

LAPORAN PRAKTIKUM
PRAKTIKUM 6
MANIPULASI TABEL



Disusun oleh:
Mohamad Solkhan Nawawi
24060123120020

PRAKTIKUM BASIS DATA
LAB E2

DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
2024

KATA PENGANTAR

Segala puji dan syukur kami panjatkan ke hadirat Allah SWT atas limpahan nikmat, rahmat, dan karunia-Nya yang senantiasa menyertai, sehingga Laporan Praktikum 6 Basis Data dengan materi “Manipulasi Tabel” ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bagian dari penugasan praktikum dalam mata kuliah Basis Data.

Pada laporan ini, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada Kak Syakira Nada Nirwana dan Kak Alya Safina atas bimbingan dan dukungannya dalam memahami materi dan langkah-langkah praktikum yang menjadi dasar penyusunan laporan ini.

Saya menyadari bahwa laporan ini masih memiliki kekurangan, baik dari segi isi maupun penyajiannya. Dengan segala keterbatasan, saya telah berupaya menyusun laporan ini sebaik mungkin berdasarkan kemampuan dan pengetahuan yang dimiliki. Oleh karena itu, saya sangat mengharapkan saran dan kritik yang membangun demi perbaikan dan kesempurnaan laporan ini di masa mendatang.

Semoga laporan ini dapat memberikan manfaat, tidak hanya bagi pembaca, tetapi juga menjadi pembelajaran berharga bagi saya sebagai penulis.

Semarang, 13 November 2024

Mohamad Solkhan Nawawi

BAB I

PENDAHULUAN

1.1. Rumusan Masalah

1.1.1. Soal

Adapun soal tercantum sebagai berikut :

1.1.1.1. Tugas 1

Melakukan proses pemasukan data kedalam tabel ruang dan tabel barang sesuai dengan data berikut :

```
mysql> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	40
A102	A	1	Ruang Tamu	18	6
A201	A	2	Lab Komputer	72	50
B101	B	1	Ruang Kuliah	60	40
B102	B	1	Ruang Kuliah	60	45
B201	B	2	Ruang Kuliah	70	50
B202	B	2	Ruang Kuliah	75	60
B203	B	2	Lab Komputer	50	40
B303	B	3	Ruang Kuliah	72	50

9 rows in set (0.00 sec)

Dan isi tabel **barang** yang sudah di-update adalah sebagaimana tampilan berikut ini:

```
mysql> select * from barang;
```

nobarang	namabarang	sumberdana	thnproduksi	satuan
108001	Meja Kuliah	PNBP	2008	Unit
109001	Meja Komputer	PNBP	2009	Unit
110001	Meja Sidang	PNBP	2010	Unit
111001	Meja Tamu	PNBP	2011	Unit
208001	Kursi Kuliah	PNBP	2008	Unit
208002	Kursi Kuliah	BMOM	2008	Unit
209001	Kursi Lipat	PNBP	2009	Unit
210001	Kursi Sidang	PNBP	2010	Unit
210002	Kursi Kuliah	Hibah	2010	Unit
211001	Kursi Tamu	BMOM	2011	Unit
211002	Kursi Kuliah	PNBP	2011	Unit
308001	LCD	PNBP	2008	Unit
309001	LCD	Hibah	2009	Unit
409001	Layar Peraga	BMOM	2009	Unit
409002	Layar Peraga	Hibah	2009	Unit
409003	Whiteboard	PNBP	2009	Unit
509001	Computer Desktop	PNBP	2009	Unit
509002	Printer Laser	PNBP	2009	Unit

18 rows in set (0.00 sec)

1.1.1.2. Tugas 2

Tugas

Dengan menggunakan tabel **ruang** dan tabel **barang** yang terakhir (setelah di-update), susun perintah untuk menghasilkan tampilan-tampilan sebagai berikut.

- Menampilkan data dari tabel **ruang** untuk ruang-ruang selain "**Ruang Seminar**" dan selain "**Ruang Tamu**" dengan luas tidak kurang dari 50.
- Menampilkan nomor ruang, fungsi, luas, dan kapasitas untuk "**Ruang Kuliah**" yang mempunyai **luas** paling sedikit 1,5 dari **kapasitas**.
- Menampilkan data dari tabel **barang** yang menggunakan sumber dana "**Hibah**" atau "**BMOM**" akan tetapi yang bukan berupa "**Kursi Kuliah**".
- Menampilkan nomor barang, nama barang, sumber dana, dan tahun produksi dengan sumber dana "**PNBP**" dan dengan tahun produksi bukan "**2009**".
- Menampilkan data dari tabel **barang** untuk semua barang dengan tahun produksi "**2008**", dan bukan berupa "**Kursi Kuliah**" dari sumber dana "**PNBP**".
- Menampilkan data dari tabel **ruang** diurutkan dari **luas** yang terbesar ke yang terkecil, dan untuk ruang dengan **luas** sama besar, tampilkan terlebih dahulu yang mempunyai **kapasitas** lebih besar.
- Menampilkan data dari tabel **barang** diurutkan menurut tahun produksi yang paling baru ke yang sebelumnya, dan jika tahun produksinya sama tampilkan terlebih dahulu menurut urutan abjad dari sumber dana.

1.2. Tujuan

Adapun tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Mampu mengisi data ke dalam tabel dan menampilkan hasilnya.
2. Mampu menampilkan data berdasarkan kriteria tertentu.
3. Mampu melakukan import data dari sistem yang berbeda.
4. Mampu melakukan pemeliharaan data dan perbandingan data.
5. Mampu melakukan pengurutan data berdasarkan kriteria tertentu.

BAB II

LANDASAN TEORI

2.1. Pengisian dan Penampilan Data Database

Pada sistem basis data, tabel adalah struktur penyimpanan utama yang terdiri dari kolom (fields) dan baris (records). Untuk menambahkan data ke dalam tabel, digunakan perintah `INSERT` dalam SQL. Data yang telah diisikan ke tabel kemudian dapat ditampilkan menggunakan perintah `SELECT`. Pengisian dan penampilan data merupakan tahap awal dalam manipulasi basis data, di mana tabel digunakan untuk menyimpan informasi yang kemudian dapat diambil dan dianalisis. Kemampuan untuk mengisikan dan menampilkan data ini menjadi dasar penting dalam pengelolaan basis data relasional karena memungkinkan pengguna untuk mengelola dan memeriksa informasi yang tersimpan secara sistematis.

2.2. Menampilkan Data Berdasarkan Kriteria Tertentu

Klausa `WHERE` dalam SQL digunakan untuk melakukan penyaringan data berdasarkan kondisi tertentu. Misalnya, pengguna dapat menampilkan data siswa yang memiliki usia lebih dari 18 tahun dengan menambahkan kondisi dalam query menggunakan klausa `WHERE`. Penggunaan operator seperti `=`, `<`, `>`, `<>` dan lainnya, yang memudahkan proses penyaringan data sesuai kebutuhan analisis. Dengan menampilkan data berdasarkan kriteria yang telah ditetapkan, pengguna dapat memperoleh subset data yang lebih spesifik, yang berguna dalam pengambilan keputusan berbasis data.

2.3. Pengurutan Data Berdasarkan Kriteria Tertentu

Pengurutan data adalah proses menampilkan data dalam urutan yang lebih terstruktur dan terorganisir, baik secara ascending (ASC) maupun descending (DESC). Dalam SQL, klausa `ORDER BY` digunakan untuk mengurutkan hasil query berdasarkan satu atau beberapa kolom, misalnya mengurutkan daftar nama siswa berdasarkan abjad atau mengurutkan data penjualan berdasarkan tanggal transaksi. Pengurutan data memudahkan pengguna dalam menganalisis dan meninjau informasi yang tersaji, sehingga informasi yang relevan lebih mudah ditemukan dan dipahami.

BAB III

PEMBAHASAN

3.1. Soal

Adapun soal tercantum sebagai berikut :

3.1.1. Tugas 1

Melakukan proses pemasukan data kedalam tabel ruang dan tabel barang sesuai dengan data berikut :

```
mysql> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	40
A102	A	1	Ruang Tamu	18	6
A201	A	2	Lab Komputer	72	50
B101	B	1	Ruang Kuliah	60	40
B102	B	1	Ruang Kuliah	60	45
B201	B	2	Ruang Kuliah	70	50
B202	B	2	Ruang Kuliah	75	60
B203	B	2	Lab Komputer	50	40
B303	B	3	Ruang Kuliah	72	50

9 rows in set (0.00 sec)

Dan isi tabel **barang** yang sudah di-update adalah sebagaimana tampilan berikut ini:

```
mysql> select * from barang;
```

nobarang	namabarang	sumberdana	thnproduksi	satuan
108001	Meja Kuliah	PNBP	2008	Unit
109001	Meja Komputer	PNBP	2009	Unit
110001	Meja Sidang	PNBP	2010	Unit
111001	Meja Tamu	BMOM	2011	Unit
208001	Kursi Kuliah	PNBP	2008	Unit
208002	Kursi Kuliah	BMOM	2008	Unit
209001	Kursi Lipat	PNBP	2009	Unit
210001	Kursi Sidang	PNBP	2010	Unit
210002	Kursi Kuliah	Hibah	2010	Unit
211001	Kursi Tamu	BMOM	2011	Unit
211002	Kursi Kuliah	PNBP	2011	Unit
308001	LCD	PNBP	2008	Unit
309001	LCD	Hibah	2009	Unit
409001	Layar Peraga	BMOM	2009	Unit
409002	Layar Peraga	Hibah	2009	Unit
409003	Whiteboard	PNBP	2009	Unit
509001	Komputer Desktop	PNBP	2009	Unit
509002	Printer Laser	PNBP	2009	Unit

