



Manajemen Sistem Basis Data menggunakan Maria DB

Menjadi Database Administrator menggunakan Maria DB

**Ardhin Primadewi, S.Si., M.Ti.
Sunarni, S.T, M.T.**

Editor : Setiya Nugroho, S.T., M.Eng

Manajemen Sistem Basis Data menggunakan Maria DB

Penulis:

Ardhin Primadewi, S. Si, M. TI

Sunarni, S.T, M.T

Editor:

Setiya Nugroho, ST., M. Eng

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. Penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan pengumuman sebagai bahan ajar; dan
- iv. Penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).

Manajemen Sistem Basis Data menggunakan Maria DB

ISBN: 978-623-7261-68-1

Hak Cipta 2022 pada Penulis

Hak penerbitan pada UNIMMA PRESS. Bagi mereka yang ingin memperbanyak sebagian isi buku ini dalam bentuk atau cara apapun harus mendapatkan izin tertulis dari penulis dan penerbit UNIMMA PRESS.

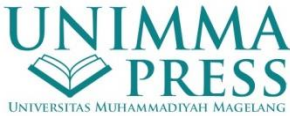
Penulis:

Ardhin Primadewi, S. Si, M. TI

Sunarni, S.T, M.T

Editor:

Setiya Nugroho, ST., M. Eng



Penerbit:

UNIMMA PRESS

Gedung Rektorat Lt. 3 Kampus 2 Universitas Muhammadiyah Magelang

Jalan Mayjend Bambang Soegeng km.05, Mertoyudan, Magelang 56172

Telp. (0293) 326945

E-Mail: unimmapress@ummgl.ac.id

Hak Cipta dilindungi Undang-undang

All Right Reserved

Cetakan I, Januari 2022

Kata Pengantar

Puji syukur yang tak terkira kami panjatkan kehadiran Alloh SWT, yang telah memberikan kami kesempatan menyelesaikan buku kami dengan judul "Manajemen Sistem Basis Data Menggunakan Maria DB".

Kami juga menyampaikan rasa terima kasih yang setinggi-tingginya bagi semua pihak yang telah membantu dan memberikan kontribusi hingga terselesainya buku ini, yang tidak dapat kami sebutkan satu-persatu.

Buku ini berisikan pengenalan SQL, teori basis data, kelengkapan basis data, DML hingga fungsi system. Buku ini selain merupakan buku pegangan wajib pada mata kuliah Manajemen Sistem Basis Data pada program Studi Teknik Informatika Universitas Muhammadiyah Magelang, juga merupakan referensi bagi para pembaca yang sedang mendalami masalah system basis data.

Kami sadar, bahwa buku ini sangat jauh dari kata sempurna, baik dari isi, penyajian maupun editorialnya. Oleh karenanya, kami sangat mengharapkan masukan kriti dan saran demi perbaikan bagi buku ini.

Akhirnya kami berharap, semoga buku ini bermanfaat, tidak hanya bagi mahasiswa Teknik Informatika Universitas Muhammadiyah Magelang, tapi juga semua yang membacanya.

Magelang, Maret 2022

Penulis

Daftar Isi

<i>Kata Pengantar</i>	<i>iv</i>
<i>Daftar Isi</i>	<i>v</i>
<i>Daftar Gambar</i>	<i>vii</i>
<i>Daftar Tabel</i>	<i>vii</i>
<i>Bab 1. Pengenalan</i>	<i>1</i>
1. <i>Basis Data / Database</i>	<i>1</i>
2. <i>Komponen Sistem Basis Data</i>	<i>1</i>
3. <i>Mengenal SQL (Structured Query Language)</i>	<i>1</i>
4. <i>Mengenal MySQL</i>	<i>3</i>
5. <i>Kelebihan MySQL</i>	<i>5</i>
6. <i>Phpmyadmin</i>	<i>6</i>
7. <i>XAMPP</i>	<i>6</i>
8. <i>Command Prompt (CMD)</i>	<i>7</i>
9. <i>Text Editor</i>	<i>7</i>
<i>Bab 2. Instalasi</i>	<i>9</i>
A. <i>MySQL pada XAMPP</i>	<i>9</i>
B. <i>Mengaktifkan Mysql Melalui CMD</i>	<i>11</i>
C. <i>Mengetahui Database Pada Mysql</i>	<i>13</i>
D. <i>Memilih Database Yang Diaktifkan</i>	<i>13</i>
E. <i>Melihat Metadata Tabel</i>	<i>13</i>
F. <i>Memulai DDL (Data Definition Language)</i>	<i>13</i>
<i>Bab 3. Type Data</i>	<i>15</i>
A. <i>Tipe Data</i>	<i>15</i>
B. <i>Constraint</i>	<i>18</i>
C. <i>Referential Intergrity Constraint</i>	<i>19</i>
D. <i>Primary Key Dan Foreign Key</i>	<i>20</i>
E. <i>Index</i>	<i>21</i>

<i>Bab 4. Data Manipulation Language</i>	22
A. <i>Memulai DML (Data Manipulation Language)</i>	22
<i>Bab 5. Lanjutan DML</i>	24
A. <i>Lanjutan DML (Data Manipulation Language)</i>	24
<i>Bab 6. Fungsi</i>	26
A. <i>Fungsi</i>	26
B. <i>Select Into Statement</i>	32
<i>Bab 7. Sub Query</i>	34
A. <i>Nested Query (Sub Query)</i>	34
B. <i>Double Nested Query (Double Sub Query)</i>	35
<i>Bab 8. JOIN</i>	36
A. <i>Join Without Join Statement</i>	36
B. <i>Inner Join dan Outer Join</i>	36
<i>Bab 9. Studi Kasus</i>	38
A. <i>Penjelasan Studi Kasus Kapal Laut</i>	38
B. <i>DDL Kapal Laut</i>	39
C. <i>DML Kapal Laut</i>	43
D. <i>DML Nested Soal Kapal Laut</i>	59
<i>Daftar Pustaka</i>	61

Daftar Gambar

Gambar 1 Hasil setelah instalasi pada direktori C.....	9
Gambar 2 Hasil setelah instalasi pada direktori mysql.....	10
Gambar 3 Direktori bin dan data di dalam direktori mysql	10
Gambar 4 Mengaktifkan MySQL dan Apache pada XAMPP control panel	11
Gambar 5 Direktori mysql pada jendela explorer.....	11
Gambar 6 Mencari jendela command prompt	11
Gambar 7 Masuk melalui command prompt menuju.....	12
Gambar 8 Setelah masuk ke database mysql.....	13
Gambar 9 Desain database kapal_laut	39

Daftar Tabel

Tabel 1 Tipe data numerik.....	15
Tabel 2 Tipe data numerik (Lanjutan)	16
Tabel 3 Tipe data string	16
Tabel 4 Tipe data date	17
Tabel 5 Tipe data BLOB	17
Tabel 6 Tipe data lain	18
Tabel 7 Operator pada SQL.....	24
Tabel 8 Operator pada SQL (Lanjutan)	25
Tabel 9 Tabel symbol format pada Fungsi Tanggal.....	30
Tabel 10 Tabel isi dari entitas Nahkoda	43
Tabel 11 Tabel isi dari entitas Kapal	43
Tabel 12 Tabel isi dari entitas Job	44
Tabel 13 Tabel isi dari entitas Mengikuti_training	44
Tabel 14 Tabel isi dari entitas Training.....	45

Bab 1. Pengenalan

1. Basis Data / Database

Merupakan himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.

2. Komponen Sistem Basis Data

- a. Perangkat Keras (Hardware) : Komputer, memori, storage (Harddisk), peripheral, dll.
- b. Sistem Operasi (Operating System) : Program yang menjalankan sistem komputer, mengendalikan resource computer dan melakukan berbagai operasi dasar sistem komputer.
- c. Basis Data (Database) : Menyimpan berbagai obyek database (struktur tabel, indeks,dll)
- d. DBMS (Database Management System) : Perangkat lunak yang memaintain data dalam jumlah besar.
- e. Pemakai (User) : Para pemakai database.
- f. Aplikasi (perangkat lunak) lain.
- g. Program lain dalam DBMS.

3. Mengenal SQL (Structured Query Language)

SQL adalah singkatan dari *Structured Query Language*. Sedangkan **pengertian SQL adalah** suatu bahasa (language) yang digunakan untuk mengakses data di dalam sebuah database relasional. SQL adalah sebuah bahasa permintaan *database* yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat

merelasikan beberapa tabel dalam database maupun merelasikan antar database.

SQL sering juga disebut dengan istilah **query**, dan bahasa SQL secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa SQL.

Dalam penggunaan SQL terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Jenis perintah SQL secara umum dibagi kepada tiga sub perintah, yaitu DDL (Data Definition Language), DML (Data Manipulation Language), dan DCL (Data Control Language). Ketiga sub perintah tersebut sangat perlu untuk dipahami bagi anda yang ingin menguasai bahasa sql dan mahir dalam pembuatan database.

