karena semua catatan terbuka untuk siapa pun yang memiliki akses terhadapnya.
3. Data-data dapat dengan mudah disalin dan dilihat oleh orang lain, tanpa ada yang tahu.
4. Sistem ini biasanya terbuka untuk hacker.
5. Membutuhkan banyak perlindungan untuk mencegah agar data tidak dicuri atau dilihat oleh orang-orang yang tidak terotorisasi (tindakan perlindungan data)
6. Virus bisa melumpuhkan seluruh database, atau mengubah hal-hal.
7. Bahaya kecurangan komputer jika tingkat kontrol dan keamanan baik internal maupun eksternal tidak benar dipertahankan.

5. Beberapa istilah yang digunakan pada basis data

- a. Enterprise
- b. Entitas
- c. Atribut (elemen data)
- d. Nilai data (data value)
- e. Kunci elemen (key data elemen)
- f. Record data

Untuk lebih jelasnya bisa dilihat dari gambar dibawah ini :



# **BASIS DATA**

## **STRUKTUR DAN KOMPONEN-KOMPONEN DBMS**

### **1. DBMS (*Database Management Systems*)**

Pengelolaan basis data tidak secara langsung dilakukan oleh pengguna akan tetapi ditangani oleh sebuah software khusus. Software ini disebut dengan Database Management System (DBMS). Perangkat lunak ini lah yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali apabila diperlukan. Selain itu DBMS ini juga yang akan mengatur mekanisme pengamanan data, pemakaian data secara bersama, keakuratan/ konsistensi data dan sebagainya.

Perangkat lunak yang termasuk kedalam DBMS adalah sebagai berikut :

1. dBase
2. FoxBase
3. Microsoft access
4. MS-SQL Server
5. Oracle Database
6. MySQL dan lainnya

### **2. Pengertian DBMS (*Database Management Systems*)**

Pengertian DBMS menurut para ahli adalah sebagai berikut :

#### 1. Menurut C.J. Date

DBMS merupakan software yang menhandel seluruh akses pada database dalam melayani kebutuhan user

#### 2. Menurut S. Attre

DBMS merupakan software yang menhandel seluruh akses pada database dalam melayani kebutuhan user

#### 3. Menurut Gordon C. Everest

DBMS merupakan software, hardware, firmware serta juga prosedur yang manage database. Firmware merupakan software yang telah menjadi sebuah modul yang tertanam pada hardware (ROM)

### **3. Tujuan DBMS (*Database Management Systems*)**

Tujuan dari DBMS antara lain :

1. Bisa dipakai atau digantikan secara bersama-sama
2. Kecepatan serta kemudahan ketika mengakses data
3. Menghemat ruang penyimpanan data
4. Untuk keamanan data
5. Menghilangkan duplikasi dan inkonsistensi data
6. Menangani data dalam jumlah yang banyak atau besar

### **4. Fungsi DBMS (*Database Management Systems*)**

Fungsi dari DBMS ialah sebagai berikut :

1. Mendefenisikan data dan kaitannya
2. Mengubah data atau memanipulasi data
3. Keamanan dan integritas data
4. Recovery dan kurasi data
5. Data dictionary
6. Untuk performance kerja

### **5. Keuntungan dan kerugian DBMS**

Keuntungan DBMS menurut Connolly dan Begg adalah :

1. Mengontrol pengulangan data
2. Data yang konsisten
3. Bisa mendapatkan informasi yang lebih banyak dan jumlah data yang sama
4. Pemakaian data secara bersama-sama
5. Meningkatkan integritas data
6. Meningkatkan keamanan
7. Penetapan standarisasi
8. Meningkatkan akses dan resporn dari data
9. Meningkatkan produktifitas
10. Meningkatkan pemeliharaan data melalui data independen
11. Meningkatkan layanan backup dan recovery

Kekurangan DBMS menurut Connolly dan Begg adalah :

1. Kompleksitas
2. Ukuran yang besar karena kompleksitas yang terjadi
3. Biaya konversi yang diperlukan dalam pemakaian DBMS yang baru
4. Biaya DBMS yang bervariasi tergantung pada lingkungan serta fungsi yang disediakan
5. Tambahan biaya untuk hardware
6. Performa kinerja menurun
7. Tingkat kegagalan yang lebih besar

## **6. Karakteristik Basis Data**

Basis data memiliki beberapa karakteristik sebagaimana berikut.

### *1) Self-describing*

Basis data tidak hanya berisi data saja, tetapi lengkap dengan definisi dari data itu sendiri. Definisi data disimpan dalam catalog sistem (meta-data) yang berisi struktur setiap file, tipe dan format penyimpanan data, serta constraint dari data. Software DBMS dapat mengekstraksi data dari catalog dan menggunakannya.

### *2) Isolation*

Program pengaksesan DBMS ditulis secara terpisah dengan file-file yang bersifat spesifik seperti abstraksi data. Sehingga program dan data berdiri sendiri.

### *3) Multi-view*

Memungkinkan user yang berbeda untuk mendapatkan perspektif (view) basis data yang berbeda.

### *4) Sharing data dan pemrosesan transaksi multi-user*

Memungkinkan sejumlah user mengakses data secara bersamaan. Untuk itu DBMS perlu menyediakan 'concurrency control software' sehingga data yang diakses valid.

## **7. Orang yang berperan dalam basis data**



Ada empat kelompok orang yang berperan secara langsung pada basisdata, yaitu :

1) Database Administrator (DBA)

Yaitu orang yang bertanggung jawab terhadap administrasi penggunaan sumber daya basis data (DBMS dan basis data). DBA bertugas :

- Mengatur otorisasi akses terhadap basis data
- Memonitor penggunaan basis data
- Melayani permintaan software dan hardware

2) Database Designer

Yaitu orang yang bertanggung jawab dalam perancangan basis data (dalam memenuhi permintaan user). Database designer bertugas :

- Mengidentifikasi data yang akan disimpan dalam basis data.
- Memilih struktur yang sesuai dalam menyajikan dan menyimpan data.

3) *End User*

Orang-orang yang pekerjaannya memerlukan akses terhadap basis data untuk keperluan membaca dan merubah data, serta membaca laporan dari data.

4) *System Analysts*

Yaitu orang-orang yang bertugas mendefinisikan kebutuhan-kebutuhan end user dan mengembangkan spesifikasi untuk transaksi-transaksi yang memenuhi keinginannya

5) *Application Programmers*

Yaitu orang-orang yang bertugas mengimplementasikan spesifikasi yang dibuat oleh system analyst menjadi program.

Latihan :

Sebuah Kedai Kopi memiliki bahan baku seperti Kopi arabika, kopi robusta, kopi liberika, susu kental manis, gula pasir, gula merah, dan sirup. Setiap bahan baku tersebut

berjumlah 5 kg, dan akan habis dalam waktu 3 hari. Dalam mendata setiap bahan baku tersebut setiap bulan harus dicatat di buku besar oleh pegawainya.

