

**BUKU AJAR Teori dan Praktik  
Database Manajement System  
Mahir Menggunakan dan  
Mengelola Database MySQL**

## KATA PENGANTAR

Puji Syukur Kehadirat tuhan yang maha kuasa, karena atas berkah, rahmat dan hidayah-Nya Buku Ajar untuk Mahasiswa dengan materi pokok Mahir menggunakan dan Mengelola Database MySQL ini dapat diselesaikan dengan baik dan tepat waktu.

Buku Ajar yang berjudul “Teori dan Praktik Database Management System” ini disajikan secara khusus untuk kalangan pelajar tingkat SMA, maupun Mahasiswa, harapan dari buku ini yaitu dapat membantu pembaca agar mampu memahami Peranan, Fungsi, tujuan dan pengelolaan dari Sistem Manajemen Basis Data (SMBD) terutama MySQL. Database Management System (DBMS) atau yang disebut dengan Sistem Manajemen Basis Data (SMBD), adalah sebuah perangkat lunak yang dirancang untuk mengelola Basis Data, beberapa contoh system manajemen Basis Data yang tersedia dan populer di Dunia baik berbayar maupun yang disediakan secara gratis antara lain Oracle, SQL Server 2000/2003, MySQL, Ms. Access dan sebagainya.

Pembahasan yang disajikan dalam buku ini secara spesifik akan membahas salah satu dari SMBD yang dianggap paling populer yaitu MySQL (*My Structured Query Language*) dan saat ini telah digunakan lebih dari 100 juta pengguna di seluruh dunia, MySQL merupakan salah satu *software* SMBD yang memiliki sifat *open source* dengan artian Setiap orang dapat menggunakan/ memanfaatkannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (*General Public License*) namun tidak diizinkan untuk menjadikannya produk turunan yang dianggap memiliki sifat komersial. Dengan segala kelebihan yang dimilikinya, antara lain handal, cepat, dan mudah untuk digunakan, MySQL menjadi pilihan utama bagi kalangan pengembang system informasi

Buku ajar yang dikembangkan ini bertujuan untuk mengembangkan proses pembelajaran secara aktif dengan aspek sasaran yaitu Afektif, Kognitif,

Psikomotorik serta melatih kemampuan berpikir kreatif Mahasiswa pada jenjang Strata 1 (Sarjana), buku ini diharapkan dapat membantu Mahasiswa dalam mengembangkan kompetensinya, agar kelak mampu memahami dan menerapkannya dalam kehidupan sehari-hari.

Akhirnya penulis berharap agar buku ajar ini dapat bermanfaat khususnya bagi pembaca dan secara umum bagi perkembangan ilmu dan pengetahuan di Indonesia khususnya dalam hal penggunaan dan pengelolaan database MySQL. Penulis menyadari bahwa buku ajar ini masih terdapat kekurangan, oleh karena itu Saran dan kritik yang bersifat memperbaiki kualitas konten dari buku ini sangat penulis harapkan. Terimakasih kepada semua pihak yang membantu dalam penyusunan buku ajar ini.

Dompu, Agustus 2023  
Penulis

## DAFTAR ISI

<b>HALAMAN SAMPUL .....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>BAGIAN 1 Dasar Teori DBMS (<i>Database Management System</i>).....</b>	<b>1</b>
A. Pengantar .....	1
B. Fasilitas DBMS.....	5
C. Komponen Pendukung DBMS .....	6
D. Manfaat, Kelebihan dan kekurangan dari	
E. penggunaan DBMS.....	7
F. Macam-macam DBMS beserta kelebihan	
dan kekurangannya.....	9
1. MySQL.....	9
2. Oracle.....	11
3. Firebird.....	13
4. Microsoft SQL Server .....	15
5. PostgreSQL.....	17
6. MongoDB .....	19
G. Hirarki Basis Data.....	20
H. Evaluasi Materi .....	22
<b>BAGIAN 2 Praktikum dasar penggunaan DBMS .....</b>	<b>23</b>
A. XAMPP sebagai software Pendukung.....	23
B. Praktikum 1: Melakukan Instalasi XAMPP .....	24
C. Praktikum 2: Akses MySQL server melalui CMD.....	25
D. Praktikum 3: Membuat dan menampilkan Database.....	26
E. Memahami Tipe-Tipe Tabel dan Field dalam MySQL .....	27
1. Tipe-tipe Tabel dalam MySQL .....	27
a) MyISAM .....	27
b) InnoDB.....	28
c) HEAP.....	28
2. Tipe-tipe Field pada MySQL .....	28
a) Tipe Numeric (TinyINT, SmallINT,	
MediumINT, INT, BIGINT, Float, Double,	
Decimal) .....	28
b) Tipe Date and Time (Date, Time, DateTime, Year).	

c) Tipe String (Text) (Char, Varchar, TinyText, Text, MediumText, Long Text .....	30
d) Tipe BLOB (Biner) (Bit, TinyBlob, Blob, MediumBlob, LongBlob).....	31
e) Tipe Data lain (Enum dan Set) .....	32
F. Aturan dasar dalam perancangan Database.....	32
1. Redudancy dan peran Primary Key .....	32
2. Tipe dan ukuran data yang tepat .....	33
3. Penamaan Identifier .....	34
G. Praktikum 4: Menghapus Database.....	34
H. Praktikum 5: Membuat, Menampilkan, dan mendeskripsikan Database .....	35
I. Praktikum 6: Mengubah Struktur Tabel pada Database .....	39
J. Praktikum 7: Menghapus Tabel pada Database .....	47
K. Praktikum 8: Menambah RECORD kedalam Tabel dengan perintah INSERT .....	47
L. Praktikum 9: Perintah SELECT untuk menampilkan RECORD.....	51
M. Evaluasi Materi .....	56
<b>DAFTAR PUSTAKA.....</b>	<b>57</b>
<b>PENUTUP .....</b>	<b>58</b>
<b>GLOSARIUM.....</b>	<b>59</b>
<b>BIODATA PENULIS .....</b>	<b>60</b>

## DAFTAR GAMBAR

Gambar 1. Analogi pemanfaatan database dalam kehidupan .....	2
Gambar 2. Komponen dalam penggunaan database .....	4
Gambar 3. Hirarki Database .....	21







## **BAGIAN 1**

### **Dasar Teori DBMS (*Database Management System*)**

**Setelah mempelajari Bagian 1 ini mahasiswa mampu:**

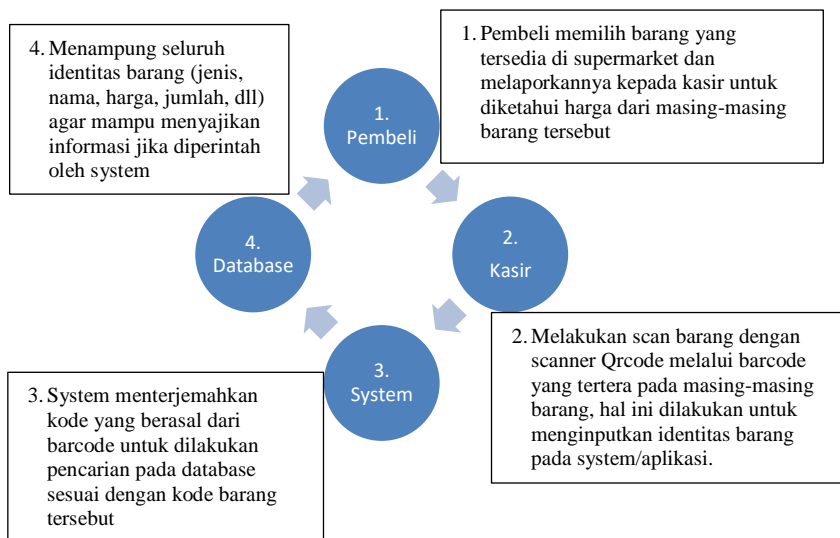
1. Memahami dasar teori, fasilitas dan komponen Database management system
2. Memahami manfaat, kelebihan dan kekurangan DBMS
3. Memahami konsep Hirarki dari basis data

#### **A. Pengantar**

Database Management System (DBMS) tidak dapat lepas dari istilah yang sering kita dengar yaitu “Basis Data”, Menurut (Indrajani, 2015), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. Sedangkan DBMS adalah sebuah perangkat lunak/ *software* yang digunakan untuk merancang serta membangun sebuah basis data. DBMS berperan dalam membantu, memelihara, dan melakukan pengolahan terhadap sekumpulan data, baik dalam jumlah yang kecil maupun jumlah yang besar. Tujuan dari pengolahan data tersebut agar seluruh data yang ter-input kedalam software DBMS dapat di manajemen secara baik dan teratur untuk dipergunakan oleh pengguna sesuai dengan kebutuhan.

Setelah kita memahami bahwa DBMS adalah sebuah perangkat lunak yang memungkinkan pengguna (user) untuk merancang serta membangun basis data yang terkomputerisasi, DBMS berperan sebagai perantara user dengan basis data, keberadaan DBMS memungkinkan user dapat menambahkan, mengurangi, memodifikasi, menghapus serta mengelola sekumpulan data yang tersimpan dalam sebuah basis data, hal ini sesuai dengan pendapat yang disampaikan oleh (Connolly & Begg, 2010) Database Management System (DBMS) adalah sebuah system perangkat lunak yang mengizinkan pengguna untuk mendefinisikan, membuat, memelihara dan mengontrol akses kedalam basis data.

Database sudah menjadi salah satu bagian yang sangat penting bagi kehidupan masyarakat modern, beberapa dari kita menemukan aktivitas yang melibatkan database dalam berinteraksi salah satu contohnya, jika kita pergi ke supermarket dan bertransaksi dengan kasir, kita akan menemukan kasir yang berinteraksi dengan system untuk mengetahui harga, dan jenis dari masing-masing barang yang kita beli, hubungan yang terjadi dari kondisi tersebut yaitu, pembeli berinteraksi dengan petugas (kasir), petugas berinteraksi dengan system, dan system berinteraksi dengan database untuk mencari harga dan jenis dari barang yang diinginkan oleh pembeli.



Gambar 1. Analogi pemanfaatan database dalam kehidupan

Analogi yang disampaikan pada gambar diatas menjelaskan interaksi antar seluruh komponen yang terkait dalam pemanfaatan database sebagai sumber data, jika yang disampaikan pada gambar tersebut database hanya mampu menampung data dengan huruf dan angka terkait dengan informasi biodata barang, teknologi dan kemampuan DBMS telah dikembangkan agar mampu menampung data gambar, klip audio, dan video stream secara digital, selain itu

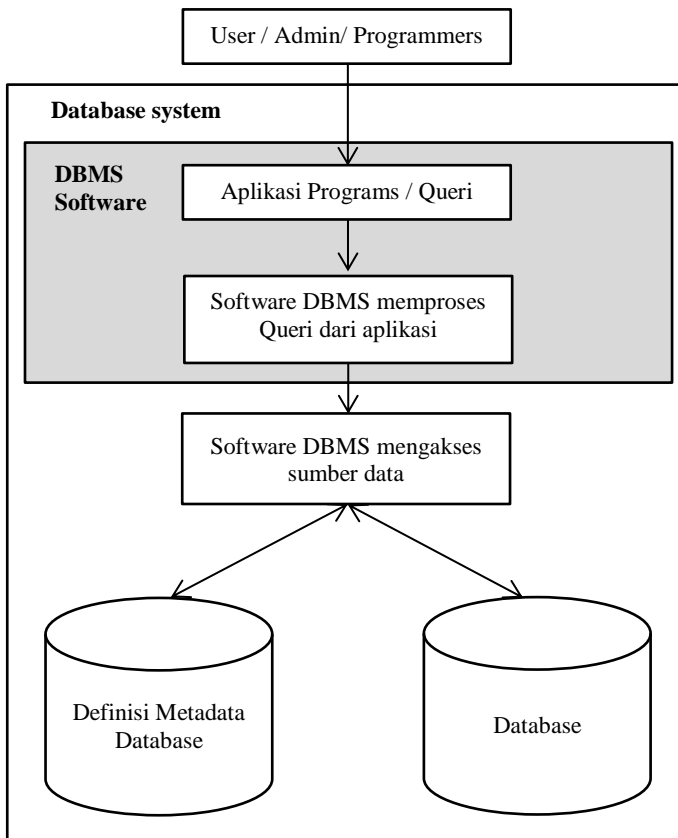
teknologi DBMS juga telah dikembangkan agar mampu digunakan secara real time untuk mengontrol proses bisnis, industry, manufaktur dan sejenisnya dengan teknik pencarian basis data yang diterapkan ke World Wide Web (aplikasi berbasis website) agar mampu meningkatkan pencarian informasi yang dibutuhkan oleh pengguna yang menjelajahi internet.

Teknologi database memiliki dampak yang sangat besar terhadap meningkatnya penggunaan computer, adalah adil jika mengatakan bahwa database telah memainkan peran penting dalam hampir semua bidang dimana computer digunakan termasuk bisnis, perdagangan, teknik, kedokteran, genetika, hukum, pendidikan dan beberapa bidang lainnya. Istilah database adalah istilah yang sangat umum digunakan, agar tidak mengalami penyimpangan dalam mendefinisikannya maka terlebih dahulu kita mendefinisikan apa itu database?

Database adalah kumpulan data terkait. Langkah awal dalam mengumpulkan data yang dimaksud adalah menyediakan rancangan strukturnya, Struktur tersebut dimaksudkan untuk mengklasifikasi masing-masing data yang akan ditampung dalam database, hal ini dilakukan karena kita memahami bahwa data yang direncanakan tersebut memiliki karakter dan sifat masing-masing. Database dirancang, dibangun dan diisi dengan data demi tujuan tertentu dari sekelompok pengguna, keberadaan Database management system dalam mendukung proses tersebut sangat dibutuhkan, mengingat DBMS adalah sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara basis data, dalam penggunaannya DBMS berperan dalam memfasilitasi proses pendefinisian, pembangunan, manipulasi dan berbagi data antar berbagai pengguna dan aplikasi.

Program aplikasi mengakses database dengan mengirimkan pertanyaan atau permintaan data ke DBMS melalui "query", query yang dimaksud adalah sebuah tindakan yang dapat menyebabkan beberapa data dibaca dan ditampilkan ke layar, selain itu query juga dapat menyebabkan data ditulis kedalam database yang telah disediakan, selain memfasilitasi hubungan antara data dengan pengguna, DBMS juga memiliki fungsi penting lainnya yaitu

melindungi database dan mempertahankannya dalam jangka waktu yang lama, perlindungan yang dimaksud adalah perlindungan terhadap kerusakan perangkat keras atau perangkat lunak dari akses yang tidak sah atau berbahaya, berikut disajikan gambar yang menjelaskan tentang hubungan antara pengguna (user), aplikasi (software), software DBMS dan database secara sederhana.



Gambar 2. Komponen dalam penggunaan database

Gambar tersebut menjelaskan hubungan antara user (pengguna), administrator, atau programmer dengan system database, skema diatas dianggap bahwa aplikasi dan database telah

dikembangkan terlebih dahulu, dalam pengembangannya aplikasi tentu mengandung Query atau yang dikenal dengan perintah untuk mengakses database, query yang diinstruksikan oleh aplikasi tersebut akan diproses oleh software DBMS yang dalam hal ini adalah MySQL, akses yang dilakukan adalah akses kedalam database dan metadata dari database tersebut, hal ini dilakukan untuk memenuhi permintaan dari aplikasi yang diinstruksikan oleh user.

## **B. Fasilitas DBMS**

Untuk mempermudah pemahaman terhadap keberadaan Database Management System (DBMS) maka akan dijelaskan fasilitas apa saja yang dapat dimanfaatkan oleh user (pengguna) dalam melakukan manajemen terhadap sekumpulan data yang disebut dengan basis data (Database), fasilitas-fasilitas tersebut diuraikan sebagai berikut:

1. DBMS memungkinkan user (pengguna) untuk mendefinisikan/ membangun basis data melalui *Data Definition Language* atau yang sering disebut dengan Perintah *DDL*. Perintah ini digunakan untuk menentukan type, dan struktur basis data yang akan dibangun.
2. DBMS memungkinkan pengguna untuk melakukan operasi penambahan (*create*), pengubahan (*modification*), penghapusan (*delete*), serta pengambilan data (*select*) dari basis data yang sebelumnya telah dirancang, fasilitas ini merupakan fasilitas yang disediakan oleh DBMS dengan sebutan *Data Manipulation Language* atau yang sering disebut dengan *DML*. Adapun standarisasi bahasa yang digunakan oleh *DBMS* adalah *Structure Query Language (SQL)* atau yang biasa disebut dengan bahasa Query.
3. DBMS memungkinkan pengguna untuk melakukan akses kendali secara penuh kedalam basis data, seperti mengatur security (keamanan) basis data untuk mengontrol client atau pengguna yang tidak berkepentingan terhadap basis data dengan tujuan menjaga integritas serta konsistensi data yang terdapat dalam basis data.