Penjelasan terkait DDL, DML dan DCL :

a. DDL (Data Definition Language)

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah Database, Query yang dimiliki DDL adalah :

- CREATE : Digunakan untuk membuat Database dan Tabel
- DROP : Digunakan untuk menghapus Tabel dan Database
- ALTER : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field (Add), mengganti nama Field (Change) ataupun menamakannya kembali (Rename), dan menghapus Field (Drop)

b. DML (Data Manipulation Language)

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. Query yang dimiliki DML adalah :

- INSERT : Digunakan untuk memasukkan data pada Tabel Database
- UPDATE : Digunakan untuk perubahan terhadap data yang ada pada Tabel Database
- DELETE : Digunakan untuk Penhapusan data pada tabel Database

c. DCL (Data Control Language)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (privileges).

Yang termasuk perintah dalam DCL ada dua, yaitu GRANT dan REVOKE.

a) GRANT adalah perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna.

Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database.

b) REVOKE adalah perintah yang digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT.

4. Mengetahui MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta

menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa kita gunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL (General Public License). MySQL Merupakan sebuah *database server* yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server.

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

Selain karena *Free*, MySQL juga merupakan DBMS yang relatif ringan dan mudah digunakan. Ada beberapa *software* yang bisa digunakan untuk menjalankan *service* MySQL yaitu WAMP, PhpTriad dan XAMPP. Namun XAMPP merupakan *software* yang paling populer saat ini. Kepopuleran ini dikarenakan XAMPP terus didukung dengan adanya *update* pada *software* nya.

XAMPP merupakan program paket yang didalamnya terdapat beberapa *software* yang digabungkan menjadi satu. Dalam XAMPP sendiri didalamnya terdapat 5 *software* yang berbeda (MySQL, Apache,

FileZilla,Mercury, Tomcat). Sehingga ketika menginstall XAMPP kita tidak perlu lagi menginstall Apache dan MySQL secara terpisah.

5. Kelebihan MySQL

Database MySQL memiliki beberapa kelebihan dibanding database lain, diantaranya :

1. MySQL merupakan *Database Management System* (DBMS)
2. MySQL sebagai *Relational Database Management System* (RDBMS) atau disebut dengan database Relational
3. MySQL merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya
4. MySQL merupakan sebuah database client
5. MySQL mampu menerima query yang bertupuk dalam satu permintaan atau Multi- Threading.
6. MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.
7. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti visual Basic dan Delphi.
8. MySQL adalah database menggunakan enkripsi password, jadi database ini cukup aman karena memiliki password untuk mengakses nya.
9. MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
10. MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unqi (Unique).

11. MySQL memiliki kecepatan dalam pembuatan table maupun peng-update an table.

6. Phpmyadmin

Phpmyadmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

PhpMyadmin dapat di download secara free di <http://www.phpmyadmin.net>. Karena berbasis web, maka phpmyadmin dapat di jalankan di banyak OS, selama dapat menjalankan webserver dan Mysql.

7. XAMPP

XAMPP adalah sebuah paket kumpulan software yang terdiri dari apache, mysql, phpmyadmin, php, Perl, Freetype2,dll. XAMPP berfungsi untuk memudahkan instalasi lingkungan php, di mana biasa nya lingkungan pengembangan web memerlukan php, apache, mysql dan phpmyadmin serta *software-software* yang terkait dengan pengembangan web. Dengan menggunakan XAMPP, kita tidak perlu menginstall aplikasi- aplikasi tsb satu persatu. Untuk master software XAMPP bisa *download* gratis di situs resminya www.apachefriends.org/en/xampp.html .

Paket aplikasi perlu di extract dan di install terlebih dahulu, dengan memilih jenis XAMPP sesuai dengan jenis OS nya. Setelah sukses menginstall XAMPP, kita dapat langsung mengaktifkan Mysql

dengan cara mengaktifkan XAMPP. Untuk mengaktifkan phpmyadmin dengan cara :

1. Aktifkan phpmyadmin pada Panel
2. Ketik : <http://localhost/phpmyadmin>

8. Command Prompt (CMD)

Command Prompt (CMD) adalah sebuah perintah dos yang terdapat pada OS Windows yang dapat memudahkan user dalam menjelajahi windows baik secara online maupun offline. *Command Prompt* secara resmi disebut *Windows Command Processor* tetapi terkadang juga disebut *command shell* atau *cmd prompt* atau bahkan disebut berdasarkan nama filenya yaitu **cmd.exe** **cmd.exe** adalah baris perintah penerjemah pada OS/2, Windows CE dan Windows NT berbasis sistem operasi (termasuk Windows 2000, Windows XP, Windows Vista, Windows Server 2003 dan Windows Server 2008). Ini merupakan analog dari COMMAND.COM dalam MS-DOS dan sistem Windows 9x atau dari Unix shell yang digunakan pada sistem Unix-like.

9. Text Editor

Text Editor adalah aplikasi yang berguna untuk mengedit teks dan skrip kode pemrograman / database. Pada mata kuliah ini dikenalkan text editor seperti Notepad++ dan sublime text. Text editor memudahkan pemrogram mengenal tag dan kode dalam berbagai bahasa pemrograman.

Program Notepad++ banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi desktop dan web. Notepad++ merupakan software gratis

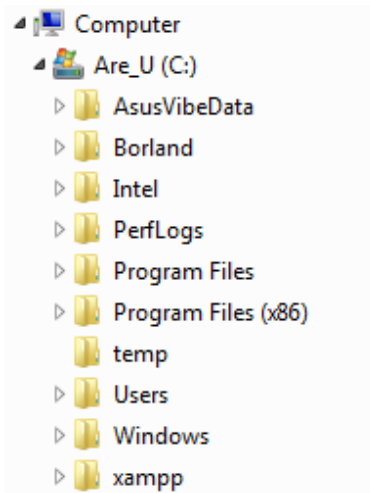
(opensource). Notepad++ dapat dijalankan di sistem operasi Win2K, Windows XP, Vista, dan Windows 7.

Program sublime text, tidak jauh berbeda dengan Notepad++. Namun memiliki kelebihan yaitu lintas platform (lintas operating system). Sublime Text adalah aplikasi text editor yang digunakan untuk membuka file apapun namun sejatinya para programmer menggunakannya untuk menulis code. Sublime text mendukung sejumlah bahasa pemrograman diantaranya C, C++, C#, PHP, CSS, HTML, ASP dan banyak lagi.

Bab 2. Instalasi

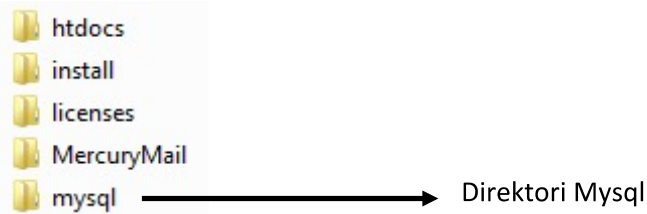
A. MySQL pada XAMPP

Pada praktikum mata kuliah SMDB ini, mahasiswa telah melakukan proses instalasi XAMPP dengan sukses. Hal yang perlu diingat ketika menginstall XAMPP adalah direktori tempat meletakkan hasil instalasi. Biasanya instalasi akan terdefault pada direktori C, namun demikian jika menginginkan *custom* saat instalasi juga dapat dilakukan (seperti direktori D atau E). Contoh hasil setelah instalasi dapat dilihat pada gambar 1 berikut ini. Di dalam folder xampp tersebut, dapat dilihat direktori mysql aktif sesuai gambar 2. Semua proses yang berjalan untuk bekerja dalam MySQL terdapat dalam direktori mysql.



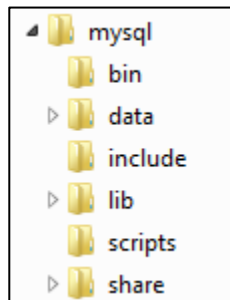
Direktori XAMPP setelah diinstall

Gambar 1 Hasil setelah instalasi pada direktori C



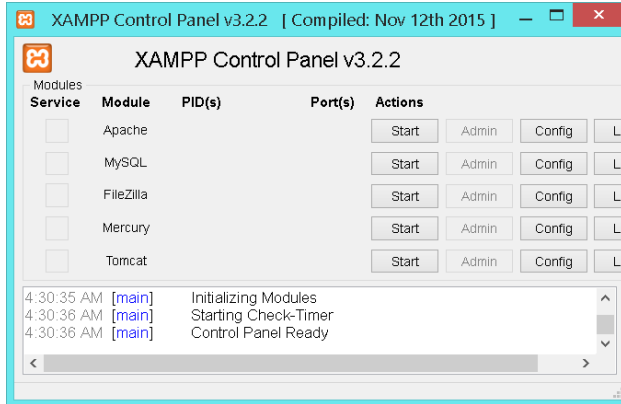
Gambar 2 Hasil setelah instalasi pada direktori mysql

Ada beberapa direktori yang perlu diketahui dalam direktori mysql yaitu bin dan data (seperti terlihat pada gambar 3). Direktori bin berisi service-service yang bisa dijalankan untuk mengakses mysql. Sedangkan direktori data berisi database dan tabel yang sudah dibuat.



Gambar 3 Direktori bin dan data di dalam direktori mysql

Untuk memulai menggunakan MYSQL pada XAMPP, pertama nyalakan service MySQL dengan menekan tombol START pada MYSQL seperti pada gambar 4. Jika ingin mengaktifkan PHPMYADMIN maka tombol START pada Apache juga ikut ditekan.



