

PETUNJUK PELAKSANAAN PRAKTIKUM

BASIS DATA

Oleh

Mirza Yogy Kurniawan

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ISLAM KALIMANTAN MUHAMMAD ARSYAD AL BANJARI
BANJARMASIN**

DAFTAR ISI

DAFTAR ISI.....	<i>i</i>
ATURAN UMUM	<i>iii</i>
KELENGKAPAN.....	<i>iii</i>
PRASYARAT	<i>iii</i>
KESELAMATAN DAN KESEHATAN KERJA (K3).....	<i>iii</i>
KONTAK.....	<i>iv</i>
SESI 1 Pengenalan MySQL dan Operasi DDL	1
TUJUAN PRAKTEK	1
PENDAHULUAN.....	1
PERCOBAAN 1.1 Menjalankan MySQL dan Operasi DDL Basis data.....	2
KESIMPULAN 1.1.....	8
LATIHAN 1.1.....	8
PERCOBAAN 1.2 Operasi DDL Tabel Bagian 1	9
KESIMPULAN 1.2.....	13
LATIHAN 1.2.....	13
PERCOBAAN 1.3 Operasi DDL Tabel Bagian 2	14
LATIHAN 1.3.....	15
SESI 2 Operasi DML.....	16
TUJUAN PRAKTEK	16
PENDAHULUAN.....	16
PERCOBAAN 2.1 Memasukkan Record dan Menampilkannya	17
KESIMPULAN 2.1.....	19
LATIHAN 2.1.....	19
PERCOBAAN 2.2 Mengubah Record	20
KESIMPULAN 2.2.....	21
LATIHAN 2.2.....	21
PERCOBAAN 2.3 Menghapus Record	22
LATIHAN 2.3.....	23
SESI 3 Seleksi Record.....	24
TUJUAN PRAKTEK	24
PENDAHULUAN.....	24
PERCOBAAN 3.1 Menampilkan Record.....	25
PERCOBAAN 3.2 Menampilkan Record dengan Kondisi Tertentu.....	27

PERCOBAAN 3.3 Menampilkan Record dengan Kondisi Between	28
PERCOBAAN 3.4 Menampilkan Record dengan Kondisi Like	29
PERCOBAAN 3.5 Menampilkan Record dengan Urutan	30
PERCOBAAN 3.6 Fungsi Statistik dan Pengelompokan	32
LATIHAN 3	34
<i>SESI 4 Relasi Tabel dan Seleksinya</i>	36
TUJUAN PRAKTEK	36
PENDAHULUAN.....	36
PERCOBAAN 4.1 Relasi Tabel Karyawan ke Tabel Bagian	37
PERCOBAAN 4.2 Relasi Tabel Karyawan ke Tabel Penggajian	41
LATIHAN 4	43
PENUTUP	43
DAFTAR PUSTAKA.....	44

ATURAN UMUM

KELENGKAPAN

1. Peserta praktikum wajib berpakaian sopan dan formal, mengenakan celana panjang/rok, kemeja, dan sepatu.
2. Peserta diwajibkan membawa laptop masing-masing dan stop kontak jika diperlukan.
3. Peserta menyiapkan kuota internet sebagai cadangan

PRASYARAT

Prasyarat yang harus dilakukan peserta sebelum pelaksanaan praktikum

1. Mengunduh dan memasang aplikasi XAMPP
(<https://www.apachefriends.org/download.html>)

KESELAMATAN DAN KESEHATAN KERJA (K3)

1. Sebelum memulai praktikum, praktikan memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik (stop kontak dan circuit breaker)
3. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera lapor ke asisten terkait atau dapat langsung melapor ke laboran.
4. Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik (sengatan listrik) secara tidak sengaja, misal kabel yang terkelupas
5. Keringkan bagian tubuh yang basah, seperti keringat atau sisa air wudhu
6. Selalu waspada terhadap bahaya listrik pada setiap aktifitas praktikum.

7. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
 - a. Jangan panik
 - b. Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikum yang tersengat arus listrik.
 - c. Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik
 - d. Beritahukan dan minta bantuan kepada laboran, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik.
8. Jangan membawa benda-benda mudah terbakar (korek api, gas, dll) ke dalam ruangan laboratorium bila tidak disyaratkan dalam modul praktikum.
9. Jangan melakukan sesuatu yang menimbulkan api, percikan api, atau panas yang berlebihan.
10. Jangan melakukan sesuatu yang menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain.
11. Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
12. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai.
13. Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain.

KONTAK

Telegram: @mirzayogy

SESI 1

Pengenalan MySQL dan Operasi DDL

TUJUAN PRAKTEK

1. Mahasiswa mampu menjalankan aplikasi MySQL melalui shell
2. Mahasiswa mampu melakukan operasi DDL menggunakan SQL

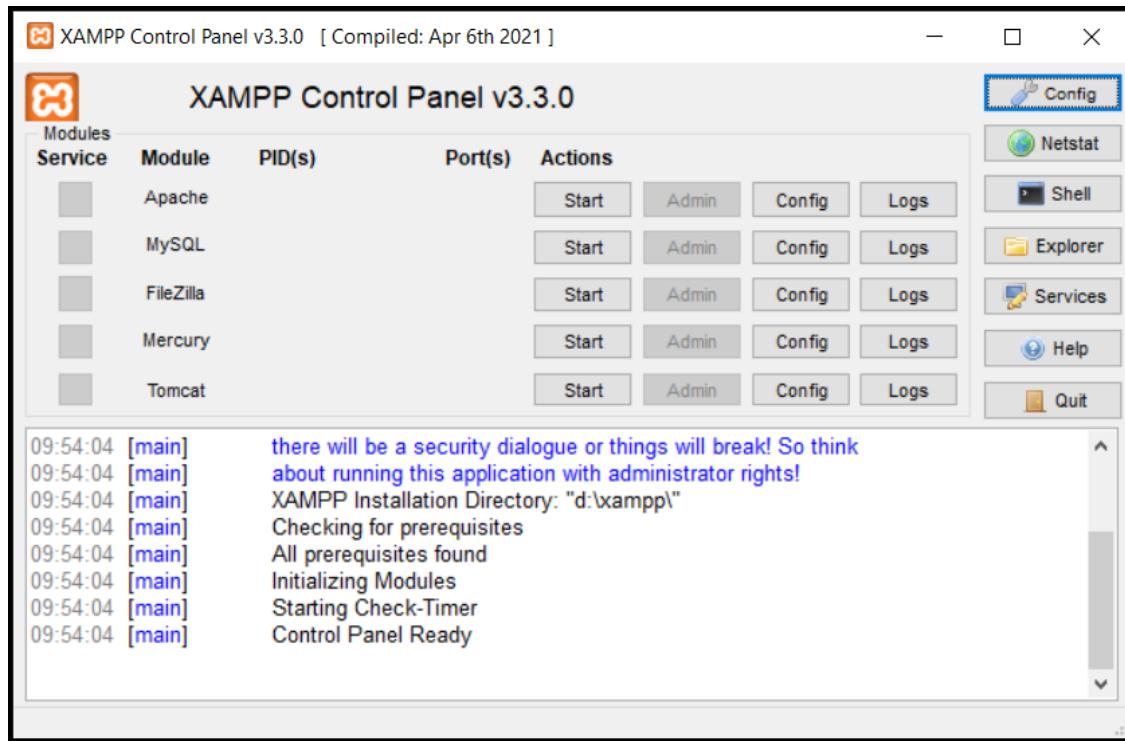
PENDAHULUAN

MySQL merupakan sebuah bahasa basis data yang bersifat open source dan gratis sehingga umum digunakan sebagai basis data dari sebuah aplikasi web. Ada beberapa tools yang bisa menjalankan MySQL, salah satu diantaranya adalah MariaDB yang sudah dipaketkan bersama virtual server Apache dalam package XAMPP. Pada package ini pengoperasian MySQL bisa menggunakan MariaDB ataupun PHPMyAdmin, akan tetapi pada praktek ini akan digunakan MariaDB agar mahasiswa paham dasar pemrograman basis data.

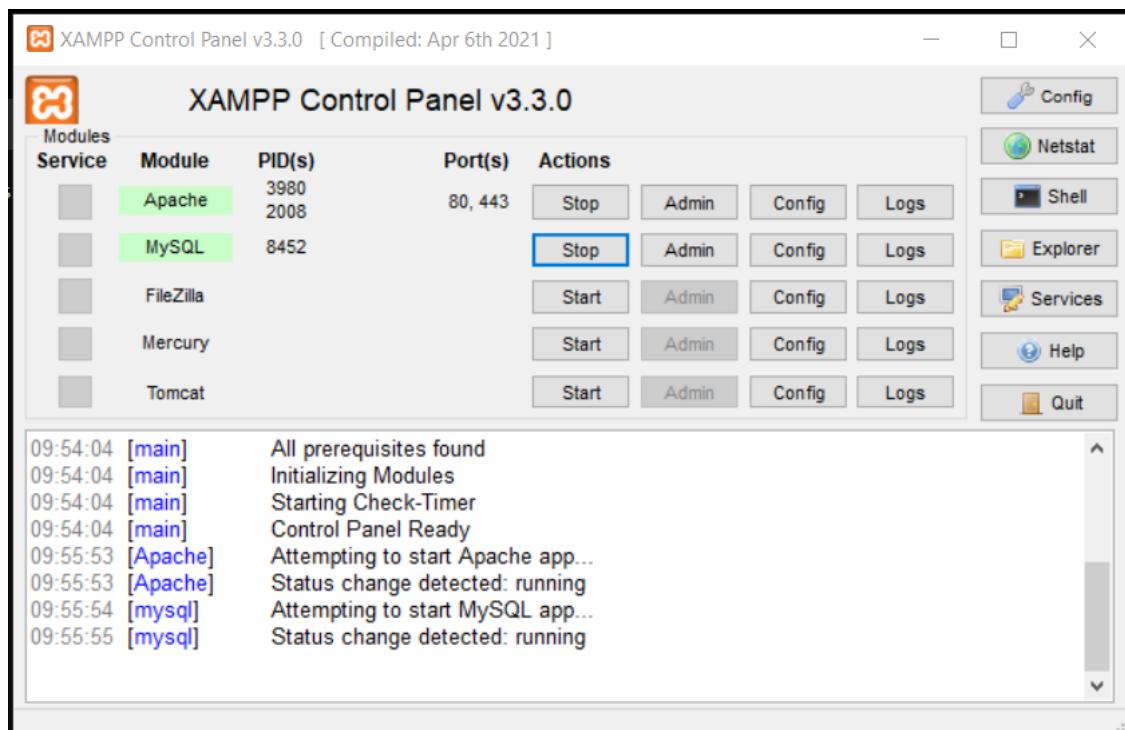
Data Definition Language (DDL) merupakan bahasa yang terstandarisasi dan digunakan untuk menciptakan, memodifikasi, dan menghapus basis data beserta objeknya seperti tabel, index, dan stogroups. DDL sifatnya mendefinisikan struktur (field, tipe data, hak akses, dll) dari sebuah database maupun tabel, belum terkait dengan isi data (record).

