

**LAPORAN PRAKTIKUM BASIS
DATA DATA MANIPULATION
LANGUAGE**



DISUSUN OLEH :

Oktario Mufti Yudha

2320506044

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS TIDAR**

2024

LAPORAN
PRAKTIKUM BASIS DATA



Diisi Mahasiswa Praktikan								
Nama Praktikan	Oktario Mufti Yudha							
NPM	2320506044							
Rombel	04							
Judul Praktikum	Kelompok Perintah SQL							
Tanggal Praktikum	07 Maret 2024							
Diisi Asisten Praktikum								
Tanggal Pengumpulan								
Catatan								

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
	Imam Adi Nata, S.Kom., M.Kom.	

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR
Genap 2023/2024

I. Tujuan Praktikum

1. Mampu menjelaskan DDL dan DML
2. Mampu menggunakan perintah Data Manipulation Language (DDL) Dasar Teori

II. Dasar Teori

Data Manipulation Language (DML) adalah komponen kunci dari bahasa SQL (Structured Query Language) yang digunakan untuk memanipulasi data dalam basis data relasional. DML memberikan kemampuan kepada pengguna untuk melakukan operasi dasar seperti menyisipkan, memperbarui, menghapus, dan mengambil data dari tabel. Perintah INSERT memungkinkan pengguna untuk menambahkan baris baru ke dalam tabel dengan menentukan nilai untuk setiap kolom. Operasi ini berguna untuk memasukkan data baru ke dalam basis data. Sebagai contoh, perintah `INSERT INTO nama_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);` akan menyisipkan data baru ke dalam tabel sesuai dengan nilai yang ditentukan.

Selanjutnya, perintah UPDATE digunakan untuk memperbarui nilai dari satu atau beberapa kolom dalam satu atau beberapa baris dalam sebuah tabel. Dengan menggunakan perintah ini, pengguna dapat mengubah nilai yang sudah ada menjadi nilai baru sesuai kebutuhan. Misalnya, perintah `UPDATE nama_tabel SET kolom1 = nilai1 WHERE kondisi;` akan mengubah nilai kolom1 dalam tabel nama_tabel di mana kondisi yang ditentukan terpenuhi.

Perintah DELETE memungkinkan pengguna untuk menghapus satu atau beberapa baris dari sebuah tabel berdasarkan kondisi yang ditentukan. Dengan menggunakan DELETE, pengguna dapat menghapus data yang tidak diinginkan atau tidak relevan dari basis data. Sebagai contoh, perintah `DELETE FROM nama_tabel WHERE kondisi;` akan menghapus baris-baris dari tabel nama_tabel di mana kondisi yang ditentukan terpenuhi.

Meskipun perintah SELECT secara teknis bukan termasuk dalam DML, namun juga merupakan bagian penting dari manipulasi data dalam SQL. Perintah SELECT digunakan untuk mengambil atau menampilkan data dari satu atau beberapa tabel dalam basis data. Ini memungkinkan pengguna untuk menentukan kolom-kolom tertentu yang ingin diambil, serta menerapkan berbagai klausa seperti WHERE,

ORDER BY, dan GROUP BY untuk memfilter dan mengurutkan data sesuai kebutuhan.

Dengan demikian, DML merupakan bagian yang sangat penting dari SQL karena memberikan kemampuan kepada pengguna untuk melakukan operasi dasar yang diperlukan untuk memanipulasi data dalam sebuah basis data relasional. Dengan menggunakan perintah-perintah DML yang sesuai, pengguna dapat dengan mudah menyisipkan, memperbarui, menghapus, dan mengambil data, yang merupakan bagian penting dari manajemen data dalam sistem basis data.

III. Metode Praktikum

A. Alat dan bahan

Alat :

1. PC (Komputer)
2. Keyboard
3. Mouse

Bahan :

1. Operating System Linux
2. File Materi Praktikum
3. Terminal
4. mysql

B. Langkah kerja

3.2.1. Perintah Insert

Perintah SQL INSERT INTO digunakan untuk menambahkan atau menyisipkan data baru ke dalam sebuah tabel dalam sebuah basis data relasional. Ini adalah salah satu perintah yang paling sering digunakan dalam SQL dan memberikan kemampuan untuk menambahkan data ke dalam tabel secara programatik.