Gambar 4 Mengaktifkan MySQL dan Apache pada XAMPP control panel

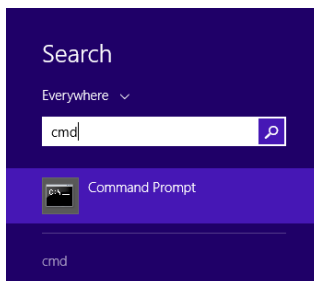
B. Mengaktifkan Mysql Melalui CMD

Masuk kedalam MySQL melalui CMD artinya masuk kedalam direktori `mysql\bin` dalam direktori hasil installan XAMPP. Pada contoh di bawah ini, direktori `mysql\bin` terletak pada partisi D. Contoh direktori `mysql` seperti pada gambar 5.



Gambar 5 Direktori mysql pada jendela explorer.

Kemudian aktifkan jendela *command prompt* seperti pada gambar 6 dari jendela *search windows*.



Gambar 6 Mencari jendela command prompt

Pada gambar 5 dapat dilihat bahwa XAMPP diinstal pada partisi D. Maka tahapan mengakses direktori mysql melalui command prompt dapat dilihat pada gambar 7 di bawah ini.



```
modul smdb ardhin - Microsoft Word
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\User>d:
D:\>
D:\>cd xampp
D:\xampp>
D:\xampp>cd mysql
D:\xampp\mysql>
D:\xampp\mysql>cd bin
D:\xampp\mysql\bin>
```

Gambar 7 Masuk melalui command prompt menuju

Sintaks :

d : (masuk ke direktori D)

cd xampp (masuk ke folder xampp dalam direktori aktif)

cd mysql (masuk ke folder mysql dalam direktori aktif)

cd bin (masuk ke folder bin dalam direktori aktif)

Setelah aktif pada folder bin. Maka aktifkan database server mysql dengan mengkasesnya langsung melewati jendela *command prompt*. Dengan perintah **mysql -u root** (dapat dilihat pada gambar 2.8) untuk user = root dan password = (kosong). Tanda sudah bisa melakukan query SQL dapat dilihat pada gambar 8.

```
D:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.19-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> _
```

Gambar 8 Setelah masuk ke database mysql

C. Mengetahui Database Pada Mysql

Untuk melihat list/daftar dari database atau tabel yang sudah dibuat pada direktori mysql :

- a) Melihat daftar database : Untuk melihat daftar database yang sudah dibuat.

Query : **SHOW** databases;

- b) Melihat daftar tabel : Untuk melihat daftar database yang ada digunakan fungsi show tables.

Query : **SHOW** tables;

D. Memilih Database Yang Diaktifkan

Perintah USE digunakan untuk memilih database yang ingin digunakan.

Query : **USE** nama_database;

E. Melihat Metadata Tabel

Digunakan untuk melihat struktur (metadata) sebuah tabel.

Query : **DESC** nama_tabel;

F. Memulai DDL (Data Definition Language)

1. Membuat database / tabel baru

- a. Membuat database

Query : **CREATE DATABASE** nama_database;

- b. Membuat tabel

Query :
CREATE
TABLE
nama_tabel(
Field1
TipeData1,
Field2 TipeData2,
.....
FieldN TipeDataN
);

2. Menghapus sebuah **database atau struktur tabel**

a. Menghapus database

Query : **DROP DATABASE** nama_database;

b. Menghapus struktur tabel

Query : **DROP TABLE** nama_tabel;

3. Merubah struktur dari sebuah tabel.

Perubahan struktur ini dapat berupa penambahan field baru, perubahan field yang sudah ada, maupun menghapus field yang sudah ada.

Query :

- **ALTER TABLE** nama_tabel **ADD** new_field typedata
- **ALTER TABLE** nama_tabel **CHANGE** field_lama field_baru typedata
- **ALTER TABLE** nama_tabel **DROP** nama_field
- **ALTER TABLE** nama_tabel **RENAME** nama_field

Bab 3. Type Data

A. Tipe Data

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau *record*) dan kolom (*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa tabel. Macam-macam Tipe Data pada MySQL, secara umum tipe-tipe data MySQL ini ada empat (4), diantaranya yaitu:

1. Tipe Data Numeric.

Tipe data numerik yaitu tipe data yang digunakan untuk menyimpan data numerik (angka).

Tabel 1 Tipe data numerik

No	Nama	Fungsi	Jangkauan	Ukuran
1	TINYINT	Menyimpan data bilangan bulat positif dan negatif.	-128 s/d 127	1 byte (8 bit).
2	SMALLINT	menyimpan data bilangan bulat positif dan negatif.	: -32.768 s/d 32.767	: 2 byte (16 bit).
3	MEDIUMINT	menyimpan data bilangan bulat positif dan negatif.	-8.388.608 s/d 8.388.607	Ukuran : 3 byte (24 bit).
4	INT	menyimpan data bilangan bulat positif dan negative	-2.147.483.648 s/d 2.147.483.647	4 byte (32 bit).
5	BIGINT	menyimpan data bilangan bulat positif dan negatif.	$\pm 9,22 \times 10^{18}$	8 byte (64 bit).
6	FLOAT	menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.	4 byte (32 bit)

Tabel 2 Tipe data numerik (Lanjutan)

No	Nama	Fungsi	Jangkauan	Ukuran
7	DOUBLE	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d - 2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit)
8	REAL	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d - 2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
9	DECIMAL	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d - 2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
10	NUMERIC	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d - 2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).

2. Tipe Data String.

Tipe data string yaitu tipe data yang digunakan untuk menyimpan data string (text).

Tabel 3 Tipe data string

No	Nama	Fungsi	Jangkauan
1	CHAR	menyimpan data string ukuran tetap.	0 s/d 255 karakter
2	VARCHAR	menyimpan data string ukuran dinamis.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
3	TINYTEXT	menyimpan data text.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
4	TEXT	menyimpan data text.	0 s/d 65.535
5	MEDIUMTEXT	menyimpan data text	0 s/d 224 - 1 karakter
6	LONGTEXT	menyimpan data text.	0 s/d 232 - 1 karakter

3. Tipe Data Date.

Tipe data date dan time yaitu tipe data yang digunakan untuk menyimpan data tanggal dan waktu.

Tabel 4 Tipe data date

No	Nama	Fungsi	Jangkauan	Ukuran
1	DATE	menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)	3 byte.
2	TIME	menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)	3 byte
3	DATETIME	menyimpan data tanggal dan waktu.	'1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'	8 byte
4	YEAR	menyimpan data tahun dari tanggal	1900 s/d 2155	1 byte

4. Tipe Data BLOB

Tipe data blob digunakan untuk menyimpan data biner.

Tabel 5 Tipe data BLOB

No	Nama	Fungsi	Jangkauan
1	BIT	Menyimpan data biner.	64 digit biner
2	TINYBLOB	menyimpan data biner/ Gambar ukuran kecil	255 byte
3	BLOB	Menyimpan data biner seperti files, gambar, suara, dll	4
4	MEDIUMBLOB	Menyimpan data biner/ Gambar ukuran sedang	224-1 byte
5	LOB	Menyimpan data biner/ Gambar ukuran besar	232- 1 byte

5. Tipe Data yang lain

Selain tipe data di atas, MySQL juga menyediakan tipe data yang lain, diantaranya adalah :

Tabel 6 Tipe data lain

No	Nama	Fungsi	Jangkauan
1	ENUM	enumerasi (kumpulan data).	sampai dengan 65535 string.
2	SET	combination (himpunan data).	sampai dengan 255 string anggota

B. Constraint

Constraint adalah batasan atau aturan yang ada pada tabel.

MySQL menyediakan beberapa tipe constraint yaitu :

1. NOT NULL

Suatu kolom yang didefinisikan dengan constraint NOT NULL tidak boleh berisi nilai NULL. Kolom yang berfungsi sebagai kunci primer (primary key) otomatis tidak boleh NULL.

2. UNIQUE

Mendefinisikan suatu kolom menjadi bersifat unik, artinya antara satu data dengan data lainnya namanya tidak boleh sama, misal alamat email.

3. PRIMARY KEY

Constraint PRIMARY KEY membentuk key yang unik untuk suatu table.

4. FOREIGN KEY

FOREIGN KEY constraint didefinisikan pada suatu kolom yang ada pada suatu table, dimana kolom tersebut juga dimiliki oleh table yang lain sebagai suatu PRIMARY KEY.

- Untuk Menghapus constraint perintah yang dapat digunakan yaitu "DROP "

Query : ALTER TABLE table DROP CONSTRAINT type (column);

- Untuk Mengaktifkan constraint perintah yang dapat

digunakan yaitu "ENABLE"

Query : ALTER TABLE table ENABLE
CONSTRAINT type (column);

- Untuk Mematikan constraint perintah yang dapat digunakan yaitu "DISABLE"

Query : ALTER TABLE table DISABLE
CONSTRAINT type (column);

C. Referential Integrity Constraint

Referential Integrity Constraint adalah aturan untuk relasi antar tabel untuk menjamin validasi hubungan antar record di dalam tabel – tabel yang terkait.