PERCOBAAN 1.1 Menjalankan MySQL dan Operasi DDL Basis data

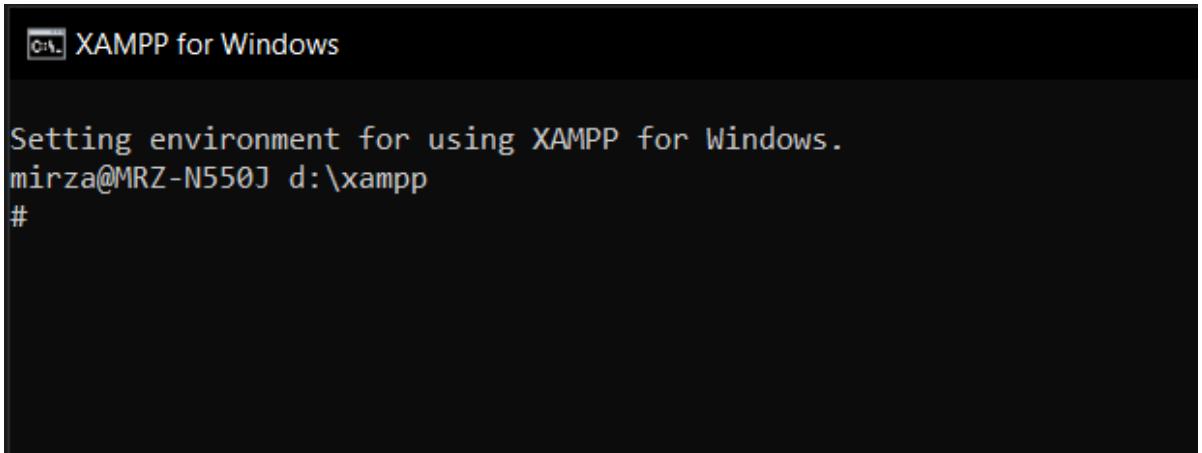
1. Jalankan aplikasi XAMPP, tekan tombol Start pada module Apache dan MySQL



2. Tampak hasilnya Module Apache dan MySQL berwarna hijau, lanjutkan dengan menekan tombol Shell pada bagian kanan.



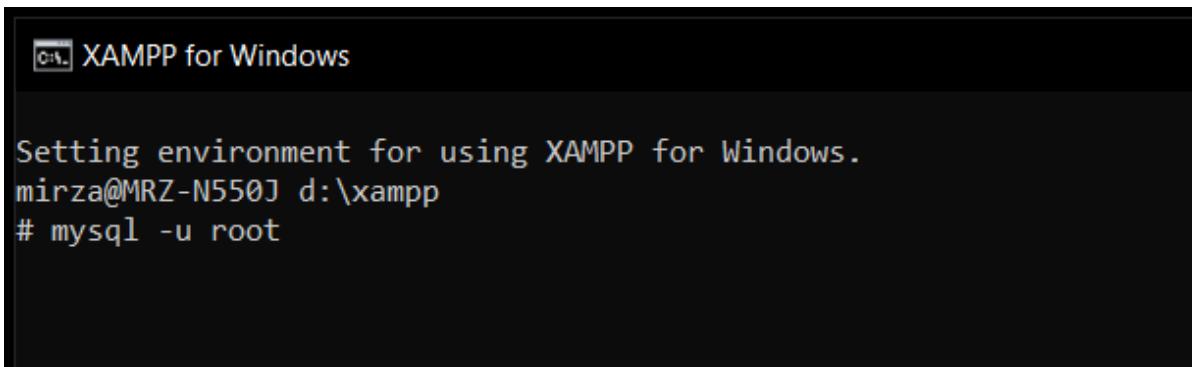
3. Muncul sebuah command prompt seperti gambar berikut.



```
Setting environment for using XAMPP for Windows.  
mirza@MRZ-N550J d:\xampp  
#
```

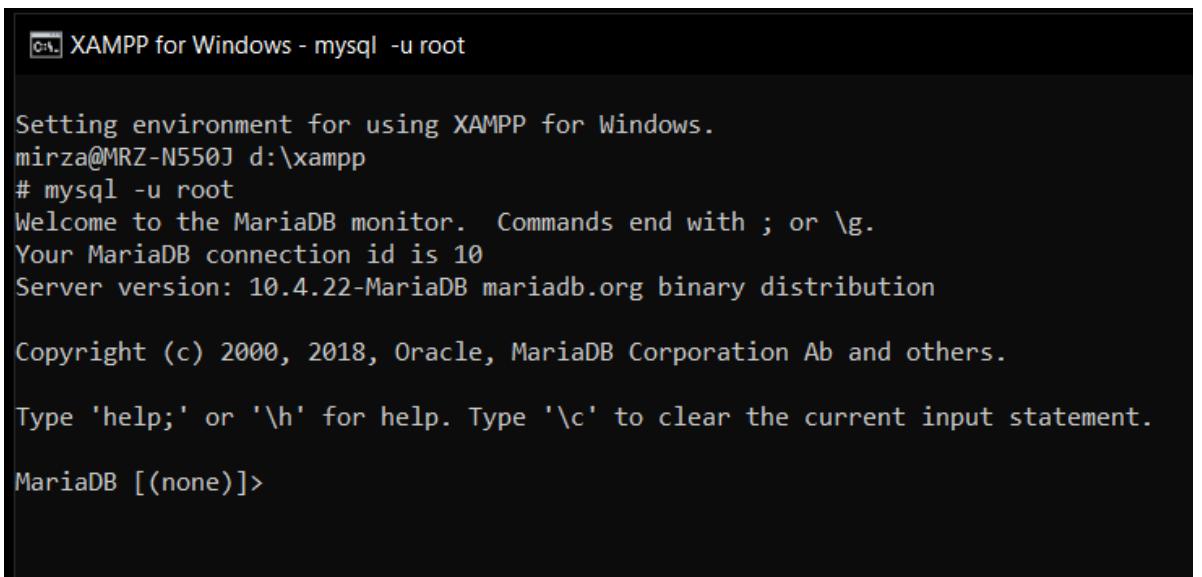
4. Ketikkan perintah

```
mysql -u root
```



```
Setting environment for using XAMPP for Windows.  
mirza@MRZ-N550J d:\xampp  
# mysql -u root
```

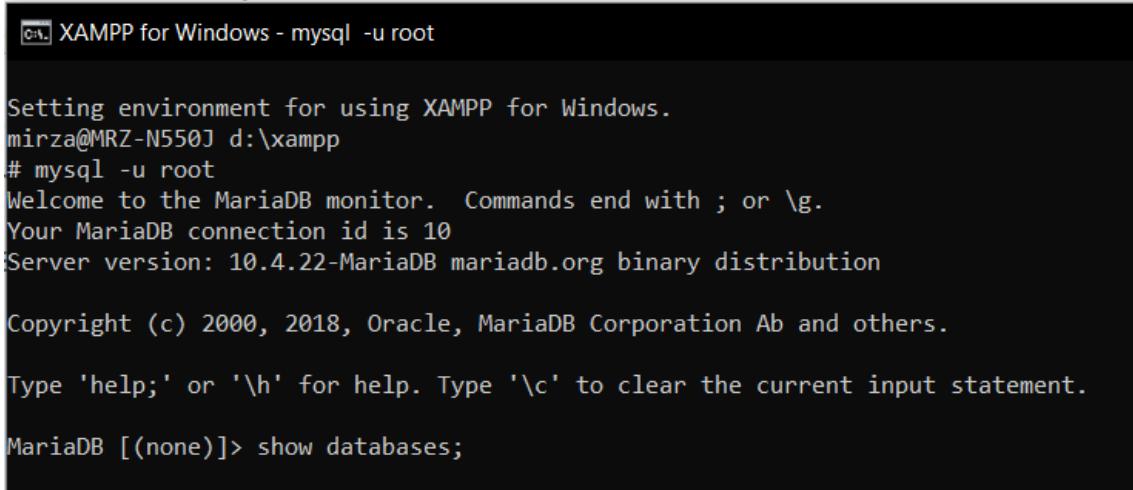
5. Berikut hasil tampilannya



```
XAMPP for Windows - mysql -u root  
Setting environment for using XAMPP for Windows.  
mirza@MRZ-N550J d:\xampp  
# mysql -u root  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 10  
Server version: 10.4.22-MariaDB mariadb.org binary distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]>
```

6. Ketikkan perintah

```
show databases;
```



```
[mysql] XAMPP for Windows - mysql -u root

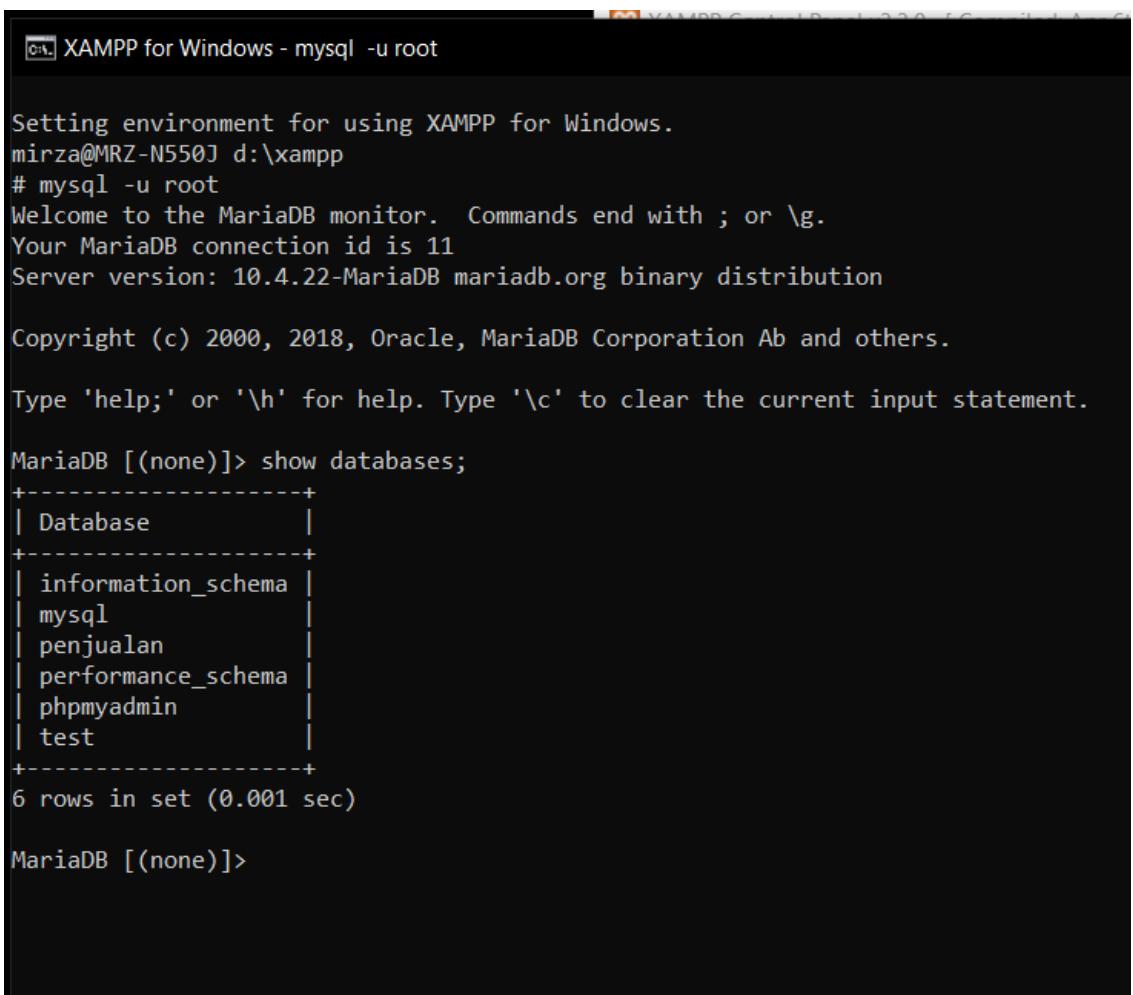
Setting environment for using XAMPP for Windows.
mirza@MRZ-N550J d:\xampp
# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.4.22-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
```