Perintah INSERT INTO memungkinkan Kita untuk secara dinamis menambahkan data baru ke dalam tabel sesuai dengan kebutuhan aplikasi atau skenario bisnis Kita. Ini adalah operasi fundamental dalam manajemen data dan digunakan dalam banyak aplikasi untuk memperbarui basis data dengan data baru. Penting untuk memastikan bahwa nilai-nilai yang dimasukkan sesuai dengan tipe data dan batasan yang diberlakukan pada kolom dalam tabel untuk menghindari kesalahan atau kegagalan dalam operasi penambahan data.

3.2.2. Perintah Select

Query SELECT merupakan salah satu perintah utama dalam bahasa SQL (Structured Query Language) yang digunakan untuk mengambil atau menampilkan data dari satu atau beberapa tabel dalam sebuah basis data. Ini adalah perintah yang paling sering digunakan dalam SQL dan memberikan kemampuan untuk mengambil informasi yang diperlukan dari basis data. Berikut adalah penjelasan lengkap tentang Query SELECT:

Penjelasan:

- ☛ SELECT: Ini adalah kata kunci yang menunjukkan bahwa Anda ingin mengambil data dari tabel tertentu. Anda dapat menentukan kolom-kolom tertentu yang ingin Anda ambil dengan menggunakan nama kolom, atau menggunakan wildcard (*) untuk mengambil semua kolom.
- ☛ kolom1, kolom2, ...: Ini adalah daftar kolom yang ingin Anda ambil dari tabel. Anda dapat menentukan satu atau beberapa kolom, atau menggunakan wildcard (*) untuk mengambil semua kolom.
- ☛ FROM: Ini adalah kata kunci yang menunjukkan dari mana Anda ingin mengambil data. Anda harus menentukan nama tabel dari mana Anda ingin mengambil data.
- ☛ nama_tabel: Ini adalah nama tabel dari mana Anda ingin mengambil data.
- ☛ WHERE: Ini adalah klausa opsional yang memungkinkan Anda untuk memfilter data berdasarkan kondisi tertentu. Hanya baris-baris yang memenuhi kondisi yang akan diambil.
- ☛ kondisi: Ini adalah ekspresi logis yang menentukan kriteria untuk memfilter baris yang akan diambil. Misalnya, WHERE kolom = nilai.
- ☛ ORDER BY: Ini adalah klausa opsional yang memungkinkan Anda untuk mengurutkan hasil berdasarkan kolom tertentu dalam urutan menaik atau menurun.

- ☛ kolom: Ini adalah nama kolom yang ingin Anda gunakan untuk mengurutkan hasil. Misalnya, ORDER BY kolom ASC untuk urutan menaik atau ORDER BY kolom DESC untuk urutan menurun.
- ☛ LIMIT: Ini adalah klausa opsional yang membatasi jumlah baris yang akan diambil dari hasil. Ini berguna terutama ketika Anda hanya ingin mengambil sejumlah kecil baris dari tabel yang besar.
- ☛ jumlah: Ini adalah jumlah maksimum baris yang akan diambil dari hasil. Misalnya, LIMIT 10 akan membatasi hasil menjadi 10 baris.

Data yang dikembalikan disimpan dalam tabel hasil, yang disebut record-set.

3.2.3. Perintah Update

Perintah SQL UPDATE adalah salah satu perintah yang penting dalam bahasa SQL yang digunakan untuk memperbarui data yang ada dalam tabel sebuah basis data relasional. Ini memungkinkan pengguna untuk mengubah nilai dari satu atau beberapa kolom dalam satu atau lebih baris sesuai dengan kriteria yang ditentukan. Perintah UPDATE memberikan fleksibilitas dalam mengelola data yang ada, baik untuk memperbaiki kesalahan, mengupdate informasi, atau mengubah status dari data dalam basis data.