- Aturan untuk Update :

Cascade : pembaharuan sebuah baris data diikuti oleh pembaharuan baris data pada tabel anak yang terelasi

Restrict : mencegah pembaharuan data jika terdapat baris data di tabel anak yang terhubung

Ignore : mengabaikan referensi. Boleh memperbarui data pada tabel parent, tetapi tidak memperbarui data pada tabel child

No Action : Tidak ada aksi apapun

- Aturan untuk Delete :

Cascade : menghapus seluruh baris data pada tabel child yang terhubung

Restrict : mencegah penghapusan jika terdapat baris data yang terhubung tabel child

Ignore : boleh menghapus data, tapi tidak akan berpengaruh pada tabel anaknya

No Action : Tidak ada aksi apapun

- Aturan untuk Insert :

Restrict : tidak boleh menambah data pada tabel child jika nilai yang dimasukkan pada kolom yang berelasi tidak terdapat pada parent tabelnya

Ignore : boleh menambah data pada tabel child meskipun nilai yang dimasukkan pada kolom yang berelasi tidak terdapat pada tabel parentnya

No Action : Tidak ada aksi apapun

D. Primary Key Dan Foreign Key

1. Membuat Primary Key

Query : **CREATE TABLE** nama_tabel>Nama_field tipe_field
PRIMARY KEY NOT NULL);

2. Membuat Foreign Key

Query : **CREATE TABLE** nama_tabel>Nama_field tipe_field
FOREIGN KEY REFERENCES NOT NULL);

3. Merubah Primary Key atau Foreign Key

Query :

- **ALTER TABLE** nama_tabel **ADD PRIMARY KEY**(nama_field);
- **ALTER TABLE** nama_tabel **ADD FOREIGN KEY**(nama_field)**REFERENCES** nama_tabel1(nama_field1);

4. Menghapus Primary Key atau Foreign Key

Query :

- **ALTER TABLE** nama_tabel **DROP PRIMARY KEY**(nama_field);
- **ALTER TABLE** nama_tabel **DROP FOREIGN KEY**(nama_field)**REFERENCES** nama_tabel1(nama_field1);

E. Index

Indeks merupakan suatu objek yang berfungsi untuk mempercepat proses pengambilan data, pengurutan dan pencarian data dari suatu tabel. Data pada tabel yang sudah diindeks akan diurutkan berdasarkan kolom indeks. dengan demikian proses pencarian data dapat lebih cepat. Saat data baru sedikit mungkin pengindekan belum terlalu berguna, namun jika data sudah mencapai ribuan atau bahkan ratusan ribu barulah terasa perbedaanya jika field tersebut belum diindeks.

Bab 4. Data Manipulation Language

A. Memulai DML (Data Manipulation Language)

1. Menambahkan Data Pada Tabel

Fungsi *Insert* berfungsi untuk mengisi data kedalam sebuah tabel.

```
Query : INSERT INTO nama_tabel  
VALUES('nilaiField1','nilaiField2', .....,'nilaiFieldN');
```

2. Merubah Data Pada Tabel

Fungsi *Update* digunakan untuk merubah isi data kolom / field dari sebuah baris / record pada sebuah tabel.

```
Query : UPDATE nama_tabel SET nama_field = 'value' WHERE  
nama_field = 'value'
```

3. Menghapus Data Pada Tabel

Fungsi *delete* digunakan untuk menghapus sebuah baris / record dalam sebuah tabel.

```
Query : DELETE nama_tabel WHERE nama_field = 'value'
```

4. Menampilkan Data Pada Tabel

Fungsi *Select* digunakan untuk menampilkan data pada tabel. Untuk menampilkan sebuah tabel ada beberapa kondisi yang bisa digunakan, antara lain :

1. SELECT ALL

Kondisi ini digunakan untuk menampilkan semua record dan semua field nya dalam sebuah tabel.

```
Query : SELECT * FROM  
nama_tabel;
```

2. SELECT FIELD

Kondisi ini digunakan untuk menampilkan semua baris / record yang ada, tetapi hanya kolom / field tertentu saja (kolom / field sesuai permintaan kondisi saja).

Query : **SELECT** field1,field2,... **FROM** nama_tabel;

3. SELECT RECORD (WHERE)

Kondisi ini digunakan untuk menampilkan isi dari baris / record tertentu saja. WHERE bisa digunakan untuk SELECT ALL maupun SELECT FIELD.

Query : **SELECT (ALL/FIELD) FROM** nama_tabel **WHERE** nama_field = 'value';

Bab 5. Lanjutan DML

A. Lanjutan DML (Data Manipulation Language)

1. Operator

Seperti halnya pemrograman java, C++ dll, SQL juga memiliki operator dasar. Operator dalam SQL adalah simbol yang digunakan untuk menginstruksi program untuk melakukan sesuatu. Akan berbeda definisi ketika kita membicarakan operator dalam kehidupan sehari-hari. Banyak sekali operator yang bisa digunakan ketika kita ingin menuliskan query. Disini kita akan membahas satu persatu operator dasar yang bisa kita gunakan dalam menuliskan query.

Pada tabel di bawah ini merupakan operator yang sering digunakan pada SQL khususnya menggunakan MySQL.

Tabel 7 Operator pada SQL

Nama	Keterangan
AND, &&	Logika AND
AS	Operator untuk memberikan nama Alias
ASC	Operator untuk mengurutkan dari kecil ke besar
BETWEEN ... AND ...	Mengecek nilai dalam range yang diapit 'AND'
BINARY	Mengubah nilai string ke biner string
&	Operasi bit AND
~	Membalik nilai bits
	Operasi bit OR
^	Operasi bit XOR
CASE	Operator Case
DESC	Operator untuk mengurutkan dari besar ke kecil
DISTINCT	Operator untuk mencegah duplikasi
DIV	Pembagian hasil nilai Integer
/	Operator pembagian
<=>	Operator logika untuk karakter NULL
=	Operator sama dengan
>=	Lebih besar atau sama dengan
>	Operator lebih besar
GROUP BY	Operator untuk mengkategorikan data

Tabel 8 Operator pada SQL (Lanjutan)

Nama	Keterangan
IN / NOT IN	Operator logika untuk mengetahui data dalam kondisi IN
IS NOT NULL	Menguji nilai NOT NULL
IS NOT	Menguji boolean benar/salah
IS NULL	Mengujin nilai NULL
IS	Menguji boolean benar/salah
<<	Operator geser kekiri
<=	Lebih kecil atau sama dengan
<	Lebih kecil
LIKE	Pencocokan pola tesks /mirip
-	Operator pengurangan
%	operator modulus
NOT BETWEEN ... AND ...	Mengecek yang nilainya tidak sama dalam range
!=, <>	Operator tidak sama dengan
NOT LIKE	Pembandingan yang tidak mirip/sama
NOT REGEXP	Pembandingan pola yang tidak sama dengan ekspresi
NOT, !	Membalik nilai
ORDER BY	Operator Logika untuk mengurutkan data
, OR	Logika OR
+	Operator penambahan
REGEXP	Pembandingan pola yang sama dengan ekspresi
>>	Geser kekanan
RLIKE	Sama dengan operator REGEXP
*	Times operator
-	Jika didepan tanda bilangan negatif
XOR	logika XOR

Bab 6. Fungsi

A. Fungsi

Fungsi merupakan suatu rutin khusus yg disediakan oleh MySQL untuk melakukan manipulasi suatu data. Bentuk Umum : `nama_fungsi([argumen1[, argumen2[, ...]])`. Argumen1, argumen2, ... adalah argumen/ parameter yang dibutuhkan oleh fungsi. Fungsi digunakan sebagai bagian dari perintah DML yaitu select. `SELECT fungsi(ekspresi) [FROM namatabel];`

Ada beberapa tipe fungsi dalam mysql, yaitu :

1. Fungsi String

MySQL memiliki banyak sekali fungsi yang berhubungan dengan operasi String. Berikut ini beberapa fungsi string yang disediakan MySQL.

a. `ASCII(x)`

Menghasilkan id_nahkoda ASCII untuk karakter pertama dalam suatu string x.

b. `CHAR(x,y,z,...)`

Menghasilkan nilai string berdasarkan id_nahkoda ASCII yang dituliskan dalam parameternya.

c. `LENGTH(X)`

Untuk mendapatkan panjang sebuah string X.

d. `CONCAT()`

Menggabungkan beberapa string dalam parameter menjadi satu string. Jika ada NULL maka hasil dari `CONCAT` adalah NULL.

e. `INSERT(X,Y,Z,J)`

- Menghasilkan string X yang telah diganti isinya dengan string J mulai dari posisi ke Y sebanyak Z.
- f. INSTR(X,Y)
Menghasilkan nilai posisi Y di dalam string X
 - g. LOCATE(X,Y,Z)
Memberikan posisi string X di dalam string Y mulai posisi ke Z.
 - h. LEFT(X,Y)
Mengambil sejumlah Y karakter dari string X, mulai posisi pertama.
 - i. RIGHT(X,Y)
Mengambil sejumlah Y karakter dari string X, mulai dari posisi paling akhir.
 - j. MID(X,Y,Z)
Mengambil data string X sejumlah Z karakter mulai dari posisi ke Y.
 - k. LTRIM(X)
Membuang spasi di sebelah kiri string X.
 - l. RTRIM(X)
Membuang spasi disebelah kanan string X.
 - m. REVERSE(X)
Membalik urutan penulisan sebuah string X.