4. DBMS memungkinkan pengguna untuk melakukan control konkurensi yang mengizinkan basis data untuk dapat berbagi akses dengan pengguna maupun basisdata lainnya.
5. DBMS memungkinkan pengguna untuk melakukan system control pemulihan yang bermaksud mengembalikan data pada keadaan yang konsisten dari sebelumnya jika terjadi kegagalan yang disebabkan oleh perangkat keras maupun perangkat lunak yang berkaitan dengan DBMS.

### **C. Komponen Pendukung DBMS**

Database Management System (DBMS) adalah sebuah perangkat lunak (*Software*) yang ketika beroperasi didukung oleh beberapa perangkat lainnya seperti *Hardware, software, Data, Procedures*, maupun *brainware*, berikut ini akan diuraikan beberapa komponen pendukung DBMS yang dimaksud.

#### **1. Hardware**

Dalam implementasinya, DBMS membutuhkan ketersediaan Hardware berupa *Computer Personal (PC)*, spesifikasi computer yang disediakan tentu disesuaikan dengan kebutuhan basis data yang akan dibangun perusahaan/ organisasi, semakin besar basis data yang direncanakan tentu akan membutuhkan computer dengan spesifikasi yang lebih tinggi.

#### **2. Software**

Kita sudah sama sama memahami bahwa DBMS adalah sebuah perangkat lunak (*software*) yang digunakan untuk membuat dan mengelola basis data, tentu ketersediaan *software* yang dimaksud merupakan syarat utama dalam merancang, dan mengelola basis data, tidak hanya itu untuk memudahkan pengelolaan basis data DBMS juga membutuhkan beberapa perangkat lunak pendukung seperti *XAMPP, MySQL Front, Browser*, dan beberapa perangkat lunak lainnya, masing-masing perangkat lunak pendukung tersebut memiliki fungsi dan peran secara masing-masing.

#### **3. Data**

Data merupakan komponen yang menjadi alasan utama dari keberadaan komponen lainnya, keberadaan DBMS dan berbagai

*software* pendukung memiliki tugas untuk menampung, mengelola, memajemen, dan mengamankan data.

#### 4. **Procedures**

Prosedur merupakan sekumpulan instruksi dan aturan yang digunakan untuk merancang dan mengelola basis data.

#### 5. **Brainware**

*Brainware* (manusia) merupakan komponen terakhir dan merupakan salah satu komponen yang dianggap paling penting karena memiliki sifat pemerintah atau pemberi tugas kepada seluruh komponen yang disediakan.

### **D. Manfaat, Kelebihan dan Kekurangan dari penggunaan DBMS**

DBMS adalah salah satu perangkat lunak yang dirancang dengan memperhatikan kebutuhan pengguna, baik dari fasilitas pengelolaan basis data, pelayanan, kompleksitas, integritas data maupun keamanannya, berbagai *software* DBMS baik yang ber-license maupun yang non-license telah memiliki banyak versi pengembangan, hal ini dilakukan untuk menutupi kekurangan dari versi sebelumnya. Berikut ini akan disajikan secara umum terkait dengan manfaat, kelebihan maupun kekurangan dari penggunaan DBMS.

#### 1. **Manfaat**

Beberapa manfaat yang dapat diperoleh dari implementasi DBMS adalah sebagai berikut:

- a) Praktis, DBMS menyediakan media penyimpanan yang secara fisik berukuran kecil namun dapat menyimpan data yang sangat banyak dibandingkan dengan konsep penyimpan data secara konvensional menggunakan buku dan arsip lain.
- b) Aplikasi yang dirancang tentu akan memanfaatkan DBMS dan diakses melalui computer, tentu secara cepat dan spesifik dapat menampilkan informasi dari data yang dibutuhkan.
- c) Pemanfaatan DBMS akan memberikan analisa terkait dengan progres input, dan ketersediaan data.
- d) Meminimalisir biaya dari pengadaan dokumen penyimpanan secara konvensional yang dilakukan secara terus menerus.

## 2. Kelebihan

Beberapa kelebihan dari DBMS adalah sebagai berikut:

- a) Mengendalikan redundansi data melalui aturan klasifikasi table dan *field* dalam basis data, *redundansi* merupakan kesamaan dari data yang terdapat pada basis data
- b) Pemusatan pengelolaan basis data dibawah satu orang/kelompok, tentu akan menjamin terpeliharanya standar dan kualitas data.
- c) Pemakaian dapat dilakukan secara bersama oleh beberapa user dengan control yang terjaga oleh aturan yang diberlakukan administrator.
- d) Meningkatkan keamanan data, DBMS memiliki fasilitas untuk melindungi basis data dari pihak yang tidak berwenang melalui autentifikasi username dan password.

## 3. Kekurangan

Beberapa kekurangan DBMS adalah sebagai berikut:

- a) Kompleksitas, merupakan kelebihan dari DBMS jika kemampuan sumberdaya manusia pengelola mempuni, dan sebaliknya dapat menjadi kekurangan jika terbatasnya kemampuan dari sumberdaya manusia yang tersedia, kompleksitas yang dimaksud adalah tersedianya multi fasilitas dalam implementasi DBMS seperti *Database Designers*, *Database Developer*, *Database Administrator*, dan *End-User*, tentu sumberdaya manusia pengelola harus memahami fungsional dari fasilitas tersebut.
- b) Biaya implementasi, dalam implementasi DBMS membutuhkan biaya yang sangat bervariasi, hal ini bergantung pada lingkungan, kebutuhan dan fungsionalitas yang diinginkan. Semakin banyak dan besar kebutuhan yang ingin dicakup oleh DBMS maka akan semakin besar biaya yang dikeluarkan, begitu pula sebaliknya, semakin terbatasnya kebutuhan dan fungsionalitas yang ingin dicakup oleh DBMS semakin sedikit biaya yang dikeluarkan. ketersediaan hardware menjadi kendala utama dalam implementasi DBMS, untuk mengharapkan implementasinya dapat sesuai dengan harapan tentu harus mengeluarkan biaya, hal ini dapat berupa biaya



tambahan untuk ketersediaan mesin (server) yang handal, dan media penyimpanan yang cukup besar agar mencapai kinerja yang diinginkan

- c) Biaya Operasional, biaya ini relative lebih kecil dari biaya implementasi, biaya operasional meliputi, biaya pelatihan karyawan untuk memahami system, biaya tenaga ahli dalam menjalankan operasional dan maintenance system.

## **E. Macam-macam DBMS beserta kelebihan dan kekurangannya**

Berbagai macam jenis DBMS yang tersedia di internet, baik yang berbayar maupun yang gratis, Antara lain MySQL, Oracle, Firebird, Microsoft SQL Server, PostgreSQL, MongoDB, berikut ini akan bahas terkait dengan macam-macam jenis DBMS beserta kelebihan dan kekurangannya.

### **1. MySQL**

MySQL adalah sebuah perangkat lunak (*software*) yang berfungsi untuk manajemen basis data, MySQL didistribusikan secara gratis dibawah license GPL (*General Public License*), dengan kata lain setiap orang/organisasi dapat memanfaatkannya secara bebas demi memenuhi kebutuhan pengelolaan basis data, namun MySQL memiliki batasan yakni pengguna tidak diizinkan untuk menciptakan produk turunan yang bersifat komersial.

MySQL pada awalnya diciptakan pada tahun 1979 oleh "Monty" Widenius, yang merupakan seorang programmer computer berasal dari Swedia, Monty mengembangkan sebuah system pengolahan basis data sederhana yang diberi nama UNIREG dibawah perusahaan bernama TcX di Swedia, system yang dibangun menggunakan koneksi low-level ISAM Database Engine, Sebagai bahan ujicoba Pada tahun 1994 TcX mulai mengembangkan sebuah perangkat lunak berbasis web, dan berencana menggunakan UNIREG sebagai sistem database Namun fasilitas yang dimiliki oleh UNIREG pada saat itu dianggap tidak tepat untuk di implementasikan pada perangkat lunak pengolah data yang bersifat dinamis seperti website.

Setelah UNIREG dianggap tidak tepat untuk perangkat lunak yang bersifat dinamis seperti aplikasi berbasis website, TcX lalu

mencari alternatif system lain, yaitu pemanfaatan mSQL (*miniSQL*) yang dikembangkan oleh David Hughes. Namun pada saat itu mSQL yang dikembangkan masih memiliki kekurangan, karena tidak mendukung konsep indexing. Monty bersama TcX akhirnya memutuskan untuk merancang dan mengembangkan konsep sistem basis data baru dengan menggabungkan antara UNIREG dan mSQL. Sehingga pada Bulan Mei Tahun 1995, sebuah RDBMS (*Relationship Database Management System*) baru, yang dinamakan MySQL dirilis, jenis RDBMS baru yang telah dikembangkan tersebut telah mendukung konsep indexing sehingga mampu dipadukan dengan perangkat lunak dinamis berbasis website.

a) Kelebihan MySQL

- 1) *Multiplatform*, MySQL merupakan salah satu perangkat lunak yang dapat dijalankan pada berbagai system operasi seperti Windows, Linux, dan Mac OS.
- 2) *Opensource*, merupakan salah satu kelebihan yang paling diunggulkan oleh para pengembang system, dimana MySQL dengan versi dasar dapat dimanfaatkan secara Gratis, meski disediakan secara gratis para pengembang system tidak perlu meragukan kualitas yang disediakan terutama dari segi fasilitas, dan keamanannya, hal ini dikarenakan MySQL telah ber-license GPL (*General Public License*), tentu MySQL yang disediakan secara gratis ini memiliki batasan fungsi dalam penggunaannya akan tetapi untuk pengembangan system skala menengah kebawah sampai saat ini tidak diragukan lagi, namun jika pengembang system membutuhkan lebih banyak fungsi dari MySQL dapat menggunakan versi *Enterprise* dengan harga yang cukup terjangkau dibandingkan dengan beberapa DBMS lainnya.
- 3) *Multi User*, adalah salah satu alasan mengapa MySQL begitu banyak diminati oleh para pengembang system, dengan kelebihan ini multi user dapat secara bersamaan dalam satu waktu melakukan aktivitas dengan basis data.

- 4) *Variasi Tipe Data*, MySQL menawarkan berbagai jenis tipe data antara lain *Integer, Float, Double, Char, Text, Date, Timestamp*, dan berbagai jenis type data lainnya. Beragamnya tipe data yang dimiliki oleh MySQL demi menunjang kebutuhan dari pengguna.
  - 5) *Supporting*, MySQL adalah salah satu DBMS yang fleksibel karena dapat diintegrasikan dengan berbagai macam bahasa pemrograman.
  - 6) *Tangguh dan Stabil*, beberapa pengembang system menganggap bahwa salah satu kelebihan dari MySQL yaitu Tangguh dan stabil dalam menampung, mengelola, juga dianggap cukup baik dalam menjaga basis data dari berbagai serangan.
- b) Kekurangan MySQL

Meski menyajikan banyak sekali kelebihan, tentu DBMS MySQL tidak lepas dari kekurangan, antara lain.

1. *Technical support*, Dukungan teknis dari MySQL dinilai tidak responsive, banyak pengembang yang beranggapan bahwa ketika berhadapan dengan masalah yang berbasis pada MySQL pengembang akan mengalami kesulitan pengaduan kepada Technical Support.
2. *Batasan Kemampuan*, MySQL merupakan salah satu DBMS yang tidak membutuhkan resource mempuni untuk beroperasi, jika harus dioperasikan pada spesifikasi computer yang tinggi (server) MySQL juga akan dapat beroperasi dengan baik. Kekurangan dari MySQL yaitu ketidakmampuannya mengolah data yang sangat besar karena batasan fasilitas, oleh karena itu MySQL dianggap tidak tepat untuk penggunaan skala perusahaan besar.

## 2. Oracle

Oracle adalah salah satu jenis *Database Management System* yang tidak kalah populer bagi para pengembang system, DBMS ini dikembangkan dengan menggunakan bahasa mesin (*Assembly*) yang kemudian digantikan oleh SQL, pada dasarnya DBMS ini memiliki fungsi dan fasilitas yang hampir sama dengan DBMS MySQL yaitu mengelola informasi secara akurat,

komprehensif dan terintegrasi, DBMS ini pertama kali dikembangkan oleh programmer computer yang bernama *Larry Ellison, Bob Miner* dan *Ed Oates* pada tahun 1977 dibawah naungan sebuah perusahaan *Software Development Laboratories (SDL)* yang sekarang (sejak tahun 1983) telah berganti nama menjadi *Oracle Corporation*. Pada tahun dimana *Oracle* dirilis yaitu 1979 DBMS ini telah mampu melakukan pengolahan data dalam jumlah yang sangat besar jika dibandingkan dengan DBMS jenis lain.

a) Kelebihan Oracle

1. *Scalability*, adalah kemampuan yang dimiliki oleh DBMS Oracle dalam menangani banyaknya user yang dalam satu waktu melakukan koneksi secara bersamaan, akses yang dilakukan secara bersamaan oleh pengguna tidak mempengaruhi performance dari DBMS ini.
2. *Reliability*, adalah kemampuan yang dimiliki oleh DBMS Oracle dalam melindungi data yang disebabkan oleh kegagalan fungsi pada system.
3. *Serviceability*, adalah kemampuan yang digunakan untuk menangani permasalahan yang dialami, baik mendeteksi, mengoreksi, maupun melakukan re-konfigurasi struktur data yang bermasalah.
4. *Availability*, adalah fasilitas yang digunakan untuk melakukan penanganan pada masalah yang dihadapi oleh system namun tetap menjaga agar servicenya tetap berjalan.
5. *Multiplatform*, sama halnya dengan DBMS MySQL, Oracle juga dapat digunakan pada beberapa jenis system operasi antara lain Windows, Unix, Linux, dan berbagai system operasi lain.
6. *Handal dan Aman*, Oracle memiliki kelebihan dibandingkan dengan DBMS lain yaitu kemampuannya menampung data yang sangat besar dan Keamanan yang cukup handal.
7. Dapat menampung tipe data yang sangat banyak antara lain text, image, sound, video, time series, dan berbagai tipe data lainnya.

#### b) Kekurangan Oracle

1. DBMS Oracle adalah DBMS yang cenderung dianggap sulit oleh sebagian pengembang karena kompleksitasnya.
2. Spesifikasi hardware yang dibutuhkan untuk mengimplementasikan DBMS Oracle adalah spesifikasi computer yang tinggi, hal ini dilakukan agar system dapat berjalan dengan stabil, namun jika diaplikasikan pada sumberdaya computer yang terbatas hal ini akan berdampak pada tidak optimalnya DBMS dalam beroperasi. Ini menjadi salah satu alasan mengapa DBMS Oracle Hanya diperuntukkan bagi perusahaan besar, karena penggunaannya akan dapat dilakukan secara maksimal dengan pengolahan data yang besar dan dukungan sumberdaya computer yang maksimal.

### 3. Firebird

*Firebird* adalah sebuah perangkat lunak pengolah basis data yang bersifat *opensource*, DBMS ini pertama kali dikembangkan oleh perusahaan yang bernama Borland sekitar tahun 2000, versi beta dari aplikasi basis datanya pada saat itu adalah *Interbase 6.0*. pada saat itu *source code* dari *Interbase 6.0*, beredar secara luas di internet, namun beberapa waktu kemudian Borland tidak lagi menetapkan versi *Interbase* secara *open source* namun bersifat komersial, hal ini menunjukkan bahwa peredaran *source code* hasil pengembangan lebih lanjut dari *Interbase* tidak dapat lagi dimanfaatkan secara bebas. Berdasarkan *source code Interbase* yang pertama kali di rilis oleh Borland programmer-programer yang tergabung kedalam sebuah tim pengembang perangkat lunak tertarik mengembangkan software pengolah basis data tersebut dan diberi nama *Firebird*.