7. Berikut hasilnya



```
[mysql] XAMPP for Windows - mysql -u root

Setting environment for using XAMPP for Windows.
mirza@MRZ-N550J d:\xampp
# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.4.22-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

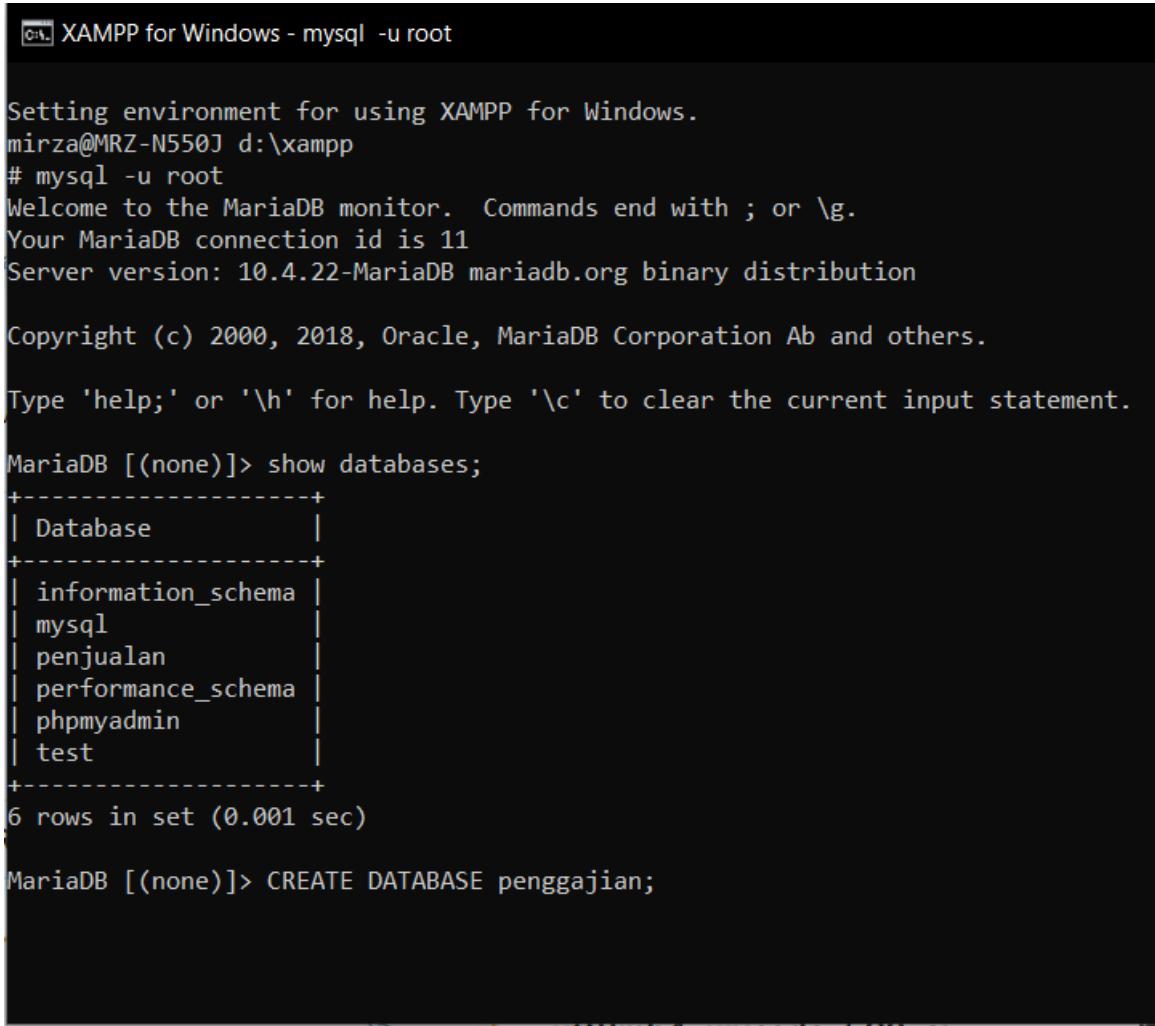
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| penjualan      |
| performance_schema |
| phpmyadmin     |
| test           |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]>
```

Perintah ini menampilkan daftar basis data yang ada pada aplikasi MySQL

8. Ketikkan perintah

```
CREATE DATABASE penggajian;
```



The screenshot shows a terminal window titled "XAMPP for Windows - mysql -u root". It displays the MySQL monitor welcome message, the version information, and the results of the "show databases;" command, which lists several existing databases. Then, the "CREATE DATABASE penggajian;" command is executed.

```
XAMPP for Windows - mysql -u root

Setting environment for using XAMPP for Windows.
mirza@MRZ-N550J d:\xampp
# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.4.22-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| penjualan      |
| performance_schema |
| phpmyadmin     |
| test           |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]> CREATE DATABASE penggajian;
```

9. Berikut adalah respon yang diharapkan, menampilkan bahwa Query OK

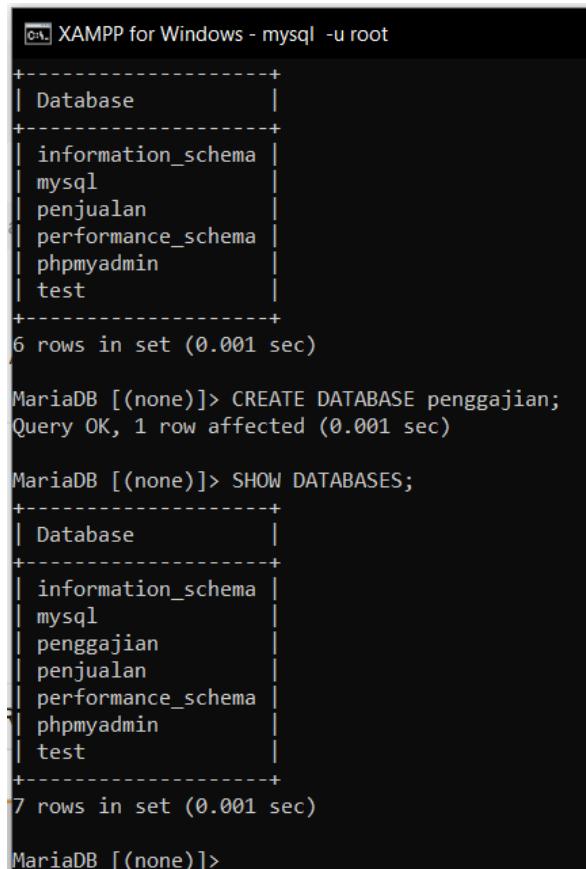
```
MariaDB [(none)]> CREATE DATABASE penggajian;
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [(none)]>
```

Perintah tersebut merupakan perintah untuk membuat sebuah basis data baru dengan nama 'penggajian'. Perintah MySQL tidak sensitif terhadap huruf besar kecil, penggunaan huruf kapital semua hanya untuk membedakan sintaks bahasa MySQL dan penamaan yang kita buat sendiri

10. Guna memastikan bahwa basis data dibuat maka bisa dijalankan kembali perintah

```
SHOW DATABASES;
```



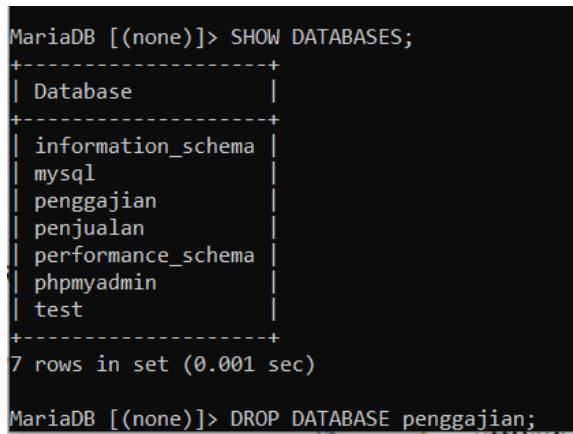
The screenshot shows a terminal window titled "XAMPP for Windows - mysql -u root". It displays the following MySQL session:

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| penjualan |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
6 rows in set (0.001 sec)  
  
MariaDB [(none)]> CREATE DATABASE penggajian;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| penggajian |  
| penjualan |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
7 rows in set (0.001 sec)  
  
MariaDB [(none)]>
```

Pada gambar dapat diamati bahwa yang tadinya hanya ada 6 rows sekarang bertambah menjadi 7 rows, dan dalam kolom 'Database' kita bisa temukan 'penggajian'

11. Berikutnya tuliskan perintah

```
DROP DATABASE penggajian;
```



The screenshot shows a terminal window titled "XAMPP for Windows - mysql -u root". It displays the following MySQL session:

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| penggajian |  
| penjualan |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
7 rows in set (0.001 sec)  
  
MariaDB [(none)]> DROP DATABASE penggajian;
```

12. Hasil yang diharapkan adalah munculnya pesan Query OK, kemudian untuk memastikan bisa kembali dijalankan perintah untuk menampilkan daftar basis data.

```
MariaDB [(none)]> DROP DATABASE penggajian;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| penjualan      |
| performance_schema |
| phpmyadmin     |
| test           |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]>
```

Pada gambar nampak bahwa basis data `penggajian` sudah tidak ada, yang menunjukkan bahwa DROP DATABASE adalah perintah untuk menghapus basis data.