Apakah kedai kopi tersebut sudah harus memiliki sebuah database untuk menyimpan data bahan bakunya? Jika tidak silahkan tuliskan alasannya, jika iya silahkan dirancang database yang tepat untuk kedai kopi tersebut

# **BASIS DATA**

## **PENGGUNA (USER) BASIS DATA**

### 1. Tingkatan pengguna basis data

a. **Database manager** => orang yang bertugas mengelola database dengan memberikan batasan tiap pengguna database. Adapun tugas dari database manager adalah :

1. Berinteraksi dengan file manager
2. Menjaga integritas dan keamanan data
3. Backup dan recovery
4. Mengendalikan konsistensi dan konkurensi data

b. **Database administrator** => bertugas untuk mengelola isi dari database, dengan mengubah isi, lokasi dan organisasi database tanpa mengganggu program aplikasi yang ada. dan bertanggung jawab terhadap kebutuhan hardware dan software, biasanya database administrator dibantu oleh staf admin. Adapun tugasnya adalah :

1. Mendefenisikan skema database
2. Mendefenisikan metode untuk mengakses data
3. Struktur penyimpanan
4. Modifikasi fisik penyimpanan dan skema data
5. Memberikan otorisasi hak akses data (granting)
6. Menentukan integritas constraints

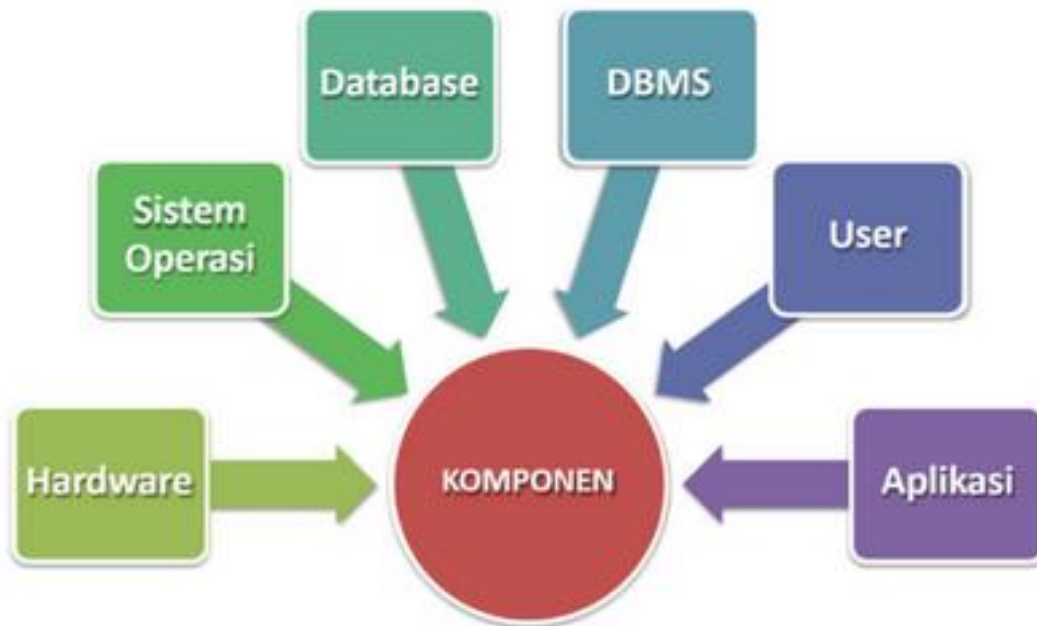
c. **Database user** => orang yang menggunakan database untuk mengisi atau hanya melihat saja, terdiri dari :

1. *Applications programmer*
2. *Specialized users*
3. *Dan lain-lain*

# BASIS DATA

## KONSEP DAN ARSITEKTUR SISTEM BASIS DATA

A. Komponen penting dalam sistem basis data adalah sebagai berikut :



Gambar 1. Komponen basis data

### 1. Hardware

Adalah perangkat keras berupa komputer dengan media penyimpanan yang digunakan untuk menyimpan data karena pada umumnya basis data memiliki ukuran yang besar.

### 2. Sistem operasi

Adalah program yang mengaktifkan dan mengfungsikan sistem komputer, mengendalikan seluruh sumber daya dalam komputer, dan melakukan operasi dasar dalam komputer meliputi input, proses dan output.

### **3. Database**

Adalah inti dari sistem basis data. Basis data menyimpan data serta struktur sistem basis data baik entitas maupun objek-objek secara detail

### **4. Database managemen System (DBMS)**

Adalah aplikasi yang dipergunakan untuk mengelola database.

Contohnya :

MySQL, SQLServer, Microsoft Access, Oracle dan lainnya

### **5. User**

Adalah yang akan menggunakan database. User terbagi atas beberapa jenis antara lain : database administrator, database designer, end-user, dan software engineering.

- a. **Database administrator (DBA)** adalah seseorang yang akan mengelola database seperti melakukan analisa sistem dan menjamin integritas data.
- b. **Database designer** adalah seseorang yang merancang database
- c. **Software engineering** adalah seorang yang mengimplementasikan database kedalam sebuah sistem

**End- user** dapat dibagi menjadi beberapa jenis yaitu :

- a. **User mahir/ casual and user** adalah user yang berintegrasi dengan sistem dan menyetakan dengan query dengan bahasa query yang telah disediakan oleh DBMS
- b. **Naïve/ parametric end user** adalah user yang mengerjakan terkait update data seperti pada bank teller, bagian reservasi dan lainnya sebagainya.
- c. **Sophisticated end user** umumnya adalah user yang melengkapi kebutuhan database seperti data analyst, engineer, bisnis analyst dan sebagainya.
- d. **Stand alone user** adalah penggunaan yang mengelola databasenya sendiri dengan menggunakan perangkat lunak yang telah tersedia dan tersedia antar muka sehingga mempermudah dalam menggunakan database tersebut (Geeksforgeeks, 2019).

## 6. Aplikasi

Database biasanya dipergunakan untuk membantu mempermudah proses pada program aplikasi. Seperti pada proses pencarian, penyusunan laporan transaksi penjualan dan pembelian atau aplikasi lainnya. Hampir seluruh perusahaan menggunakan aplikasi contohnya aplikasi rental mobi, aplikasi jual beli, aplikasi penyewaan, dan aplikasi lainnya.

## B. DATA MODEL

Data model adalah sekumpulan konsep yang digunakan untuk menjelaskan struktur dari basis data (database structure) dan memberikan gambaran tingkat-tingkat abstraksi data (data abstraction).

Data model juga mencakup sekumpulan operasi (basic dan user-defined) yang dapat dilakukan terhadap data yang dihimpun dalam basis data.