Sintaksis dasar dari perintah UPDATE melibatkan dua kata kunci utama, yaitu UPDATE dan SET. Kata kunci UPDATE menunjukkan bahwa perintah yang akan dijalankan adalah untuk memperbarui data dalam tabel, sementara kata kunci SET digunakan untuk menetapkan nilai baru untuk kolom-kolom yang akan diperbarui. Klausa WHERE adalah opsional, namun sangat penting karena memungkinkan untuk membatasi baris-baris mana yang akan diperbarui berdasarkan kriteria tertentu.

Proses menggunakan perintah UPDATE dimulai dengan menentukan tabel yang akan diperbarui, diikuti dengan menetapkan nilai baru untuk kolom-kolom yang ingin diperbarui dengan menggunakan klausa SET. Kondisi yang akan membatasi baris-baris yang akan diperbarui kemudian ditentukan menggunakan klausa WHERE. Ini memungkinkan untuk memperbarui data secara selektif berdasarkan kriteria yang spesifik, seperti nama karyawan, nomor identitas, atau status tertentu.

Penggunaan perintah UPDATE harus dilakukan dengan hati-hati, terutama ketika menentukan klausa WHERE, karena kesalahan dalam penulisan kondisi dapat mengakibatkan perubahan yang tidak diinginkan pada data yang ada. Oleh karena itu, sebaiknya selalu melakukan uji coba terlebih dahulu pada data yang tidak kritis sebelum menjalankan perintah UPDATE pada data yang penting. Dengan pemahaman yang baik tentang sintaks dan fungsi perintah UPDATE, pengguna dapat memanfaatkannya secara efektif untuk memelihara dan memperbarui data dalam basis data mereka sesuai dengan kebutuhan.

3.2.4. Perintah Delete

Perintah DELETE pada MySQL digunakan untuk menghapus satu atau lebih baris data dari sebuah tabel dalam sebuah basis data relasional. Perintah DELETE merupakan salah satu perintah kritis dalam SQL yang memungkinkan pengguna untuk menghapus data yang tidak diinginkan atau tidak relevan dari basis data.

Sintaksis dasar dari perintah DELETE melibatkan kata kunci utama DELETE FROM, diikuti dengan nama tabel yang akan dihapus baris-datanya. Pengguna dapat juga menentukan kondisi dengan menggunakan klausa WHERE untuk membatasi baris-baris mana yang akan dihapus. Tanpa klausa WHERE, perintah DELETE akan menghapus semua baris dari tabel, sehingga perlu dilakukan dengan hati-hati.

Perintah DELETE secara umum digunakan dalam skenario ketika data yang ada sudah tidak lagi relevan atau dibutuhkan dalam basis data, atau ketika data tersebut tidak diinginkan lagi. Contohnya, jika ada entri ganda dalam sebuah tabel, perintah DELETE dapat digunakan untuk menghapus satu dari entri tersebut. Demikian juga, jika ada data yang sudah usang atau tidak relevan, perintah DELETE dapat membantu membersihkan tabel dari data tersebut.

Namun, perlu diingat bahwa perintah DELETE bersifat permanen, artinya data yang dihapus tidak dapat dikembalikan dengan mudah. Oleh karena itu, sebelum menjalankan perintah DELETE, sangat disarankan untuk membuat cadangan (backup) data yang akan dihapus terlebih dahulu. Hal ini akan membantu untuk menghindari kehilangan data yang tidak disengaja atau

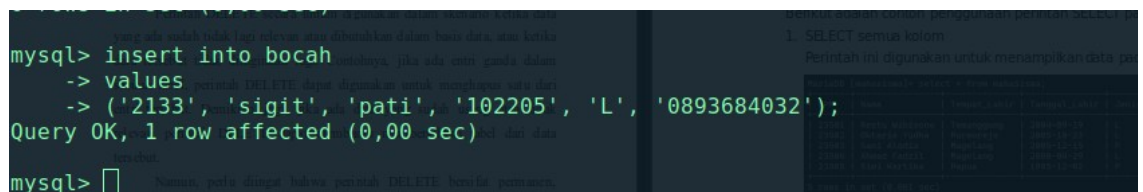
memungkinkan untuk memulihkan data jika terjadi kesalahan. Selain itu, penggunaan perintah DELETE juga harus dilakukan dengan hati-hati, terutama ketika menggunakan klausa WHERE. Kondisi yang tidak tepat dapat menyebabkan penghapusan data yang tidak diinginkan atau bahkan menghapus semua data dalam sebuah tabel. Oleh karena itu, selalu penting untuk memastikan bahwa klausa WHERE dibuat dengan cermat dan telah diuji sebelum menjalankan perintah DELETE dalam produksi.