Pengembangan demi pengembangan terus dilakukan, kemampuan *Firebird* dari versi ke versi terus menunjukkan fitur-fitur mempunyai yang terdapat dalam standar ANSI SQL-99 dan SQL-2003. RDBMS ini telah dilakukan ujicoba untuk kompatibelnya terhadap system operasi dan terbukti dapat berjalan dengan baik pada system operasi Linux, Windows

maupun sejumlah platform unix. *Firebird* yang merupakan hasil pengembangan dari *source code* milik *Interbase 6.0* yang dikembangkan oleh borlan kini pengelolaan dan pengembangannya berada dibawah naungan *FirebirdSQL Foundation*. *Software* Basis data ini dapat digunakan untuk kebutuhan personal maupun keperluan bisnis skala menengah dengan penggunaan license IPL (*Interbase Public License*) dan IDPL (*Initial Developer's Public License*). Pemberlakuan license IPL karena pada dasarnya *Firebird* adalah hasil pengembangan dari *source code* milik *Interbase*, sedangkan license IDP adalah license yang diberlakukan untuk sebagian *code* dan konsep yang dikembangkan oleh *Firebird foundation*.

a) Kelebihan *Firebird*

1. *License dan penggunaan yang Fleksibel*, (1) *firebird* dapat digunakan secara gratis untuk kebutuhan personal maupun kelompok namun tidak diperkenankan untuk melakukan klaim kepemilikan terhadap *source code*, untuk mempertegas hal tersebut dikatakan bahwa file license yang terdapat dalam folder *source code Firebird* tidak diperkenankan untuk dihapus atau dimodifikasi. (2) *Firebird* mengizinkan setiap orang yang hendak melakukan modifikasi terhadap *source code* yang tentunya hasil dari modifikasi tersebut harus dilicense-kan dengan license yang sama, sehingga tidak dapat diperjual belikan. (3). Bagi pengembang perangkat lunak *DBMS* dapat memanfaatkan *source code Firebird*, untuk membuat software *DBMS* lainnya dengan catatan bagian dari perangkat lunak tersebut yang memanfaatkan *source code Firebird* harus tetap menggunakan license yang sama (IPL dan IDPL) sedangkan bagian lain yang dikembangkan dapat memilih license yang berbeda.
2. *Community developer*, Perangkat lunak *DBMS Firebird* memiliki komunitas pengembang meski tidak sebesar komunitas *MySQL* dan *Oracle*, dengan aktifnya komunitas ini kemampuan, dan keamanan dari *DBMS* ini dapat terus dikembangkan, tidak hanya itu komunitas ini juga dapat

dijadikan sebagai media untuk bertukar permasalahan yang dihadapi selama penggunaan DBMS *Firebird*.

3. *MultiOS dan Minimum Resources, Firebird* merupakan jenis DBMS yang dapat digunakan pada setiap system operasi, baik Linux, Windows, MacOS dan Unix, dalam implementasinya *Firebird* tidak membutuhkan resources/sumberdaya computer yang memadai dalam implementasinya (Minimum Memory 1 Megabyte), penggunaan dengan spesifikasi computer yang minimal tidak membuat kemampuan *firebird* menjadi tidak maksimal.

b) Kekurangan *Firebird*

1. *Minimum User*, terdapat banyak kelebihan yang dimiliki oleh *Firebird*, akan tetapi pengguna dari DBMS ini relative lebih sedikit dibandingkan dengan DBMS lainnya. Beberapa factor yang dianggap sebagai penyebabnya antara lain, *Firebird* tidak didukung oleh beberapa perusahaan besar yang bergerak pada Teknologi Informasi seperti Oracle, IBM, Microsoft, dan sebagainya sehingga membatasi perkembangan fitur dan promosi dari DBMS ini.
2. *Website official* dari *Firebird* memiliki tampilan yang tidak meyakinkan bahwa DBMS ini memiliki fitur, layanan, fasilitas, keamanan dan masa depan yang baik pada bidang teknologi informasi, bahkan beberapa pengguna menganggapnya telah ketinggalan jaman, hal ini menjadi salah satu alasan utama yang membuat sedikitnya peminat dari DBMS ini.

#### 4. Microsoft SQL Server

Pengembang system tentu sudah merasa tidak asing lagi dengan nama perangkat lunak DBMS *Microsoft SQL Server*, perangkat lunak ini adalah salah satu program unggulan yang dikembangkan oleh Microsoft dengan menggunakan Transact SQL sebagai Query utamanya yang dimana query ini merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Perangkat lunak DBMS jenis ini dirancang

secara khusus untuk arsitektur system informasi yang berbasis client server, beberapa fitur andalan yang dimiliki oleh *Microsoft SQL Server* yaitu memiliki kemampuan untuk membuat basis data mirror serta clustering.

*SQL (Structure Query Language)* adalah salah satu bahasa yang digunakan oleh computer untuk mengakses data-data yang tersimpan dalam sebuah basis data, sampai saat ini hampir seluruh perangkat lunak DBMS sudah menggunakan *SQL* untuk membantu melakukan koreksi dan implementasi dari pengolahan dan management basis data.

*a) Kelebihan Microsoft SQL Server*

1. Bekerja dengan sangat baik pada system operasi berbasis Windows, karena sama-sama merupakan produk yang dikembangkan oleh *Microsoft*.
2. Terpusat, DBMS ini memiliki pengelolaan yang dapat dilakukan secara terpusat, sehingga tidak memungkinkan terjadinya kesalahan dalam pengelolaan, manajemen, dan implementasi basis data.
3. *Fitur Recovery dan Restore data*, DBMS ini memiliki fasilitas yang digunakan untuk membantu dalam memecahkan permasalahan data yang hilang atau rusak akibat dari kesalahan management yang dilakukan.
4. *Keamanan*, Fasilitas keamanan yang dimiliki oleh DBMS *Microsoft SQL Server* dianggap baik karena terbukti aman dari ujicoba serangan yang telah dilakukan.

*b) Kekurangan Microsoft SQL Server*

1. *Berbayar*, untuk dapat menggunakan produk *Microsoft SQL Server* *Microsoft* telah memasang harga yang cukup tinggi bagi para penggunanya.
2. *Tidak multi OS*, *Microsoft SQL Server* hanya dapat digunakan pada system operasi yang berbasis Windows.
3. *DBMS Microsoft SQL Server* terkenal dengan pasangan bahasa pemrograman yang mendukungnya yaitu *.net*, selain itu DBMS ini tidak akan mampu bekerja secara maksimal ketiga dipadukan dengan bahasa pemrograman lainnya.



4. *DBMS Microsoft SQL Server* dianggap tidak mampu untuk di implementasikan pada skala data yang besar tidak seperti halnya dengan *DBMS Oracle*.

## 5. PostgreSQL

PostgreSQL atau (*Post-gress-SQL*) adalah sebuah perangkat lunak DBMS yang dikembangkan oleh mahasiswa dan staff programmer University of California, Berkeley dibawah arahan Professor Michael Stonebraker. Perangkat lunak yang dikembangkan bersifat *Opensource*, DBMS ini disebarluaskan secara bebas dan gratis menurut *Perjanjian License BSD*. Perangkat lunak ini adalah salah satu perangkat lunak DBMS yang paling banyak digunakan saat ini, selain MySQL dan Oracle. Banyaknya peminat dari DBMS ini karena DBMS ini menyediakan beberapa fitur yang berguna untuk kebutuhan replikasi basis data. fitur-fitur yang disediakan antara lain DB Mirror, PGPool, Slony, PGCluster, dan lainnya.

Versi pertama yang dikembangkan oleh PostgreSQL dianggap tidak memenuhi standar ANSI-SQL92 atau dengan kata lain PostgreSQL tidak memenuhi standar pengolahan data berskala besar, pengembangan demi pengembangan dilakukan untuk mencapai standar tersebut, versi PostgreSQL yang diluncurkan setelah itu adalah versi PostgreSQL yang telah mencapai standar ANSI-SQL selain itu juga PostgreSQL yang diluncurkan pada waktu itu dianggap melebihi standar ANSI-SQL, yang berarti kemampuan PostgreSQL dalam pengolahan data dengan skala besar melebihi kemampuan DBMS lain diantaranya adalah Oracle. Pada saat bersamaan PostgreSQL dinobatkan oleh Linux Award sebagai DBMS pilihan dengan performa dan kemampuan yang optimal dan menempatkan Oracle pada tempat ke-dua.

Perbedaan yang paling mendasar antara postgres dengan perangkat lunak DBMS lainnya terdapat pada kemampuan DBMS PostgreSQL yang memungkinkan user dapat mendefinisikanSQL-nya secara sendiri terutama untuk pembuatan function. Hal ini dimungkinkan karena informasi yang terdapat pada PostgreSQL tidak hanya tabel dan kolom melainkan type, fungsi, cara akses, dan beberapa informasi lain. Keseluruhan fasilitas tersebut

dihimpun dalam bentuk class yang memungkinkan user untuk merubahnya. Dengan model class seperti ini PostgreSQL dianggap lebih mudah untuk dikembangkan pada tingkatan user, selain itu PostgreSQL dapat mendefinisikan sebuah tabel sebagai turunan tabel lain. Dimana pada DBMS konvensional hal itu hanya bisa dilakukan dengan mengganti source code-nya atau menggunakan modul tambahan.

*a) Kelebihan PostgreSQL*

1. *PostgreSQL* adalah perangkat lunak DBMS yang mendapat dukungan antar muka dari berbagai bahasa pemrograman seperti Perl, PHP, C++, Python, Java dan beberapa bahasa pemrograman lain.
2. *PostgreSQL* dapat beroperasi pada banyak system operasi antara lain (Windows, Linux, UNIX (AIX, HP-UX, BSD, SGI IRIX, Solaris, Mac OS X, Tru64).
3. *PostgreSQL* mendukung sebagian besar standar SQL dan menawarkan banyak fitur seperti Complex SQL queries, SQL Sub-selects, Triggers, Foreign keys, Trigger, Transactions, Views, Multiversion concurrency control (MVCC), Hot Standby (as of 9.0), Streaming Replication (as of 9.0).
4. *PostgreSQL* memiliki arsitektur multiproses (forking) yang berarti memiliki stabilitas yang lebih tinggi dari DBMS lainnya.
5. *PostgreSQL* telah menyediakan banyak fasilitas yang dimiliki oleh DBMS komersial pada umumnya.
6. *PostgreSQL* memiliki tipe data geometri (seperti titik, garis, lingkaran, poligon) yang dimungkinkan dapat bermanfaat bagi user yang mengembangkan aplikasi ilmiah tertentu.
7. *PostgreSQL* juga memiliki kemampuan untuk membuat konektivitas dengan database lain seperti *pgdump*, *Interbase*, *pgaccess* dan hampir semua database yang terdapat pada system operasi *Linux*.

*b) Kekurangan PostgreSQL*

1. Tidak begitu populer dikalangan programmer web karena dianggap tidak begitu cocok jika diimplementasikan pada

aplikasi berbasis website dibandingkan dengan DBMS MySQL.

2. Pengembangan DBMS *PostgreSQL* tidak memperhatikan kecepatan pengolahan data serta dianggap kurang unggul dalam ketersediaan fungsi built-in.
3. Model dan Arsitektur dari *PostgreSQL* dengan multiproses yang dimilikinya dianggap sulit diterapkan pada system operasi berbasis Windows, sebab Windows menggunakan konsep thread-oriented kecuali dengan memanfaatkan lapisan emulasi Cygwin.

## 6. MongoDB

MongoDB berasal dari kata "*Humongous*" yang merupakan sebuah *Document-Oriented Database* yang bersifat *OpenSource* yang awalnya dibuat dengan bahasa C++, Perangkat lunak ini dikembangkan oleh *10gen* sejak Oktober 2007, namun baru dipublikasikan pada Februari 2009, Selain karena performanya 4 kali lebih cepat dibandingkan MySQL serta mudah diaplikasikan, karena telah tergabung juga sebagai modul PHP, MongoDB berbeda dengan perangkat lunak DBMS yang bersifat Relational seperti *MySQL*, *Oracle*, *PostgreSQL*, dan *Microsoft SQL Server*.

MongoDB adalah perangkat lunak basis data yang bersifat NoSQL atau yang sering disebut dengan istilah Not Only SQL yang merupakan istilah system manajemen basis data yang berbeda dari konsep manajemen basis data relasional pada umumnya yang memanfaatkan SQL, Dalam implementasinya MongoDB tidak mengenal yang namanya tabel, kolom dan baris, jadi tidak ada schema dalam MongoDB (schema-less), unit paling kecil dari MongoDB adalah document, sedangkan kumpulan dari document adalah collection, seperti halnya dalam database relasional document itu ibarat record dan collection pada sebuah tabel. document dalam MongoDB dapat memiliki atribut yang berbeda-beda dengan document yang lainnya walaupun dalam satu collection.

a) *Kelebihan MongoDB*

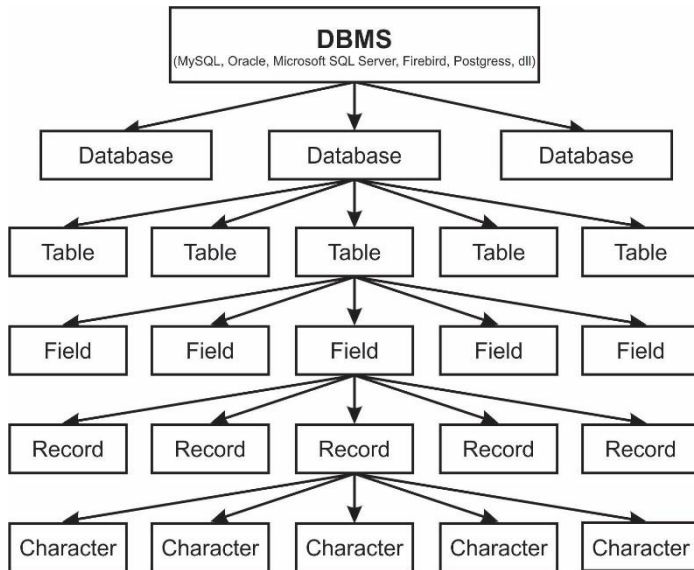
1. Performa yang ditawarkan MongoDB lebih cepat disebabkan oleh *Memcached* dan format dokumennya yang berbentuk seperti JSON.
2. Pengguna tidak perlu membuat struktur tabel, karena MongoDB akan otomatis membuat struktur tabelnya pada saat proses insert (fleksibel skema).
3. MongoDB juga sudah mendukung C, C++, C#, Erlang, Haskell, Java, JavaScript, .NET(C# F#, PowerShell), Lips, Perl, PHP, Python, Ruby dan Scala.
4. *Cross-platform*, sehingga dapat digunakan di Windows, Linux, OS X dan Solaris.
5. Proses CRUD (*Create, Read, Update, Delete*) terasa sangat ringan.
6. *Map/Reduce*, akan sangat membantu ketika kita melakukan operasi agregasi. Dimana semua entry datangnya dari collection dan outputnya pun akan menjadi collection juga. Kalau di MySQL biasanya kita menggunakan query GROUP BY.
7. *GridFS*, spesifikasi yang digunakan untuk menyimpan data yang sangat besar.

b) *Kekurangan MongoDB*

1. Belum banyak hosting yang support.
2. Fleksibilitas dalam query (sebagai contoh tidak adanya JOIN).

## F. Hirarki Basis Data

Hirarki merupakan sebuah gambaran yang digunakan untuk memahami sebuah konsep yang kompleks, dimana konsep tersebut diuraikan kedalam elemen-elemen secara hirarkis berdasarkan tingkatannya agar dapat memahami struktur, keterkaitan, dan interaksi dalam sebuah suatu system. Hirarki dalam basis data diurutkan dari *perangkat lunak DBMS, Database, Table, Field, Record Data*, dan *Character* seperti yang diuraikan pada gambar berikut ini:



Gambar 3. Hirarki Database

Untuk memahami gambar diatas akan diberikan penjelasan sebagai berikut:

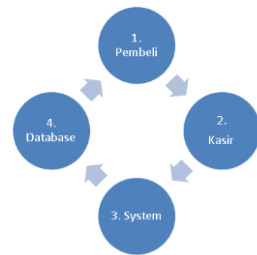
1. *DBMS*, seperti yang telah dijelaskan sebelumnya komponen ini adalah komponen utama yang harus disediakan dalam membangun sebuah basis data, baik yang bersifat gratis maupun berbayar, DBMS adalah sebuah perangkat lunak yang digunakan untuk membuat, mengelola, memanajemen dan mengamankan basis data, seperti MySQL, Oracle, Firebird, Microsoft SQL Server, dan PostgreSQL.
2. *Database*, merupakan sebuah media yang di rancang menggunakan salah satu dari perangkat lunak DBMS, media yang dirancang tersebut akan digunakan untuk menampung data yang menjadi informasi penting dalam pengelolaan.
3. *Table*, merupakan bagian terpenting dari keberadaan database, tabel dapat berjumlah banyak maupun sedikit dalam sebuah database, tabel dalam database tersusun oleh field, dan record data.