2. Fungsi Tanggal

Selain fungsi string, MySQL juga memiliki fungsi-fungsi yang berhubungan dengan operasi tanggal dan waktu (date and time). Berikut ini beberapa fungsi tanggal dan waktu yang disediakan MySQL. Format tanggal dalam MySQL : yyyy-mm-dd HH:ii:ss

Keterangan :

yyyy : Tahun
mm : Bulan
dd : Tanggal
HH : Jam
ii : Menit
ss : Detik

Berikut ini beberapa fungsi tanggal yang disediakan MySQL.

a. CURDATE()

Menampilkan tanggal sekarang dari sistem. Contoh :

```
mysql> SELECT CURDATE();
+-----+
| CURDATE() |
+-----+
| 2012-04-19 |
+-----+
```

b. CURTIME()

Menampilkan waktu sekarang dari sistem. Contoh :

```
mysql> SELECT CURTIME();
+-----+
| CURTIME() |
+-----+
| 02:01:15 |
+-----+
```

c. CURRENT_TIMESTAMP() / NOW()

Menampilkan waktu saat ini, tanggal dan jam. Contoh :

```
mysql> SELECT CURRENT_TIMESTAMP();
+-----+
| CURRENT_TIMESTAMP() |
+-----+
| 2012-04-19 02:01:30 |
+-----+
```

d. DATE_FORMAT(x, SimbolFormat)

Mengkonversi data tanggal sesuai dengan format yang diinginkan. Contoh :

```
mysql> SELECT DATE_FORMAT('2012-04-19', '%M %D %Y');
+-----+
| DATE_FORMAT('2012-04-19', '%M %D %Y') |
+-----+
| April 19th 2012 |
+-----+
```

e. DAY_NAME(X)

Menampilkan nama hari dari tanggal yang menjadi argumen dalam X. Contoh :

```
mysql> SELECT DAYNAME('2012-04-19');
+-----+
| DAYNAME('2012-04-19') |
+-----+
| Thursday                |
+-----+
```

f. DAYOFMONTH(DATE)

Menampilkan nomor hari dari tanggal yang menjadi argumen.

Contoh :

```
mysql> SELECT DAYOFMONTH('2012-04-19');
+-----+
| DAYOFMONTH('2012-04-19') |
+-----+
|                            | 19 |
+-----+
```

g. MONTH(DATE)

Menampilkan bagian bulan dari tanggal yang menjadi argumen.

```
mysql> SELECT MONTH('2012-04-19');
+-----+
| MONTH('2012-04-19') |
+-----+
|                       | 4 |
+-----+
```

h. YEAR(DATE)

Menampilkan bagian tahun dari tanggal yang menjadi argumen.

```
mysql> SELECT YEAR('2012-04-19');
+-----+
| YEAR('2012-04-19') |
+-----+
|                     | 2012 |
+-----+
```

i. HOUR(TIME)

Menghasilkan bagian jam dari waktu yang menjadi argumen.

Contoh :

```
mysql> SELECT HOUR('10:20:11');
+-----+
| HOUR('10:20:11') |
+-----+
|                   | 10 |
+-----+
```

j. MINUTE(TIME)

Menghasilkan bagian menit dari waktu yang menjadi argumen. Contoh :

```
mysql> SELECT MINUTE('10:20:11');
+-----+
| MINUTE('10:20:11') |
+-----+
|                    20 |
+-----+
```

k. SECOND(TIME)

Menghasilkan bagian detik dari waktu yang menjadi argumen.

Contoh :

```
mysql> SELECT SECOND('10:20:11');
+-----+
| SECOND('10:20:11') |
+-----+
|                    11 |
+-----+
```

Tabel 9 Tabel symbol format pada Fungsi Tanggal

Simbol Format	Arti
%M	Nama bulan secara penuh
%m	Nomor bulan
%b	Nama bulan disingkat
%W	Nama hari secara lengkap
%D	Nomor hari dalam bulan
%Y	Tahun dalam format 4 digit
%y	Tahun dengan 2 digit
%j	Nomor hari dalam 1 tahun
%a	Nama hari disingkat
%d	Nomor hari dalam 1 bulan
%r	Waktu dalam format 12 jam
%T	Waktu dalam format 24 jam
%H	Jam dalam format 24 jam
%h	Jam dalam format 12 jam
%S	Detik

3. Fungsi Agregat

Fungsi agregat pada MySQL digunakan untuk mengelompokan data. Adapun yang tergolong didalam fungsi agregat, yaitu SUM, MAX, MIN, AVG dan COUNT.

a. SUM (nama_field)

Mendapatkan nilai total dari suatu kolom dalam sebuah tabel atau ekspresi.

- b. AVG (nama_field)
Untuk mendapatkan nilai rata-rata sebuah kolom pada tabel atau ekspresi.
- c. MAX (nama_field)
Untuk mendapatkan nilai maksimum dari sebuah kolom dalam tabel atau ekspresi.
- d. MIN (nama_field)
Untuk mendapatkan nilai minimum sebuah kolom pada tabel atau ekspresi.
- e. COUNT (X)
Untuk menghitung jumlah record dari suatu kolom atau tabel X.

4. Fungsi Aritmatika

Fungsi aritmatika pada MySQL digunakan untuk melakukan operasi matematika pada data. Adapun yang termasuk dalam fungsi aritmatika, yaitu :

- a. Penjumlahan (+)
Untuk menjumlahkan dua buah data numerik.
- b. Pengurangan (-)
Untuk mengurangi dua buah data numerik.
- c. Perkalian (x)
Untuk melakukan perkalian dua buah data numerik.
- d. Pembagian (/)
Untuk melakukan pembagian dua buah data numerik.
- e. Pembagian Sisa (%)
Untuk mendapatkan sisa pembagian dari suatu operasi pembagian bilangan numerik.

5. Fungsi Sistem

a. DATABASE()

Mengetahui database yang sedang digunakan.

b. LAST_INSERT_ID()

Menampilkan info data terakhir yang digenerate oleh MySQL pada kolom bertipe AUTO INCREMENT (AI).

c. SESSION_USER

Menampilkan informasi pemakai yang sedang melakukan akses ke database.

B. Select Into Statement

SELECT INTO adalah sebuah perintah yang ada pada MySQL yang digunakan untuk memindahkan data dari satu tabel ke tabel yang lain atau dari satu database ke database yang lain. SELECT INTO sebenarnya juga dapat memilah field yang mana sajakah yang ingin kita pindahkan dengan cara menyebutkan nama fieldnya.

Struktur SQL yang digunakan adalah `select * into tabelBaru from tabelLama;` dan bila hanya digunakan untuk memindah kolom-kolom tertentu saja maka Syntax yang digunakan adalah `select namaKolom1, namaKolom2, namaKolom3 into tabelBaru from tabelLama;`

Berikut adalah beberapa contoh penggunaan SELECT INTO:

- Membuat backup sebuah tabel

```
SELECT * INTO backup_transaksi FROM transaksi;
```

- Membuat backup sebuah tabel ke dalam database yang baru

```
SELECT * INTO backup_trsndskdi IN 'backup_database.mdb'  
FROM transaksi;
```


- Membuat backup beberapa kolom sebuah tabel

```
SELECT ID,NamaPelanggan INTO backup_transaksi FROM  
transaksi;
```

Bab 7. Sub Query

A. Nested Query (Sub Query)

Sub Query atau Nested query ini sangat bermanfaat untuk memudahkan kita membuat query dengan teknik *pipelining* dimana hasil dari sebuah query akan digunakan lagi dalam query yang lain. Nested query adalah sebuah query yang ada di dalam query (query bersarang).

Nested query menghubungkan 1 atau lebih tabel yang dihubungkan secara virtual untuk mendapatkan query seperti yang diinginkan pengguna.

Select ___a___ From ___b___

Where ___c___ IN / NOT IN (Select ___c___ From ___d___

Where ___e___);

Keterangan :

a : merupakan atribut / record yang ingin ditampilkan

b : merupakan tabel dari atribut / record yang ingin ditampilkan

c : atribut yang menjadi foreign key antar tabel yang dihubungkan

d : merupakan tabel dari atribut/record yang memiliki atribut sebagai foreign key pada tabel b

e : kondisi yang diberikan dalam query

B. Double Nested Query (Double Sub Query)

Double Nested query menghubungkan 3 atau lebih tabel yang dihubungkan secara virtual untuk mendapatkan query seperti yang diinginkan pengguna.