18 rows in set (0.00 sec)

3.1.2. Tugas 2

Tugas

Dengan menggunakan tabel **ruang** dan tabel **barang** yang terakhir (setelah di-update), susun perintah untuk menghasilkan tampilan-tampilan sebagai berikut.

- Menampilkan data dari tabel **ruang** untuk ruang-ruang selain "**Ruang Seminar**" dan selain "**Ruang Tamu**" dengan luas tidak kurang dari 50.
- Menampilkan nomor ruang, fungsi, luas, dan kapasitas untuk "**Ruang Kuliah**" yang mempunyai **luas** paling sedikit 1,5 dari **kapasitas**.
- Menampilkan data dari tabel **barang** yang menggunakan sumber dana "**Hibah**" atau "**BMOM**" akan tetapi yang bukan berupa "**Kursi Kuliah**".
- Menampilkan nomor barang, nama barang, sumber dana, dan tahun produksi dengan sumber dana "**PNBP**" dan dengan tahun produksi bukan "**2009**".
- Menampilkan data dari tabel **barang** untuk semua barang dengan tahun produksi "**2008**", dan bukan berupa "**Kursi Kuliah**" dari sumber dana "**PNBP**".
- Menampilkan data dari tabel **ruang** diurutkan dari **luas** yang terbesar ke yang terkecil, dan untuk ruang dengan **luas** sama besar, tampilkan terlebih dahulu yang mempunyai **kapasitas** lebih besar.
- Menampilkan data dari tabel **barang** diurutkan menurut tahun produksi yang paling baru ke yang sebelumnya, dan jika tahun produksinya sama tampilkan terlebih dahulu menurut urutan abjad dari sumber dana.

3.2. Screenshot Input dan Output Program Beserta Penjelasan

3.2.1. Tugas 1

3.2.1.1. Manipulasi Tabel Ruang

Pada Tugas 1, kita akan terlebih dahulu mengecek apakah kita sudah memiliki tabel ruang dan barang pada *database* inventory. Untuk itu, kita akan menggunakan query sebagai berikut :

```
24060123120020> show databases;
+-----+
| Database |
+-----+
| information_schema |
| inventory          |
| mysql              |
| organisasi         |
| performance_schema |
| sakila              |
| sys                 |
| world               |
+-----+
8 rows in set (0.09 sec)
```

Melalui perintah `show databases` kita dapat mengetahui *database* apa saja yang kita miliki, kemudian kita akan mengaktifkan *database* inventory melalui query berikut :

```
24060123120020> use inventory;
Database changed
```

Setelah *database* inventory berhasil diaktifkan, kita dapat menampilkan tabel yang ada di dalam *database* inventory dengan menggunakan query sebagai berikut :

```
24060123120020> show tables;
+-----+
| Tables_in_inventory |
+-----+
| ruang                |
+-----+
1 row in set (0.03 sec)
```

Melalui perintah `show tables`, kita dapat mengetahui bahwa kita sudah memiliki tabel ruang didalamnya, untuk mengetahui struktur tabel ruang, kita bisa menggunakan query berikut :

```
24060123120020> describe ruang;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| noruang    | char(4)       | NO   | PRI | NULL    |       |
| blokgedung | char(1)       | NO   |     | NULL    |       |
| lantai     | char(1)       | NO   |     | NULL    |       |
| fungsi     | varchar(30)   | YES  |     | NULL    |       |
| luas       | int unsigned  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Melalui query describe ruang, kita dapat mengetahui bahwa dalam tabel ruang terdapat beberapa kolom yaitu noruang dengan tipe data char, blokgedung dengan tipe data char, lantai dengan tipe data char, fungsi dengan tipe data varchar, dan luas dengan tipe data integer. Setelah mengetahui struktur dari tabel ruang, selanjutnya kita dapat mengecek apakah dalam tabel ruang sudah terdapat isi data, berikut adalah query yang dapat digunakan :

```
24060123120020> select * from ruang;  
Empty set (0.00 sec)
```

Melalui perintah select * from ruang, kita dapat menampilkan isi data dari tabel ruang. Pada tabel ruang, diperoleh hasil Empty set, yang berarti belum ada baris data didalam tabel ruang tersebut. Untuk melakukan pemasukan data kita memiliki beberapa cara yang pertama yaitu dengan menggunakan query :