Operasi dasar (basic) disediakan oleh sistem basis data : insert, delete, update dan retrieve

User defined operation dibuat oleh perancang, contohnya adalah operasi menghitung IPK mahasiswa

Data abstraction adalah deskripsi mengenai struktur basis data yang mudah dimengerti oleh user, atau bisa mencakup detil dari penyimpanan data yang biasanya perlu diketahui oleh perancang basis data.

Database structure mencakup data type (*type of name is string*), relationship (*customer has relationship with account*), dan *constraint*.

## C. TINGKATAN ABSTRAKSI

1. **Physical level** menggambarkan bagaimana suatu record disimpan secara fisik.
2. **Logical level** menggambarkan bagaimana suatu record disimpan dalam basis data dan menggambarkan hubungan (*relationship*) antar data.

3. **View level** suatu *application* program, tidak akan memperlihatkan data type juga dapat menyembunyikan informasi (misalnya gaji) yang mempunyai tingkat security tertentu yang terkait dengan otoritas user.

#### **D. Kategori Model Data (Konsep Data Model)**

1. **Higt-level or Conceptual data model** adalah konsep yang mudah dimengerti oleh end –user. Menggunakan konsep entitas, attributes dan relationships
2. **Entity** adalah representasi obyek dalam dunia nyata misalnya mahasiswa, dosen dan lainnya atau obyek dalam konsep misalnya makhluk hidup.
3. **Relationship adalah** hubungan antara beberapa jenis entitas misalnya hubungan antara entitas pegawai dengan entitas proyek adalah pelaksana proyek

#### **Latihan**

1. Sebutkan dan Jelaskan defenisi dari tingkatan level abstraksi data (penjelasan menggunakan bahasa sendiri bukan bahasa buku dan sejenisnya).
2. Sebutkan contoh dari masing-masing tingkatan level dari abstraksi data didalam sebuah basis data
3. Buatlah sebuah contoh relasi antar tabel dengan minimal 3 buah tabel



4. Sebuah perusahaan XYZ menginginkan sebuah database untuk mengelola data pegawainya, sebagai seorang Database designer silahkan dirancang database sesuai keinginan dari perusahaan tersebut.

# BASIS DATA

## RELATIONAL

### 1. BASIS DATA RELASIONAL

Pada model relasional, basis data akan disebar kedalam bentuk 2 dimensi. Yang mana setiap tabel terdiri atas baris dan kolom.

Contoh basis data yang mengilustrasikan model data relasional secara lebih nyata, contohnya basis data sederhana yang berkaitan dengan Pendidikan Perguruan Tinggi, misalnya **Basis Data Akademik**.

Basis data tersebut terdiri dari atas 3 tabel utama yaitu

- a. Tabel Mahasiswa
- b. Tabel dosen
- c. Tabel kuliah

Sebelum implementasi basis data dapat dilakukan (dengan menggunakan salah satu perangkat lunak DBMS yang ada), maka terlebih dahulu kita harus memiliki contoh data yang akan digunakan. Contoh data ini harus dianalisis untuk mengetahui karakteristik masing-masing tabel data. Pengetahuan karakteristik ini penting, karna hal ini akan dijadikan dasar untuk penetapan struktur masing-masing tabel.

Berikut contoh data yang akan digunakan (data hanya sebagai contoh yang tentunya belum menggambarkan kenyataan sesungguhnya)

- a. Data mahasiswa, dengan isi sebagai berikut :

| <b>Nim/ NPM</b> | <b>Nama Mahasiswa</b> | <b>Alamat Mahasiswa</b> | <b>Tgl Lahir</b> |
|-----------------|-----------------------|-------------------------|------------------|
| 10081001        | Ali Akbar             | Jl. Merak               | 02 Januari 1997  |
| 10081002        | Budi Haryanto         | Jl. Merpati             | 13 Desember 1998 |
| 10071003        | Indah Susanti         | Jl. Bangau              | 05 Februari 1997 |
| 10071004        | Hasanudin             | Jl. Balam               | 25 Maret 1997    |

Dari data diatas dapat diketahui karakteristik tabel sebagai berikut :

1. Data mahasiswa tersebut memiliki 4 (buah) Kolom data
2. Kolom pertama berisi data angka tetapi tidak menunjukkan suatu jumlah (disebut alfa numerik) dengan lebar (banyak nya karakter) tetap yaitu 6 karakter (dijit)
3. Kolom kedua berisi data karakter (string atau text) dengan lebar maksimum 30 karakter. Angka 30 ini merupakan perkiraan bahwa untuk nama mahasiswa yang ada dan nama mahasiswa yang akan menyusul untuk ditambahkan tidak melebihi 30 karakter
4. Kolom ketiga berisi data karakter (string atau text) dengan lebar maksimum 60 karakter. 60 karakter merupakan jumlah digit terpanjang pada data yang sudah ada dan perkiraan dari data yang akan menyusul untuk diinputkan
5. Kolom keempat merupakan data penanggalan

b. Data Dosen, dengan isi sebagai berikut :

| <b>Nama Dosen</b>                     | <b>Alamat Dosen</b>           |
|---------------------------------------|-------------------------------|
| Bambang Kurniawan, A.Md, S.Kom, M.Kom | Jl. Garuda Sakti              |
| Refni Wahyuni, S.Kom, M.TI, MTA       | Perumahan Sidomulyo Residence |
| Uci Rahmalisa, S.Kom, M.TI            | Jalan Sentosa                 |

Karakteristik dari data diatas adalah :

1. Data dosen diatas memiliki 2 buah kolom data
2. Kolom Pertama berisi data karakter (string atau text) dengan lebar maksimum 40 karakter. Angka 40 ini merupakan perkiraan bahwa untuk nama dosen yang ada dan nama dosen yang akan menyusul untuk ditambahkan tidak melebihi 37 karakter
3. Kolom kedua berisi data karakter (string atau text) dengan lebar maksimum 60 karakter. Angka 60 ini merupakan perkiraan bahwa untuk alamat dosen yang ada dan alamat dosen yang akan menyusul untuk ditambahkan tidak melebihi 60 karakter

c. Data kuliah, dengan isi sebagai berikut :

| <b>Kode Kuliah</b> | <b>Nama Kuliah</b> | <b>SKS</b> | <b>Semester</b> |
|--------------------|--------------------|------------|-----------------|
| IF-110             | Basis Data         | 3          | 2               |
| IF-310             | Struktur Data      | 3          | 3               |

|        |                  |   |   |
|--------|------------------|---|---|
| KU-234 | Bahasa Indonesia | 2 | 2 |
| MA-115 | Bahasa Inggris   | 2 | 2 |

Karakteristik dari data diatas adalah :

1. Data kuliah diatas memiliki 4 buah kolom data
2. Kolom pertama berisi data karakter (string atau text) dengan lebar tetap yaitu 6 karakter (termasuk tanda minus)
3. Kolom kedua berisi data karakter (string atau text) dengan lebar maksimum 30 karakter
4. Kolom ketiga berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat (bukan pecahan)
5. Kolom keempat berisi data angka yang juga menunjukkan jumlah dengan jenis bilangan bulat (bukan pecahan)

Dari karakteristik masing-masing data diatas maka dapat ditetapkan struktur masing-masing tabel. Secara minimal yang harus ditentukan dalam struktur tabel adalah

- a. **nama kolom (field),**
- b. **tipe data,**
- c. **lebar (size data/ banyaknya karakter yang ada)**

untuk nama kolom harus menggunakan satu kata dan tidak boleh menggunakan spasi atau tanda minus, akan tetapi boleh menggunakan underscore( \_ ).