4. *Field*, adalah bagian dari tabel yang berupa aturan dalam penulisan (input) record data. atau dengan istilah lain *Field* merupakan kolom dari sebuah tabel yang memiliki ukuran type data tertentu untuk ditampung.
5. *Record*, merupakan informasi yang ditampung oleh *Field*. *Record* data terbentuk dari kumpulan *character* yang diinputkan.
6. *Character*, adalah bagian data yang terkecil dari basis data, dapat berupa karakter numeric, huruf, maupun karakter khusus (special characters) yang membentuk suatu item (1 item yang terbentuk dari satu/lebih *character* disebut sebagai 1 *Record data*).

## G. Evaluasi Materi

Jawablah soal-soal berikut ini dengan benar:

1. Gambar disamping adalah gambar tentang pemanfaatan database dalam kehidupan sehari hari. berdasarkan alur antar komponen dari gambar tersebut, komunikasi berlangsung dari pembeli menuju kasir, system dan database, dapatkah anda menjelaskan alur balasan dari hasil respon database tersebut?



2. Dapatkah anda menyebutkan komponen-komponen yang mendukung keberadaan DBMS? Serta jelaskan konsep dukungan yang diberikan terhadap DBMS?
3. “DBMS berperan sebagai perantara user dengan basis data, keberadaan DBMS memungkinkan user dapat menambahkan, mengurangi, memodifikasi, menghapus serta mengelola system aplikasi yang telah dikembangkan sebelumnya” berikan pendapat anda tentang teori tersebut?
4. Dapatkah anda menyebutkan data seperti apa yang dapat ditampung serta dikelola oleh DBMS?
5. Dapatkah anda memberikan perbandingan terkait dengan keunggulan dan kekurangan dari berbagai macam DBMS?

## BAGIAN 2

### Praktikum dasar penggunaan DBMS

**Setelah mempelajari Bagian 2 ini mahasiswa mampu:**

1. Melakukan Praktikum instalasi XAMPP dan mengakses MySQL Server melalui CMD Windows
2. Melakukan Praktikum dalam Membuat dan menampilkan database
3. Memahami tipe-tipe tabel dan field (tipe data) dalam MySQL
4. Melakukan praktikum menghapus, membuat, menampilkan dan mendeskripsikan database
5. Melakukan Praktikum dalam Mengubah, menghapus, menambah, dan menampilkan RECORD pada database

#### **A. XAMPP sebagai software Pendukung**

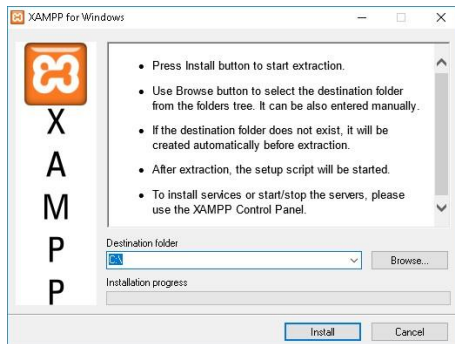
XAMPP adalah sebuah perangkat lunak (*software*) yang berlicense *GNU General Public License* atau dengan istilah lain, perangkat lunak ini dapat diperoleh dan digunakan secara gratis, XAMPP merupakan uraian dari istilah Multi system operasi, Apache, MySQL, PHP dan Perl. Perangkat lunak ini dikembangkan oleh sebuah perusahaan yang bernama *Apache Friends* dengan tujuan membuat server pribadi baik server website, database server, maupun beberapa kebutuhan pembangunan server lainnya. Dalam buku ini akan membahas terkait dengan pemanfaatan XAMPP pada system operasi windows, untuk mendukung aktivitas praktikum mahasiswa dalam membuat dan mengelola basis data melalui DBMS MySQL. Untuk memulai langkah instalasi XAMPP anda diwajibkan untuk memperoleh perangkat lunaknya terlebih dahulu melalui situs resmi <http://www.apachefriends.org/download.html>. Setelah mengunjungi laman resmi tersebut anda akan dihadapkan dengan pilihan berbagai jenis XAMPP sesuai dengan system operasi yang anda gunakan, dalam buku ini penulis menggunakan XAMPP dengan versi 2.5.8.

Sebelum anda memahami materi-materi yang disajikan dalam buku ini, terlebih dahulu anda memasang perangkat lunak XAMPP, dalam perangkat lunak tersebut terdapat DBMS MySQL yang digunakan sebagai database server, Pada buku ini anda akan dibimbing secara bertahap untuk dapat mempraktikkan setiap materi yang disampaikan.

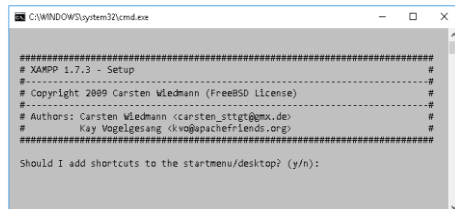
XAMPP memiliki 3 bagian penting dalam mendukung kinerja pengembang (developer) dalam penggunaannya yaitu (1) *Htdocs*, adalah lokasi yang berbentuk folder dari hasil instalasi XAMPP, folder tersebut digunakan untuk menampung seluruh file seperti file PHP, HTML, File lain yang terkait, (2) *PhpMyadmin* adalah sebuah layanan yang digunakan untuk mengakses, membuat, menghapus serta mengelola database, (3) *Control Panel* yang merupakan sebuah layanan pengelolaan XAMPP, terkait dengan start dan stop dari servicenya.

## B. Praktikum 1: Melakukan instalasi XAMPP

- ▶ Setelah mendapatkan software XAMPP melalui situs resmi, anda dapat memulai instalasinya dengan melakukan klik ganda pada master aplikasi tersebut. Gambar disamping menjelaskan bahwa hasil instalasi software XAMPP ditujukan pada **Partisi C:\**

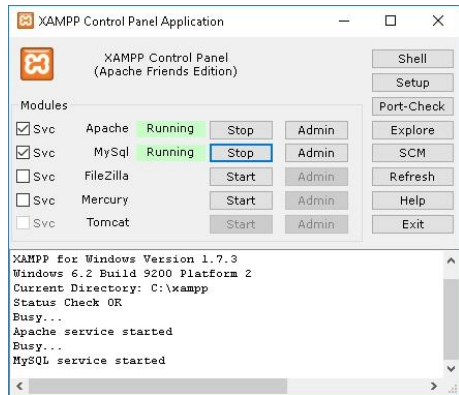


- ▶ Klik Install untuk memulai proses instalasi XAMPP Gambar disamping menjelaskan bahwa **shortcuts** dari hasil install akan terdapat pada start menu atau Desktop, ketik **huruf y** lalu Enter untuk mengakhiri proses install XAMPP





- ▶ Untuk mengoperasikan XAMPP lakukan klik ganda pada shortcuts XAMPP yang terdapat pada Desktop atau start menu. Gambar disamping merupakan tampilan dari control panel aplikasi XAMPP, yang berfungsi untuk menjalankan service dari 5 fasilitas yang disediakan, pada praktikum ini anda perlu mengaktifkan **service Apache** dan **MySQL** saja, lalu klik tombol Start dari dua service tersebut.



### C. Praktikum 2: Akses MySQL Server melalui Command Prompt

- ▶ Langkah pertama yang dilakukan yaitu mengakses Command Prompt dengan menekan tombol START → RUN lalu ketikkan cmd, dan tekan tombol Enter.
- ▶ Selanjutnya Ketikkan perintah berikut pada Command Prompt

```

C:\WINDOWS\system32\cmd.exe - C:\xampp\mysql\bin\mysql -u root -p
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Fathir>C:\xampp\mysql\bin\mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.41 Source distribution

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

mysql> _

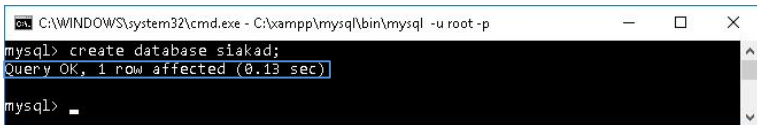
```

- ▶ Langkah yang dilakukan diatas digunakan untuk mengakses MySQL melalui Command Prompt, perintah `C:\xampp\mysql\bin` menunjukkan bahwa lokasi MySQL dapat diakses melalui Partisi, Folder dan Sub Folder tersebut, sedangkan perintah `MySQL -u root -p` adalah perintah yang menunjukkan bahwa untuk

mengakses MySQL membutuhkan username “root” serta password yang secara default dikosongkan.

#### D. Praktikum 3: Membuat dan menampilkan Database

- ▶ Untuk membuat database dapat menggunakan perintah `create database <nama database>`; yang dalam praktikum ini kita akan membuat sebuah database dengan nama “siakad”.

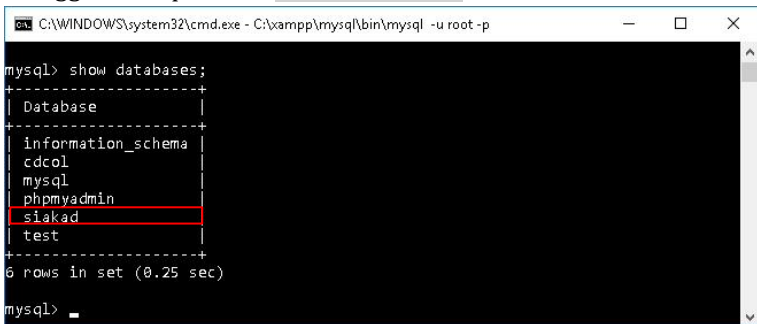


```
C:\WINDOWS\system32\cmd.exe - C:\xampp\mysql\bin\mysql -u root -p
mysql> create database siakad;
Query OK, 1 row affected (0.13 sec)

mysql> _
```

Baris informasi dalam kotak biru pada gambar diatas menunjukan bahwa perintah `create database siakad;` telah berhasil dieksekusi (database siakad telah berhasil diciptakan) dengan waktu eksekusi perintah selama 0.13 detik.

- ▶ Selanjutnya, untuk menampilkan database yang telah dibuat dapat menggunakan perintah `show databases;`

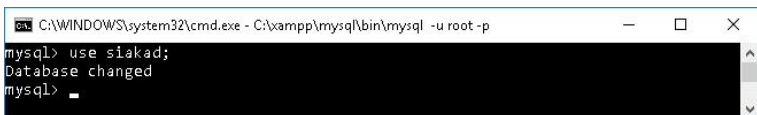


```
C:\WINDOWS\system32\cmd.exe - C:\xampp\mysql\bin\mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| mysql     |
| phpmyadmin |
| siakad    |
| test      |
+-----+
6 rows in set (0.25 sec)

mysql> _
```

Lingkaran merah yang terdapat pada gambar diatas adalah nama database “siakad” yang sebelumnya telah dibuat.

- ▶ Untuk dapat melakukan pengelolaan terhadap database siakad terlebih dahulu mengetikan format perintah `use <nama database>`; atau `use siakad;`



```
C:\WINDOWS\system32\cmd.exe - C:\xampp\mysql\bin\mysql -u root -p
mysql> use siakad;
Database changed

mysql> _
```

Setelah berhasil melakukan praktikum diatas, dapat dipastikan bahwa database siacad sudah siap untuk dikelola lebih lanjut.

## **E. Memahami Tipe-tipe Tabel dan Field dalam MySQL**

Sebelum melakukan pengelolaan terhadap database yang telah di buat sebelumnya terlebih dahulu anda harus memahami Tipe-tipe Tabel dan Field dalam MySQL.

### **1. Tipe-tipe Tabel dalam MySQL**

Berdasarkan Hirarki sebuah basisdata, Pembuatan tabel adalah langkah yang selanjutnya dilakukan setelah pembuatan Database, dalam MySQL pengguna dapat menentukan tipe pada setiap tabel yang dibuat hal ini bergantung pada konfigurasi yang dilakukan pada awal install aplikasi, dalam implementasinya MySQL memiliki 3 tipe data yang utama, antara lain MyISAM, InnoDB, HEAP, BDB, Archieve, CSV, NDB Table, dan Federated. Pada bagian ini akan dibahas 3 (tiga) tipe data yang paling populer yaitu MyISAM, InnoDB, dan HEAP. Penetapan tabel pada MySQL yang diinstal melalui aplikasi XAMPP secara default tipe tabelnya MyISAM.

#### **a) MyISAM**

Tipe tabel MyISAM adalah tipe tabel yang dianggap paling sederhana dari tipe tabel lain, selain itu stabil dan mudah untuk digunakan Serta memiliki kemampuan dalam pengindeksan teks yang lengkap. Tipe tabel MyISAM terbagi kedalam 3 jenis tabel MyISAM, yaitu:

- 1) **MyISAM Static**, adalah jenis tabel yang digunakan untuk kolom dalam tabel yang ber-ukuran pasti (fixed). Istilah berukuran pasti menjelaskan bahwa dalam kolom tidak memiliki tipe data seperti VARCHAR, TEXT dan BLOB.
- 2) **MyISAM Dynamic**, adalah jenis tabel yang bersifat dinamis, karena mampu menampung tipe data seperti VARCHAR, TEXT dan BLOB.
- 3) **MyISAM Compressed**, adalah jenis tabel yang dihasilkan dari kompresi antara tabel MyISAM Static dan Dynamic, hasil kompresi yang dilakukan akan membuat jenis tabel ini lebih kecil dari segi kapasitas/ukuran. Tabel yang telah

terkompresi memiliki batasan operasi seperti perintah INSERT, UPDATE dan DELETE.

**b) InnoDB**

Berbeda dengan tipe tabel MyISAM, tipe tabel ini memiliki kelebihan dalam mendukung proses transaksi antar tabel, mendukung Row-Level-Locking, Mendukung Foreign- Key Constraints, serta Crash recovery.

**c) HEAP**

Tipe tabel HEAP adalah tipe tabel yang memanfaatkan RAM (*Random Acces Memory*) sebagai media penyimpananya, karena sifat memory yang temporary (sementara), tabel yang dibuat dengan jenis ini secara otomatis akan terhapus (hilang) jika server MySQL dimatikan.

**2. Tipe-tipe Field pada MySQL**

Dalam MySQL terdapat beberapa tipe data yang paling umum atau sering digunakan oleh programmer untuk setiap kolom (field) pada tabel antara lain:

**1. Tipe Numeric**, merupakan tipe data (karakter) yang digunakan untuk menampung data numeric (angka), data yang ditampung dimungkinkan untuk dikenai operasi aritmatika seperti operasi penjumlahan, pengurangan, perkalian, serta operasi pembagian. Berikut ini akan dibahas field (kolom) yang termasuk kedalam tipe numeric:

**1) TINYINT (Tiny Integer)**, adalah tipe data yang digunakan untuk menampung bilangan bulat positif dan bilangan bulat negatif, tipe data ini memiliki jangkauan bilangan antara -128 s/d 127, besaran ukurannya yaitu **1 byte (8 bit)**, dengan jumlah maksimal angka yang mampu ditampung yaitu **0 s/d 255**.

**2) SMALLINT (Small Integer)**, adalah tipe data yang digunakan untuk menampung bilangan bulat positif dan bilangan bulat negatif, tipe data ini memiliki jangkauan bilangan antara -32.768 s/d 32.767, besaran ukurannya yaitu **2 byte (16 bit)**, dengan jumlah maksimal angka yang mampu ditampung yaitu **0 s/d 65.535**.