```
24060123120020> insert into ruang  
-> set  
-> noruang = "A101",  
-> blokgedung = "A",  
-> lantai = 1,  
-> fungsi = "Ruang Seminar",  
-> luas = 60;  
Query OK, 1 row affected (0.01 sec)
```

Melalui perintah insert into ruang, kita melakukan pemasukan data kedalam tabel ruang. Melalui, cara pemasukan data dengan query ini kita perlu menyebutkan nama kolom lalu dilakukan pemasukan data yang diinginkan dengan menggunakan format penulisan nama_kolom = nilai data. Untuk menampilkan, hasil pemasukan data yang kita buat, dapat kita tampilkan melalui query berikut :

```
24060123120020> select * from ruang;  
+-----+-----+-----+-----+-----+-----+  
| noruang | blokgedung | lantai | fungsi      | luas | kapasitas |  
+-----+-----+-----+-----+-----+-----+  
| A101    | A          | 1      | Ruang Seminar | 60    | 0          |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

Melalui perintah select * from ruang, kita dapat menampilkan baris data yang berada didalam tabel ruang. Untuk melakukan pemasukan data, terdapat cara lain yaitu melalui query berikut :

```
24060123120020> insert into ruang  
-> (noruang, blokgedung, lantai, fungsi, luas)  
-> values  
-> ("A102", "A", 1, "Ruang Tamu", 18);  
Query OK, 1 row affected (0.01 sec)
```


Dengan cara penulisan query ini, kita menentukan terlebih dahulu kolom yang akan kita lakukan pemasukan data, lalu diikuti dengan values lalu kita sebutkan nilai data yang ingin kita masukkan, berurutan sesuai dengan urutan kolom yang kita tulis sebelumnya. Berikut adalah hasil dari pemasukan data dari query diatas :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	0
A102	A	1	Ruang Tamu	18	0

```
2 rows in set (0.00 sec)
```

Lalu, terdapat cara ketiga untuk melakukan pemasukan data yaitu melalui query berikut :

```
24060123120020> insert into ruang
-> values ("A201", "A", 2, "Lab Komputer", 72);
Query OK, 1 row affected (0.01 sec)
```

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas
A101	A	1	Ruang Seminar	60
A102	A	1	Ruang Tamu	18
A201	A	2	Lab Komputer	72

```
3 rows in set (0.00 sec)
```

Dengan menggunakan cara penulisan query cara ketiga ini, kita tidak perlu untuk menuliskan nama kolom dalam query, kita hanya perlu menuliskan nilai data yang akan kita masukkan. Namun dengan penulisan ini, kita harus menuliskan nilai data secara berurutan sesuai dengan struktur tabel, sehingga kita perlu mengetahui struktur tabelnya terlebih dahulu. Keunggulan dari cara ketiga ini, kita bisa melakukan pemasukan data lebih dari satu sekaligus. Berikut adalah query untuk memasukkan data lebih dari satu :

```
24060123120020> insert into ruang
-> values
-> ("B101", "B", 1, "Ruang Kuliah", 60),
-> ("B303", "B", 3, "Ruang Kuliah", 72);
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Melalui query ini, kita juga harus melakukan pemasukan data secara berurutan sesuai dengan urutan kolom dalam struktur tabel. Untuk menambahkan lebih dari satu, kita hanya perlu menekan enter, dan memasukkan data sesuai dengan urutan yang sama. Berikut adalah query untuk menampilkan hasil dari pemasukan data yang kita lakukan dengan cara ketiga :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas
A101	A	1	Ruang Seminar	60
A102	A	1	Ruang Tamu	18
A201	A	2	Lab Komputer	72
B101	B	1	Ruang Kuliah	60
B303	B	3	Ruang Kuliah	72

```
5 rows in set (0.00 sec)
```

Dalam struktur tabel ruang, kita belum memiliki kolom kapasitas, sesuai dengan perintah soal, kita perlu menambahkan satu buah kolom baru dengan nama kapasitas, berikut adalah query yang kita gunakan :

```
24060123120020> alter table ruang
-> add kapasitas int not null default 0;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Melalui query alter table ruang, kita menambahkan kolom dengan perintah add, lalu dilanjutkan dengan nama kolom yang kita inginkan, kemudian tipe data dari kolom tersebut, serta optionalitas null atau not null, kita juga dapat menambahkan default sebagai nilai yang akan otomatis ditambahkan ketika kolom berhasil di buat. Berikut adalah query untuk menampilkan hasil dari penambahan kolom kapasitas :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	0
A102	A	1	Ruang Tamu	18	0
A201	A	2	Lab Komputer	72	0
B101	B	1	Ruang Kuliah	60	0
B303	B	3	Ruang Kuliah	72	0

```
5 rows in set (0.00 sec)
```