Dari karakteristik diatas dapat disimpulkan dari fakta yang ada pada masing-masing data, struktur dari ketiga tabel dalam basis data akademik yang dapat kita tentukan adalah sebagai berikut :

- a. **Tabel mahasiswa , dengan struktur :**

| <b>Nama Kolom</b> | <b>Tipe</b>      | <b>Lebar</b> | <b>Untuk Menampung Data</b>  |
|-------------------|------------------|--------------|------------------------------|
| <b>Nim</b>        | <b>Character</b> | <b>6</b>     | <b>Nomor induk mahasiswa</b> |
| <b>nama_mhs</b>   | <b>Character</b> | <b>30</b>    | <b>Nama setiap mahasiswa</b> |

|                   |                  |           |                                       |
|-------------------|------------------|-----------|---------------------------------------|
| <b>alamat_mhs</b> | <b>Character</b> | <b>60</b> | <b>Alamat setiap mahasiswa</b>        |
| <b>tgl_lahir</b>  | <b>Date</b>      | <b>8</b>  | <b>Tanggal lahir setiap mahasiswa</b> |

b. **Tabel Dosen**, dengan struktur :

| <b>Nama Kolom</b> | <b>Tipe</b>      | <b>Lebar</b> | <b>Untuk Menampung Data</b> |
|-------------------|------------------|--------------|-----------------------------|
| <b>nama_dos</b>   | <b>Character</b> | <b>35</b>    | <b>Nama setiap dosen</b>    |
| <b>alamat_dos</b> | <b>Character</b> | <b>60</b>    | <b>Alamat setiap dosen</b>  |

c. **Tabel kuliah**, dengan struktur :

| <b>Nama Kolom</b> | <b>Tipe</b>      | <b>Lebar</b> | <b>Untuk Menampung Data</b>                    |
|-------------------|------------------|--------------|--|
| <b>kode_kul</b>   | <b>Character</b> | <b>6</b>     | <b>Kode kuliah</b>                             |
| <b>nama_kul</b>   | <b>Character</b> | <b>30</b>    | <b>Nama matakuliah</b>                         |
| <b>Sks</b>        | <b>Integer</b>   | <b>1</b>     | <b>Jumlah bobot satuan kredit semester</b>     |
| <b>semester</b>   | <b>Integer</b>   | <b>1</b>     | <b>Semester diselenggarakannya perkuliahan</b> |

## BASIS DATA

# NORMALISASI

**Normalisasi** adalah proses penyusunan tabel-tabel yang tidak redudan (dobel), yang dapat menyebabkan anomaly pada saat terjadi operasi manipulasi data seperti **tambah, hapus, dan ubah.**

**Anomali** adalah masalah yang timbul dalam pembuatan pembuatan tabel. Anomali adalah proses pada basis data yang mempunyai efek samping yang tidak diharapkan.

**Contoh :** ketidakkonsistenan data, suatu data hilang pada saat dihapus dan lain-lain.

### **Fungsi Normalisasi :**

1. Dapat digunakan sebagai metodologi dalam menciptakan desain database
2. Dapat digunakan sebagai verifikasi terhadap hasil desain yang telah dibuat, baik menggunakan ER-Model atau menggunakan model relasi, seperti yang anda buat diatas atau dari model yang lain

### **Tujuan Normalisasi :**

1. Normaliasi diperlukan agar kerelasian dalam basis data menjadi mudah dimengerti, mudah dipelihara, mudah memprosesnya, dan mudah untuk dikembangkan sesuai dengan kebutuhan baru.
2. Normalisasi mengharuskan relasi pada suatu basis data dalam bentuk normal.
3. Bentuk normal sendiri merupakan seperangkat aturan yang dikenakan dan harus dipenuhi oleh relasi basis data pada setiap level normalisasi

### **Jenis-jenis Anomali**

#### **1. Anomali peremajaan**

Anomali ini terjadi bila ada perubahan pada sejumlah data yang mubazir, tetapi tidak seluruhnya diubah

Contoh : Tabel Pesanan

| <b>Pemasok</b> | <b>Kota</b> | <b>Barang</b> | <b>Jumlah</b> |
|----------------|-------------|---------------|---------------|
| Budi           | Bangkinang  | AC            | 6             |
| Ani            | Pekanbaru   | Radio         | 3             |
| Meri           | Pelalawan   | TV 40 inc     | 7             |
| Rahma          | Siak        | Kipas Angin   | 5             |

Seandainya budi pindah dari kota Bangkinang pindah ke Bengkalis maka perubahan data hanya dilakukan pada data pertama menjadi :

Tabel pesanan

| Pemasok | Kota      | Barang      | Jumlah |
|---------|-----------|-------------|--------|
| Budi    | Bengkalis | AC          | 6      |
| Ani     | Pekanbaru | Radio       | 3      |
| Meri    | Pelalawan | TV 40 inc   | 7      |
| Rahma   | Siak      | Kipas Angin | 5      |

Maka dapat terlihat bahwa data tentang pesanan atas nama Budi tidak sama dengan data tabel pesanan yang awal yang menyebabkan ketidak konsistenan data.

## 2. Anomali Penvisipan

Anomali ini terjadi pada saat penambahan data yang mana ada nya elemen kosong dan elemen tersebut justru menjadi key.