13. Ketikkan perintah untuk keluar dari aplikasi MySQL

QUIT

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| penjualan      |
| performance_schema |
| phpmyadmin     |
| test           |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]> QUIT
Connection closed.
```

14. Berikut hasilnya.

```
MariaDB [(none)]> QUIT
Bye

mirza@MRZ-N550J d:\xampp
#
```

KESIMPULAN 1.1

1. Menjalankan MySQL diawali dengan menjalankan aplikasi XAMPP, tekan tombol Start pada Apache dan MySQL, kemudian klik tombol Shell

2. Menjalankan MySQL pada Shell dengan mengetikkan perintah

```
mysql -u root
```

3. Menampilkan daftar basis data dengan perintah

```
SHOW DATABASES;
```

4. Membuat basis data dengan perintah

```
CREATE DATABASE nama_basis_data;
```

5. Menghapus basis data dengan perintah

```
DROP DATABASE nama_basis_data;
```

6. Keluar dari MySQL dengan perintah

```
QUIT
```

LATIHAN 1.1

1. Buat tangkapan layar dari hasil tiap percobaan dan jadikan satu dalam sebuah dokumen Word/PDF
2. Buat 3 (tiga) basis data, dan tampilkan hasilnya kemudian hapuskan salah satu diantaranya. Sajikan jawaban dengan gambar tangkapan layar dan penjelasan pada tiap tahapan yang anda kerjakan pada file yang sama dengan nomor 1

PERCOBAAN 1.2 Operasi DDL Tabel Bagian 1

1. Menggunakan perintah CLS membersihkan layar Shell, dengan catatan bahwa posisi sedang tidak membuka MySQL atau setelah mengetikkan QUIT

The screenshot shows two terminal windows. The left window is titled 'XAMPP for Windows' and displays MySQL commands and their results. It starts with a 'SHOW DATABASES' command showing seven databases: mysql, penggajian, penjualan, performance_schema, phpmyadmin, and test. Then it runs a 'DROP DATABASE penggajian;' command, followed by another 'SHOW DATABASES' command showing six databases. Finally, it exits with 'QUIT'. The right window is also titled 'XAMPP for Windows' and shows a command prompt with the user 'mirza@MRZ-N550J' and the path 'd:\xampp'. It ends with a '#'. The bottom of the left window shows the command 'mirza@MRZ-N550J d:\xampp # CLS'.

```
| mysql  
| penggajian  
| penjualan  
| performance_schema  
| phpmyadmin  
| test  
+-----+  
7 rows in set (0.001 sec)  
  
MariaDB [(none)]> DROP DATABASE penggajian;  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| penjualan |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
6 rows in set (0.001 sec)  
  
MariaDB [(none)]> QUIT  
Bye  
  
mirza@MRZ-N550J d:\xampp  
#
```

2. Buka kembali MySQL kemudian buat kembali basis data dengan nama 'penggajian'

The screenshot shows a terminal window titled 'XAMPP for Windows - mysql -u root'. It starts with a MySQL connection command 'mysql -u root'. The server then displays its welcome message, including the connection id (14), server version (10.4.22-MariaDB), and copyright information. Finally, it executes a 'CREATE DATABASE penggajian;' command, which is successful.

```
mirza@MRZ-N550J d:\xampp  
# mysql -u root  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 14  
Server version: 10.4.22-MariaDB mariadb.org binary distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> CREATE DATABASE penggajian;  
Query OK, 1 row affected (0.001 sec)
```

3. Pilih basis data 'penggajian' dengan perintah

```
USE penggajian
```

```
MariaDB [(none)]> USE penggajian;
Database changed
MariaDB [penggajian]>
```

4. Dapat diamati bahwa terjadi perubahan pada MariaDB yang tadinya tertulis (none) yang bermakna tidak ada basis data yang sedang dipilih, sekarang berubah menjadi `penggajian`
5. Tampilkan daftar tabel yang ada pada basis data `penggajian` dengan perintah
SHOW TABLES;

```
MariaDB [(none)]> USE penggajian;
Database changed
MariaDB [penggajian]> SHOW TABLES;
```

6. Berikut hasilnya

```
MariaDB [(none)]> USE penggajian;
Database changed
MariaDB [penggajian]> SHOW TABLES;
Empty set (0.001 sec)

MariaDB [penggajian]>
```

Tampak bahwa responnya adalah Empty set, karena basis datanya baru dibuat sehingga memang belum ada isi tabelnya.

7. Lanjutkan dengan menuliskan perintah

```
CREATE TABLE karyawan (
    nik VARCHAR(255),
    nama VARCHAR(255)
) Engine = InnoDB;
```

```

MariaDB [penggajian]> CREATE TABLE karyawan(
-> nik VARCHAR(255),
-> nama VARCHAR(255)
-> ) Engine = InnoDB;
Query OK, 0 rows affected (0.442 sec)

MariaDB [penggajian]>

```

8. Setelah tabel berhasil dibuat, jalankan kembali perintah untuk menampilkan daftar tabel pada basis data `penggajian`

```

MariaDB [penggajian]> CREATE TABLE karyawan(
-> nik VARCHAR(255),
-> nama VARCHAR(255)
-> ) Engine = InnoDB;
Query OK, 0 rows affected (0.442 sec)

MariaDB [penggajian]> SHOW TABLES;
+-----+
| Tables_in_penggajian |
+-----+
| karyawan           |
+-----+
1 row in set (0.000 sec)

MariaDB [penggajian]>

```

Bisa dilihat respon dari aplikasi menampilkan sebuah kolom `Tables_in_penggajian` dengan isi berupa sebuah tabel dengan nama `karyawan`.

9. Tampilkan struktur tabel karyawan dengan menggunakan perintah
`DESCRIBE karyawan;`

```

MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nik   | varchar(255) | YES  |     | NULL    |       |
| nama  | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.117 sec)

MariaDB [penggajian]>

```

10. Hapus tabel 'karyawan' dengan perintah

```
DROP TABLE karyawan;
```

```
MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nik   | varchar(255) | YES  |     | NULL    |       |
| nama  | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.013 sec)

MariaDB [penggajian]> DROP TABLE karyawan;
Query OK, 0 rows affected (0.213 sec)

MariaDB [penggajian]>
```

11. Coba jalankan perintah untuk menampilkan tabel, dan perintah untuk menampilkan struktur tabel karyawan, kemudian amati hasilnya

```
MariaDB [penggajian]> DROP TABLE karyawan;
Query OK, 0 rows affected (0.213 sec)

MariaDB [penggajian]> SHOW TABLES;
Empty set (0.000 sec)

MariaDB [penggajian]> DESCRIBE karyawan;
ERROR 1146 (42S02): Table 'penggajian.karyawan' doesn't exist
MariaDB [penggajian]>
```

KESIMPULAN 1.2

1. Perintah memilih basis data untuk kemudian dioperasikan

```
USE nama_basis_data;
```

2. Perintah menampilkan daftar tabel dalam suatu basis data

```
SHOW TABLES;
```

3. Perintah untuk membuat tabel

```
CREATE TABLE nama_tabel ( ↴  
  nama_kolom_1 TIPE_DATA, ↴  
  nama_kolom_2 TIPE_DATA ↴  
) Engine = InnoDB; ↴
```

4. Perintah untuk menampilkan struktur dari tabel

```
DESCRIBE nama_tabel;
```

5. Perintah untuk menghapus tabel

```
DROP TABLE nama_tabel;
```

LATIHAN 1.2

1. Buat tangkapan layar dari hasil tiap percobaan dan jadikan satu dalam dokumen Word/PDF Latihan 1.1
2. Buat sebuah basis data yang berisikan 3 (tiga) tabel dengan masing-masing tabel minimal berisikan 3 (tiga) kolom.
3. Eksplorasi jenis tipe data yang bisa digunakan selain VARCHAR.
4. Hapus salah satu dari tabel tersebut.
5. Sajikan jawaban dengan gambar tangkapan layar (screenshot) pada tiap tahapan yang anda kerjakan dan deskripsikan.

PERCOBAAN 1.3 Operasi DDL Tabel Bagian 2

1. Buat kembali tabel `karyawan` pada basis data `penggajian`.

```
MariaDB [penggajian]> CREATE TABLE karyawan(
-> nik VARCHAR(255),
-> nama VARCHAR(255)
-> ) Engine = InnoDB;
Query OK, 0 rows affected (0.249 sec)
```

2. Tuliskan perintah berikut, perhatikan respon dan dampaknya pada tabel `karyawan`

```
MariaDB [penggajian]> ALTER TABLE karyawan ADD alamat VARCHAR(255);
Query OK, 0 rows affected (0.244 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [penggajian]>
```

```
MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nik   | varchar(255) | YES  |     | NULL    |       |
| nama  | varchar(255) | YES  |     | NULL    |       |
| alamat | text        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.009 sec)

MariaDB [penggajian]>
```

3. Tuliskan perintah berikut, perhatikan respon dan dampaknya pada tabel `karyawan`

```
MariaDB [penggajian]> ALTER TABLE karyawan CHANGE COLUMN alamat alamat_lengkap TEXT;
Query OK, 0 rows affected (0.117 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nik        | varchar(255) | YES  |     | NULL    |       |
| nama       | varchar(255) | YES  |     | NULL    |       |
| alamat_lengkap | text        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.008 sec)

MariaDB [penggajian]>
```

4. Tuliskan perintah berikut, perhatikan respon dan dampaknya pada tabel 'karyawan'

```
MariaDB [penggajian]> ALTER TABLE karyawan DROP COLUMN alamat_lengkap;
Query OK, 0 rows affected (0.131 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| nik   | varchar(255) | YES  |      | NULL    |       |
| nama  | varchar(255)  | YES  |      | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.010 sec)

MariaDB [penggajian]>
```

5. Tuliskan perintah berikut, perhatikan respon dan dampaknya pada tabel 'karyawan'

```
MariaDB [penggajian]> ALTER TABLE karyawan MODIFY nik VARCHAR(255) NOT NULL PRIMARY KEY;
Query OK, 0 rows affected (0.625 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [penggajian]> DESCRIBE karyawan;
+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| nik   | varchar(255) | NO   | PRI | NULL    |       |
| nama  | varchar(255)  | YES  |      | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.011 sec)

MariaDB [penggajian]>
```

6. Tuliskan perintah berikut, perhatikan respon dan dampaknya pada tabel 'karyawan'

```
MariaDB [penggajian]> ALTER TABLE karyawan DROP PRIMARY KEY;
Query OK, 0 rows affected (0.706 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

LATIHAN 1.3

1. Jelaskan fungsi dari perintah – perintah yang sudah diketik pada percobaan 1.3
2. Terapkan pada tabel yang sudah dibuat pada latihan 1.2
3. Tampilan hasil tangkapan layarnya.
4. Kumpulkan semua hasil percobaan dan latihan dalam bentuk Word/PDF latihan dalam 1 (satu) file dan dikumpulkan pada Google Form yang disediakan

SESI 2

Operasi DML

TUJUAN PRAKTEK

1. Mahasiswa mampu menjalankan operasi Data Manipulation Language (DML)
2. Mahasiswa mampu memahami kamus data dan menuangkannya menjadi tabel
3. Mahasiswa mampu menerapkan perintah INSERT, UPDATE, DELETE

PENDAHULUAN

Data Manipulation Language (DML) merupakan bahasa terstruktur yang digunakan pada SQL untuk memanipulasi record atau isi data dari sebuah tabel. DML hanya dapat digunakan setelah DDL terjadi. Diantara perintah yang paling umum digunakan adalah SELECT untuk menampilkan record, INSERT untuk menambahkan record baru, UPDATE untuk mengubah record, dan DELETE untuk menghapus record.