- 3) **MEDIUMINT (Medium Integer)**, adalah tipe data yang digunakan untuk menampung bilangan bulat positif dan bilangan bulat negatif, tipe data ini memiliki jangkauan bilangan antara **-8.388.608 s/d 8.388.607**, besaran ukurannya yaitu **3 byte (24 bit)**, dengan jumlah maksimal angka yang mampu ditampung yaitu **0 s/d 16.777.215**.
- 4) **INT (Integer)**, adalah tipe data yang digunakan untuk menampung bilangan bulat positif dan bilangan bulat negatif tipe data ini memiliki jangkauan bilangan antara **-2.147.483.648 s/d 2.147.483.647**, besaran ukurannya yaitu **4 byte (32 bit)**, dengan jumlah maksimal angka yang mampu ditampung yaitu **0 s/d 4.294.967.295**.
- 5) **BIGINT (Big Integer)**, adalah tipe data yang digunakan untuk menampung bilangan bulat positif dan bilangan bulat negatif, tipe data ini memiliki jangkauan bilangan antara **-9.223.372.036.854.775.808 s/d 9.223.372.036.854.775.807**, besaran ukurannya yaitu **8 byte (64 bit)**, dengan jumlah maksimal angka yang mampu ditampung yaitu **0 s/d 18.446.744.073.709.551.615**.
- 6) **FLOAT (Floating Point)**, adalah tipe data yang digunakan untuk menampung bilangan pecahan positif dan pecahan negatif presisi tunggal, tipe data ini memiliki jangkauan bilangan antara **-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38**, besaran ukurannya yaitu **4 byte (32 bit)**, penulisan tipe data ini ditandai dengan terdapatnya angka desimal dibelakang tanda "titik". Contoh 0.6 atau 3.30.
- 7) **DOUBLE**, adalah tipe data yang sejenis dengan Float digunakan untuk menampung bilangan pecahan positif dan pecahan negatif presisi ganda, tipe data ini memiliki jangkauan bilangan yang lebih panjang dibandingkan dengan tipe data Float yaitu antara **-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308**, besaran ukurannya yaitu **8 byte (64 bit)**.
- 8) **DECIMAL**, adalah tipe data yang digunakan untuk menampung bilangan pecahan positif dan pecahan negatif,

tipe data ini memiliki jangkauan bilangan yang lebih panjang dibandingkan dengan tipe data Float yaitu antara - **1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308**, besaran ukurannya yaitu **8 byte (64 bit)**.

2. **Tipe Date and Time**, adalah tipe data yang digunakan untuk menampung data satuan waktu (Tahun, Bulan, Tanggal, Jam, Menit, dan Detik). Berikut ini akan disajikan field (kolom) pada MySQL yang termasuk kedalam tipe data Date and Time.
  - 1) **DATE**, adalah tipe data yang digunakan untuk menampung data tanggal, dengan format (Tahun-Bulan-Hari) atau **(YYYY-MM-DD)**, tipe data ini memiliki jangkauan antara **1000-01-01 s/d 9999-12-31**, besaran ukurannya yaitu **3 byte**.
  - 2) **TIME**, adalah tipe data yang digunakan untuk menampung data waktu, dengan format (Jam-Menit-Detik) atau **(HH:MM:SS)**, tipe data ini memiliki jangkauan antara - **838:59:59 s/d +838:59:59**, besaran ukurannya yaitu **3 byte**.
  - 3) **DATETIME**, tipe data ini adalah perpaduan dari 2 (dua) tipe data sebelumnya yang digunakan untuk menampung data waktu (Tahun, Bulan, Tanggal, Jam, Menit, dan Detik), dengan format (Tahun-Bulan-Tanggal Jam:Menit:Detik) atau **(YYYY-MM-DD HH:MM:SS)**, tipe data ini memiliki jangkauan antara **'1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'**, besaran ukurannya yaitu **8 byte**.
  - 4) **YEAR**, adalah tipe data yang digunakan untuk menampung data tahun saja, dengan format **(YYYY)**, tipe data ini memiliki jangkauan antara **1900 s/d 2155**, besaran ukurannya yaitu **1 byte**.
3. **Tipe String (Text)**, adalah tipe data yang digunakan untuk menampung data **string (text)**. Tipe data ini dimungkinkan untuk dikenai operasi aritmatika seperti penjumlahan, pengurangan, perkalian serta pembagian, berikut tipe field pada MySQL yang termasuk kedalam tipe string:

- 1) **CHAR**, adalah tipe data yang digunakan untuk menampung data string dengan ukuran tetap, tipe data ini memiliki jangkauan antara **0 s/d 255 karakter**.
  - 2) **VARCHAR**, adalah tipe data yang digunakan untuk menampung data string dengan ukuran dinamis, tipe data ini memiliki jangkauan antara **0 s/d 65.535 karakter**.
  - 5) **TINYTEXT**, adalah tipe data yang digunakan untuk menampung data text dengan ukuran kecil, tipe data ini memiliki jangkauan antara **0 s/d 255 karakter**, besar ukurannya yaitu **255 byte**.
  - 3) **TEXT**, adalah tipe data yang digunakan untuk menampung data text dengan ukuran sedang, tipe data ini memiliki jangkauan antara **0 s/d 65.535 ( $2^{16}-1$ ) karakter**, besar ukurannya yaitu **64 KB**.
  - 4) **MEDIUM TEXT**, adalah tipe data yang digunakan untuk menampung data text dengan ukuran besar, tipe data ini memiliki jangkauan antara **0 s/d 16.777.216 ( $2^{24}-1$ ) karakter**, besar ukurannya yaitu **16 MB**.
  - 6) **LONG TEXT**, adalah tipe data yang digunakan untuk menampung data text dengan ukuran sangat besar, tipe data ini memiliki jangkauan antara **0 s/d 4.294.967.295 ( $2^{32}-1$ ) karakter**, besar ukurannya yaitu **4 GB**.
4. **Tipe BLOB (Biner)**, adalah tipe data yang digunakan untuk menampung data versi binary dari suatu file atau object. BLOB merupakan singkatan dari Binary Large Object, dimana karakter akan disimpan dalam bentuk bit, klasifikasi besar dan kecilnya huruf akan dibedakan sesuai dengan nilai karakternya. Contoh A tidak sama dengan a. berikut ini akan dijelaskan tipe field pada MySQL yang termasuk dalam tipe data BLOB:
- 1) **BIT**, adalah tipe data yang digunakan untuk menampung data biner dengan ukuran sangat kecil yaitu 64 digit biner.
  - 2) **TINYBLOB**, adalah tipe data yang digunakan untuk menampung data biner dengan **ukuran kecil** yaitu **255 byte**.

- 3) **BLOB**, adalah tipe data yang digunakan untuk menampung data biner dengan ukuran **sedang** yaitu **65.535 atau  $2^{16} - 1$  byte**.
- 4) **MEDIUMBLOB**, adalah tipe data yang digunakan untuk menampung data biner dengan **ukuran besar** yaitu **16.777.215 atau  $2^{24} - 1$  byte**.
- 5) **LONGBLOB**, adalah tipe data yang digunakan untuk menampung data biner dengan **ukuran sangat besar** yaitu **4.294.967.295 atau  $2^{32} - 1$  byte**.

## 5. Tipe Data Lain

Selain tipe data yang dijelaskan diatas, MySQL juga memiliki beberapa tipe data lain diantaranya yaitu:

- 1) **ENUM**, adalah tipe data yang digunakan untuk menampung nilai dari data yang telah ditentukan sebelumnya, pilihan ini dapat berisi 1 s/d 65.535 pilihan string. Contoh pada data pilihan Jurusan: `enum('TIK', 'Bahasa Inggris', 'Agama Islam', 'PGSD')`, **dimana satu orang user hanya dapat memilih 1 jurusan saja**.
- 2) **SET**, adalah tipe data yang digunakan untuk menampung nilai dari data yang telah ditentukan sebelumnya, pilihan ini dapat berisi 1 s/d 64 pilihan string. Contoh pada data pilihan hoby: `SET('Membaca', 'Berenang', 'Olahraga', 'Travelling')`, **dimana satu orang user dapat memilih lebih dari 1 Hoby**.

## F. Aturan dasar dalam perancangan Database

Setelah memahami tipe Tabel dan Tipe Field yang terdapat pada DBMS MySQL buku ini akan mengajak anda memahami beberapa aturan dasar dalam merancang sebuah database.

1. **Redudancy dan peran *Primary Key***, Dalam database terdapat field (kolom), yang mengandung record (data), record data tersebut tidak boleh bernilai ganda. Contoh: terdapat **tabel Mahasiswa** dengan field dan record data sebagai berikut:

Nama Mahasiswa	Jurusan	JK	Agama	Alamat
Arif	Ekonomi	L	Islam	Mataram
Arif	Ekonomi	L	Islam	Mataram



Record data pada **tabel mahasiswa** diatas menunjukkan terjadinya redundancy data dimana terdapat kesamaan antara record. **Pertanyaan: Apakah kesamaan record tersebut mungkin terjadi?? YA, Sangat Mungkin terjadi (terdapat dua orang mahasiswa yang memiliki nama, jurusan, JK, agama dan alamat yang sama). Lalu bagaimana mencegahnya??** Perancang database harus membuat sebuah kolom “unik” pada tabel mahasiswa tersebut dengan bertujuan untuk membedakan antar record, kolom “unik” inilah yang disebut dengan “**Primary Key**” berikut contoh tabel yang menggunakan “**Primary Key**”:

NIM	Nm_Mahasiswa	Jurusan	JK	Agama	Alamat
201601001	Arif	Ekonomi	L	Islam	Mataram
201601002	Arif	Ekonomi	L	Islam	Mataram

Field NIM pada tabel tersebut adalah sebuah *Primary Key* yang berguna untuk membedakan record. **Kesimpulan 1: walaupun dalam tabel tersebut memiliki dua orang mahasiswa dengan identitas nama, jurusan, JK agama dan alamat yang sama namun akan dibedakan oleh keberadaan field NIM. Kesimpulan 2: setiap tabel harus memiliki field yang unik sebagai *Primary Key* agar tidak terjadi redundancy data.**

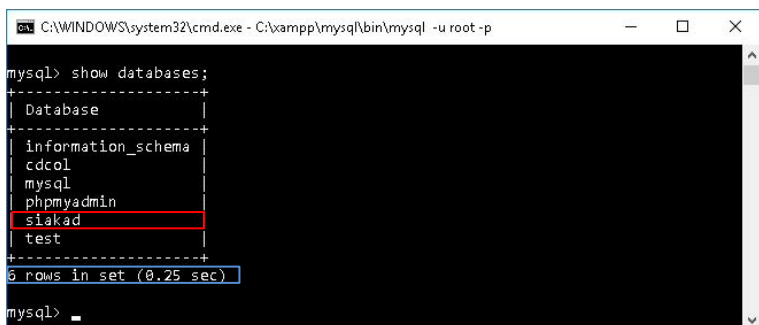
2. **Tipe dan ukuran data yang tepat**, dalam pemilihan tipe data, perancang database harus memikirkan bahwa tipe data dan ukurannya yang akan di Set pada setiap Field adalah tipe data dengan ukuran yang seharusnya. **Contoh: field Agama tidak mungkin ber-tipe data INTEGER, dan field JK tidak mungkin ber-tipe data TINYTEXT dengan panjang 255 karakter.** Berikut ini akan diberikan contoh penentuan tipe data, ukuran dan keterangan lain yang harus diperhatikan:

Field	NIM	Nm_Mahasiswa	Jurusan	JK	Agama	Alamat
Tipe data	Char	Varchar	Varchar	Varchar	Varchar	Text
Primary Key	Ya	Tidak	Tidak	Tidak	Tidak	Tidak
Ket- NULL	Harus	Harus	Harus	Harus	Boleh	Harus
	Berisi	Berisi	Berisi	Berisi	Kosong	Berisi
Ukuran	9	40	15	2	8	65.535

3. **Penamaan Identifier**, dalam database, tabel, dan kolom pada DBMS MySQL bersifat sangat sensitive, yang berarti perancang harus secara konsisten dalam menetapkan identifier. Contoh: penerapan lower-case, UPPER-CASE, camelCase, dsb. selain itu maksimum karakter untuk identifier tersebut berjumlah 64 karakter, dan sebaiknya menghindari penggunaan karakter khusus seperti *üüü*.

#### G. Praktikum 4: Menghapus Database

- ▶ Struktur Perintah SQL yang digunakan untuk Menghapus Database adalah **DROP DATABASE <nama database>**; sebelum melakukan penghapusan pada database sebaiknya anda memastikan terlebih dahulu nama database yang akan dihapus dengan mengetikkan kembali perintah **SHOW DATABASES**; seperti berikut ini:



```
C:\WINDOWS\system32\cmd.exe - C:\xampp\mysql\bin\mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcpl      |
| mysql     |
| phpmysqladmin |
| siakad    |
| test      |
+-----+
6 rows in set (0.25 sec)
mysql>
```

Nama Database “*siakad*” yang dilingkari merah diatas adalah nama database yang akan di hapus dengan perintah **DROP DATABASE siakad**; gambar berikut ini telah menunjukkan bahwa database dengan nama “*siakad*” telah berhasil dihapus.

Baris informasi dalam kotak biru pada gambar diatas menunjukkan bahwa perintah show databases; telah berhasil menampilkan 6 baris database dengan waktu eksekusi perintah selama 0.25 detik.

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

Untuk melanjutkan kembali praktikum yang telah direncanakan, silahkan anda membuat kembali database dengan nama “*siakad*” dengan perintah **CREATE DATABASE siakad;** lalu masuk kedalam database “*siakad*” tersebut untuk dapat mengelola database tersebut dengan perintah **USE siakad;**

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> create database siakad;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| phpmyadmin |
| siakad |
| test |
+-----+
6 rows in set (0.00 sec)

mysql> use siakad;
Database changed
mysql> _
```

## H. Praktikum 5: Membuat, Menampilkan, dan mendeskripsikan Database

Setelah menjalankan perintah **USE siakad;** seperti gambar diatas langkah selanjutnya yaitu memahami Struktur perintah yang digunakan untuk membuat sebuah tabel dalam database seperti contoh berikut:

```
CREATE TABLE <nama_tabel> (  
Field1 Tipe-data (Panjang-data) Info-NULL,  
Field2 Tipe-data (Panjang-data) Info-NULL,
```

```
Dst...
PRIMARY KEY (Field_key)
);
```

Setelah memahami struktur perintah SQL diatas, selanjutnya akan dipraktikan dalam pembuatan tabel yang sebenarnya dengan struktur tabel yang direncanakan sebagai berikut:

#### Rancangan Tabel mahasiswa

No	Field	Tipe Data	Panjang Data
1	id_mhs*	Varchar	10
2	nm_mhs	Varchar	45
3	alamat	Text	-
4	tmpt_lahir	Varchar	15
5	tgl_lahir	Date	-
6	jk	Varchar	2
7	agama	Varchar	10
8	asal_sekolah	Varchar	35
9	prodi	Varchar	25
10	email	Varchar	25
11	handphone	char	12

Rancangan tabel diatas akan di terapkan pada database dengan perintah SQL sebagai berikut:

```
CREATE TABLE mahasiswa (
  id_mhs VARCHAR (10) NOT NULL,
  nm_mhs VARCHAR (30) NOT NULL,
  alamat TEXT,
  tmpt_lahir VARCHAR (15),
  tgl_lahir DATE,
  jk VARCHAR (2),
  agama VARCHAR (10),
  asal_sekolah VARCHAR (35),
  prodi VARCHAR (25),
  email VARCHAR (25),
  handphone CHAR (12),
  PRIMARY KEY (id_mhs)
);
```

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> CREATE TABLE mahasiswa (
-> id_mhs VARCHAR (10) NOT NULL,
-> nm_mhs VARCHAR (30) NOT NULL,
-> alamat TEXT,
-> tmpt_lahir VARCHAR (15),
-> tgl_lahir DATE,
-> jk VARCHAR (2),
-> agama VARCHAR (10),
-> asal_sekolah VARCHAR (35),
-> prodi VARCHAR (25),
-> email VARCHAR (25),
-> handphone CHAR (12),
-> PRIMARY KEY (id_mhs)
-> );
Query OK, 0 rows affected (0.16 sec)
```

Keterangan perintah:

- ▶ **CREATE TABLE** adalah perintah dasar SQL untuk membuat sebuah tabel dalam database siacad.
- ▶ **Mahasiswa** adalah nama tabel yang akan dibuat pada database siacad.
- ▶ **id\_mhs, nm\_mhs, alamat, tmpt\_lahir, tgl\_lahir, jk, agama, asal\_sekolah, prodi, email, Handphone** adalah nama field dari tabel mahasiswa yang dibuat pada database siacad.
- ▶ **VARCHAR, TEXT, DATE dan CHAR** adalah Tipe data yang ditetapkan pada masing-masing field yang terdapat pada tabel mahasiswa.
- ▶ **NOT NULL** adalah opsi yang menandakan bahwa field tersebut tidak boleh dibiarkan kosong ketika proses peng-Inputan kedalam tabel mahasiswa.
- ▶ **10, 30, 15, 2, 35, 25, 12** adalah angka yang digunakan untuk menentukan panjang maksimal dari masing-masing field yang terdapat pada tabel mahasiswa.
- ▶ **TEXT dan DATE** adalah tipe data yang panjang maksimalnya tidak perlu ditentukan.
- ▶ **PRIMARY KEY** adalah perintah yang digunakan untuk menentukan salah satu field yang akan dijadikan sebagai kunci (Primary Key) dalam tabel mahasiswa.