Contoh : Tabel Kursus

| Jumlah Siswa | Kursus            | Biaya Kursus |
|--------------|-------------------|--------------|
| 25           | Robotika          | Rp 1.500.000 |
| 14           | Jaringan          | Rp 1.000.000 |
| 20           | Pemograman Web    | Rp 1.600.000 |
| 15           | Pemograman Mobile | Rp 1.700.000 |

Dilihat dari data diatas, seandainya akan ditambahkan data baru, misalnya dibukanya kursus Desain Grafis dengan biaya Rp. 1.500.000 akan tetapi peserta kursusnya belum ada, sehingga akan membentuk tabel seperti dibawah ini :

Tabel Kursus

| Jumlah Peserta | Kursus            | Biaya Kursus |
|----------------|-------------------|--------------|
| 25             | Robotika          | Rp 1.500.000 |
| 14             | Jaringan          | Rp 1.000.000 |
| 20             | Pemograman Web    | Rp 1.600.000 |
| 15             | Pemograman Mobile | Rp 1.700.000 |
|                | Desain Grafis     | Rp 1.500.000 |

## 3. Anomali Penghapusan

Anomali ini terjadi apabila dalam satu baris/ tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang. Contoh pada

table kursus data Jumlah Peserta 25 akan dihapus dengan data kursus Robotika karena sudah tidak ikut kursus lagi sehingga akibatnya data kursus Robotika dan biaya kursus sebesar Rp 1.500.000 akan ikut terhapus



# MODUL

## STRUCTURED QUERY LANGUAGE (SQL)

**Bahasa query** adalah bahasa yang termasuk dalam kategori bahasa tingkat tinggi (*high level language*) yang digunakan user untuk mendapatkan informasi/ data dari basis data. Sedangkan Structured query language merupakan komponen bahasa relational database system. SQL merupakan bahasa baku, non procedural dan berorientasi himpunan. SQL dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan hingga pemeliharaan kinerja basis data.

Bahasa SQL memiliki beberapa bagian penting, yaitu :

1. Data Definition Language (DDL)

DDL menyediakan perintah-perintah untuk mendefinisikan skema relasi, menghapus relasi, serta memodifikasi skema relasi

2. Data Manipulation Language (DML)

DML mencakup bahasa SQL untuk menyisipkan rekaman pada relasi, menghapus rekaman pada relasi, serta memodifikasi rekaman pada relasi.

3. Data Control Language (DCL)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, contohnya manipulasi user dan hak akses. Yang termasuk perintah dalam DCL ada dua, yaitu GRANT dan REVOKE.

### 1. DATA DEFINITION LANGUAGE (DDL)

Penulisan perintah SQL pada kelompok ini terdiri dari Create, Alter dan Drop :

- a. **Create**

Perintah **Create** digunakan untuk membuat basis data dan tabel pendukung didalam membangun basis data tersebut. Contoh sintak penggunaan Create :

1. Untuk membuat database baru :

***CREATE DATABASE databasename;***

***Contoh : create database db\_akademik***

2. Untuk membuat tabel baru :

```
CREATE TABLE table_name  
(column1 datatype,  
column2 datatype,  
column3 datatype  
.....  
);
```

```
Contoh : create table nasabah  
(no_nasabah char (10),  
nm_nasabah varchar (30),  
alamat_nasabah varchar (30),  
primary key (no_nasabah);
```

**b. Alter**

Perintah ini digunakan untuk mengubah struktur tabel yang terdapat didalam basis data. Hal ini terjadi apabila terjadi penambahan atau penghapusan suatu field atau atribut key (Primary/ Foreign), tanpa mendefenisikan struktur baru pada tabel. Contoh sintak penggunaan Alter :

1. Sintak menambah kolom baru :

```
ALTER TABLE tabel_name ADD column_name datatype;
```

**Contoh :**

*Menambahkan kolom alamat pada tabel mahasiswa dengan tipe data varchar panjang field 25, dengan sintak seperti dibawah ini :*

```
Alter table mahasiswa ADD alamat varchar(25);
```

2. Sintak menghapus kolom :

```
ALTER TABLE tabel_name DROP COLUMN column_name;
```

**Contoh :** *Alter table mahasiswa DROP COLUMN alamat;*

3. Sintak merubah kolom :

```
ALTER TABLE table_name ALTER COLUMN column_name datatype;
```

**Contoh :** *ALTER TABLE mahasiswa ALTER COLUMN alamat;*

c. **Drop**

perintah ini digunakan untuk menghapus table yang terdapat didalam basis data, contoh penulisannya :

1. Sintak menghapus database

**DROP DATABASE databasename;**

**Contoh :**

**DROP DATABASE db\_akademik;**

2. Sintak menghapus table

**DROP TABLE table\_name;**

**Contoh :**

**DROP DATABASE alamat;**

## 2. DATA MANIPULATION LANGUAGE (DML)

Data Manipulation Language (DML) merupakan sintak SQL yang digunakan untuk melakukan pengolahan record atau manipulasi data pada table dalam suatu basis data. Kelompok dari DML adalah sebagai berikut : insert, select, update, dan delete. Contoh penulisan sintaknya adalah :

a. **Insert**

Perintah ini digunakan untuk melakukan entry atau penambahan suatu record pada tabel dalam basis data. Aturan umum penulisan yang digunakan adalah :

1. Sintak pertama

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

2. Sintak kedua :

INSERT INTO table\_name VALUES (value1, value2, value3, ...);

Contoh : menambahkan satu record pada tabel mahasiswa :

| NIM     | Nama mahasiswa | Alamat    |
|---------|----------------|-----------|
| 1021001 | Budi           | Pekanbaru |

|         |       |           |
|---------|-------|-----------|
| 1021002 | Ani   | Pelalawan |
| 1021003 | Ilham | Kuansing  |

Sintak untuk menambahkan satu record pada tabel diatas adalah :

***INSERT INTO mahasiswa (nim, nama\_mahasiswa, alamat) Values ('10211004','Redo','Kampar');***

maka hasil dari penambahan record pada tabel diatas adalah

| NIM      | Nama mahasiswa | Alamat    |
|----------|----------------|-----------|
| 1021001  | Budi           | Pekanbaru |
| 1021002  | Ani            | Pelalawan |
| 1021003  | Ilham          | Kuansing  |
| 10211004 | Redo           | Kampar    |

#### b. Select

Perintah ini digunakan untuk memilih record yang akan ditampilkan berdasarkan data pada tabel dalam basis data. Aturan umumnya adalah sebagai berikut:

1. Sintak untuk menampilkan salah satu kolom yang ada pada tabel :

***SELECT COLUMN1, COLUMN2,... FROM table\_name;***

2. Sintak untuk menampilkan semua kolom yang ada pada tabel :

***SELECT \*FROM table\_name;***

#### c. Update

Perintah ini untuk mengubah data dalam suatu tabel pada field tertentu dengan record baru berdasarkan suatu field sebagai criteria perubahan record-nya.

1. Sintak perintah update

***UPDATE table\_name SET column1= value1, column2=value2, ... WHERE condition;***

***Contoh :***

***UPDATE mahasiswa SET nama\_mahasiswa ='Riko' WHERE alamat = 'Pekanbaru';***

#### d. Delete

Perintah SQL ini digunakan untuk menghapus data dalam suatu table, berdasarkan suatu field sebagai kriteria penghapusan record-nya.