Sintaks penulisan untuk tiap perintah berbeda, dan perlu diperhatikan penggunaan huruf besar pada praktik ini hanya bertujuan untuk membedakan antara perintah SQL dengan penamaan dan nilai record. Perlu diperhatikan juga record dengan tipe data date, varchar, ataupun text, harus diapit dengan tanda petik (').

PERCOBAAN 2.1 Memasukkan Record dan Menampilkannya

1. Berdasarkan percobaan sesi 1, buat tabel `karyawan` pada basis data `penggajian` dengan struktur sebagai berikut

No	Field	Type	Null	Length	Keterangan
1	nik	Varchar	No	20	Primary Key
2	nama	Varchar	Yes	50	
3	tanggal_mulai	Date	Yes		
4	gaji_pokok	Int	Yes	11	
5	status_karyawan	Enum	Yes	'TETAP', 'KONTRAK', 'MAGANG'	

```
MariaDB [penggajian]> CREATE TABLE karyawan(
    -> nik VARCHAR(20) NOT NULL PRIMARY KEY,
    -> nama VARCHAR(50) NULL,
    -> tanggal_mulai DATE NULL,
    -> gaji_pokok INT(11) NULL,
    -> status_karyawan ENUM('TETAP', 'KONTRAK', 'MAGANG')
    -> );
Query OK, 0 rows affected (0.035 sec)
```

```
MariaDB [penggajian]> 
```

2. Tampilkan hasil create tabel tersebut

```
MariaDB [penggajian]> DESC karyawan;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nik   | varchar(20) | NO  | PRI | NULL | 
| nama  | varchar(50)  | YES |     | NULL | 
| tanggal_mulai | date | YES |     | NULL | 
| gaji_pokok | int(11) | YES |     | NULL | 
| status_karyawan | enum('TETAP', 'KONTRAK', 'MAGANG') | YES |     | NULL | 
+-----+-----+-----+-----+-----+
5 rows in set (0.007 sec)
```

3. Tampilkan isi record dari tabel `karyawan` dengan sintaks

```
SELECT * FROM nama_tabel;
```

```
MariaDB [penggajian]> SELECT * FROM karyawan;
Empty set (0.001 sec)
```

4. Respon dari MySQL menunjukkan bahwa record masih kosong karena memang tabelnya baru di-create.

5. Masukkan record dengan sintaks

```
INSERT INTO nama_tabel VALUES  
(nilai_field1, nilai_field2, ..., nilai_fieldN);
```

```
MariaDB [penggajian]> INSERT INTO karyawan VALUES  
    -> '0001',  
    -> 'WACHID',  
    -> '2011-01-05',  
    -> 3100000,  
    -> 'TETAP'  
    -> );  
Query OK, 1 row affected (0.002 sec)
```

6. Tampilkan isi record dari tabel `karyawan`

```
MariaDB [penggajian]> SELECT * FROM karyawan;  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |  
+-----+-----+-----+-----+-----+  
1 row in set (0.001 sec)
```

7. Masukkan record berikut menggunakan perintah yang serupa

nik	nama	tanggal_mulai	gaji_pokok	Status_karyawan
0002	DWI	2011-01-05	3100000	TETAP
0003	TRIO	2011-01-05	2900000	TETAP
0004	ARBA	2015-09-09	2400000	KONTRAK
0005	PANCA	2019-09-09	2200000	KONTRAK
0006	SITI	2021-09-16	1500000	MAGANG

```
MariaDB [penggajian]> SELECT * FROM karyawan;  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |  
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |  
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |  
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |  
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |  
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |  
+-----+-----+-----+-----+-----+  
6 rows in set (0.001 sec)
```

KESIMPULAN 2.1

1. Selain varchar dan text, MySQL mengenal tipe data int, date, enum dan banyak lagi.
2. Perintah untuk menampilkan seluruh field dari seluruh record pada sebuah tabel

```
SELECT * FROM nama_tabel
```

3. Perintah untuk memasukkan seluruh field dari sebuah record

```
INSERT INTO nama_tabel VALUES  
(nilai_field1, nilai_field2,..., nilai_fieldN);
```

4. Untuk memasukkan nilai dengan tipe data varchar, text, dan date harus diawali dan diakhiri tanda petik ('), sedangkan untuk int justru tidak boleh menggunakan tanda petik.

LATIHAN 2.1

1. Buat tangkapan layar dari hasil tiap percobaan dan latihan dan jadikan satu dalam dokumen Word/PDF Latihan 2
2. Jalankan perintah berikut, amati hasil dan jelaskan apa yang terjadi

```
INSERT INTO karyawan VALUES ('0007','SAPTO');
```

3. Jalankan perintah berikut, amati hasil dan jelaskan apa yang terjadi

```
INSERT INTO karyawan (nik,nama) VALUES ('0008','WINDU');
```

4. Jalankan perintah berikut, amati hasil dan jelaskan apa yang terjadi

```
INSERT INTO karyawan (nik,nama,status_karyawan) VALUES  
( '0009', 'NINO', 'TETAP') ,  
( '0010', 'TENTEN', 'KONTRAK') ,  
( '0011', 'EVAN', 'MAGANG') ;
```

5. Tambahkan data bebas minimal 5 (lima) record

6. Buat tangkapan layar dari hasil tiap latihan dan latihan dan jadikan satu file

PERCOBAAN 2.2 Mengubah Record

1. Update record dengan sintaks

```
UPDATE nama_tabel SET  
field1 = nilai_field1,  
field2 = nilai_field2,  
...  
fieldN = nilai_fieldN
```

2. Perlu diperhatikan bahwa sintaks tersebut perlu ditambahkan pemeriksaan kondisi dengan perintah WHERE yang menentukan record mana yang ingin di-update, jika tidak ada WHERE maka perintah tersebut akan merubah seluruh record.
3. Sintaks UPDATE memasangkan langsung antara nama field dengan nilai field-nya, berbeda dengan sintaks INSERT yang bisa langsung memasukkan nilai field tanpa menyebutkan nama fieldnya.
4. Jalankan perintah, amati hasilnya.

```
MariaDB [penggajian]> UPDATE karyawan SET  
-> status_karyawan = 'MAGANG'  
-> WHERE status_karyawan is NULL;
```

5. Jalankan perintah, amati hasilnya.

```
MariaDB [penggajian]> UPDATE karyawan SET  
-> tanggal_mulai = '2022-01-05',  
-> gaji_pokok = 1000000  
-> WHERE tanggal_mulai is NULL;
```

6. Jalankan perintah, amati hasilnya.

```
MariaDB [penggajian]> UPDATE karyawan SET  
-> tanggal_mulai = '2018-08-08',  
-> gaji_pokok = 1300000,  
-> status_karyawan = 'KONTRAK'  
-> WHERE nik = '0008';
```

KESIMPULAN 2.2

Mengubah record dengan sintaks

```
UPDATE nama_tabel SET  
field1 = nilai_field1,  
field2 = nilai_field2,  
...  
fieldN = nilai_fieldN
```

LATIHAN 2.2

1. Buat tangkapan layar dari hasil tiap percobaan dan latihan dan jadikan satu dalam dokumen Word/PDF Latihan 2
2. Dengan memperhatikan contoh lakukan perubahan data berikut
 - a. Update tanggal masuk menjadi 2019-05-05 untuk karyawan dengan nama Panca
 - b. Ubah gaji pokok dari karyawan dengan nik 0001 menjadi 3200000
 - c. Ubah seluruh karyawan dengan status KONTRAK menjadi TETAP dan yang memiliki status MAGANG menjadi KONTRAK
 - d. Naikkan gaji pokok semua karyawan yang memiliki gaji dibawah 2.000.000 menjadi 2.050.000

PERCOBAAN 2.3 Menghapus Record

1. Delete record dengan sintaks

```
DELETE FROM nama_tabel
```

2. Perlu diperhatikan bahwa sintaks tersebut sama seperti UPDATE dimana perlu ditambahkan pemeriksaan kondisi dengan perintah WHERE, jika tidak ada WHERE maka perintah tersebut akan menghapus seluruh record.

3. Tampilkan seluruh data, misal:

```
MariaDB [penggajian]> SELECT * FROM karyawan;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP
| 0010 | TENTEN | 2022-01-05 | 1000000 | KONTRAK
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG
+-----+-----+-----+-----+
10 rows in set (0.006 sec)
```

4. Hapus record karyawan dengan nik 0010

```
MariaDB [penggajian]> DELETE FROM karyawan WHERE nik='0010';
Query OK, 1 row affected (0.018 sec)
```

5. Tampilkan kembali seluruh data

```
MariaDB [penggajian]> SELECT * FROM karyawan;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG
+-----+-----+-----+-----+
9 rows in set (0.002 sec)
```

LATIHAN 2.3

1. Tampilkan seluruh data, kemudian hapus record karyawan dengan nik 0008, tampilkan kembali seluruh data.
2. Tampilkan seluruh data, kemudian hapus record karyawan dengan status KONTRAK, tampilkan kembali seluruh data.
3. Tampilkan seluruh data, kemudian hapus record karyawan dengan gaji pokok dibawah 2.500.000, tampilkan kembali seluruh data.