Setelah kolom kapasitas berhasil kita buat, kita dapat mengisi nilai dari kapasitas sehingga sesuai dengan soal. Untuk mengisi nilai dari kapasitas, berikut adalah query yang diperlukan :

```
24060123120020> update ruang
-> set kapasitas = 40
-> where noruang = "A101";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Melalui perintah update ruang, kita dapat memperbarui isi dari kapasitas dengan perintah set kapasitas lalu di isi dengan nilai baru yang sesuai dengan perintah where yang menjadi syarat baris mana yang ingin dilakukan perubahan data. Berikut adalah query untuk menampilkan hasil perubahan dari perintah update yang kita lakukan :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	40
A102	A	1	Ruang Tamu	18	0
A201	A	2	Lab Komputer	72	0
B101	B	1	Ruang Kuliah	60	0
B303	B	3	Ruang Kuliah	72	0

```
5 rows in set (0.00 sec)
```

Melalui perintah di atas, tabel pada kolom kapasitas dengan noruang A101 berubah menjadi 40 sesuai dengan query yang kita masukkan. Kita dapat mengulangi langkah-langkah ini secara berulang hingga didapatkan hasil berikut :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	40
A102	A	1	Ruang Tamu	18	6
A201	A	2	Lab Komputer	72	50
B101	B	1	Ruang Kuliah	60	40
B303	B	3	Ruang Kuliah	72	50

```
5 rows in set (0.00 sec)
```

Untuk menghasilkan tabel dengan isi data sesuai soal pada Tugas 1, kita dapat melakukan pemasukan data dengan menggunakan query sebagai berikut :

```
24060123120020> insert into ruang
-> values
-> ("B102", "B", 1, "Ruang Kuliah", 60, 45),
-> ("B201", "B", 2, "Ruang Kuliah", 70, 50),
-> ("B202", "B", 2, "Ruang Kuliah", 75, 60),
-> ("B203", "B", 2, "Lab Komputer", 50, 40);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Melalui perintah insert into ruang ini, kita menambahkan beberapa data sekaligus. Data yang dimasukkan ini sesuai dengan urutan kolom pada struktur tabel. Sehingga, setelah dilakukan pemasukan data dapat ditampilkan semua data dalam tabel ruang. Berikut adalah query untuk menampilkan isi dari tabel ruang :

```
24060123120020> select * from ruang;
```

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A101	A	1	Ruang Seminar	60	40
A102	A	1	Ruang Tamu	18	6
A201	A	2	Lab Komputer	72	50
B101	B	1	Ruang Kuliah	60	40
B102	B	1	Ruang Kuliah	60	45
B201	B	2	Ruang Kuliah	70	50
B202	B	2	Ruang Kuliah	75	60
B203	B	2	Lab Komputer	50	40
B303	B	3	Ruang Kuliah	72	50

```
9 rows in set (0.00 sec)
```

Melalui perintah `select * from ruang`, kita menampilkan semua kolom dari tabel ruang beserta dengan isi semua baris data yang sudah sesuai dengan permintaan pada soal.

3.2.1.2. Manipulasi Tabel Barang

Berikutnya, kita akan membuat tabel barang, yang sesuai dengan permintaan pada soal Tugas 1. Berikut adalah query untuk membuat tabel barang melalui query berikut :

```
24060123120020> create table barang (
-> nobarang char(6) not null primary key,
-> namabarang varchar(30),
-> sumberdana varchar(20),
-> thnproduksi int,
-> satuan varchar(20)
-> );
Query OK, 0 rows affected (0.03 sec)
```

Melalui perintah `create table barang`, kita membuat tabel baru bernama barang, lalu didalam terdapat kolom nobarang sebagai primary key, namabarang, sumberdana, thnproduksi, dan satuan beserta dengan atribut yang sesuai dengan permintaan soal. Berikut adalah query untuk menampilkan struktur dari tabel barang yang sudah kita buat :

```
24060123120020> describe barang;
```

Field	Type	Null	Key	Default	Extra
nobarang	char(6)	NO	PRI	NULL	
namabarang	varchar(30)	YES		NULL	
sumberdana	varchar(20)	YES		NULL	
thnproduksi	int	YES		NULL	
satuan	varchar(20)	YES		NULL	

```
5 rows in set (0.00 sec)
```