1. Sintak perintah untuk menghapus salah satu record:

***DELETE FROM table\_name WHERE conditionan:***

***Contoh :***

***DELETE FROM mahasiswa WHERE nama\_mahasiswa ='Budi';***

2. Sintak untuk menghapus seluruh record pada tabel:

***DELETE FROM table\_name;***

***Contoh : DELETE FROM mahasiswa***

### 3. DATA CONTROL LANGUAGE (DCL)

Data control language (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna basis data. Penulisan perintah terdiri dari : grant dan revoke.

#### a. GRANT

Perintah ini digunakan oleh seorang administrator basis data untuk memberikan hak aksesnya kepada pengguna tertentu, agar dapat mengakses suatu tabel dalam basis data. Hak akses tersebut adalah insert, delete, update dan select. Aturan penulisan dari **GRANT** adalah sebagai berikut :

1. Sintak :

***GRANT hak\_akses ON nama\_table To Pengguna\_tertentu;***

***Contoh :***

- Pemberian hak akses **INSERT** dan **DELETE** pada tabel mahasiswa kepada Budi

***GRANT INSERT, DELETE ON mahasiswa to Budi;***

- Pemberian hak akses **UPDATE** pada tabel mahasiswa kepada Raisa

***GRANT UPDATE ON mahasiswa TO Raisa;***

#### b. REVOKE

perintah ini digunakan oleh seorang administrator basis data, untuk membatalkan/ menghentikan hak akses yang telah diberikan kepada pengguna tertentu. Berikut aturannya :

1. Sintak:

***REVOKE hak\_akses ON nama\_tabel FROM pengguna\_tertentu;***

***Contoh :***

- Membatalkan hak akses **INSERT** dan **DELETE** pada tabel mahasiswa

kepada Budi

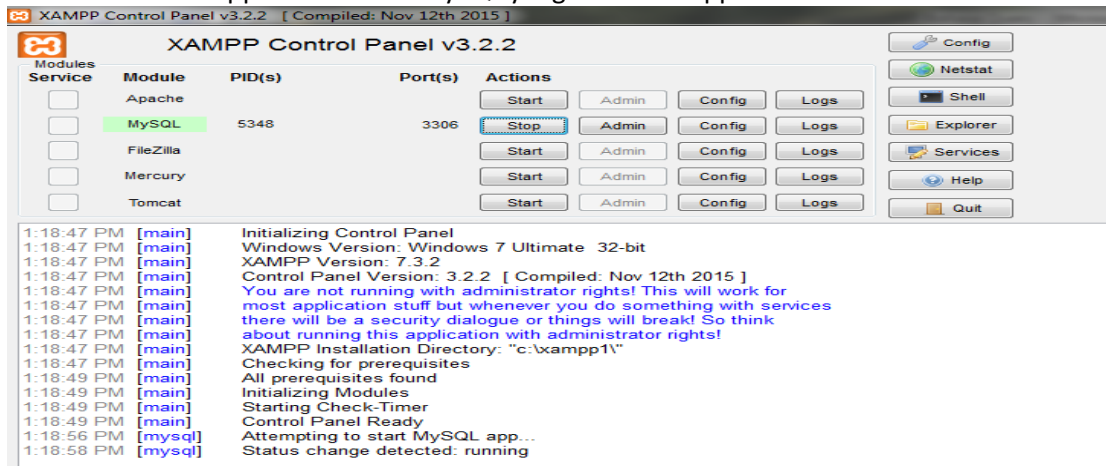
***REVOKE INSERT, DELETE ON mahasiswa FROM Budi;***

- Membatalkan hak akses UPDATE pada tabel mahasiswa kepada Raisa  
***REVOKE UPDATE ON mahasiswa FROM Raisa;***

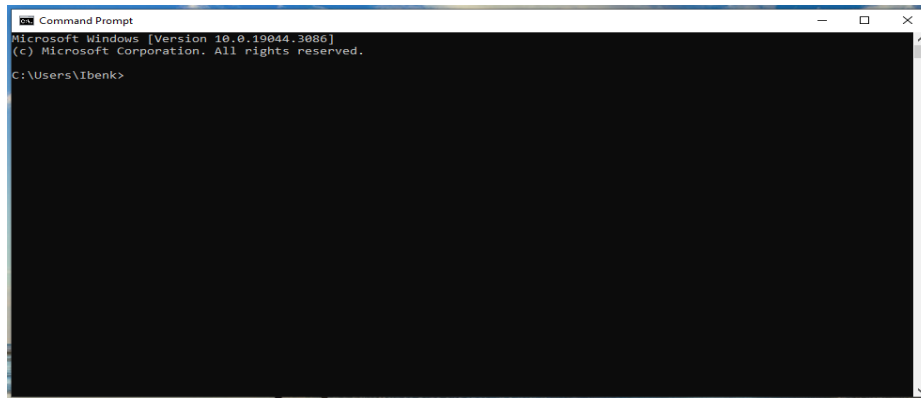
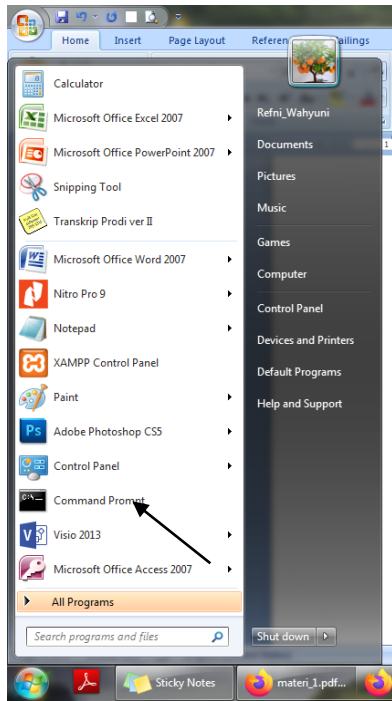
## MODUL PRAKTIKUM