SESI 3

Seleksi Record

TUJUAN PRAKTEK

1. Mahasiswa mampu menampilkan seluruh record dari tabel
2. Mahasiswa mampu menyajikan record dengan kustomisasi menggunakan sintaks WHERE, BETWEEN, LIKE, ORDER BY, GROUP BY, SUM, dan COUNT

PENDAHULUAN

Seleksi data merupakan teknik utama dalam penyajian data menjadi informasi yang jauh lebih bermakna bagi pengguna. Teknik ini jika dikuasai dengan baik maka akan berhasil memunculkan keunggulan penyimpanan data kedalam basis data dibandingkan jika hanya disimpan ke dalam program pengolah angka semacam Microsoft Excel. Meskipun program pengolah angka lebih familiar bagi pengguna awam, penerapan basis data mampu menyajikan data lebih cepat terlebih jika datanya sudah mencapai ribuan yang mana akan susah dioperasikan dalam program pengolah angka.

Keunggulan basis data juga bisa ditunjukkan dalam hal penyajian datanya dalam banyak bahasa pemrograman, sehingga dengan kemampuan yang tepat, basis data bisa memberikan informasi yang menunjang penggunanya dalam mengambil keputusan.

PERCOBAAN 3.1 Menampilkan Record

1. Berikut sintaks untuk menampilkan seluruh field dari seluruh record

```
SELECT * FROM nama_tabel;
```

```
MariaDB [penggajian]> SELECT * FROM karyawan;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
+-----+-----+-----+-----+
9 rows in set (0.001 sec)
```

2. Jika ingin menampilkan field tertentu saja, maka ganti simbol bintang menjadi field yang ingin ditampilkan

```
SELECT field1,field3 FROM nama_tabel;
```

```
MariaDB [penggajian]> SELECT nama, status_karyawan FROM karyawan;
+-----+-----+
| nama | status_karyawan |
+-----+-----+
| WACHID | TETAP |
| DWI | TETAP |
| TRIO | TETAP |
| ARBA | KONTRAK |
| PANCA | KONTRAK |
| SITI | MAGANG |
| WINDU | KONTRAK |
| NINO | TETAP |
| EVAN | MAGANG |
+-----+-----+
9 rows in set (0.005 sec)
```

3. Sintaks ini jika lebih lengkap bisa ditulis dengan

```
SELECT nama_tabel.field1,nama_tabel.field3
FROM nama_tabel;
```

Sintaks ini biasa digunakan jika ingin memanggil beberapa tabel sekaligus yang akan dibahas lebih lanjut pada sesi 4

```
MariaDB [penggajian]> SELECT karyawan.nama, karyawan.nik FROM karyawan;
+-----+-----+
| nama | nik |
+-----+-----+
| WACHID | 0001 |
| DWI | 0002 |
| TRIO | 0003 |
| ARBA | 0004 |
| PANCA | 0005 |
| SITI | 0006 |
| WINDU | 0008 |
| NINO | 0009 |
| EVAN | 0011 |
+-----+-----+
9 rows in set (0.004 sec)
```

4. Sintaks ini jika lebih bisa juga ditulis dengan memberikan alias menggunakan kata kunci ‘AS’

```
SELECT alias_tabel.field1 AS nama_field1,
alias_tabel.field3 AS nama_field3
FROM nama_tabel AS alias_tabel;
```

```
MariaDB [penggajian]> SELECT K.nama AS nama_lengkap,
-> K.nik AS nomor_induk_karyawan
-> FROM karyawan AS K;
+-----+-----+
| nama_lengkap | nomor_induk_karyawan |
+-----+-----+
| WACHID | 0001 |
| DWI | 0002 |
| TRIO | 0003 |
| ARBA | 0004 |
| PANCA | 0005 |
| SITI | 0006 |
| WINDU | 0008 |
| NINO | 0009 |
| EVAN | 0011 |
+-----+-----+
9 rows in set (0.004 sec)
```

PERCOBAAN 3.2 Menampilkan Record dengan Kondisi Tertentu

- Menampilkan record dengan kondisi tertentu dengan menggunakan kata kunci ‘WHERE’

```
SELECT * FROM nama_tabel WHERE nama_field1 = kondisi
```

- Perintah berikut adalah menampilkan record karyawan hanya yang karyawan ‘TETAP’

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE status_karyawan = 'TETAP';
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
+-----+-----+-----+-----+
4 rows in set (0.003 sec)
```

- Perintah berikut menampilkan record karyawan yang gaji pokoknya lebih kecil dari atau sama dengan 1.500.000

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE gaji_pokok <= 1500000;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
+-----+-----+-----+-----+
4 rows in set (0.004 sec)
```

- Perintah berikut menampilkan record karyawan yang status karyawannya masih MAGANG dan gaji pokoknya lebih kecil dari atau sama dengan 1.500.000

```
MariaDB [penggajian]> SELECT * FROM karyawan
    -> WHERE status_karyawan='MAGANG'
    -> AND
    -> gaji_pokok <= 1500000;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
+-----+-----+-----+-----+
2 rows in set (0.008 sec)
```

PERCOBAAN 3.3 Menampilkan Record dengan Kondisi Between

- Menampilkan record dengan kondisi BETWEEN akan menampilkan record dengan kondisi range yang dituliskan sesudah BETWEEN dan dipisahkan dengan AND.

```
SELECT * FROM nama_tabel WHERE kondisi BETWEEN  
nilai_1 AND nilai_2
```

- Perintah berikut menampilkan record karyawan dengan gaji pokok antara 1.500.000 dan 3.000.000

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE gaji_pokok BETWEEN  
-> 1500000 AND 3000000;  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |  
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |  
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |  
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.007 sec)
```

- Perintah berikut menampilkan record karyawan dengan tanggal masuk antara tahun 2015 dan 2021

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE YEAR(tanggal_mulai)  
-> BETWEEN 2015 AND 2021;  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |  
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |  
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |  
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.004 sec)
```

PERCOBAAN 3.4 Menampilkan Record dengan Kondisi Like

1. Pencarian nilai dengan tipe data VARCHAR atau TEXT bisa menggunakan perintah LIKE yang berarti ‘menyerupai’. Berbeda dengan tanda sama dengan (=) perintah ini akan memuat record dengan kondisi yang ditentukan meskipun memiliki nilai tidak sama persis. Penggunaan kata kunci LIKE biasanya juga disertai dengan simbol WILDCARD (%).
2. Berikut adalah perintah yang digunakan untuk memanggil record dengan nama karyawan berawalan huruf ‘W’

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE nama LIKE "W%";  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |  
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.006 sec)
```

3. Berikut adalah perintah yang digunakan untuk memanggil record dengan nama karyawan berakhiran huruf ‘A’

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE nama LIKE "%A";  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |  
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.003 sec)
```

4. Berikut adalah perintah yang digunakan untuk memanggil record dengan nama karyawan yang mengandung kata ‘IN’

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE nama LIKE "%IN%";  
+-----+-----+-----+-----+-----+  
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |  
+-----+-----+-----+-----+-----+  
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |  
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.001 sec)
```

PERCOBAAN 3.5 Menampilkan Record dengan Urutan

- Menampilkan record dengan urutan tertentu bisa dilakukan dengan menggunakan kata kunci ORDER BY yang bisa diterjemahkan sebagai ‘urut berdasarkan’, sehingga sintaksnya kurang lebih sebagai berikut.

```
SELECT * FROM nama_tabel ORDER BY nama_field1
```

- Berikut perintah untuk menampilkan record karyawan yang diurutkan berdasarkan nama

```
MariaDB [penggajian]> SELECT * FROM karyawan ORDER BY nama;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
+-----+-----+-----+-----+
9 rows in set (0.008 sec)
```

- Jika tidak dituliskan secara spesifik maka field akan diurutkan dari kecil ke besar, jika ingin disebutkan secara spesifik bisa menggunakan kata kunci ASC sebagai singkatan dari ascending, sedangkan kebalikannya bisa menggunakan kata kunci DESC sebagai singkatan dari descending.

```
MariaDB [penggajian]> SELECT * FROM karyawan ORDER BY nama DESC;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
+-----+-----+-----+-----+
9 rows in set (0.010 sec)
```

4. Penggunaan WHERE bersamaan dengan ORDER BY harus mendahuluikan WHERE.

```
MariaDB [penggajian]> SELECT * FROM karyawan WHERE status_karyawan = 'TETAP'
-> ORDER BY gaji_pokok;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
+-----+-----+-----+-----+
4 rows in set (0.003 sec)
```

Contoh salah, jika dituliskan ORDER BY dulu dilanjutkan WHERE

```
MariaDB [penggajian]> SELECT * FROM karyawan ORDER BY gaji_pokok
-> WHERE status_karyawan = 'TETAP';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'WHERE status_karyawan = 'TETAP'' at line 2
```

5. Penggunaan beberapa pengurutan field untuk ORDER BY bisa menggunakan tanda koma.

```
MariaDB [penggajian]> SELECT * FROM karyawan ORDER BY nama, status_karyawan;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
+-----+-----+-----+-----+
9 rows in set (0.001 sec)
```

```
MariaDB [penggajian]> SELECT * FROM karyawan
-> ORDER BY status_karyawan, nama;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
+-----+-----+-----+-----+
9 rows in set (0.002 sec)
```

PERCOBAAN 3.6 Fungsi Statistik dan Pengelompokkan

1. Pada MySQL terdapat fungsi COUNT yang digunakan untuk menampilkan kuantitas dari record

```
MariaDB [penggajian]> SELECT COUNT(*) FROM karyawan;
+-----+
| COUNT(*) |
+-----+
|      9   |
+-----+
1 row in set (0.004 sec)
```

2. Fungsi ini bisa juga dikombinasikan dengan alias

```
MariaDB [penggajian]> SELECT COUNT(*) AS kuantitas FROM karyawan;
+-----+
| kuantitas |
+-----+
|      9   |
+-----+
1 row in set (0.002 sec)
```

3. Selain itu terdapat juga fungsi penjumlahan SUM untuk menghitung jumlah pada tipe data yang bisa dihitung penjumlahannya

```
MariaDB [penggajian]> SELECT SUM(gaji_pokok) as jumlah FROM karyawan;
+-----+
| jumlah   |
+-----+
| 18500000 |
+-----+
1 row in set (0.006 sec)
```

4. Bisa juga langsung dikombinasikan secara bersamaan

```
MariaDB [penggajian]> SELECT COUNT(*) as kuantitas,
    -> SUM(gaji_pokok) as jumlah
    -> FROM karyawan;
+-----+-----+
| kuantitas | jumlah   |
+-----+-----+
|      9   | 18500000 |
+-----+-----+
```

5. Fungsi tersebut sangat berguna ketika tabel sudah memiliki sangat banyak record dan tidak mudah untuk menghitungnya secara manual.

6. Terdapat juga fungsi MAX dan MIN untuk menampilkan nilai terkecil dan terbesar