Dengan memahami dengan baik fungsi, sintaks, dan potensi risiko dari perintah DELETE, pengguna dapat menggunakan alat ini secara efektif untuk membersihkan, memelihara, dan mengelola data dalam basis data MySQL mereka sesuai dengan kebutuhan aplikasi atau skenario bisnis.

IV. Hasil dan Analisis

4.1. Perintah Insert

Perintah untuk memasukkan sebagai berikut :



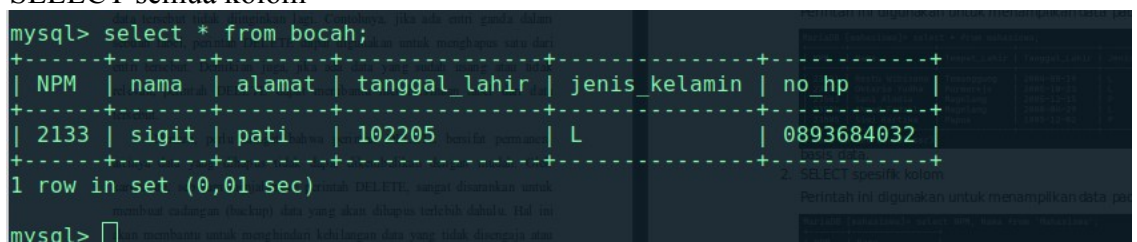
```
mysql> insert into bocah
-> values
-> ('2133', 'sigit', 'pati', '102205', 'L', '0893684032');
Query OK, 1 row affected (0,00 sec)
```

Perintah SQL untuk memasukkan data berhasil jika respon yang tampil setelah dieksekusi adalah “Query OK, 1 row affected”.

4.2. Perintah Select

Berikut adalah contoh penggunaan perintah SELECT pada tabel anggota :

1. SELECT semua kolom

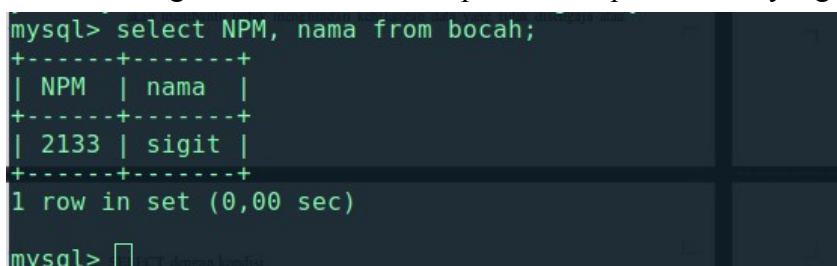


```
mysql> select * from bocah;
+-----+-----+-----+-----+-----+-----+
| NPM   | nama  | alamat | tanggal_lahir | jenis_kelamin | no_hp   |
+-----+-----+-----+-----+-----+-----+
| 2133  | sigit | pati   | 102205         | L              | 0893684032 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,01 sec)
```

Perintah ini digunakan untuk menampilkan data pada semua kolom dalam basis data.