Bab 8. JOIN

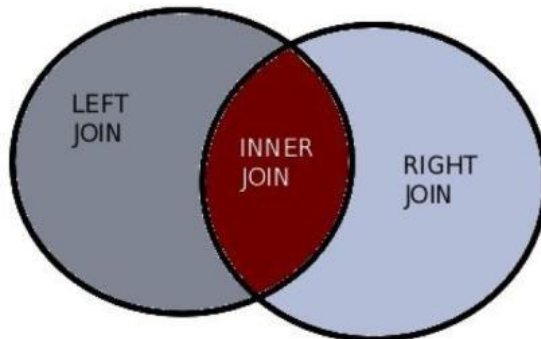
A. Join Without Join Statement

Dalam basis data relasional dimungkinkan untuk mengakses satu atau lebih tabel dalam waktu yang bersamaan. Penggunaan dua tabel atau lebih dalam satu buah baris query biasa disebut dengan JOIN. Query :

```
SELECT <tabel1.field1>,<tabel2.field2>          FROM
      <tabel1>,                                <tabel2>
      WHERE
      <key.tabel1> = <key.tabel2>
```

B. Inner Join dan Outer Join

Di dalam database, ada kalanya kita membutuhkan data dari beberapa tabel yang saling berhubungan. Untuk mendapatkan data dari beberapa tabel tersebut dapat digunakan perintah join pada perintah SQL. Gambaran hasil join :



1. Inner Join

Inner join merupakan jenis join yang paling umum yang dapat digunakan pada semua database. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada di tabel, letak relasinya

setelah pada perintah ON pada join. Hasil dari inner join adalah gabungan kedua tabel yang memiliki data join yang sama. Query:

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> INNER JOIN <tabel2>  
ON <key.tabel1> = <key.tabel2>
```

2. Outer Join

Outer join merupakan join yang sedikit berbeda dengan inner join. Outer join akan menghasilkan record-record yang ada baik ada pasangannya pada tabel lain atau tidak. Outer join dibedakan menjadi 2 jenis, yaitu :

a) Left Outer Join

Left join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set A, terlepas dari apakah nilai kunci ada dalam data set B. Query :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> LEFT JOIN <tabel2> ON  
<key.tabel1> = <key.tabel2>
```

b) Right Outer Join

Right join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set B, terlepas dari apakah nilai kunci ada dalam data set A. Query :

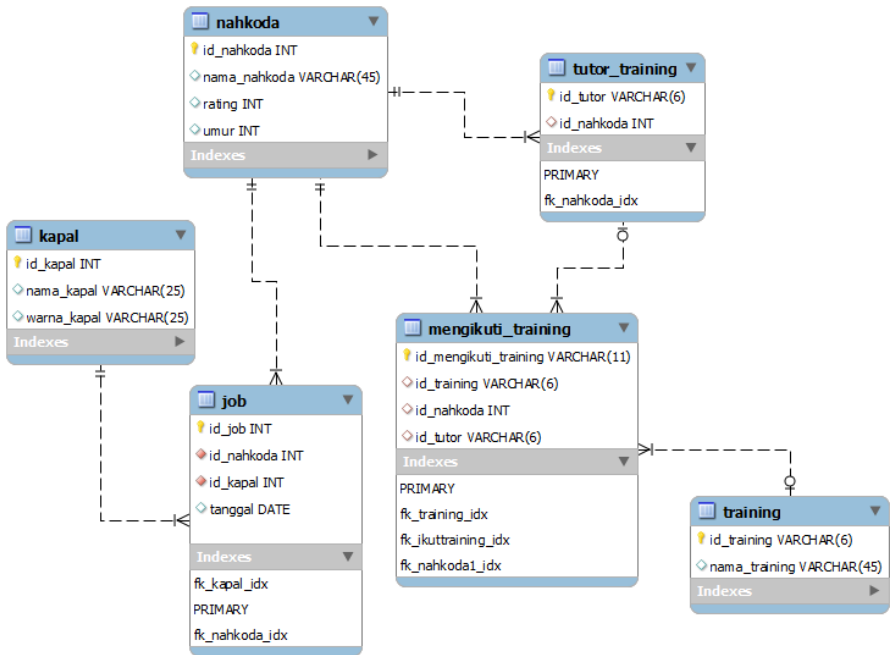
```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> RIGHT JOIN <tabel2>  
ON <key.tabel1> = <key.tabel2>
```

Bab 9. Studi Kasus

A. Penjelasan Studi Kasus Kapal Laut

Pada bab ini dibuat sebuah studi kasus dimana pada operasional sebuah kapal laut terdapat nama kapal laut pada tabel kapal, biodata singkat nahkoda pada tabel nahkoda, sekilas training nahkoda pada tabel training. Ketiga tabel diatas merupakan tabel master sebagai tabel utama yang diakses oleh tabel lain dalam sebuah relasi.

Selanjutnya terdapat tabel job sebagai tabel yang menyimpan jadwal nahkoda berlayar dengan kapal tertentu. Tabel tutor training merupakan tabel yang berisi daftar nahkoda senior yang berhak memberikan training pada nahkoda junior. Tabel mengikuti_training merupakan tabel yang menyimpan jadwal nahkoda junior yang mengikuti training



Gambar 9 Desain database kapal_laut

B. DDL Kapal Laut

```
create database kapal_laut;
use kapal_laut;
```

```

# TABEL NAHKODA
create table nahkoda (
id_nahkoda int (3) not null,
nama_nahkoda char(15) not null,
rating int (2),
umur int (2),
PRIMARY KEY (id_nahkoda)
);
show tables;
desc nahkoda;

insert into nahkoda values ('22', 'Mulyadi','7','45');
insert into nahkoda values ('29', 'Antoni','1','28');
insert into nahkoda values ('31', 'Eko Lelono','8','55');
insert into nahkoda values ('32', 'Mike','8','35');
insert into nahkoda values ('58', 'Lewis','8','44');
insert into nahkoda values ('64', 'Saputro','9','63');
insert into nahkoda values ('71', 'Eka Hadi','3','29');
insert into nahkoda values ('74', 'Dede H','3','30');
insert into nahkoda values ('85', 'Hanan S','5','37');
insert into nahkoda values ('95', 'Surendra','5','35');
select * from nahkoda;

```

```

# TABEL KAPAL
create table kapal (
id_kapal int (3) not null,
nama_kapal char(15) not null,
warna_kapal char (10),
PRIMARY KEY (id_kapal)
);
show tables;
desc kapal;

insert into kapal values ('101', 'Jaya Loka','Merah');
insert into kapal values ('102', 'Jaya Loka','Biru');
insert into kapal values ('103', 'Makmur Abadi','Hijau');
insert into kapal values ('104', 'Bintang Ufuk','Hijau');
insert into kapal values ('105', 'Bintang Ufuk','Merah');
select * from kapal;

```



```
#TABEL JOB
create table job (
id_nahkoda int (3) not null,
id_kapal int (3) not null,
tanggal date,
PRIMARY KEY (id_nahkoda,id_kapal)
);
show tables;
desc job;

insert into job values ('22','101','2016-9-5');
insert into job values ('22','102','2016-9-8');
insert into job values ('22','103','2016-9-15');
insert into job values ('22','104','2016-9-20');
insert into job values ('31','102','2016-10-8');
insert into job values ('31','103','2016-10-15');
insert into job values ('31','104','2016-10-2');
insert into job values ('64','101','2016-10-5');
insert into job values ('64','102','2016-11-8');
insert into job values ('74','103','2016-11-15');
insert into job values ('29','104','2016-11-20');
insert into job values ('32','101','2016-11-5');
insert into job values ('58','104','2016-12-20');
insert into job values ('85','102','2016-12-8');
insert into job values ('85','103','2016-12-15');
insert into job values ('95','101','2016-12-5');
select * from job;
```

```

#TABEL training
create table training (
id_training char (5) not null,
nama_training char(40) not null,
PRIMARY KEY (id_training)
);
show tables;
desc training;

insert into training values ('TN_01','Training Nahkoda Muda');
insert into training values ('TN_02','Training Manajemen Kapal');
insert into training values ('TN_03','Training Mesin Kapal');
insert into training values ('TN_04','Training Nahkoda Menengah');
insert into training values ('TN_05','Training Nahkoda Mahir');
select * from training;

#TABEL mengikuti_training
create table mengikuti_training (
id_training char (5) not null,
id_nahkoda int (3) not null,
PRIMARY KEY (id_training,id_nahkoda)
);
show tables;
desc mengikuti_training;

insert into mengikuti_training values ('TN_01','29');
insert into mengikuti_training values ('TN_02','29');
insert into mengikuti_training values ('TN_03','29');
insert into mengikuti_training values ('TN_02','71');
insert into mengikuti_training values ('TN_03','71');
insert into mengikuti_training values ('TN_05','85');
insert into mengikuti_training values ('TN_05','95');
select * from mengikuti_training;

```

1. Database KAPAL_LAUT

Tabel 10 Tabel isi dari entitas Nahkoda

Tabel :		Nahkoda	
id_nahkoda	nama_nahkoda	rating	umur
22	Mulyadi	7	45
29	Antoni	1	28
31	Eko Lelono	8	55
32	Mike	8	35
58	Lewis	8	44
64	Saputro	9	63
71	Eka Hadi	3	29
74	Dede H	3	30
85	Hanan S	5	37
95	Surendra	5	35

Tabel 11 Tabel isi dari entitas Kapal

Tabel :		Kapal
id_kapal	nama_kapal	warna_kapal
101	Jaya Loka	Merah
102	Jaya Loka	Biru
103	Makmur Abadi	Hijau
104	Bintang Ufuk	Hijau
105	Bintang Ufuk	Merah

Tabel 12 Tabel isi dari entitas Job

Tabel :		Job
id_nahkoda	nama_kapal	tanggal
22	101	2016-9-5
22	102	2016-9-8
22	103	2016-9-15
22	104	2016-9-20
31	102	2016-10-8
31	103	2016-10-15
31	104	2016-10-20
64	101	2016-10-5
64	102	2016-11-8
74	103	2016-11-15
29	104	2016-11-20
32	101	2016-11-5
58	104	2016-12-20
85	102	2016-12-8
85	103	2016-12-15
95	101	2016-12-5

Tabel 13 Tabel isi dari entitas Mengikuti_training

Tabel :		Mengikuti_training
id_training	id_nahkoda	
TN_01	29	
TN_02	29	
TN_02	71	
TN_03	29	
TN_03	71	
TN_05	85	
TN_05	95	