### Langkah –langkah

1. Silahkan buka xampp dan aktifkan MySQL yang ada di Xampp



2. Buka halaman Command Prompt pada star, pilih command Prompt



3. Masukkan sintak berikut seperti yang terlihat dilayar command prompt

```
Command Prompt - mysql -root
C:\Users\Ibenk>cd..
C:\Users>cd..
C:\>xampp\mysql\bin
'xampp\mysql\bin' is not recognized as an internal or external command,
operable program or batch file.
C:\>cd/xampp/mysql/bin
C:\xampp\mysql\bin>mysql -root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.38-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

Jika keluar teks seperti diatas **welcome to the MariaDB monitor** berarti command prompt sudah terhubung dengan MySQL yang ada di Xampp.

selanjutnya kita akan membuat database terlebih dahulu dengan perintah :

4. Membuat database baru, masukkan sintak berikut:

**ate database dbmahasiswa //dbmahasiswa** adalah nama databasenya

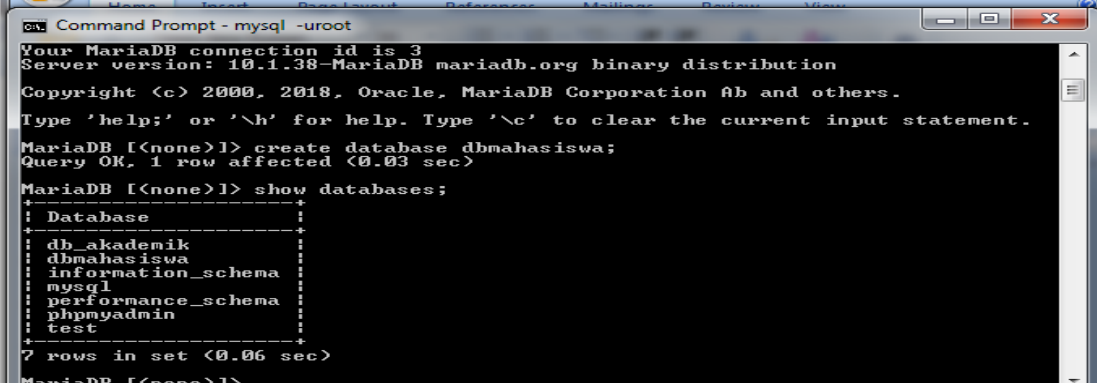
setelah itu tekan enter maka akan muncul seperti pada gambar dibawah :

```
Command Prompt - mysql -v -u root -p
C:\xampp\mysql\bin>mysql -v -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.1.38-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> create database dbmahasiswa;
-----
create database dbmahasiswa
-----
Query OK, 1 row affected (0.00 sec)
MariaDB [(none)]>
```



untuk menampilkan database yang ada di MySQL yang ada di xampp yaitu dengan mengetikkan sintak berikut :

***show databases;*** tekan **enter** : maka akan tampil semua database yang ada di xampp tersebut



```
Command Prompt - mysql -uroot
Your MariaDB connection id is 3
Server version: 10.1.38-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> create database dbmahasiswa;
Query OK, 1 row affected (0.03 sec)

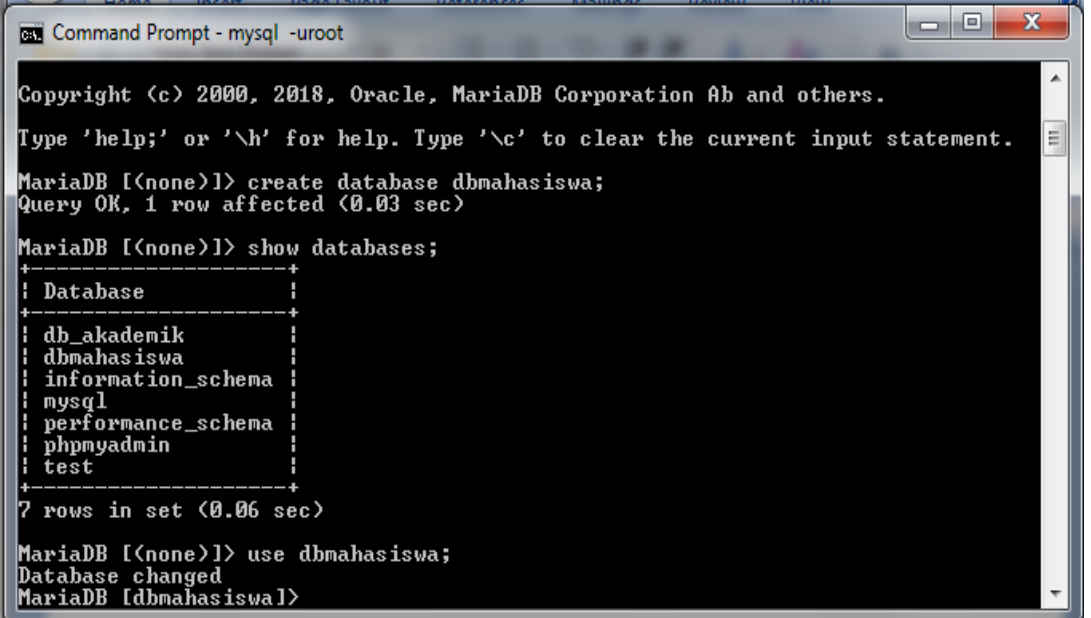
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| db_akademik |
| dbmahasiswa |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
7 rows in set (0.06 sec)

MariaDB [(none)]>
```

5. Membuat tabel baru

Untuk selanjutnya ketikkan sintak : ***use dbmahasiswa;***

Sintak diatas menyatakan bahwa dalam kasus ini kita menggunakan database **dbmahasiswa**



```
Command Prompt - mysql -uroot
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> create database dbmahasiswa;
Query OK, 1 row affected (0.03 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| db_akademik |
| dbmahasiswa |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
7 rows in set (0.06 sec)

MariaDB [(none)]> use dbmahasiswa;
Database changed
MariaDB [dbmahasiswa]>
```

Setelah itu masukkan sintak berikut :

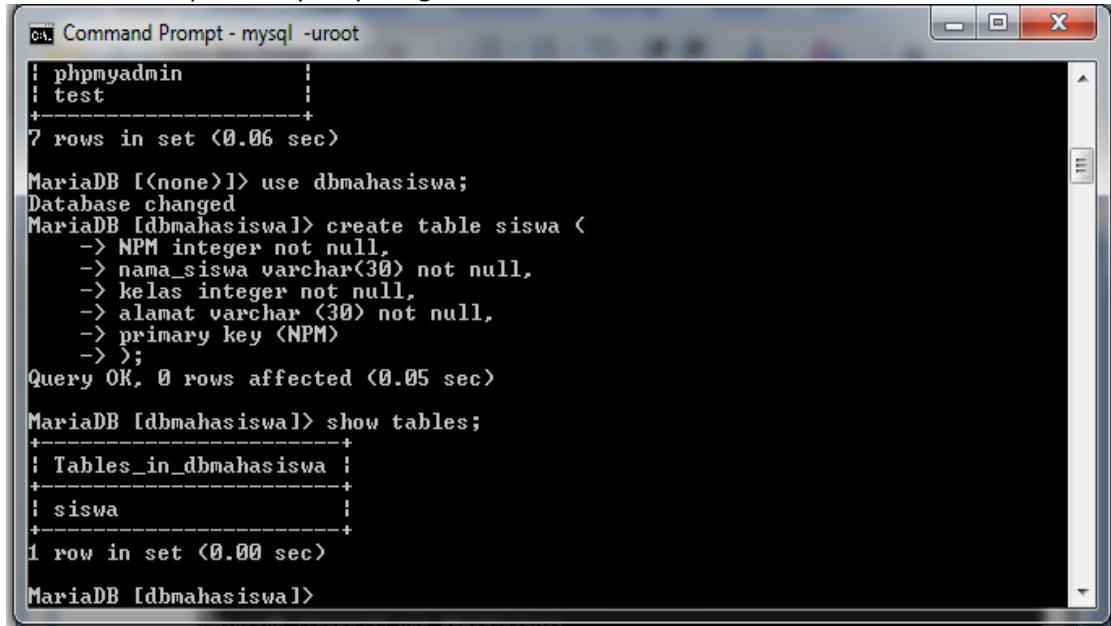
***Create table mahasiswa (***  
***NPM integer not null,***  
***nama\_mahasiswa varchar(30) not null,***  
***kelas integer not null,***  
***alamat varchar (30) not null,***

**primary key (NPM)**  
);

Untuk melihat hasil dari tabel yang sudah dibuat maka ketikkan sintak berikut :

**Show tables;**

Maka akan tampil hasilnya seperti gambar dibawah :