2. SELECT spesifik kolom

Perintah ini digunakan untuk menampilkan data pada kolom yang spesifik.



```
mysql> select NPM, nama from bocah;
+-----+-----+
| NPM   | nama  |
+-----+-----+
| 2133  | sigit |
+-----+-----+
1 row in set (0,00 sec)
```


3. SELECT dengan kondisi

SELECT dengan kondisi digunakan untuk mencari data-data secara spesifik, misal kita ingin melihat data anggota yang memiliki jenis kelamin perempuan saja maka kita gunakan kolom sebagai kondisi pada perintah SELECT. Letak kondisi pada perintah SELECT berada pada bagian belakang perintah setelah nama tabel ditambahkan beberapa kalimat perintah dengan dimulai “WHERE” lalu dilanjutkan kondisi yang diinginkan. Penggunaan SELECT dengan kondisi dapat dilihat pada contoh berikut :

```
mysql> select NPM, nama from bocah where jenis_kelamin = 'L';
+-----+-----+
| NPM   | nama   |
+-----+-----+
| 2133  | Elsigit |
+-----+-----+
1 row in set (0,00 sec)
```

4.3. Perintah Update

Contoh perintah update mengubah nama pada mahasiswa dengan NPM = 2133 adalah sebagai berikut:

```
mysql> update bocah set nama = 'sigit rendang' where NPM = '2133';
Query OK, 1 row affected (0,03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select NPM, nama from bocah;
+-----+-----+
| NPM   | nama   |
+-----+-----+
| 2133  | sigit rendang |
+-----+-----+
1 row in set (0,00 sec)

mysql>
```

4.4. Perintah Delete

Berikut adalah contoh penggunaan perintah DELETE pada table mahasiswa. Kita akan menghapus data mahasiswa dengan NPM ‘2133’ sehingga perintah yang kita gunakan seperti dibawah:

```
mysql> delete from bocah where NPM = '2133';
Query OK, 1 row affected (0,04 sec)

mysql> select NPM, nama from bocah;
Empty set (0,00 sec)

mysql>
```

V. Soal

1. Membuat Tabel

```
mysql> create table Mahasiswa (  
-> NPM char(5) primary key,  
-> Nama varchar(25) not null,  
-> Tempat_Lahir varchar(25) not null,  
-> Tanggal_Lahir date not null,  
-> Jenis_Kelamin enum ('L', 'P') not null,  
-> No_Hp varchar(25) not null  
-> );  
Query OK, 0 rows affected (0,30 sec)  
  
mysql> desc Mahasiswa;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| NPM | char(5) | NO | PRI | NULL | |  
| Nama | varchar(25) | NO | | NULL | |  
| Tempat_Lahir | varchar(25) | NO | | NULL | |  
| Tanggal_Lahir | date | NO | | NULL | |  
| Jenis_Kelamin | enum('L', 'P') | NO | | NULL | |  
| No_Hp | varchar(25) | NO | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
6 rows in set (0,02 sec)  
  
mysql> 
```

(Gambar 5.1.1)

- 'create table' berfungsi untuk membuat sebuah tabel baru dalam database, yang diikuti dengan nama tabel yang ingin dibuat 'Mahasiswa', yang selanjutnya akan diikuti dengan definisi kolom-kolom yang ada dalam tabel tersebut.
- 'NPM char(5) primary key' akan mendefinisikan NPM (Nomor Pokok Mahasiswa) yang memiliki tipe data char dengan tepat 5 karakter, dan baris ini berfungsi untuk menjadi primary key dengan perintah dan ditambahkan pada akhir baris.
- 'Nama varchar(25) not null' akan mendefinisikan kolom Nama dalam tabel mahasiswa, yang memiliki tipe data varchar dengan maksimal 25 karakter, not null menandakan bahwa kolom harus diisi, sehingga setiap baris harus mengisi kolom Nama.
- 'Tempat_Lahir varchar(25) not null' akan mendefinisikan kolom Tempat Lahir dalam tabel mahasiswa, yang memiliki tipe data varchar dengan maksimal 25 karakter, not null menandakan bahwa kolom harus diisi, sehingga setiap baris harus mengisi kolom Nama.
- 'Tanggal_Lahir date not null' akan mendefinisikan kolom Tanggal Lahir dalam tabel mahasiswa, yang memiliki tipe data date untuk mendefinisikan tanggal, not null menandakan bahwa kolom harus diisi, sehingga setiap baris harus

mengisi kolom Nama.