```
MariaDB [penggajian]> SELECT COUNT(*) AS kuantitas,
-> SUM(gaji_pokok) AS jumlah,
-> MIN(gaji_pokok) AS terkecil,
-> MAX(gaji_pokok) AS terbesar
-> FROM karyawan;
+-----+-----+-----+
| kuantitas | jumlah | terkecil | terbesar |
+-----+-----+-----+
|         9 | 18500000 | 1000000 | 3100000 |
+-----+-----+-----+
1 row in set (0.005 sec)
```

7. Record dapat dikelompokkan dengan menggunakan perintah GROUP BY, berikut menampilkan data dengan dan tanpa GROUP BY

```
MariaDB [penggajian]> SELECT * FROM karyawan;
+-----+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG |
+-----+-----+-----+-----+
9 rows in set (0.003 sec)
```

```
MariaDB [penggajian]> SELECT * FROM karyawan GROUP BY status_karyawan;
+-----+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG |
+-----+-----+-----+-----+
3 rows in set (0.004 sec)
```

Record yang ditampilkan sudah terkelompok berdasarkan status karyawan, sehingga hanya menampilkan satu record per satu kelompok. Tampilan seperti ini mungkin bisa menyebabkan kesalahan karena seolah datanya lebih sedikit, karena pada prakteknya GROUP BY biasa digunakan bersamaan dengan fungsi statistik

8. GROUP BY bisa dikombinasikan bersama dengan fungsi statistik

```
MariaDB [penggajian]> SELECT status_karyawan,
-> COUNT(*) AS kuantitas
-> FROM karyawan
-> GROUP BY status_karyawan;
+-----+-----+
| status_karyawan | kuantitas |
+-----+-----+
| TETAP          |      4 |
| KONTRAK        |      3 |
| MAGANG         |      2 |
+-----+-----+
3 rows in set (0.004 sec)
```

LATIHAN 3

1. Tampilkan data karyawan pada field nama, status karyawan, dan gaji pokoknya saja

```
+-----+-----+-----+
| nama    | status_karyawan | gaji_pokok |
+-----+-----+-----+
| WACHID  | TETAP          | 3100000  |
| DWI     | TETAP          | 3100000  |
| TRIO    | TETAP          | 2900000  |
| ARBA    | KONTRAK        | 2400000  |
| PANCA   | KONTRAK        | 2200000  |
| SITI    | MAGANG         | 1500000  |
| WINDU   | KONTRAK        | 1300000  |
| NINO    | TETAP          | 1000000  |
| EVAN    | MAGANG         | 1000000  |
+-----+-----+-----+
```

2. Tampilkan semua field tabel karyawan yang mulai bekerja tahun 2011 saja

```
+-----+-----+-----+-----+-----+
| nik   | nama    | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+-----+
| 0001  | WACHID  | 2011-01-05   | 3100000  | TETAP          |
| 0002  | DWI     | 2011-01-05   | 3100000  | TETAP          |
| 0003  | TRIO    | 2011-01-05   | 2900000  | TETAP          |
+-----+-----+-----+-----+-----+
```

3. Tampilkan semua field tabel karyawan yang mulai bekerja antara 2011 dan 2015

```
+-----+-----+-----+-----+-----+
| nik   | nama    | tanggal_mulai | gaji_pokok | status_karyawan |
+-----+-----+-----+-----+-----+
| 0001  | WACHID  | 2011-01-05   | 3100000  | TETAP          |
| 0002  | DWI     | 2011-01-05   | 3100000  | TETAP          |
| 0003  | TRIO    | 2011-01-05   | 2900000  | TETAP          |
| 0004  | ARBA    | 2015-09-09   | 2400000  | KONTRAK        |
+-----+-----+-----+-----+-----+
```

4. Tampilkan nama karyawan yang mengandung huruf A

nama
WACHID
ARBA
PANCA
EVAN

5. Tampilkan semua field tabel karyawan yang mulai bekerja antara 2011 dan 2015
urutkan tanggalnya dari besar ke kecil dan namanya dari kecil ke besar

nik	nama	tanggal_mulai	gaji_pokok	status_karyawan
0004	ARBA	2015-09-09	2400000	KONTRAK
0002	DWI	2011-01-05	3100000	TETAP
0003	TRIO	2011-01-05	2900000	TETAP
0001	WACHID	2011-01-05	3100000	TETAP

6. Tampilkan kuantitas karyawan per tahun mulai

tahun_mulai	kuantitas
2011	3
2015	1
2018	1
2019	1
2021	1
2022	2

SESI 4

Relasi Tabel dan Seleksinya

TUJUAN PRAKTEK

1. Mahasiswa mampu memahami konsep basis data relasional
2. Mahasiswa mampu menyajikan data dari tabel yang terhubung

PENDAHULUAN

Basis data relasional merupakan penyusunan basis data yang menghubungkan antar beberapa tabel dan biasanya merupakan hasil normalisasi data. Bertujuan untuk memberikan konsistensi kepada data dan mengurangi redundansi pada basis data.

Pada praktek ini tidak membahas proses normalisasi melainkan langsung mencontohkan sebuah kasus sederhana penggajian yang menghubungkan 3 (tiga) tabel yang dianggap representatif dan mampu menunjukkan fitur utama dari basis data relasional

PERCOBAAN 4.1 Relasi Tabel Karyawan ke Tabel Bagian

1. Hubungan antar tabel bisa diekspresikan dengan menambahkan Primary Key (PK) dari tabel lain menjadi Foreign Key (FK) pada tabel yang akan dihubungkan.
2. Misalkan pada gambar berikut terdapat 2 (dua) tabel yaitu 'karyawan' dan 'bagian', dengan masing masing memiliki PK

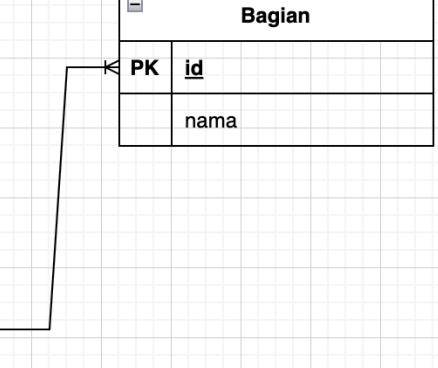
Karyawan	
PK	<u>nik</u>
	nama
	tanggal_mulai
	gaji_pokok
	status_karyawan

Bagian	
PK	<u>id</u>
	nama

3. Berikutnya tabel 'karyawan' dan 'bagian' akan dihubungkan dengan logika bahwa SATU karyawan hanya boleh memiliki SATU bagian sedangkan SATU bagian bisa memiliki BEBERAPA karyawan, maka hubungan ini disebut dengan relasi ONE TO MANY dan digambarkan sebagai berikut

Karyawan	
PK	<u>nik</u>
	nama
	tanggal_mulai
	gaji_pokok
	status_karyawan
FK	bagian_id

Bagian	
PK	<u>id</u>
	nama



4. Terlihat bahwa tabel 'karyawan' terhubung ke tabel 'bagian' melalui sebuah FK berupa bagian_id, karena dalam recordnya 1 (satu) karyawan akan memiliki 1 (satu) bagian_id. Sedangkan sebaliknya pada tabel 'bagian' tidak terdapat field baru karena 1 (satu) bagian bisa dimiliki oleh beberapa karyawan.

5. Buat sebuah tabel `bagian` dengan struktur sebagai berikut

```
MariaDB [penggajian]> DESC bagian;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+
| id   | int(11)   | NO  | PRI | NULL    | auto_increment |
| nama | varchar(255)| NO  |     | NULL    |                |
+-----+-----+-----+-----+-----+
2 rows in set (0.021 sec)
```

6. Isi record tabel `bagian` dengan nilai sebagai berikut

```
MariaDB [penggajian]> SELECT * FROM bagian;
+---+---+
| id | nama   |
+---+---+
| 1  | Marketing |
| 2  | HRD      |
| 3  | Manager  |
+---+---+
3 rows in set (0.001 sec)
```

7. Tambahkan field bagian_id pada tabel `karyawan` dengan perintah berikut

```
MariaDB [penggajian]> ALTER TABLE karyawan ADD bagian_id INT NULL DEFAULT 1;
Query OK, 0 rows affected (0.034 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [penggajian]> DESC karyawan;
+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+
| nik        | varchar(20)    | NO  | PRI | NULL    |                |
| nama       | varchar(50)    | YES  |     | NULL    |                |
| tanggal_mulai | date          | YES  |     | NULL    |                |
| gaji_pokok | int(11)        | YES  |     | NULL    |                |
| status_karyawan | enum('TETAP','KONTRAK','MAGANG') | YES  |     | NULL    |                |
| bagian_id  | int(11)        | YES  |     | 1       |                |
+-----+-----+-----+-----+-----+
6 rows in set (0.004 sec)
```

Perintah ini menambah sebuah field baru bernama bagian_id yang tipenya disamakan dengan tipe pada PK milik tabel Bagian yaitu INT. Nilai default-nya diatur menjadi 1 sehingga tidak terdapat data null pada tabel Karyawan. Meskipun demikian NULL bersifat YES karena bisa saja ketika karyawan baru masuk harus menunggu penempatan terlebih dahulu dan tidak langsung ditetapkan pada salah satu bagian.

8. Tampilkan semua field dan record dari tabel Karyawan

```
MariaDB [penggajian]> SELECT * FROM karyawan;
+-----+-----+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan | bagian_id |
+-----+-----+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP | 1 |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP | 1 |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP | 1 |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK | 1 |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK | 1 |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG | 1 |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK | 1 |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP | 1 |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG | 1 |
+-----+-----+-----+-----+-----+
```

9. Update sehingga karyawan dengan nik 0001 memiliki bagian_id = 3 dan nik 0002 memiliki nilai bagian_id = 2

```
+-----+-----+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok | status_karyawan | bagian_id |
+-----+-----+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 | TETAP | 3 |
| 0002 | DWI | 2011-01-05 | 3100000 | TETAP | 2 |
| 0003 | TRIO | 2011-01-05 | 2900000 | TETAP | 1 |
| 0004 | ARBA | 2015-09-09 | 2400000 | KONTRAK | 1 |
| 0005 | PANCA | 2019-09-09 | 2200000 | KONTRAK | 1 |
| 0006 | SITI | 2021-09-16 | 1500000 | MAGANG | 1 |
| 0008 | WINDU | 2018-08-08 | 1300000 | KONTRAK | 1 |
| 0009 | NINO | 2022-01-05 | 1000000 | TETAP | 1 |
| 0011 | EVAN | 2022-01-05 | 1000000 | MAGANG | 1 |
+-----+-----+-----+-----+-----+
```

10. Untuk mempersingkat tampilan maka hanya beberapa bagian field saja yang akan ditampilkan, kombinasikan perintah dengan menggunakan alias agar memudahkan untuk memanggil tabel bagian yang nantinya terhubung