Tabel 14 Tabel isi dari entitas Training

TABEL :	Training
id_training	nama_training
TN_01	Training Nahkoda Muda
TN_02	Training Manajemen Kapal
TN_03	Training Mesin Kapal
TN_04	Training Nahkoda Menengah
TN_05	Training Nahkoda Mahir

2. Query Nahkoda

- a. Usia Nahkoda tertua yang tersimpan dalam database
 select max(umur) from nahkoda;

id_nahkoda	nama_nahkoda	rating	umur
64	Saputro	9	63

- b. Nahkoda dengan ID tertinggi yang tersimpan dalam database

Select max(id_nahkoda) from nahkoda;

id_nahkoda	nama_nahkoda	rating	umur
95	Surendra	5	35

- c. Jumlah Nahkoda yang terdaftar dalam database
 select count(*) from nahkoda;

10

- d. Nahkoda yang memiliki "di" dalam namanya

Select*from nahkoda where nama_nahkoda like '%di%';

id_nahkoda	nama_nahkoda	rating	umur
22	Mulyadi	7	45
71	Eka Hadi	3	29

- e. Nahkoda yang memiliki huruf E dalam namanya

Select*from nahkoda where nama_nahkoda like '%E%';

id_nahkoda	nama_nahkoda	rating	umur
31	Eko Lelono	8	55
32	Mike	8	35
58	Lewis	8	44
71	Eka Hadi	3	29
74	Dede H	3	30
95	Surendra	5	35

- f. Nahkoda yang mendekati masa pensiun (≥ 58 tahun)

Select * from nahkoda where umur ≥ 58 ;

id_nahkoda	nama_nahkoda	rating	umur
64	Saputro	9	63

- g. Nahkoda muda yang baru bergabung (antara 23 sampai 35 tahun)

SELECT * from nahkoda where umur between 23 and 35;

id_nahkoda	nama_nahkoda	rating	umur
29	Antoni	1	28
32	Mike	8	35
71	Eka Hadi	3	29
74	Dede H	3	30
95	Surendra	5	35

- h. Nahkoda dengan prestasi baik (rating ≥ 7)

Select * from nahkoda where rating ≥ 7 ;

id_nahkoda	nama_nahkoda	rating	umur
22	Mulyadi	7	45
31	Eko Lelono	8	55
32	Mike	8	35
58	Lewis	8	44
64	Saputro	9	63

- i. Nahkoda muda butuh training (umur ≤ 35 tahun dan rating ≤ 2)

Select * from nahkoda where umur ≤ 35 and rating ≥ 7 ;

id_nahkoda	nama_nahkoda	rating	umur
32	Mike	8	35

- j. Nahkoda muda butuh training (umur<=35 tahun dan rating<=2)

Select * from nahkoda where umur <= 35 and rating <= 2;

id_nahkoda	nama_nahkoda	rating	umur
29	Antoni	1	28

- k. Rating terendah nahkoda yang disimpan dalam database
select min(rating) from nahkoda;

1

- l. Rating tertinggi nahkoda yang disimpan dalam database
select max(rating) from nahkoda;

9

- m. Nahkoda yang berumur 36 sampai dengan 54 tahun
SELECT id_nahkoda, nama_nahkoda, umur
from nahkoda where umur between 36 and 54;

id_nahkoda	nama_nahkoda	usia_nahkoda
22	Mulyadi	45
58	Lewis	44
85	Hanan S	37

- n. Nahkoda dengan rasio baik (perbandingan umur dan rating <=10)

select * from nahkoda
having umur/rating<=10;

id_nahkoda	nama_nahkoda	rating	umur
22	Mulyadi	7	45
31	Eko Lelono	8	55
32	Mike	8	35
58	Lewis	8	44
64	Saputro	9	63
71	Eka Hadi	3	29
74	Dede H	3	30
85	Hanan S	5	37
95	Surendra	5	35

- o. Tampilkan data nahkoda tertua yang disimpan di database

```
select * from nahkoda
  where umur IN (select max(umur) from
select * from nahkoda
  where umur >=ALL (select umur from nahkoda);
```

id_nahkoda	nama_nahkoda	rating	umur
64	Saputro	9	63

- p. Data Nahkoda yang tidak dapat job

```
select * from nahkoda
  where id_nahkoda NOT IN (select id_nahkoda from
job);
```

id_nahkoda	nama_nahkoda	rating	umur
71	Eka Hadi	3	29

- q. Data Kapal yang mendapat job dan tersimpan dalam database

```
select * from kapal
  where id_kapal IN (select id_kapal from job);
```

id_kapal	nama_kapal	warna_kapal
101	Jaya Loka	Merah
102	Jaya Loka	Biru
103	Makmur Abadi	Hijau
104	Bintang Ufuk	Hijau

- r. Data Kapal yang tidak mendapat job

```
select * from kapal
```

```
where id_kapal NOT IN (select id_kapal from job);
```

id_kapal	nama_kapal	warna_kapal
101	Jaya Loka	Merah

- s. Tampilkan data nahkoda dan jobnya diurutkan

```
id_nahkoda
```

```
select * from nahkoda
```

```
where id_nahkoda IN (select distinct(id_nahkoda)
```

```
from job);
```

id_nahkoda	nama_nahkoda	rating	umur
22	Mulyadi	7	45
29	Antoni	1	28
31	Eko Lelono	8	55
32	Mike	8	35
58	Lewis	8	44
64	Saputro	9	63
74	Dede H	3	30
85	Hanan S	5	37
95	Surendra	5	35

- t. Tampilkan data nahkoda dan jobnya diurutkan rating

```
select rating, id_nahkoda, nama_nahkoda, umur
```

```
from nahkoda
```

```
where id_nahkoda IN (select id_nahkoda from job)
```

```
ORDER BY rating desc;
```

rating	id_nahkoda	nama_nahkoda	umur
9	64	Saputro	63
8	31	Eko Lelono	55
8	32	Mike	35
8	58	Lewis	44
7	22	Mulyadi	45
5	85	Hanan S	37
5	95	Surendra	35
3	74	Dede H	30
1	29	Antoni	28

3. Query Kapal

- a. Jumlah Nahkoda yang terdaftar dalam database dengan nama alias Jumlah_Nahkoda

```
select count(*) AS Jumlah_Nahkoda from kapal;
```

Jumlah_Nahkoda
5

- b. Jumlah kapal berwarna hijau

```
select count(*) from kapal where warna_kapal='Hijau';
```

2

- c. Jumlah kapal JAYA LOKA

```
select count(*) from kapal where nama_kapal='Jaya Loka';
```

2

- d. Tampilkan data kapal BINTANG UFUK

```
select * from kapal where nama_kapal='Bintang Ufuk';
```

id_kapal	nama_kapal	warna_kapal
104	Bintang Ufuk	Hijau
105	Bintang Ufuk	Merah

- e. tampilkan id_kapal untuk kapal MAKMUR ABADI

```
select id_kapal from kapal where nama_kapal='Makmur Abadi';
```

id_kapal
103

- f. Tampilkan nama kapal yang tersimpan dalam database

```
select distinct(nama_kapal) from kapal;
```

nama_kapal
Jaya Loka
Makmur
Abadi
Bintang Ufuk

- g. Tampilkan warna kapal yang ada pada database
 select distinct(warna_kapal) from kapal;

warna_kapal
Merah
Biru
Hijau

4. Query Job

- a. Nahkoda paling banyak job
 SELECT id_nahkoda, count(*) as masing2
 From job
 GROUP BY id_nahkoda;

id_nahkoda	masing2
22	4
29	1
31	3
32	1
58	1
64	2
74	1
85	2
95	1

- b. Nahkoda yang dapat job paling banyak
 SELECT id_nahkoda, count(*) as 'masing2'
 from job
 GROUP BY id_nahkoda
 ORDER BY 'masing2' ASC
 LIMIT 1;

id_nahkoda	masing2
22	4

- c. Jumlah job yang tersimpan dalam database
 select id_nahkoda from nahkoda
 where id_nahkoda IN (select id_nahkoda from job);

id_nahkoda
22
29
31
32
58
64
74
85
95

- d. Jumlah job yang tersimpan dalam database dengan nama alias `Banyak_job`
`select count(*) AS banyak_job from job;`

banyak_job
16

- e. Kapal yang mendapat job dan tersimpan dalam database
`select distinct(nama_kapal) from kapal`
`where id_kapal IN (select id_kapal from job);`

nama_kapal
Jaya Loka
Makmur Abadi
Bintang Ufuk

- f. Menampilkan job sesuai urutan tanggal job tersebut, khusus yang dilaksanakan bulan september
`select tanggal, id_nahkoda, id_kapal`
`from job`
`ORDER BY tanggal ASC`
`LIMIT 4;`

tanggal	id_nahkoda	id_kapal
2016-9-5	22	101
2016-9-8	22	102
2016-9-15	22	103
2016-9-20	22	104
2016-12-5	95	101
2016-12-8	85	102
2016-12-15	85	103
2016-12-20	58	104
2016-11-5	32	101
2016-11-8	64	102
2016-11-15	74	103
2016-11-20	29	104
2016-10-5	64	101
2016-10-8	31	102
2016-10-15	31	103
2016-10-20	31	104

5. Query Join

a. Data Nahkoda sesuai job per tanggal nya

```

select b.tanggal,a.id_nahkoda, a.nama_nahkoda, a.rating,
a.umur
from nahkoda a, job b
where a.id_nahkoda=b.id_nahkoda;