- 'Jenis_Kelamin enum('L','P') not null' akan mendefinisikan kolom Jenis Kelamin dalam tabel mahasiswa, yang memiliki tipe data enum yang hanya bisa memiliki nilai 'L' atau 'P', not null menandakan bahwa kolom harus diisi, sehingga setiap baris harus mengisi kolom Nama.
- 'No_Hp varchar(25) not null' akan mendefinisikan kolom Nomor Handpone dalam tabel mahasiswa, yang memiliki tipe data varchar dengan maksimal 25 karakter, not null menandakan bahwa kolom harus diisi, sehingga setiap baris harus mengisi kolom Nama.

2. Membuat Perintah SQL DML untuk Mengisi Tabel

```
mysql> insert into Mahasiswa
-> values
-> ('23501', 'Restu Wibisono', 'Temanggung', '2004/09/19', 'L', '085156476828'),
-> ('23502', 'Oktario Yudha', 'Purworejo', '2005/10/23', 'L', '088347627267'),
-> ('23503', 'Sani Alodia', 'Magelang', '2005/12/15', 'P', '081238572834'),
-> ('23504', 'Ahmad Fadzil', 'Magelang', '2000/04/29', 'L', '088748392759'),
-> ('23505', 'Siwi Kartika', 'Papua', '1995/12/02', 'P', '08525132582');
Query OK, 5 rows affected, 5 warnings (0,03 sec)
Records: 5 Duplicates: 0 Warnings: 5

mysql> select * from mahasiswa;
ERROR 1146 (42S02): Table 'kontrasun.mahasiswa' doesn't exist
mysql> select * from Mahasiswa;
+-----+-----+-----+-----+-----+-----+
| NPM | Nama | Tempat_Lahir | Tanggal_Lahir | Jenis_Kelamin | No_Hp |
+-----+-----+-----+-----+-----+-----+
| 23501 | Restu Wibisono | Temanggung | 2004-09-19 | L | 085156476828 |
| 23502 | Oktario Yudha | Purworejo | 2005-10-23 | L | 088347627267 |
| 23503 | Sani Alodia | Magelang | 2005-12-15 | P | 081238572834 |
| 23504 | Ahmad Fadzil | Magelang | 2000-04-29 | L | 088748392759 |
| 23505 | Siwi Kartika | Papua | 1995-12-02 | P | 08525132582 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql>
```

(Gambar 5.2.1)

- 'insert into Mahasiswa' berfungsi untuk perintah utama dalam proses penambahan data kedalam tabel 'Mahasiswa'.
- 'Values' berfungsi untuk kata kunci yang digunakan untuk menentukan nilai yang dimasukkan ke dalam tabel. Nilai yang dimasukkan sesuai dengan urutan kolom-kolom tabel yang dibuat.
- Pada baris berikutnya berfungsi untuk memasukkan nilai-nilai kedalam tabel 'Mahasiswa'. Nilai-nilai yang masukkan sesuai dengan urutan kolom-kolom yang telah dibuat.

3. Membuat Perintah SQL DML untuk Mengubah nama kolom ke-3

```
mysql> update Mahasiswa set Nama = 'Sunny Alodia' where NPM = '23503';
Query OK, 1 row affected (0,02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select NPM, Nama from Mahasiswa;
+-----+-----+
| NPM   | Nama      |
+-----+-----+
| 23501 | Restu Wibisono |
| 23502 | Oktario Yudha  |
| 23503 | Sunny Alodia   |
| 23504 | Ahmad Fadzil   |
| 23505 | Siwi Kartika   |
+-----+-----+
5 rows in set (0,00 sec)

mysql>
```

(Gambar 5.3.1)

- 'update' berfungsi untuk memperbarui atau mengubah data yang sudah ada.
- 'Mahasiswa' adalah nama tabel yang akan diupdate.
- 'set `Nama` = 'Sunny Alodia'' adalah bagian dari perintah update yang menentukan kolom mana akan diperbarui dan nilai baru apa yang akan diberikan dalam kolom 'Nama'.
- 'where NPM = '23503'' adalah klausa yang digunakan untuk menentukan baris mana yang diperbarui. Hanya baris yang ditetapkan yang akan diubah.
- 'select NPM, Nama from `Mahasiswa`;' adalah perintah yang digunakan untuk menampilkan data dari tabel 'Mahasiswa' akan tetapi hanya untuk menampilkan kolom 'NPM' dan 'Nama'.