Tanda baca Titik-koma (;) adalah perintah yang wajib digunakan untuk mengakhiri baris perintah SQL.

Setelah struktur tabel mahasiswa terbentuk melalui perintah diatas, selanjutnya akan ditampilkan tabel apa saja yang terdapat dalam database siacad melalui perintah **SHOW TABLES;**

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> show tables;
+-----+
| Tables_in_siacad |
+-----+
| mahasiswa        |
+-----+
1 row in set (0.14 sec)

mysql>
```

Gambar diatas menunjukkan bahwa pada database siacad terdapat satu tabel dengan nama mahasiswa, selanjutnya untuk melihat deskripsi dari tabel mahasiswa tersebut secara detail maka diketikkan perintah **DESC mahasiswa;**

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL | |
| nm_mhs | varchar(30) | NO | | NULL | |
| alamat | text | YES | | NULL | |
| tmpt_lahir | varchar(15) | YES | | NULL | |
| tgl_lahir | date | YES | | NULL | |
| jk | varchar(2) | YES | | NULL | |
| agama | varchar(10) | YES | | NULL | |
| asal_sekolah | varchar(35) | YES | | NULL | |
| prodi | varchar(25) | YES | | NULL | |
| email | varchar(25) | YES | | NULL | |
| handphone | char(12) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.22 sec)
```

Perintah **DESC** adalah singkatan dari kata **DESCRIBE** yang bermakna menguraikan secara detail tentang struktur tabel yang terdapat dalam database.

Keterangan struktur:

- ▶ Bahwa dalam tabel mahasiswa terdapat 11 (Sebelas) field.
- ▶ Kunci (Primary Key) dari tabel mahasiswa tersebut adalah **id\_mhs**.
- ▶ Untuk dua field ter-atas **id\_mhs** dan **nm\_mhs** secara default tidak boleh dibiarkan kosong ketika proses penginputan karena telah diuraikan **NO** pada kolom **NULL**.

- ▶ Untuk 9 (sembilan) field lain (**alamat, tmpt\_lahir, tgl\_lahir, jk, agama, asal\_sekolah, prodi, email, handphone**) boleh dibiarkan kosong ketika proses penginputan karena telah diuraikan Yes pada kolom NULL.

Tidak Lengkapnya data pada saat proses input kedalam tabel, akan membuat cacat pada informasi yang ditampung dalam tabel.

## I. Praktikum 6: Mengubah Struktur Tabel pada Database

Dalam DBMS MySQL sebuah tabel dapat dirubah atau dimodifikasi untuk memenuhi kebutuhan pengguna dengan bentuk struktur perintahnya sebagai berikut:

```
ALTER TABLE <nama_tabel> <alter_options>;
```

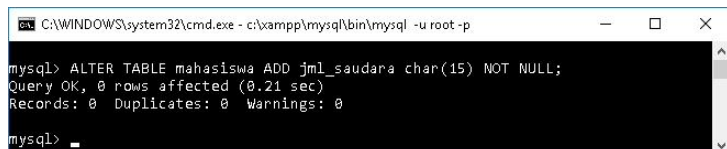
Keterangan Perintah:

- ▶ **ALTER TABLE** adalah perintah SQL yang digunakan untuk mengubah struktur tabel dalam database.
- ▶ **nama\_tabel** adalah nama tabel yang terdapat dalam database yang akan dirubah strukturnya yang dalam hal ini telah ditentukan bahwa nama tabel dalam database siacad yang dijadikan sebagai bahan latihan yaitu tabel mahasiswa.
- ▶ **alter\_options** adalah pilihan yang digunakan untuk merubah struktur tabel, diantaranya sebagai berikut:
  - ▶ **ADD (definisi\_field\_baru)**

Adalah opsi dari perintah SQL yang digunakan untuk menambahkan field baru pada tabel yang terdapat dalam database.

Struktur Perintah:

```
ALTER TABLE <nama tabel> ADD <field yang ingin ditambahkan><tipe_data><(panjang_data)> <keterangan NULL>;
```



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> ALTER TABLE mahasiswa ADD jml_saudara char(15) NOT NULL;
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> _
```

Informasi dari gambar diatas menunjukkan bahwa query untuk menambahkan field baru dengan nama "jml\_saudara" dengan

tipe data “char” dan panjang data “15” karakter dengan sifat data tidak boleh kosong “NOT NULL” telah berhasil diproses seperti yang ditunjukkan pada gambar dibawah ini.

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL | |
| nm_mhs | varchar(30) | NO | | NULL | |
| alamat | text | YES | | NULL | |
| tmpt_lahir | varchar(15) | YES | | NULL | |
| tgl_lahir | date | YES | | NULL | |
| jk | varchar(2) | YES | | NULL | |
| agama | varchar(10) | YES | | NULL | |
| asal_sekolah | varchar(35) | YES | | NULL | |
| prodi | varchar(25) | YES | | NULL | |
| email | varchar(25) | YES | | NULL | |
| handphone | char(12) | YES | | NULL | |
| jml_saudara | char(15) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.01 sec)

mysql>
```

► **ADD INDEX (nama\_index)**

Adalah opsi dari perintah SQL yang digunakan untuk menambahkan index pada tabel dalam database.

Struktur Perintah:

**ALTER TABLE <nama tabel> ADD INDEX <(field yang dijadikan index)>;**

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> ALTER TABLE mahasiswa ADD INDEX (id_mhs);
Query OK, 0 rows affected (0.24 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> _
```

Informasi dari gambar diatas menjelaskan bahwa dalam tabel mahasiswa telah ditetapkan (*id\_mhs*) sebagai indexnya, yang berarti pencarian data dalam tabel mahasiswa berdasarkan index yang telah ditetapkan (*id\_mhs*), Index bertujuan untuk mempercepat proses (Query) pencarian data berdasarkan kolom.

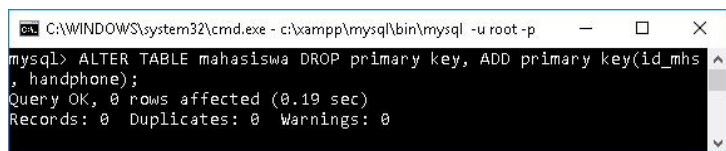


► **ADD PRIMARY KEY (Field\_kunci)**

Adalah opsi dari perintah SQL yang digunakan untuk menambahkan primary key (Kunci) pada tabel dalam database.

Struktur Perintah:

```
ALTER TABLE <nama_tabel> DROP <primary key awal>,  
ADD <primary key baru><(field primary key baru 1, field  
primary key baru 2);
```



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p  
mysql> ALTER TABLE mahasiswa DROP primary key, ADD primary key(id_mhs  
, handphone);  
Query OK, 0 rows affected (0.19 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Primary key adalah sebuah nilai dalam database yang digunakan untuk mengidentifikasi baris yang terdapat dalam tabel, perintah SQL yang dieksekusi dengan CMD diatas adalah struktur perintah yang digunakan untuk menambahkan primary key menjadi dua yaitu *id\_mhs* dengan *handphone*, (kondisi sebelumnya telah diketahui bahwa tabel mahasiswa hanya memiliki satu primary key saja yaitu *id\_mhs*) untuk itu, baris perintah diatas digunakan untuk menambahkan primary key menjadi dua, tapi sebelum itu primary key awal harus di hapus terlebih dahulu menggunakan perintah DROP lalu di ADD kembali dengan dua usulan primary key *id\_mhs* dan *handphone* dan untuk hasilnya dapat dilihat melalui deskripsi tabel mahasiswa sebagai berikut:

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL | |
| nm_mhs | varchar(30) | NO | | NULL | |
| alamat | text | YES | | NULL | |
| tmpt_lahir | varchar(15) | YES | | NULL | |
| tgl_lahir | date | YES | | NULL | |
| jk | varchar(2) | YES | | NULL | |
| agama | varchar(10) | YES | | NULL | |
| asal_sekolah | varchar(35) | YES | | NULL | |
| prodi | varchar(25) | YES | | NULL | |
| email | varchar(25) | YES | | NULL | |
| handphone | char(12) | NO | PRI | NULL | |
| jml_saudara | char(15) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+

```

Gambar diatas menunjukkan bahwa dalam tabel mahasiswa telah terdapat dua Primary Key (*id\_mhs* dan *handphone*).

► **CHANGE** (Field\_yang\_diubah definisi\_field\_baru)

Adalah opsi dari perintah SQL yang digunakan untuk mengubah field lama menjadi field baru pada tabel yang terdapat dalam database.

Struktur Perintah:

```

ALTER TABLE <nama_tabel> CHANGE <field_lama>
<field_baru> <tipe_data><(panjang_data)>;

```

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> ALTER TABLE mahasiswa CHANGE jk jns_kelamin VARCHAR(8);
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql>

```

Baris perintah SQL pada gambar diatas menjelaskan bahwa field *jk* dengan tipe data VARCHAR dengan panjang data 2 karakter pada tabel mahasiswa dirubah menggunakan opsi CHANGE menjadi *jns\_kelamin* dengan tipe data VARCHAR dengan panjang data 8 karakter seperti yang dideskripsikan pada gambar berikut ini.

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs     | varchar(10)   | NO   | PRI | NULL     |       |
| nm_mhs     | varchar(30)   | NO   |     | NULL     |       |
| alamat     | text          | YES  |     | NULL     |       |
| tmpt_lahir | varchar(15)   | YES  |     | NULL     |       |
| tgl_lahir  | date          | YES  |     | NULL     |       |
| jns_kelamin | varchar(8)    | YES  |     | NULL     |       |
| agama      | varchar(10)   | YES  |     | NULL     |       |
| asal_sekolah | varchar(35)  | YES  |     | NULL     |       |
| prodi      | varchar(25)   | YES  |     | NULL     |       |
| email      | varchar(25)   | YES  |     | NULL     |       |
| handphone  | char(12)      | NO   | PRI | NULL     |       |
| jml_saudara | char(15)      | NO   |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.01 sec)

```

► **MODIFY (definisi\_field)**

Adalah opsi dari perintah SQL yang digunakan untuk mengubah definisi dari sebuah field lama pada tabel dalam database.

Struktur Perintah:

```

ALTER TABLE <nama_tabel> MODIFY <nama field yang ingin dirubah><tipe_data><(panjang data)><keterangan NULL>;

```

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> ALTER TABLE mahasiswa MODIFY jml_saudara INT(8) NOT NULL;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> _

```

Opsi perintah MODIFY memiliki karakter yang tidak begitu berbeda dengan opsi perintah CHANGE yang telah dijelaskan sebelumnya, jika opsi perintah CHANGE hanya mampu mengubah nama field dan panjang karakter, opsi perintah MODIFY mampu melakukan yang lebih dari hal tersebut yaitu mengubah definisi dari sebuah field. Baris perintah pada gambar diatas menjelaskan bahwa field jml\_saudara dengan tipe data Char dan panjang data 15 karakter dirubah menjadi field jml\_saudara dengan tipe data INT (Integer) dan panjang

data 8 karakter seperti yang dideskripsikan pada gambar berikut ini.

```
mysql> desc mahasiswa;
```

Field	Type	Null	Key	Default	Extra
id_mhs	varchar(10)	NO	PRI	NULL	
nm_mhs	varchar(30)	NO		NULL	
alamat	text	YES		NULL	
tmpt_lahir	varchar(15)	YES		NULL	
tgl_lahir	date	YES		NULL	
jns_kelamin	varchar(8)	YES		NULL	
agama	varchar(10)	YES		NULL	
asal_sekolah	varchar(35)	YES		NULL	
prodi	varchar(25)	YES		NULL	
email	varchar(25)	YES		NULL	
handphone	char(12)	NO	PRI	NULL	
jml_saudara	int(8)	NO		NULL	

```
12 rows in set (0.02 sec)
```

► **RENAME TO** (nama\_tabel\_baru)

Adalah opsi dari perintah SQL yang digunakan untuk mengganti nama tabel yang terdapat dalam database

Struktur Perintah:

```
ALTER TABLE <nama_tabel_lama> RENAME TO <nama_tabel_baru>;
```

```
mysql> ALTER TABLE mahasiswa RENAME TO mhs;
Query OK, 0 rows affected (0.00 sec)

mysql> _
```

Baris perintah pada gambar diatas menjelaskan bahwa nama tabel *mahasiswa* dirubah menjadi tabel *mhs* dengan menggunakan opsi **RENAME TO**.

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> desc mahasiswa;
ERROR 1146 (42502): Table 'siakad.mahasiswa' doesn't exist
mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL | |
| nm_mhs | varchar(30) | NO | | NULL | |
| alamat | text | YES | | NULL | |
| tmpt_lahir | varchar(15) | YES | | NULL | |
| tgl_lahir | date | YES | | NULL | |
| jns_kelamin | varchar(8) | YES | | NULL | |
| agama | varchar(10) | YES | | NULL | |
| asal_sekolah | varchar(35) | YES | | NULL | |
| prodi | varchar(25) | YES | | NULL | |
| email | varchar(25) | YES | | NULL | |
| handphone | char(12) | NO | PRI | NULL | |
| jml_saudara | int(8) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.03 sec)

mysql>

```

Gambar diatas menjelaskan bahwa ketika tabel *mahasiswa* dideskripsikan dengan perintah *desc mahasiswa;* terjadi kesalahan yang memunculkan informasi *ERROR 1146 (42502): Table "siakad.mahasiswa" doesn't exist* dengan artian bahwa deskripsi tabel *mahasiswa* yang diperintahkan melalui perintah *desc mahasiswa;* tidak dapat dipenuhi karena nama tabel *mahasiswa* pada database *siakad* sudah dirubah menjadi tabel *mhs*, dan selanjutnya perintah *desc mhs;* mampu menampilkan deskripsi dari tabel *mhs* yang sebelumnya adalah *tabel mahasiswa*.

► DROP (nama\_field)

Adalah opsi dari perintah SQL yang digunakan untuk menghapus nama field pada sebuah tabel.

Struktur Perintah:

```
ALTER TABLE <nama_tabel> DROP <nama_field yang ingin dihapus>;
```

Gambar berikut menampilkan deskripsi tabel *mhs* yang masih mengandung 12 field.

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL |  |
| nm_mhs | varchar(30) | NO |  | NULL |  |
| alamat | text | YES |  | NULL |  |
| tmpt_lahir | varchar(15) | YES |  | NULL |  |
| tgl_lahir | date | YES |  | NULL |  |
| jns_kelamin | varchar(8) | YES |  | NULL |  |
| agama | varchar(10) | YES |  | NULL |  |
| asal_sekolah | varchar(35) | YES |  | NULL |  |
| prodi | varchar(25) | YES |  | NULL |  |
| email | varchar(25) | YES |  | NULL |  |
| handphone | char(12) | NO | PRI |  |  |
| jml_saudara | int(8) | NO |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.11 sec)

mysql>
```

Baris perintah SQL pada gambar berikut ini menjelaskan bahwa field jml\_saudara akan dihapus dengan opsi perintah DROP.

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p

mysql> ALTER TABLE mhs DROP jml_saudara;
Query OK, 0 rows affected (0.32 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

Hasil eksekusi dari perintah DROP pada gambar diatas menunjukkan bahwa field jml\_saudara telah berhasil dihapus, hal ini terbukti dari deskripsi tabel mhs pada gambar berikut yang sudah tidak menampilkan field jml\_saudara dengan total 11 field yang berhasil ditampilkan.