Dengan perintah `describe barang`, kita dapat mengetahui struktur dari tabel barang. Berikutnya setelah tabel barang berhasil dibuat, kita akan melakukan pemasukan data melalui perintah berikut :

```

24060123120020> insert into barang
-> values
-> ("108001", "Meja Kuliah", "PNBP", 2008, "Unit"),
-> ("109001", "Meja Komputer", "PNBP", 2009, "Unit"),
-> ("110001", "Meja Sidang", "PNBP", 2010, "Unit"),
-> ("111001", "Meja Tamu", "BMOM", 2011, "Unit"),
-> ("208001", "Kursi Kuliah", "PNBP", 2008, "Unit"),
-> ("208002", "Kursi Kuliah", "BMOM", 2008, "Unit"),
-> ("209001", "Kursi Lipat", "PNBP", 2009, "Unit"),
-> ("210001", "Kursi Sidang", "PNBP", 2010, "Unit"),
-> ("210002", "Kursi Kuliah", "Hibah", 2010, "Unit"),
-> ("211001", "Kursi Tamu", "BMOM", 2011, "Unit"),
-> ("211002", "Kursi Kuliah", "PNBP", 2011, "Unit"),
-> ("308001", "LCD", "PNBP", 2008, "Unit"),
-> ("309001", "LCD", "Hibah", 2009, "Unit"),
-> ("409001", "Layar Peraga", "BMOM", 2009, "Unit"),
-> ("409002", "Layar Peraga", "Hibah", 2009, "Unit"),
-> ("409003", "Whiteboard", "PNBP", 2009, "Unit"),
-> ("509001", "Komputer Desktop", "PNBP", 2009, "Unit"),
-> ("509002", "Printer Laser", "PNBP", 2009, "Unit");
Query OK, 18 rows affected (0.01 sec)
Records: 18 Duplicates: 0 Warnings: 0

```

Melalui perintah `insert into barang`, kita menambahkan beberapa data sekaligus. Data yang dimasukkan ini memakai cara ketiga sehingga urutan dari memasukkan nilai harus sesuai dengan urutan struktur data pada tabel `barang`. Selanjutnya kita dapat menampilkan tabel `barang` yang sudah dibuat dengan menggunakan query sebagai berikut :

```

24060123120020> select * from barang;

```

nobarang	namabarang	sumberdana	thnproduksi	satuan
108001	Meja Kuliah	PNBP	2008	Unit
109001	Meja Komputer	PNBP	2009	Unit
110001	Meja Sidang	PNBP	2010	Unit
111001	Meja Tamu	BMOM	2011	Unit
208001	Kursi Kuliah	PNBP	2008	Unit
208002	Kursi Kuliah	BMOM	2008	Unit
209001	Kursi Lipat	PNBP	2009	Unit
210001	Kursi Sidang	PNBP	2010	Unit
210002	Kursi Kuliah	Hibah	2010	Unit
211001	Kursi Tamu	BMOM	2011	Unit
211002	Kursi Kuliah	PNBP	2011	Unit
308001	LCD	PNBP	2008	Unit
309001	LCD	Hibah	2009	Unit
409001	Layar Peraga	BMOM	2009	Unit
409002	Layar Peraga	Hibah	2009	Unit
409003	Whiteboard	PNBP	2009	Unit
509001	Komputer Desktop	PNBP	2009	Unit
509002	Printer Laser	PNBP	2009	Unit

```

18 rows in set (0.00 sec)

```

Melalui perintah `select * from barang`, kita menampilkan semua isi data dari tabel `barang`. Setelah kita tampilkan, tabel `barang` sudah sesuai dengan permintaan dari Soal Tugas 1.

3.2.2. Tugas 2

3.2.2.1. Nomor 1

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select * from ruang
-> where
-> fungsi != "Ruang Seminar" AND
-> fungsi != "Ruang Tamu" AND
-> luas >= 50;
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select * from ruang`, menampilkan semua kolom yang berada di tabel `ruang`.
2. Perintah `where` , memberikan kondisi yang harus dipenuhi untuk suatu baris atau record data di tampilkan. Kondisi yang harus dipenuhi yaitu baris data kolom fungsi tidak boleh sama dengan “Ruang Seminar” dan baris data kolom fungsi tidak boleh sama dengan “Ruang Tamu” serta baris data pada kolom tabel luasnya harus lebih dari sama dengan 50.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

noruang	blokgedung	lantai	fungsi	luas	kapasitas
A201	A	2	Lab Komputer	72	50
B101	B	1	Ruang Kuliah	60	40
B102	B	1	Ruang Kuliah	60	45
B201	B	2	Ruang Kuliah	70	50
B202	B	2	Ruang Kuliah	75	60
B203	B	2	Lab Komputer	50	40
B303	B	3	Ruang Kuliah	72	50

7 rows in set (0.01 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 1.