4. Membuat Perintah SQL DML untuk Menghapus data ke-4

```
mysql> delete from Mahasiswa where NPM = '23504';
Query OK, 1 row affected (0,03 sec)

mysql> select NPM, Nama from Mahasiswa;
+-----+-----+
| NPM   | Nama      |
+-----+-----+
| 23501 | Restu Wibisono |
| 23502 | Oktario Yudha  |
| 23503 | Sunny Alodia   |
| 23505 | Siwi Kartika   |
+-----+-----+
4 rows in set (0,00 sec)

mysql>
```

(Gambar 5.4.1)

- 'delete from `Mahasiswa` where `NPM` = '23504';' berfungsi untuk menghapus baris dari tabel 'Mahasiswa' dimana nilai kolom 'NPM' '23504'. Yang bisa diartikan baris dengan npm dengan nilai 23504 akan dihapus.
- 'select NPM, Nama from `Mahasiswa`;' adalah perintah yang digunakan untuk menampilkan data dari tabel 'Mahasiswa' akan tetapi hanya untuk menampilkan kolom 'NPM' dan 'Nama'.

5. Tampilkan Semua data dan kolom


```
mysql> select * from Mahasiswa;
+-----+-----+-----+-----+-----+-----+
| NPM   | Nama           | Tempat_Lahir | Tanggal_Lahir | Jenis_Kelamin | No_Hp       |
+-----+-----+-----+-----+-----+-----+
| 23501 | Restu Wibisono | Temanggung   | 2004-09-19    | L             | 085156476828 |
| 23502 | Oktario Yudha  | Purworejo    | 2005-10-23    | L             | 088347627267 |
| 23503 | Sunny Alodia   | Magelang     | 2005-12-15    | P             | 081238572834 |
| 23505 | Siwi Kartika   | Papua        | 1995-12-02    | P             | 08525132582  |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,00 sec)

mysql>
```

(Gambar 5.5.1)

- 'select' adalah kunci untuk memilih atau mengambil data dari tabel.
- '*' berfungsi untuk menandai memilih semua kolom dalam tabel. Dengan tanda ini perintah 'select' akan mengambil semua kolom yang ada dalam tabel.
- 'from mahasiswa' berfungsi untuk menentukan tabel mana yang akan dipakai untuk mengambil data.

6. Tampilkan data NPM, Nama dan No._Hp dengan jenis kelamin L

```
mysql> select NPM, Nama, No_Hp, Jenis_Kelamin from Mahasiswa where Jenis_Kelamin = 'L';
+-----+-----+-----+-----+
| NPM   | Nama           | No_Hp       | Jenis_Kelamin |
+-----+-----+-----+-----+
| 23501 | Restu Wibisono | 085156476828 | L             |
| 23502 | Oktario Yudha  | 088347627267 | L             |
+-----+-----+-----+-----+
2 rows in set (0,00 sec)

mysql>
```

(Gambar 5.6.1)

- 'select' adalah perintah untuk memilih kolom yang akan ditampilkan.
- 'NPM, Nama, No._Hp, Jenis_Kelamin' adalah daftar kolom yang akan ditampilkan dalam hasil.
- 'from Mahasiswa' adalah untuk menunjuk tabel mana yang akan ditampilkan, atau bisa juga untuk mengambil data dari mana yang akan ditampilkan.
- 'where Jenis_Kelamin' berfungsi untuk memilih hanya jenis kelamin 'L' saja yang akan ditampilkan.

VI. Kesimpulan

Kesimpulan dari praktikum yang telah dilakukan adalah dalam dasar dari pemrograman bahasa Python terdapat macam perintah seperti perintah 'print ()' yang digunakan untuk menampilkan teks sesuai isi dalam perintah 'print ()' tersebut kedalam output dari program yang telah dibuat.