```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mhs | varchar(10) | NO | PRI | NULL |  |
| nm_mhs | varchar(30) | NO |  | NULL |  |
| alamat | text | YES |  | NULL |  |
| tmpt_lahir | varchar(15) | YES |  | NULL |  |
| tgl_lahir | date | YES |  | NULL |  |
| jns_kelamin | varchar(8) | YES |  | NULL |  |
| agama | varchar(10) | YES |  | NULL |  |
| asal_sekolah | varchar(35) | YES |  | NULL |  |
| prodi | varchar(25) | YES |  | NULL |  |
| email | varchar(25) | YES |  | NULL |  |
| handphone | char(12) | NO | PRI |  |  |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.01 sec)

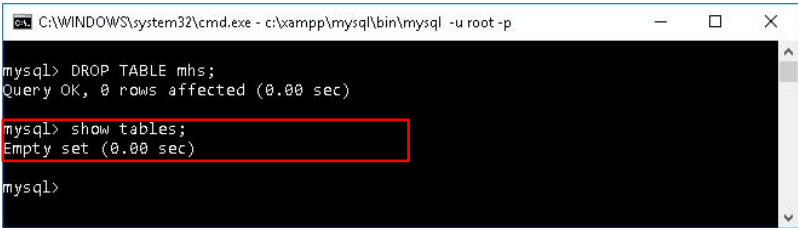
mysql>
```

## J. Praktikum 7: Menghapus Tabel pada Database

Sebuah tabel dalam database tidaklah bersifat permanen, dapat dimodifikasi strukturnya, serta dapat di hapus secara permanen, berikut ini akan ditunjukkan bentuk perintah SQL untuk menghapus sebuah tabel dalam database:

```
DROP TABLE <nama_tabel>;
```

Dalam contoh berikut ini akan diberikan contoh penghapusan tabel *mhs* dalam database *siakad*.



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> DROP TABLE mhs;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
Empty set (0.00 sec)

mysql>
```

Baris perintah *DROP TABLE mhs;* pada gambar diatas dilakukan untuk menghapus tabel *mhs* pada database *siakad* dan hasil eksekusi menampilkan informasi *Query OK*. Yang berarti tabel *mhs* telah berhasil dihapus, sedangkan baris perintah dalam kotak merah bertujuan untuk menampilkan kembali daftar tabel yang tersedia dalam database *siakad* dan menampilkan informasi *Empty set* yang berarti bahwa sudah tidak tersedia tabel apapun dalam database *siakad*.

## K. Praktikum 8: Menambah RECORD kedalam Tabel dengan perintah INSERT

*Record* data adalah sebuah nilai atau data yang akan menempati struktur tabel yang dirancang sebelumnya, **bentuk pertama** dari perintah SQL yang digunakan untuk menambahkan *record* kedalam tabel *mhs* adalah sebagai berikut:

```
INSERT INTO <nama_tabel> VALUES ('nilai_1','nilai_2','nilai_3',...');
```

Berikut ini akan diberikan contoh dalam menambahkan *record* kedalam tabel *mhs* yang telah dirancang sebelumnya:

```
INSERT INTO mhs VALUES ('C724201901', 'Arifin Ilham', 'Kecamatan Dompu', 'Bima', '1980-10-28', 'Pria', 'Islam', 'SMA Negeri 1 Dompu', 'PTI', 'arifinilham@gmail.com', '085253190337');
```

```
mysql> INSERT INTO mhs VALUES ('C724201901', 'Arifin Ilham', 'Kecamatan Dompu', 'Bima', '1980-10-28', 'Pria', 'Islam', 'SMA Negeri 1 Dompu', 'PTI', 'arifinilham@gmail.com', '085253190337');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Informasi pada kotak merah yang terdapat pada gambar diatas menunjukkan bahwa perintah SQL yang digunakan untuk menambah *record* data kedalam tabel *mhs* telah berhasil dieksekusi dengan baik, tentu untuk memastikan bahwa record data yang diinputkan tersebut telah menempati setiap *field* yang disediakan dalam tabel terlebih dahulu kita harus menampilkan isi tabel dengan struktur perintah SQL sebagai berikut:

```
SELECT * from <nama_tabel>;
```

Dengan uraian: Perintah SELECT adalah salah satu perintah SQL yang digunakan untuk menampilkan isi dari salah satu tabel dalam database, tanda "\*" (bintang) merupakan opsi yang digunakan untuk menampilkan seluruh field dari sebuah tabel dalam database, sedangkan opsi <nama\_tabel> adalah opsi dari nama tabel pada database yang akan ditampilkan isinya.

```
SELECT * from mhs;
```

```
mysql> select * from mhs;
+-----+-----+-----+-----+-----+-----+-----+
| id_mhs | nm_mhs | alamat | tmp_t_lahir | tgl_lahir | jns_kel |
amin | agama | asal_sekolah | prodi | email | handphone |
+-----+-----+-----+-----+-----+-----+-----+
| C724201901 | Arifin Ilham | Kecamatan Dompu | Bima | 1980-10-28 | Pria |
| Islam | SMA Negeri 1 Dompu | PTI | arifinilham@gmail.com | 085253190337 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Hasil eksekusi dari perintah SQL yang terdapat pada gambar diatas menunjukkan bahwa 11 (sebelas) field dalam tabel *mhs* telah memiliki



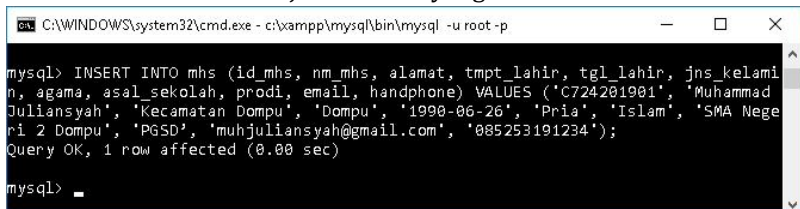
isi (record) sesuai dengan data (record) yang diinputkan melalui perintah INSERT sebelumnya.

**Bentuk yang kedua** dari perintah yang digunakan untuk menambah RECORD data dengan perintah INSERT antara lain sebagai berikut:

```
INSERT INTO <nama_tabel> (field_1, field_2, field_3,...)  
VALUES ('nilai_1','nilai_2','nilai_3',...');
```

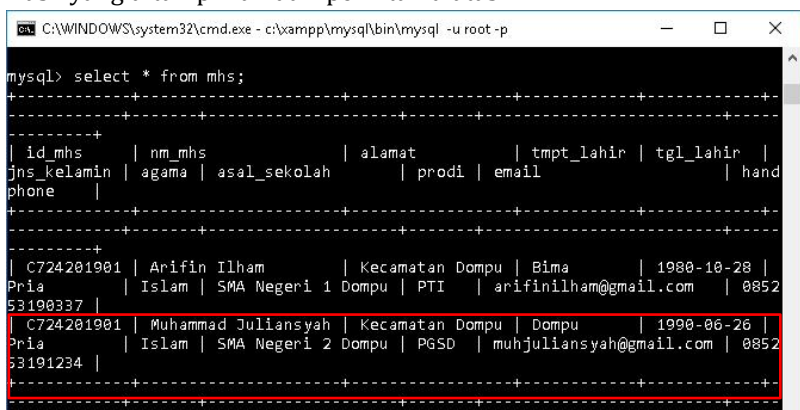
```
INSERT INTO mhs (id_mhs, nm_mhs, alamat, tmpt_lahir, tgl_lahir,  
jns_kelamin, agama, asal_sekolah, prodi, email, handphone) VALUES  
( 'C724201901', 'Muhammad Juliansyah', 'Kecamatan Dompu',  
'Dompu', '1990-06-26', 'Pria', 'Islam', 'SMA Negeri 2 Dompu', 'PGSD',  
'muhjuliansyah@gmail.com', '085253191234');
```

Berbeda dengan perintah INSERT sebelumnya yang tidak perlu mendefinisikan field dari tabel tapi langsung menguraikan nilai RECORD data yang ingin diinputkan, struktur perintah INSERT diatas menunjukkan bahwa field terlebih dahulu didefinisikan lalu RECORD data ditulis berdasarkan jumlah field yang telah didefinisikan.



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p  
mysql> INSERT INTO mhs (id_mhs, nm_mhs, alamat, tmpt_lahir, tgl_lahir, jns_kelamin,  
agama, asal_sekolah, prodi, email, handphone) VALUES ('C724201901', 'Muhammad  
Juliansyah', 'Kecamatan Dompu', 'Dompu', '1990-06-26', 'Pria', 'Islam', 'SMA Nege  
ri 2 Dompu', 'PGSD', 'muhjuliansyah@gmail.com', '085253191234');  
Query OK, 1 row affected (0.00 sec)  
mysql>
```

Hasil yang ditampilkan dari perintah diatas:



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p  
mysql> select * from mhs;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+  
| id_mhs | nm_mhs | alamat | tmpt_lahir | tgl_lahir | jns_kelamin | agama | asal_sekolah | prodi | email | hand  
phone |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+  
| C724201901 | Arifin Ilham | Kecamatan Dompu | Bima | 1980-10-28 | Pria | Islam | SMA Negeri 1 Dompu | PTI | arifinilham@gmail.com | 0852  
53190337 |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+  
| C724201901 | Muhammad Juliansyah | Kecamatan Dompu | Dompu | 1990-06-26 | Pria | Islam | SMA Negeri 2 Dompu | PGSD | muhjuliansyah@gmail.com | 0852  
53191234 |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+
```

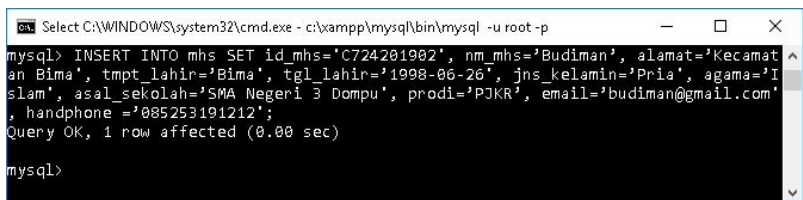
Gambar diatas menunjukkan bahwa eksekusi dari perintah INSERT dengan bentuk yang kedua telah berhasil menambahkan RECORD data sesuai kotak merah pada tabel *mhs*.

**Bentuk yang ketiga** dari perintah yang digunakan untuk menambah RECORD data dengan perintah INSERT antara lain sebagai berikut:

```
INSERT INTO <nama_tabel>  
SET field_1='nilai_1', field_2='nilai_2', field_3='nilai_3',...;
```

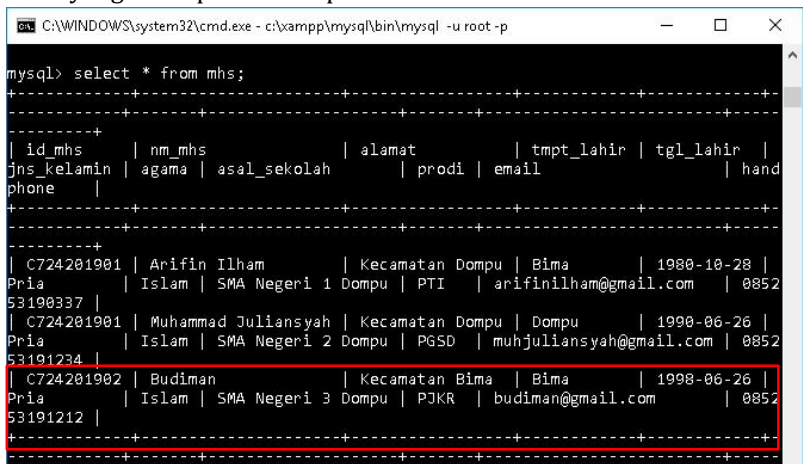
```
INSERT INTO mhs SET id_mhs='C724201902', nm_mhs='Budiman',  
alamat='Kecamatan Bima', tmpt_lahir='Bima', tgl_lahir='1998-06-26',  
jns_kelamin='Pria', agama='Islam', asal_sekolah='SMA Negeri 3  
Dompu', prodi='PJKR', email='budiman@gmail.com', handphone  
='085253191212';
```

Perintah INSERT jenis ini menggunakan opsi SET untuk mendefinisikan field secara langsung dengan pasangan RECORDnya.



```
mysql> INSERT INTO mhs SET id_mhs='C724201902', nm_mhs='Budiman', alamat='Kecamat  
an Bima', tmpt_lahir='Bima', tgl_lahir='1998-06-26', jns_kelamin='Pria', agama='I  
slam', asal_sekolah='SMA Negeri 3 Dompu', prodi='PJKR', email='budiman@gmail.com'  
, handphone ='085253191212';  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```

Hasil yang ditampilkan dari perintah diatas:



```
mysql> select * from mhs;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+  
| id_mhs | nm_mhs | alamat | tmpt_lahir | tgl_lahir | jns_kelamin | agama | asal_sekolah | prodi | email | handphone |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+  
| C724201901 | Arifin Ilham | Kecamatan Dompu | Bima | 1980-10-28 | Pria | Islam | SMA Negeri 1 Dompu | PTI | arifinilham@gmail.com | 085253190337 |  
| C724201901 | Muhammad Juliansyah | Kecamatan Dompu | Dompu | 1990-06-26 | Pria | Islam | SMA Negeri 2 Dompu | PGSD | muhjuliansyah@gmail.com | 085253191234 |  
| C724201902 | Budiman | Kecamatan Bima | Bima | 1998-06-26 | Pria | Islam | SMA Negeri 3 Dompu | PJKR | budiman@gmail.com | 085253191212 |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+
```

Gambar diatas menunjukkan bahwa eksekusi dari perintah INSERT dengan bentuk yang ketiga telah berhasil menambahkan RECORD data sesuai kotak merah pada tabel *mhs*.

#### L. Praktikum 9: Perintah SELECT untuk menampilkan RECORD

Pembahasan pada praktikum diatas telah menguraikan fungsi dari perintah SELECT yang digunakan untuk menampilkan RECORD dari sebuah tabel dalam database, bentuk umum dari perintah tersebut sebagai berikut:

```
SELECT * FROM <nama_tabel>;
```

Perintah select bersifat dinamis dan dapat dikembangkan sesuai dengan kebutuhan pengguna, berikut contoh dari pengembangan perintah SELECT:

1. Menampilkan RECORD dari dua field pada tabel

Berikut ini dijelaskan struktur perintahnya:

```
SELECT <nama_field_1>, <nama_field_2> FROM <nama_tabel>;
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs FROM mhs;
```

Jika perintah SQL diatas berhasil dieksekusi maka menampilkan hasil berikut ini.



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> SELECT id_mhs, nm_mhs FROM mhs;
+-----+-----+
| id_mhs | nm_mhs |
+-----+-----+
| C724201901 | Arifin Ilham |
| C724201901 | Muhammad Juliansyah |
| C724201902 | Budiman |
+-----+-----+
3 rows in set (0.00 sec)
mysql>
```

Terlihat pada gambar diatas hanya menampilkan dua field dari tabel *mhs* yaitu *id\_mhs* dan *nm\_mhs*, tidak hanya mampu menyeleksi RECORD berdasarkan field, perintah SELECT juga mampu menyeleksi RECORD berdasarkan sub RECORD seperti yang dicontohkan dalam pembahasan berikut ini.

2. Menampilkan RECORD dari tiga field dengan informasi sub RECORD pada tabel

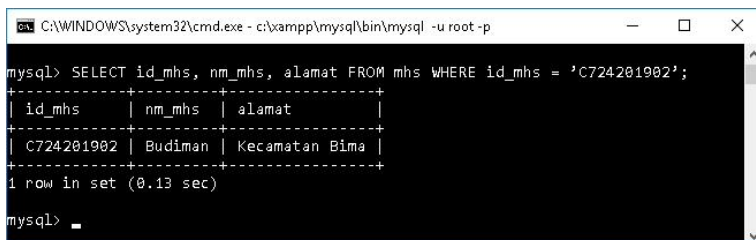
Berikut ini dijelaskan struktur perintahnya:

```
SELECT <nama_field_1>, <nama_field_2>, <nama_field_3> FROM  
<nama_tabel> WHERE <nama_field_yang_menjadi_key> = 'nilai  
field_key';
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs, alamat FROM mhs WHERE id_mhs =  
'C724201902';
```

Jika perintah SQL diatas berhasil dieksekusi maka menampilkan hasil berikut ini.