3.2.2.2. Nomor 2

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select noruang, fungsi, luas, kapasitas from ruang
-> where
-> fungsi = "Ruang Kuliah" AND
-> luas >= (kapasitas * 1.5);
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select noruang, fungsi, luas, kapasitas from ruang`, akan menampilkan kolom `noruang`, `fungsi`, `luas`, dan `kapasitas` yang berada di tabel `ruang`.
2. Perintah `where` , memberikan kondisi yang harus di penuhi untuk suatu baris atau record data di tampilkan. Kondisi yang harus dipenuhi yaitu baris data kolom `fungsi` harus sama dengan “Ruang Kuliah” dan baris data kolom `luas` besarnya harus lebih dari sama dengan 1,5 kali baris data kolom `kapasitas`.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

```
+-----+-----+-----+-----+
| noruang | fungsi      | luas | kapasitas |
+-----+-----+-----+-----+
| B101    | Ruang Kuliah | 60   | 40        |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 2.

3.2.2.3. Nomor 3

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select * from barang
-> where
-> (sumberdana = "Hibah" OR sumberdana = "BMOM") AND
-> namabarang != "Kursi Kuliah";
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select * from barang`, akan menampilkan semua kolom yang berada di tabel `barang`.
2. Perintah `where` , memberikan kondisi yang harus di penuhi untuk suatu baris atau record data di tampilkan. Kondisi yang harus dipenuhi yaitu baris data kolom `sumberdana` harus sama dengan “Hibah” atau baris data kolom `sumberdana`

sama dengan “BMOM”. Dan nama barang tidak sama dengan “Kursi Kuliah”.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

nobarang	namabarang	sumberdana	thnproduksi	satuan
111001	Meja Tamu	BMOM	2011	Unit
211001	Kursi Tamu	BMOM	2011	Unit
309001	LCD	Hibah	2009	Unit
409001	Layar Peraga	BMOM	2009	Unit
409002	Layar Peraga	Hibah	2009	Unit

5 rows in set (0.01 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 3.

3.2.2.4. Nomor 4

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select nobarang, namabarang, sumberdana, thnproduksi from barang
-> where
-> sumberdana = "PNBP" AND thnproduksi != 2009;
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select nobarang, namabarang, sumberdana, thnproduksi from barang`, akan menampilkan kolom nobarang, namabarang, sumberdana, thnproduksi yang berada di tabel barang.
2. Perintah `where` , memberikan kondisi yang harus di penuhi untuk suatu baris atau record data di tampilkan. Kondisi yang harus dipenuhi yaitu baris data kolom sumberdana harus sama dengan “PNBP” dan baris data kolom thnproduksi tidak sama dengan 2009.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

nobarang	namabarang	sumberdana	thnproduksi
108001	Meja Kuliah	PNBP	2008
110001	Meja Sidang	PNBP	2010
208001	Kursi Kuliah	PNBP	2008
210001	Kursi Sidang	PNBP	2010
211002	Kursi Kuliah	PNBP	2011
308001	LCD	PNBP	2008

6 rows in set (0.00 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 4.

3.2.2.5. Nomor 5

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select * from barang
-> where
-> thnproduksi = 2008 AND
-> (namabarang != "Kursi Kuliah" AND sumberdana = "PNBP");
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select * from barang`, akan menampilkan semua kolom yang berada di tabel barang.
2. Perintah `where` , memberikan kondisi yang harus di penuhi untuk suatu baris atau record data di tampilkan. Kondisi yang harus dipenuhi yaitu baris data kolom `thnproduksi` sama dengan 2008 dan nama barang tidak sama dengan “Kursi Kuliah” dan baris data kolom `sumberdana` harus sama dengan “PNBP”.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

nobarang	namabarang	sumberdana	thnproduksi	satuan
108001	Meja Kuliah	PNBP	2008	Unit
308001	LCD	PNBP	2008	Unit

2 rows in set (0.00 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 5.

3.2.2.6. Nomor 6

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select * from ruang  
-> order by luas desc, kapasitas desc;
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select * from ruang`, akan menampilkan semua kolom yang berada di tabel ruang.
2. Perintah `order by`, mengatur urutan ditampilkannya baris data untuk suatu baris atau record data di tampilkan. Urutannya yaitu berdasarkan baris data kolom luas yang di tampilkan dari yang terbesar hingga yang terkecil, jika terdapat kesamaan nilai pada luas, maka diurutkan berdasarkan kapasitas dari yang terbesar hingga terkecil.

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

noruang	blokgedung	lantai	fungsi	luas	kapasitas
B202	B	2	Ruang Kuliah	75	60
A201	A	2	Lab Komputer	72	50
B303	B	3	Ruang Kuliah	72	50
B201	B	2	Ruang Kuliah	70	50
B102	B	1	Ruang Kuliah	60	45
A101	A	1	Ruang Seminar	60	40
B101	B	1	Ruang Kuliah	60	40
B203	B	2	Lab Komputer	50	40
A102	A	1	Ruang Tamu	18	6

9 rows in set (0.00 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 6.

3.2.2.7. Nomor 7

Berikut adalah masukan Query untuk yang digunakan untuk menampilkan tabel :