```
MariaDB [penggajian]> SELECT K.nik, K.nama, K.bagian_id
-> FROM karyawan AS K;
+-----+-----+-----+
| nik | nama | bagian_id |
+-----+-----+-----+
| 0001 | WACHID | 3 |
| 0002 | DWI | 2 |
| 0003 | TRIO | 1 |
| 0004 | ARBA | 1 |
| 0005 | PANCA | 1 |
| 0006 | SITI | 1 |
| 0008 | WINDU | 1 |
| 0009 | NINO | 1 |
| 0011 | EVAN | 1 |
+-----+-----+-----+
```

11. Tambahkan field nama milik tabel bagian (B.nama), dan tambahkan perintah LEFT JOIN yang menunjukkan penyatuan kepada tabel bagian dengan kondisi dimana field bagian_id milik karyawan (K.bagian_id) sama dengan field id pada bagian (B.id)

```
MariaDB [penggajian]> SELECT K.nik, K.nama, K.bagian_id, B.nama
-> FROM karyawan AS K
-> LEFT JOIN bagian AS B
-> ON B.id = K.bagian_id;
+-----+-----+-----+-----+
| nik | nama | bagian_id | nama |
+-----+-----+-----+-----+
| 0001 | WACHID |         3 | Manager |
| 0002 | DWI    |         2 | HRD     |
| 0003 | TRIO   |         1 | Marketing |
| 0004 | ARBA   |         1 | Marketing |
| 0005 | PANCA  |         1 | Marketing |
| 0006 | SITI   |         1 | Marketing |
| 0008 | WINDU  |         1 | Marketing |
| 0009 | NINO   |         1 | Marketing |
| 0011 | EVAN   |         1 | Marketing |
+-----+-----+-----+-----+
```

12. Bisa ditambahkan alias nama_bagian agar kolom nama tidak ambigu antara nama karyawan dan nama bagian

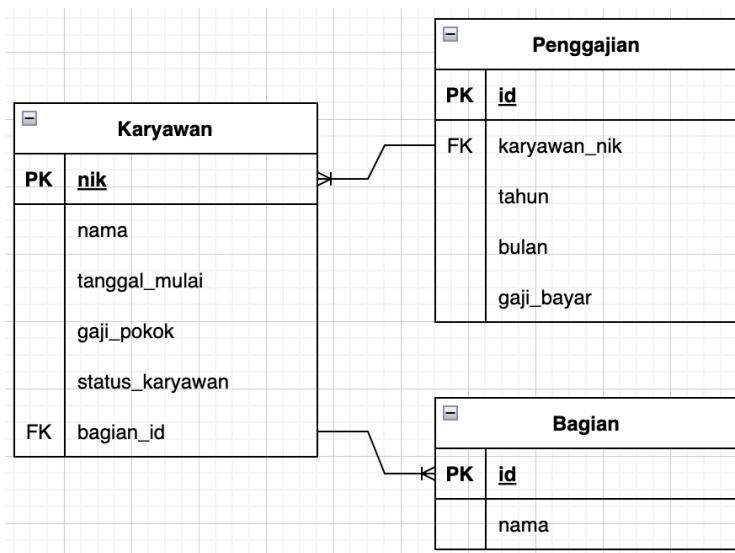
```
MariaDB [penggajian]> SELECT K.nik, K.nama, B.nama AS nama_bagian
-> FROM karyawan AS K
-> LEFT JOIN bagian AS B
-> ON B.id = K.bagian_id;
+-----+-----+-----+
| nik | nama | nama_bagian |
+-----+-----+-----+
| 0001 | WACHID | Manager |
| 0002 | DWI    | HRD     |
| 0003 | TRIO   | Marketing |
| 0004 | ARBA   | Marketing |
| 0005 | PANCA  | Marketing |
| 0006 | SITI   | Marketing |
| 0008 | WINDU  | Marketing |
| 0009 | NINO   | Marketing |
| 0011 | EVAN   | Marketing |
+-----+-----+-----+
```

13. Perintah JOIN juga bisa dikombinasikan dengan GROUP BY

```
MariaDB [penggajian]> SELECT B.nama AS nama_bagian, COUNT(*) AS kuantitas
-> FROM karyawan AS K
-> LEFT JOIN bagian AS B
-> ON B.id = K.bagian_id
-> GROUP BY B.nama;
+-----+-----+
| nama_bagian | kuantitas |
+-----+-----+
| HRD         |      1 |
| Manager     |      1 |
| Marketing   |      7 |
+-----+-----+
```

PERCOBAAN 4.2 Relasi Tabel Karyawan ke Tabel Penggajian

1. Perintah JOIN bisa memiliki kombinasi selain LEFT JOIN diantaranya ada RIGHT JOIN dan INNER JOIN, perintah ini berkaitan dengan nilai NULL pada FK atau terjadinya tidak ada pasangan yang bisa ada pada tabel, hal ini bisa dipelajari lebih lanjut pada link berikut
https://www.w3schools.com/mysql/mysql_join_inner.asp
2. Pada percobaan ini akan dibuat sebuah tabel `penggajian` yang memiliki relasi ONE TO MANY terhadap `karyawan` dimana satu record penggajian hanya berisikan satu karyawan sedangkan karyawan akan memiliki beberapa record penggajian.
3. Field yang dibuat yaitu id dengan INT AUTO_INCREMENT, karyawan_nik yang merupakan FK dan pastikan sama tipenya dengan nik pada `karyawan`, field tahun dan bulan yang menunjukkan tahun dan bulan pembayaran gaji, dan field gaji_bayar yang menunjukkan besaran gaji yang dibayarkan pada bulan dan tahun tersebut. Meskipun field gaji_pokok sudah dimiliki tabel `karyawan` namun besaran gaji bisa berubah ubah sesuai bulan dan tahunnya sehingga perlu dibuat field tersendiri.
4. Berikut diagram yang menunjukkan relasi antar tabel yang ada pada basis data `penggajian`



5. Buat sebuah tabel `penggajian` dengan struktur sebagai berikut

```
MariaDB [penggajian]> DESC penggajian;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| karyawan_nik | varchar(20) | YES | | NULL | |
| tahun | year(4) | YES | | NULL | |
| bulan | char(2) | YES | | NULL | |
| gaji_bayar | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.009 sec)
```

6. Dengan asumsi record karyawan seperti berikut

```
MariaDB [penggajian]> SELECT nik, nama, tanggal_mulai, gaji_pokok
-> FROM karyawan;
+-----+-----+-----+-----+
| nik | nama | tanggal_mulai | gaji_pokok |
+-----+-----+-----+-----+
| 0001 | WACHID | 2011-01-05 | 3100000 |
| 0002 | DWI | 2011-01-05 | 3100000 |
| 0003 | TRIO | 2011-01-05 | 2900000 |
| 0004 | ARBA | 2015-09-09 | 2400000 |
| 0005 | PANCA | 2019-09-09 | 2200000 |
| 0006 | SITI | 2021-09-16 | 1500000 |
| 0008 | WINDU | 2018-08-08 | 1300000 |
| 0009 | NINO | 2022-01-05 | 1000000 |
| 0011 | EVAN | 2022-01-05 | 1000000 |
+-----+-----+-----+-----+
```

Maka input record penggajian

```
MariaDB [penggajian]> SELECT * FROM penggajian;
+-----+-----+-----+-----+
| id | karyawan_nik | tahun | bulan | gaji_bayar |
+-----+-----+-----+-----+
| 1 | 0001 | 2015 | 07 | 1000000 |
| 2 | 0002 | 2015 | 07 | 1000000 |
| 3 | 0001 | 2015 | 08 | 1000000 |
| 4 | 0002 | 2015 | 08 | 1000000 |
| 5 | 0001 | 2015 | 09 | 1200000 |
| 6 | 0002 | 2015 | 09 | 1200000 |
| 7 | 0003 | 2015 | 09 | 1000000 |
+-----+-----+-----+-----+
```

Pada tabel ini dapat diceritakan bahwa pada bulan Juli dan Agustus 2015 masih hanya ada 2 (dua) karyawan yaitu karyawan dengan nik 0001 dan 0002. Masing-masing dibayar 1.000.000, meskipun jika dilihat pada tabel kondisi sekarang dimana gaji karyawan nik 0001 dan 0002 adalah 3.100.000. Kemudian pada bulan September 2015 gaji karyawan lama naik, dan diinputkan gaji karyawan baru sebesar 1.000.000

7. Dengan menggunakan teknik yang sama pada percobaan 4.1 bisa ditampilkan seperti berikut

nama	tahun	bulan	gaji_bayar
WACHID	2015	07	1000000
DWI	2015	07	1000000
WACHID	2015	08	1000000
DWI	2015	08	1000000
WACHID	2015	09	1200000
DWI	2015	09	1200000
TRIO	2015	09	1000000

8. Bisa juga dihitung jumlah bayar per tahunnya

tahun	bulan	jumlah_bayar
2015	07	2000000
2015	08	2000000
2015	09	3400000

LATIHAN 4

Sebagaimana Percobaan 4.2 poin ke-6, buat sendiri cerita berdasarkan data yang dimiliki dengan isian record data pengajian minimal sebanyak 20 record.

PENUTUP

Petunjuk praktikum ini disusun dengan tujuan mengenalkan sintaks dan fitur dasar dari basis data relasional dengan menggunakan MySQL. Selain yang sudah dipelajari masih ada Data Control Language (DCL), TRIGGER, VIEW, SUBQUERY dan lainnya. Pada kasus dunia nyata tentunya terdapat kasus dan sintaks yang jauh lebih rumit. Mahasiswa diharapkan siap mengadaptasi dan mengembangkan pengetahuannya tentang basis data melalui perkuliahan dan akan lebih baik lagi jika terjun pada kasus nyata.

DAFTAR PUSTAKA

Tim Penyusun. (2016, 10 8). *Modul Praktikum Basis Data MySQL*. Retrieved from EPrints

Mercu Buana Yogyakarta: <http://eprints.mercubuana-yogya.ac.id/24/1/MODUL%20PRAKTIKUM%20BASIS%20DATA.pdf>

Saputro, H. (2012). *Modul Pembelajaran Praktek Basis Data (MySQL)*. Retrieved from
Repositori Udinus: https://repository.dinus.ac.id/docs/ajar/materi_1.pdf

Tim Penyusun. (2019). *Modul Praktikum Basis Data*. Retrieved from https://stiki-indonesia.ac.id/wp-content/uploads/2019/03/modul/MODUL_SK/Modul%20Praktikum%20Basis%20Data.pdf