```
ca. Command Prompt - mysql -uroot
+-----+
| phpmyadmin |
| test       |
+-----+
7 rows in set (0.06 sec)

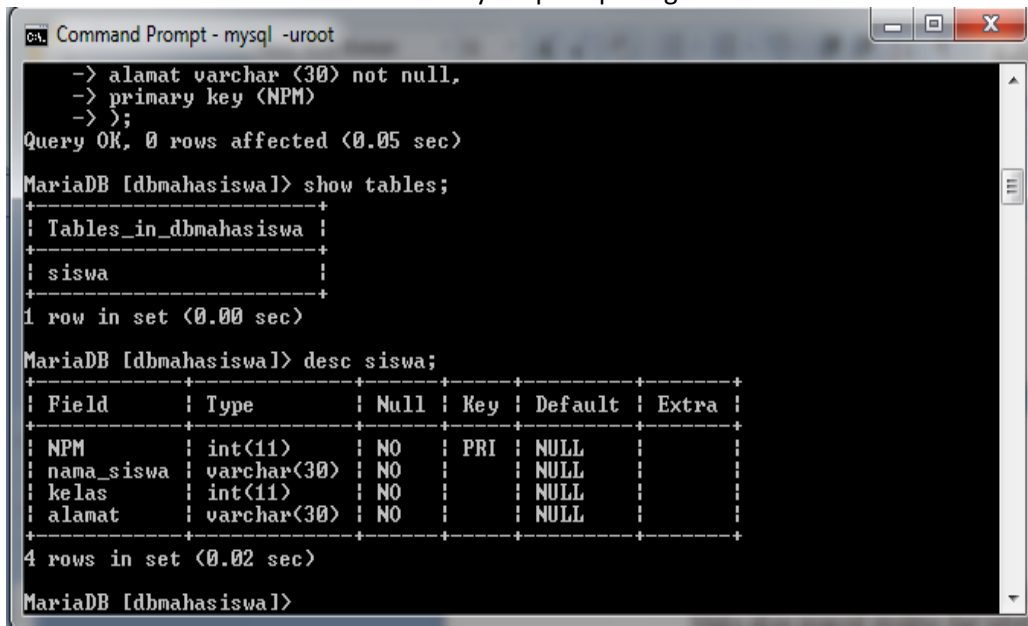
MariaDB [(none)]> use dbmahasiswa;
Database changed
MariaDB [dbmahasiswa]> create table siswa (
  -> NPM integer not null,
  -> nama_siswa varchar(30) not null,
  -> kelas integer not null,
  -> alamat varchar (30) not null,
  -> primary key (NPM)
  -> );
Query OK, 0 rows affected (0.05 sec)

MariaDB [dbmahasiswa]> show tables;
+-----+
| Tables_in_dbmahasiswa |
+-----+
| siswa                  |
+-----+
1 row in set (0.00 sec)

MariaDB [dbmahasiswa]>
```

6. Untuk melihat struktur dari tabel ketikkan sintak : **desc siswa; enter**

Maka akan muncul struktur dari tabel nya seperti pada gambar :



```
ca. Command Prompt - mysql -uroot
  -> alamat varchar (30) not null,
  -> primary key (NPM)
  -> );
Query OK, 0 rows affected (0.05 sec)

MariaDB [dbmahasiswa]> show tables;
+-----+
| Tables_in_dbmahasiswa |
+-----+
| siswa                  |
+-----+
1 row in set (0.00 sec)

MariaDB [dbmahasiswa]> desc siswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NPM        | int(11)   | NO   | PRI | NULL    |       |
| nama_siswa | varchar(30) | NO   |     | NULL    |       |
| kelas      | int(11)   | NO   |     | NULL    |       |
| alamat     | varchar(30) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

MariaDB [dbmahasiswa]>
```

7. Memasukkan data pada tabel dengan sintak seperti ini :

**insert into siswa (NPM, nama\_siswa, kelas, alamat)**  
**values (1,'Budi',2,'Pekanbaru');** lalu enter

untuk melihat data yang sudah dimasukkan ketikkan sintak berikut :

*select \* from siswa;*

```
Command Prompt - mysql -uroot
MariaDB [dbmahasiswa] > desc siswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NPM        | int(11)       | NO   | PRI | NULL    |       |
| nama_siswa | varchar(30)   | NO   |     | NULL    |       |
| kelas      | int(11)       | NO   |     | NULL    |       |
| alamat     | varchar(30)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

MariaDB [dbmahasiswa] > insert into siswa (NPM, nama_siswa, kelas, alamat)
-> values (10211001, 'Budi', 2, 'Pekanbaru');
Query OK, 1 row affected (0.02 sec)

MariaDB [dbmahasiswa] > select * from siswa;
+-----+-----+-----+-----+
| NPM      | nama_siswa | kelas | alamat |
+-----+-----+-----+-----+
| 10211001 | Budi       | 2     | Pekanbaru |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [dbmahasiswa] >
```

8. Update data pada tabel dengan sintak :

*update siswa set nama\_siswa = 'Refni',  
kelas =6,  
alamat ='Kampar'  
Where NPM = 10211001; enter*

```
ca. Command Prompt - mysql -uroot
MariaDB [dbmahasiswa] > select * from siswa;
+-----+-----+-----+-----+
| NPM      | nama_siswa | kelas | alamat |
+-----+-----+-----+-----+
| 10211001 | Budi       | 2     | Pekanbaru |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [dbmahasiswa] > update siswa set nama_siswa='Refni',
-> kelas = 6,
-> alamat = 'Kampar'
-> where NPM = 10211001;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [dbmahasiswa] > select * from;
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 '' a
t line 1
MariaDB [dbmahasiswa] > select * from siswa;
+-----+-----+-----+-----+
| NPM      | nama_siswa | kelas | alamat |
+-----+-----+-----+-----+
| 10211001 | Refni      | 6     | Kampar |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [dbmahasiswa] >
```