```

tanggal	id_nahkoda	nama_nahkoda	rating	umur
05/09/2016	22	Mulyadi	7	45
08/09/2016	22	Mulyadi	7	45
15/09/2016	22	Mulyadi	7	45
20/09/2016	22	Mulyadi	7	45
20/11/2016	29	Antoni	1	28
08/10/2016	31	Eko Lelono	8	55
15/10/2016	31	Eko Lelono	8	55
02/10/2016	31	Eko Lelono	8	55
05/11/2016	32	Mike	8	35
20/12/2016	58	Lewis	8	44
05/10/2016	64	Saputro	9	63
08/11/2016	64	Saputro	9	63
15/11/2016	74	Dede H	3	30
08/12/2016	85	Hanan S	5	37
15/12/2016	85	Hanan S	5	37
05/12/2016	95	Surendra	5	35

b. Data Kapal yang mendapat job dan tersimpan dalam database diurutkan pada tanggal job nya

```

select b.tanggal, a.id_kapal, a.nama_kapal, a.warna_kapal
from kapal a, job b
where a.id_kapal=b.id_kapal
ORDER BY tanggal;

```

tanggal	id_kapal	nama_kapal	warna_kapal
05/09/2016	101	Jaya Loka	Merah
08/09/2016	102	Jaya Loka	Biru
15/09/2016	103	Makmur Abadi	Hijau
20/09/2016	104	Bintang Ufuk	Hijau
02/10/2016	104	Bintang Ufuk	Hijau
05/10/2016	101	Jaya Loka	Merah
08/10/2016	102	Jaya Loka	Biru
15/10/2016	103	Makmur Abadi	Hijau
05/11/2016	101	Jaya Loka	Merah
08/11/2016	102	Jaya Loka	Biru
15/11/2016	103	Makmur Abadi	Hijau
20/11/2016	104	Bintang Ufuk	Hijau
05/12/2016	101	Jaya Loka	Merah
08/12/2016	102	Jaya Loka	Biru
15/12/2016	103	Makmur Abadi	Hijau
20/12/2016	104	Bintang Ufuk	Hijau

- c. **Tampilkan data kapal yang berlayar 15 september 2016**
 select b.tanggal, a.id_kapal, a.nama_kapal, a.warna_kapal
 from kapal a, job b
 where a.id_kapal=b.id_kapal AND tanggal ='2016-9-5';

tanggal	id_kapal	nama_kapal	warna_kapal
05/09/2016	101	Jaya Loka	Merah

- d. **Tampilkan record pada tabel job yang nahkodanya memiliki rating 9 ?**

select b.tanggal,a.id_nahkoda, a.nama_nahkoda
 from nahkoda a, job b
 where a.id_nahkoda=b.id_nahkoda AND a.rating='9';

tanggal	id_nahkoda	nama_nahkoda
05/10/2016	64	Saputro
08/11/2016	64	Saputro

- e. Tampilkan job yang nahkodanya memiliki rating 7 sampai 9 diurutkan dari tanggal terlama

select b.*

from nahkoda a, job b

where a.id_nahkoda=b.id_nahkoda AND a.rating

BETWEEN 7 AND 9

order by tanggal asc;

id_nahkoda	id_kapal	tanggal
22	101	05/09/2016
22	102	08/09/2016
22	103	15/09/2016
22	104	20/09/2016
31	104	02/10/2016
64	101	05/10/2016
31	102	08/10/2016
31	103	15/10/2016
32	101	05/11/2016
64	102	08/11/2016
58	104	20/12/2016

- f. Tampilkan data peserta training beserta nama training yang tersimpan di database

select b.id_nahkoda, c.nama_nahkoda, a.nama_training

from training a, mengikuti_training b, nahkoda c

where a.id_training=b.training,

b.id_nahkoda=c.id_nahkoda ;

id_nahkoda	nama_nahkoda	nama_training
29	Antoni	Training Nahkoda Muda
29	Antoni	Training Manajemen Kapal
71	Eka Hadi	Training Manajemen Kapal
29	Antoni	Training Mesin Kapal
71	Eka Hadi	Training Mesin Kapal
85	Hanan S	Training Nahkoda Mahir
95	Surendra	Training Nahkoda Mahir

g. Data Nahkoda yang berlayar pada bulan November

```
select b.tanggal,a.id_nahkoda, a.nama_nahkoda
from nahkoda a, job b
where a.id_nahkoda=b.id_nahkoda
LIMIT 4;
```

tanggal	id_nahkoda	nama_nahkoda	rating	umur
05/09/2016	22	Mulyadi	7	45
08/09/2016	22	Mulyadi	7	45
15/09/2016	22	Mulyadi	7	45
20/09/2016	22	Mulyadi	7	45

h. Tampilkan id_nahkoda yang mengikuti training dan nama_training yang diikuti

```
SELECT mengikuti_training.id_nahkoda,
training.nama_training
FROM training RIGHT JOIN mengikuti_training
ON training.id_training=mengikuti_training.id_training;
```

id_nahkoda	nama_training
29	Training Nahkoda Muda
29	Training Manajemen Kapal
71	Training Manajemen Kapal
29	Training Mesin Kapal
71	Training Mesin Kapal
85	Training Nahkoda Mahir
95	Training Nahkoda Mahir

i. Data Nahkoda keseluruhan agar terlihat Nahkoda yang tidak mendapat job

```
select job.tanggal,nahkoda.id_nahkoda,
nahkoda.nama_nahkoda, nahkoda.rating, nahkoda.umur
from nahkoda LEFT JOIN job
ON nahkoda.id_nahkoda=job.id_nahkoda;
```

tanggal	id_nahkoda	nama_nahkoda	rating	umur
05/09/2016	22	Mulyadi	7	45
08/09/2016	22	Mulyadi	7	45
15/09/2016	22	Mulyadi	7	45
20/09/2016	22	Mulyadi	7	45
20/11/2016	29	Antoni	1	28
08/10/2016	31	Eko Lelono	8	55
15/10/2016	31	Eko Lelono	8	55
02/10/2016	31	Eko Lelono	8	55
05/11/2016	32	Mike	8	35
20/12/2016	58	Lewis	8	44
05/10/2016	64	Saputro	9	63
08/11/2016	64	Saputro	9	63
NULL	71	Eka Hadi	3	29
15/11/2016	74	Dede H	3	30
08/12/2016	85	Hanan S	5	37
15/12/2016	85	Hanan S	5	37
05/12/2016	95	Surendra	5	35

D. DML Nested Soal Kapal Laut

1. Tampilkan data kapal yang berlayar yang berwarna hijau !

```
select * from kapal where id_kapal in (select id_kapal from job where warna_kapal ='hijau');
```

id_kapal	nama_kapal	warna_kapal
103	Makmur Abadi	Hijau
104	Bintang Ufuk	Hijau

2. Tampilkan rata rata umur nahkoda yang tidak mengikuti training dengan nama alias 'Rata-Rata Umur Nahkoda yg Tidak Training'!

```
select avg(umur) as 'Rata-Rata Umur Nahkoda yg Tidak Training' from nahkoda where id_nahkoda not in (select id_nahkoda from mengikuti_training);
```

Rata-Rata Umur Nahkoda yg Tidak Training
45.3333

3. Tampilkan data nahkoda yang mengikuti training dengan umur nahkoda > 25 !

```
select * from nahkoda where id_nahkoda in (select id_nahkoda from mengikuti_training where umur>25);
```

id_nahkoda	nama_nahkoda	rating	umur
29	Antoni	1	28
71	Eka Hadi	3	29
85	Hanan S	5	37
95	Surendra	5	35

4. Tampilkan nama training yang diikuti nahkoda dengan id training antara TN_02 sampai TN_04 !

```
select nama_training from training where id_training in
(select id_training from mengikuti_training where id_training
between 'TN_02' and 'TN_04');
```

nama_training
Training Manajemen Kapal
Training Mesin Kapal

5. Tampilkan nama kapal dan warna kapal yang berlayar dibulan oktober dengan nama alias 'Nama Kapal yg Berlayar Bulan 10'!

```
select nama_kapal,warna_kapal from kapal where id_kapal in
(select id_kapal from job where tanggal like '%-10-%');
```

nama_kapal	warna_kapal
Jaya Loka	Merah
Jaya Loka	Biru
Makmur Abadi	Hijau
Bintang Ufuk	Hijau

Daftar Pustaka

Cheung, W. and Babin, G. (2006) 'A metadatabase-enabled executive information system (Part A): A flexible and adaptable architecture', *Decision Support Systems*, 42(3), pp. 1589–1598. doi: 10.1016/j.dss.2006.01.005.

Kroenke, D. M. et al. (2017) *Database Concept. Eight Edit.* Hoboken, New Jersey: Pearson.

Lotfy, A., El Tazi, N. and El Gamal, N. (2016) 'SCI-F: Social-corporate data integration framework', *ACM International Conference Proceeding Series*, 11-13-July, pp. 328–333. doi: 10.1145/2938503.2938505.

Simarmata, J. and Paryudi, I. (2006) *Basis Data.*

Vajirakachorn, T. and Chongwatpol, J. (2017) 'Application of business intelligence in the tourism industry: A case study of a local food festival in Thailand', *Tourism Management Perspectives*, 23, pp. 75–86. doi: 10.1016/j.tmp.2017.05.003.