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p  
mysql> SELECT id_mhs, nm_mhs, alamat FROM mhs WHERE id_mhs = 'C724201902';  
+-----+-----+-----+  
| id_mhs | nm_mhs | alamat |  
+-----+-----+-----+  
| C724201902 | Budiman | Kecamatan Bima |  
+-----+-----+-----+  
1 row in set (0.13 sec)  
mysql>
```

Terlihat pada gambar diatas hanya menampilkan tiga field dari tabel *mhs* yaitu *id\_mhs*, *nm\_mhs*, dan *alamat*, field yang ditampilkan tersebut berdasarkan nilai dari sub RECORD dari field *id\_mhs* = 'C724201902'.

3. Menampilkan RECORD dari tiga field dengan klasifikasi berdasarkan string menggunakan perintah LIKE

Berikut ini dijelaskan struktur perintahnya:

```
SELECT <nama_field_1>, <nama_field_2>, <nama_field_3> FROM  
<nama_tabel> WHERE <nama_field_yang_diseleksi> LIKE 'nilai  
string_dari_field_yang_diseleksi';
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs, alamat FROM mhs WHERE alamat LIKE  
'Dompu';
```

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p

mysql> SELECT id_mhs, nm_mhs, alamat FROM mhs WHERE alamat LIKE '%Dompu%';
+-----+-----+-----+
| id_mhs | nm_mhs | alamat |
+-----+-----+-----+
| C724201901 | Arifin Ilham | Kecamatan Dompu |
| C724201901 | Muhammad Juliansyah | Kecamatan Dompu |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

4. Menampilkan RECORD dari empat field dengan klasifikasi berdasarkan string menggunakan operator penghubung  
Berikut ini dijelaskan struktur perintahnya:

```

SELECT <nama_field_1>, <nama_field_2>, <nama_field_3>,
<nama_field_3> FROM <nama_tabel> WHERE <nama_field_
yang_diseleksi_1> = 'nilai_field_yang_diseleksi_1' && <nama_field_
yang_diseleksi_2> LIKE 'nilai_string_dari_field_yang_diseleksi_2';

```

Jika diuraikan sesuai dengan struktur diatas, maka:

```

SELECT id_mhs, nm_mhs, alamat, email FROM mhs WHERE alamat
= 'Kecamatan Dompu' && email LIKE '%gmail.com';

```

```

C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p

mysql> SELECT id_mhs, nm_mhs, alamat, email FROM mhs WHERE alamat = 'Kecamatan Dompu' && email LIKE '%gmail.com';
+-----+-----+-----+-----+
| id_mhs | nm_mhs | alamat | email |
+-----+-----+-----+-----+
| C724201901 | Arifin Ilham | Kecamatan Dompu | arifinilham@gmail.com |
| C724201901 | Muhammad Juliansyah | Kecamatan Dompu | muhjuliansyah@gmail.com |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

Hasil eksekusi perintah diatas menjelaskan bahwa field yang ditampilkan adalah field *id\_mhs*, *nm\_mhs*, *email* dari tabel *mhs* dimana field yang diseleksi adalah field *alamat* dengan RECORD yang berisi semua alamat dari 'Kecamatan Dompu' yang memiliki *email* 'gmail.com', dua field yang diseleksi diatas dihubungkan dengan operator penghubung "&& (AND)". Operator tersebut akan memeriksa dua kondisi (kondisi pertama adalah field alamat harus berisi RECORD "Kecamatan Dompu", kondisi kedua adalah

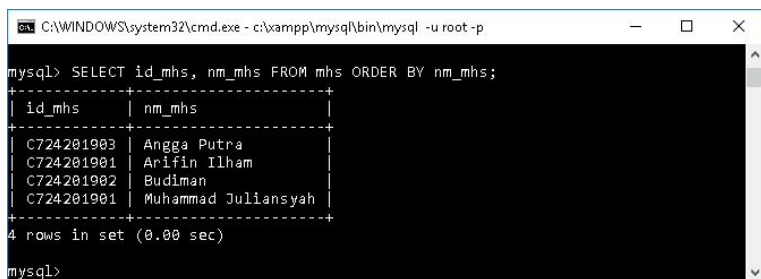
field email harus berisi RECORD “gmail.com”), jika salah satu kondisi tersebut tidak terpenuhi, atau dengan kata lain tidak mengandung RECORD dengan alamat Kecamatan Dompu untuk field pertama atau tidak mengandung RECORD gmail.com pada field email, maka perintah SQL diatas tidak dapat menampilkan informasi apapun, hal ini disebabkan karena operator penghubung && (AND) memiliki karakter bahwa kedua kondisi harus bernilai TRUE barulah menghasilkan nilai TRUE, namun jika salah satu bernilai TRUE dan salah satunya bernilai FALSE maka akan menghasilkan nilai FALSE untuk kondisi yang diperiksa.

5. Menampilkan RECORD secara urut berdasarkan salah satu field (nm\_mhs) dengan perintah ORDER BY

```
SELECT <nama_field_1>, <nama_field_2> FROM <nama_tabel>  
ORDER BY <nama_field_yang_ingin_diurutkan>;
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs;
```



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p  
mysql> SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs;  
+-----+-----+  
| id_mhs | nm_mhs |  
+-----+-----+  
| C724201003 | Angga Putra |  
| C724201001 | Arifin Ilham |  
| C724201002 | Budiman |  
| C724201001 | Muhammad Juliansyah |  
+-----+-----+  
4 rows in set (0.00 sec)  
mysql>
```

Hasil eksekusi dari perintah diatas menunjukkan bahwa RECORD yang ditampilkan adalah RECORD dalam field id\_mhs dan nm\_mhs, RECORD ditampilkan secara berurut menggunakan perintah ORDER BY berdasarkan pada nm\_mhs dari abjad awal A sampai dengan abjad akhir Z.

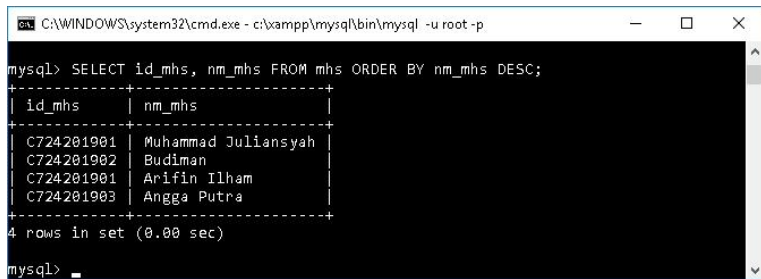
6. Menampilkan RECORD secara DESCENDING

Berikut ini dijelaskan struktur perintahnya:

```
SELECT <nama_field_1>, <nama_field_2> FROM <nama_tabel>  
ORDER BY <nama_field_yang_ingin_diurutkan>DESC;
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs DESC;
```



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs DESC;
+-----+-----+
| id_mhs | nm_mhs |
+-----+-----+
| C724201901 | Muhammad Juliansyah |
| C724201902 | Budiman |
| C724201901 | Arifin Ilham |
| C724201903 | Angga Putra |
+-----+-----+
4 rows in set (0.00 sec)
mysql>
```

Berbeda dengan perintah ORDER BY Hasil eksekusi dari perintah diatas menunjukkan bahwa RECORD yang ditampilkan adalah RECORD dalam field id\_mhs dan nm\_mhs, RECORD ditampilkan secara berurut menggunakan perintah ORDER BY berdasarkan pada nm\_mhs namun diurutkan secara DESCENDING dari abjad akhir Z sampai dengan abjad awal A.

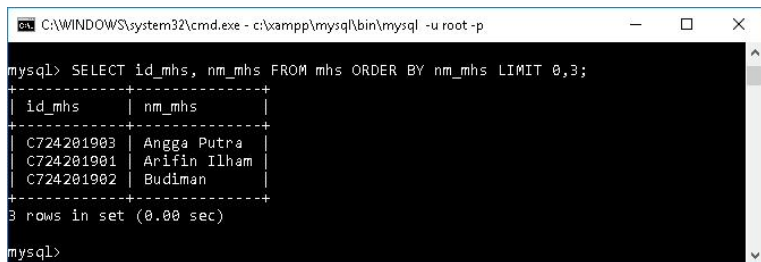
7. Menampilkan RECORD dengan perintah ORDER BY namun dibatasi jumlah RECORDnya dengan perintah LIMIT

Berikut ini dijelaskan struktur perintahnya:

```
SELECT <nama_field_1>, <nama_field_2> FROM <nama_tabel>
ORDER BY <nama_field_yang_ingin_diurutkan> LIMIT 0,3;
```

Jika diuraikan sesuai dengan struktur diatas, maka:

```
SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs LIMIT 0,3;
```



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -u root -p
mysql> SELECT id_mhs, nm_mhs FROM mhs ORDER BY nm_mhs LIMIT 0,3;
+-----+-----+
| id_mhs | nm_mhs |
+-----+-----+
| C724201903 | Angga Putra |
| C724201901 | Arifin Ilham |
| C724201902 | Budiman |
+-----+-----+
3 rows in set (0.00 sec)
mysql>
```

Limit adalah salah satu opsi dari SQL yang digunakan untuk membatasi jumlah RECORD dari sebuah tabel dalam database, angka 0,3 merupakan batasan jumlah RECORD yang ditampilkan dari tabel mhs, yaitu sebanyak 3 RECORD sesuai pada hasil query dari gambar diatas.

## M. Evaluasi Materi

**Jawablah soal-soal berikut ini dengan benar:**

1. Dalam pembuatan basisdata tentu user akan merancang field, dapatkan anda memberikan penjelasan terkait dengan resiko atas kesalahan dalam penentuan type dan panjang data dalam perancangan basis data?
2. Bagaimanakah peran dan fungsi primary key dalam sebuah tabel pada database?
3. Redudansi merupakan kesamaan dari data yang terdapat pada basis data, dapatkan anda menunjukkan serta memberikan penjelasan terkait dengan Redudansi yang terjadi dalam code berikut:

```
INSERT INTO mhs SET id_mhs='C724201902', nm_mhs='Arifin',  
alamat='Kecamatan Dompu', tmpt_lahir='Bima', tgl_lahir='1998-06-  
26', jns_kelamin='Pria', agama='Islam', asal_sekolah='SMA Negeri 3  
Dompu', prodi='PJKR', email='budiman@gmail.com', handphone  
='085253191212';
```

Dengan:

```
INSERT INTO mhs SET id_mhs='C724201902', nm_mhs='Budiman',  
alamat='Kecamatan Bima', tmpt_lahir='Bima', tgl_lahir='1998-06-26',  
jns_kelamin='Pria', agama='Islam', asal_sekolah='SMA Negeri 3  
Dompu', prodi='PJKR', email='budiman@gmail.com', handphone  
='085253191212';
```

4. Jika Field id\_mhs dan handphone dalam dua code diatas adalah Primary Key maka berikanlah penjelasan terkait dengan Record dari dua field tersebut (Jika memiliki kesamaan akan seperti apa? Dan jika memiliki perbedaan akan seperti apa?)



## DAFTAR PUSTAKA

1. Almahdy Indra. (2006). *Sistem Informasi Manajemen dan Database management systems*. Universitas Mercubuana. Tidak Dipublikasikan.
2. Ben-Gan, Itzik, et al. (2006). *Inside Microsoft SQL Server 2005: T-SQL Programming*. Microsoft Press. ISBN 0-7356-2197-7.
3. Brookshear J Gleen. (2003). *Computer Science: Suatu Pengantar Edisi Ke-7*. Jakarta: Erlangga. ISBN 979-741-009-9.
4. Connolly, T., & Begg, C. (2010). *Database Systems: a Practical Approach to design, implementation, and management*. 5<sup>th</sup> Edition. America: Pearson Education.
5. Delaney, Kalen, et al. (2007). *Inside SQL Server 2005: Query Tuning and Optimization*. Microsoft Press. ISBN 0-7356-2196-9.
6. Elmasri, & Navathe. (2011). *Fundamentals Of Database System Sixth Edition*. Addison-Wisley. ISBN 10: 0-136-08620-9.
7. Indrajani. (2015). *Database Design (Case Study All in One)*. Jakarta: PT Elex Media Komputindo.
8. Lance Delano, Rajesh George et al. (2005). *Wrox's SQL Server 2005 Express Edition Starter Kit (Programmer to Programmer)*. Microsoft Press. ISBN 0-7645-8923-7.
9. Mardapi, Djemari. (2012). *Pengukuran Penilaian & Evaluasi Pendidikan*. Yogyakarta: Nuha Medika Litera.
10. Munandar, Utami. (2009). *Pengembangan Kreativitas Anak Berbakat*. Jakarta: Rineka Cipta.
11. Solichin Achmad. (2010). *MySQL 5 Dari pemula hingga mahir Versi 1.0*. Jakarta: Universitas Budi Luhur. Tidak Dipublikasikan.

## PENUTUP

Demikian yang dapat penulis paparkan dalam buku yang berjudul “*Teori dan Praktik Database Management System*” dengan materi yang disampaikan antara lain (1) Bagian 1. Dasar Teori DBMS (Database Management System) dengan sub materi Pengantar, fasilitas, komponen pendukung, Manfaat, Kelebihan kekurangan, Macam-macam, Hirarki basis data dalam DBMS (Database Management System). (2) Bagian 2. Praktikum dasar penggunaan DBMS dengan sub materi XAMPP sebagai software Pendukung, Praktikum 1: Melakukan Instalasi XAMPP, Praktikum 2: Akses MySQL server melalui CMD, Praktikum 3: Membuat dan menampilkan Database, Memahami Tipe-Tipe Tabel dan Field dalam MySQL, Aturan dasar dalam perancangan Database, Praktikum 4: Menghapus Database, Praktikum 5: Membuat, Menampilkan, dan mendeskripsikan dalam pada Database, Praktikum 6: Mengubah Struktur Tabel pada Database, Praktikum 7: Menghapus Tabel pada Database, Praktikum 8: Menambah RECORD kedalam Tabel dengan perintah INSERT, Praktikum 9: Perintah SELECT untuk menampilkan RECORD, selain itu buku ini juga dilengkapi dengan soal latihan dimasing-masing bagian untuk mengevaluasi hasil belajar peserta didik

Dalam penyusunan buku ajar ini tentu masih banyak terdapat kekurangan dan kelemahannya, hal ini tidak lepas dari keterbatasan pengetahuan dan kurangnya rujukan atau referensi yang ada hubungannya dengan judul buku ini. Menyadari bahwa penulis masih jauh dari kata sempurna, kedepannya penulis akan lebih fokus dan details dalam menjelaskan tentang buku di atas dengan sumber – sumber yang lebih banyak yang tentunya dapat di pertanggungjawabkan. Penulis banyak berharap para pembaca yang akan membaca buku ini dapat memberikan kritik dan saran yang membangun kepada penulis demi kesempurnaannya buku ini dan dan penulisan buku di kesempatan-kesempatan berikutnya. Semoga buku ini dapat berguna bagi penulis pada khususnya juga para pembaca. Terimakasih.

## GLOSARIUM

<b>Table</b>	Kumpulan data yang diorganisasikan ke dalam baris (record) dan kolom (field).
<b>Field</b>	Adalah kolom dari sebuah table. Kolom ini memiliki ukuran dan type data tertentu.
<b>Record</b>	Adalah kumpulan data yang saling berhubungan menginformasikan tentang suatu informasi.
<b>Key</b>	Adalah field yang dapat berperan sebagai kunci dalam operasi tabel.
<b>SQL</b>	Adalah bahasa yang digunakan untuk mengakses salah satu DBMS MySQL.
<b>Primary Key</b>	Adalah nilai yang digunakan untuk mengidentifikasi baris dalam tabel
<b>Entitas</b>	Adalah orang, tempat, kejadian atau konsep yang informasinya telah dicatat.
<b>Query</b>	Adalah perintah SQL yang digunakan untuk membaca dan menulis pada database.
<b>Null</b>	Null adalah attribute yang digunakan untuk menyatakan nilai yang belum siap atau kosong
<b>Index</b>	Field lain yang digunakan untuk mengurutkan Record pada DataBase (dapat Primary atau Secondary Index)
<b>File</b>	Merupakan kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, atribute yang sama, namun berbeda-beda data valuenya
<b>Redundansi</b>	Adalah duplikasi data dalam beberapa file data sehingga data yang sama di simpan di dalam lebih dari 1 lokasi.