```
24060123120020> select * from barang  
-> order by thnproduksi desc, sumberdana;
```

Melalui Query yang telah diberikan, telah terjadi operasi sebagai berikut :

1. Perintah `select * from barang`, akan menampilkan semua kolom yang berada di tabel barang.
2. Perintah `order by`, mengatur urutan ditampilkannya baris data untuk suatu baris atau record data di tampilkan. Urutannya yaitu berdasarkan baris data kolom `thnproduksi` yang di tampilkan dari tahun terbaru hingga terlama (yang terbesar hingga yang terkecil), jika terdapat kesamaan nilai pada `thnproduksi`, maka diurutkan berdasarkan `sumberdana` berdasarkan abjad (A-Z).

Lalu tabel yang ditampilkan setelah masukan Query dijalankan, maka akan menampilkan tabel berikut :

```
24060123120020> select * from barang
-> order by thnproduksi desc, sumberdana;
```

nobarang	namabarang	sumberdana	thnproduksi	satuan
111001	Meja Tamu	BMOM	2011	Unit
211001	Kursi Tamu	BMOM	2011	Unit
211002	Kursi Kuliah	PNBP	2011	Unit
210002	Kursi Kuliah	Hibah	2010	Unit
110001	Meja Sidang	PNBP	2010	Unit
210001	Kursi Sidang	PNBP	2010	Unit
409001	Layar Peraga	BMOM	2009	Unit
309001	LCD	Hibah	2009	Unit
409002	Layar Peraga	Hibah	2009	Unit
109001	Meja Komputer	PNBP	2009	Unit
209001	Kursi Lipat	PNBP	2009	Unit
409003	Whiteboard	PNBP	2009	Unit
509001	Komputer Desktop	PNBP	2009	Unit
509002	Printer Laser	PNBP	2009	Unit
208002	Kursi Kuliah	BMOM	2008	Unit
108001	Meja Kuliah	PNBP	2008	Unit
208001	Kursi Kuliah	PNBP	2008	Unit
308001	LCD	PNBP	2008	Unit

18 rows in set (0.00 sec)

Dapat diamati bahwa tabel yang ditampilkan sudah sesuai dengan permintaan dari soal nomor 7.

BAB IV

PENUTUP

4.1. Kesimpulan

Melalui praktikum di atas, dapat dipahami bahwa dalam manipulasi tabel merupakan hal yang penting dalam pengelolaan basis data, dengan mempelajari manipulasi tabel, kita dapat melakukan operasi dalam basis data dengan lebih fleksibel. Penggunaan perintah dasar MySQL membantu dalam membangun, mengelola, dan memodifikasi basis data sesuai kebutuhan.

Pada praktikum kali ini, telah dipelajari cara memasukkan dan menampilkan tabel sesuai dengan kriteria tertentu, melakukan pengurutan dengan kriteria tertentu dan melakukan update data sesuai dengan syarat kondisi tertentu.. Selain itu, juga dipelajari cara membuat tabel baru dari tabel yang sebelumnya belum ada, melengkapi tabel dengan kolom, serta melakukan modifikasi dan peremajaan tabel seperti menambah kolom baru beserta dengan atributnya. Harapannya, ilmu yang didapatkan melalui praktikum ini dapat memberikan pemahaman yang lebih mendalam tentang pengelolaan basis data serta manipulasi tabel basis data, serta mendukung aplikasi praktis pada proyek atau pembelajaran selanjutnya.

4.2. Saran

Sebagai praktikan, saya harus memperdalam kembali konsep pengelolaan basis data pada MySQL, khususnya dalam penggunaan query untuk manipulasi data, operator pembandingan, dan logika. Pembelajaran manipulasi tabel perlu ditingkatkan agar proses pengelolaan basis data menjadi lebih efisien dan terstruktur. Dengan pemahaman yang lebih baik, saya berharap dapat lebih lancar dalam menyelesaikan studi kasus yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

Setiyowati, S. K., & Siswanti, S. (2021). *Perancangan Basis Data & Pengenalan SQL Server Management Studio*. Lembaga Penelitian Dan Pengabdian Kepada Masyarakat Universitas Dian Nuswantoro.

Papuangan, M., Safi, M., Sugiyani, Y., Qurania, A., & Sumanto. (2023). *Sistem Basis Data*. CV. KERANJANG TEKNOLOGI MEDIA.