

LAPORAN PRAKTIKUM

PRAKTIKUM DATABASE

Disusun untuk Memenuhi Matakuliah Praktikum Database
Dibimbing oleh : Sulaibatul Aslamiyah, M.Kom



Oleh:

M RAVI ACHYAR TRISTA ZHAID

1124102193

PROGRAM STUDI S1 TEKNIK INFORMATIKA
SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI
2025

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM

Matakuliah : PRATIKUM DATABASE
Oleh : M RAVI ACHYAR TRISTA ZHAID
NIM : 1124102193

Telah disahkan pada

Hari : Rabu
Tanggal : 23 April 2025

Mengetahui/ Menyetujui :

Dosen Pengampu Mata Kuliah

Sulaibatul Aslamiah, M.Kom

NIDN. 0712058304

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan inayah Nya sehingga saya dapat menyelesaikan Laporan Praktikum Database MySQL Modul 4. Terima kasih saya ucapkan kepada Ibu Dosen Sulaibatul Aslamiyah, M.Kom yang telah mengajar pratikum database.

Saya menyadari, bahwa laporan yang saya buat ini masih jauh dari kata sempurna baik segi penyusunan, bahasa, maupun penulisannya. Oleh karena itu, saya sangat mengharapkan kritik dan saran yang membangun dari Dosen Pengampu Mata Kuliah guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang.

Banyuwangi, 23 April 2025

M Ravi Achyar Trista Zhaid

MODUL PRAKTIKUM 1

1.1 Capaian Praktikum Pertemuan 1

Dalam praktikum ini, mahasiswa berhasil mencapai beberapa kompetensi dasar dalam pengelolaan basis data relasional menggunakan MySQL

1.2 Indikator Capaian

Mahasiswa menunjukkan kemampuan analisis terhadap struktur dan isi data melalui query SQL yang efisien dan akurat.

1.3 Landasan Teori

Stored Procedure adalah kumpulan perintah SQL yang disimpan dalam database dan dapat dijalankan berulang kali. Stored procedure mempermudah otomasi dan meningkatkan efisiensi dalam pengolahan data.

1.4 Pelaksanaan Praktikum

1.4.1 Uji Coba

Langkah pertama yang dilakukan dalam praktikum ini adalah pembuatan database dan tabel.

1. Membuat database kantor

Database db_penjualan berhasil dibuat menggunakan perintah:

```
mysql> CREATE DATABASE db_penjualan;  
Query OK, 1 row affected (0.00 sec)  
  
mysql>  
mysql> USE db_penjualan;  
Database changed  
mysql>
```

2. Membuat table customer

Langkah pertama yang diuji adalah pembuatan dan pengisian data pada tabel customer. Setelah tabel dibuat, data dimasukkan untuk memastikan bahwa data customer berhasil dimasukkan.

```
mysql> -- Tabel customer  
mysql> CREATE TABLE customer (  
->     id INT PRIMARY KEY,  
->     nama VARCHAR(100)  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

Melihat struktur table customer

Untuk melihat struktur table customer dengan `_desc customer;`

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
nama	varchar(100)	YES		NULL	
2 rows in set (0.07 sec)					

Query untuk Pemasukan Data Customer:

Setelah struktur tabel dibuat, langkah berikutnya adalah memasukkan data ke dalam tabel.

Pemasukan data dapat dilakukan menggunakan perintah **INSERT INTO** yang digunakan untuk menambahkan baris data ke dalam tabel

```
mysql> -- Data customer
mysql> INSERT INTO customer (id, nama) VALUES
  -> (1, 'Andi'),
  -> (2, 'Budi'),
  -> (3, 'Citra'),
  -> (4, 'Dedi');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Menampilkan data dalam tabel customer:

Untuk memastikan bahwa data telah berhasil dimasukkan ke dalam tabel, Anda dapat menggunakan perintah **SELECT** untuk menampilkan semua data yang ada dalam tabel.

id	nama
1	Andi
2	Budi
3	Citra
4	Dedi
4 rows in set (0.00 sec)	

Hasil Uji Coba:

- Data berhasil dimasukkan ke dalam tabel customer.
- Data yang dimasukkan adalah customer dengan id dan nama yang sesuai dengan yang diinginkan

3. Membuat table transaksi

langkah berikutnya adalah membuat tabel transaksi dan mengisi data transaksi yang dilakukan oleh customer. Pada tahap ini juga dilakukan uji coba untuk memastikan relasi antar tabel berjalan dengan baik.

```
mysql> -- Tabel transaksi
mysql> CREATE TABLE transaksi (
  ->   id INT PRIMARY KEY,
  ->   customer_id INT,
  ->   tanggal DATE,
  ->   jumlah DECIMAL(10,2),
  ->   FOREIGN KEY (customer_id) REFERENCES customer(id)
  -> );
Query OK, 0 rows affected (0.02 sec)
```

Melihat struktur table transaksi:

Untuk melihat struktur table customer dengan `_desc transaksi;`

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    |       |
| customer_id | int           | YES  | MUL | NULL    |       |
| tanggal   | date          | YES  |     | NULL    |       |
| jumlah     | decimal(10,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Query untuk Pemasukan Data Transaksi:

Setelah struktur tabel dibuat, langkah berikutnya adalah memasukkan data ke dalam tabel. Pemasukan data dapat dilakukan menggunakan perintah `INSERT INTO` yang digunakan untuk menambahkan baris data ke dalam tabel

```
mysql> -- Data transaksi
mysql> INSERT INTO transaksi (id, customer_id, tanggal, jumlah) VA
LUES
  -> (101, 1, '2024-01-10', 50000),
  -> (102, 2, '2024-02-15', 75000),
  -> (103, 1, '2024-03-20', 100000),
  -> (104, 3, '2024-04-01', 60000);
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Melihat struktur table transaksi

Untuk memastikan bahwa data telah berhasil dimasukkan ke dalam tabel, Anda dapat menggunakan perintah **SELECT** untuk menampilkan semua data yang ada dalam tabel.

```
+-----+-----+-----+-----+
| id | customer_id | tanggal | jumlah |
+-----+-----+-----+-----+
| 101 | 1 | 2024-01-10 | 50000.00 |
| 102 | 2 | 2024-02-15 | 75000.00 |
| 103 | 1 | 2024-03-20 | 100000.00 |
| 104 | 3 | 2024-04-01 | 60000.00 |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Uji coba :

- Data detail transaksi berhasil dimasukkan.
- Data pada tabel detail_transaksi berhasil berelasi dengan baik dengan tabel barang dan transaksi berdasarkan id_barang dan id_transaksi

4. Membuat table barang

Selanjutnya, uji coba dilakukan untuk tabel barang. Tabel ini menyimpan data mengenai barang yang dijual. Pengujian dilakukan dengan memastikan bahwa data barang dapat dimasukkan dengan benar.

```
mysql> -- Tabel barang
mysql> CREATE TABLE barang (
->   id INT PRIMARY KEY,
->   nama_barang VARCHAR(100)
-> );
Query OK, 0 rows affected (0.02 sec)
```

Melihat struktur table barang

Untuk melihat struktur table customer dengan **desc barang;**

```
mysql> desc barang;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
nama_barang	varchar(100)	YES		NULL	

2 rows in set (0.00 sec)

Query untuk Pemasukan Data barang:

Setelah struktur tabel dibuat, langkah berikutnya adalah memasukkan data ke dalam tabel. Pemasukan data dapat dilakukan menggunakan perintah INSERT INTO yang digunakan untuk menambahkan baris data ke dalam table

```
mysql> -- Data barang
mysql> INSERT INTO barang (id, nama_barang) VALUES
-> (1, 'Pulpen'),
-> (2, 'Buku Tulis'),
-> (3, 'Penghapus');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Melihat struktur table transaksi

Untuk memastikan bahwa data telah berhasil dimasukkan ke dalam tabel, Anda dapat menggunakan perintah **SELECT** untuk menampilkan semua data yang ada dalam tabel.

```
mysql> select * from barang;
+----+-----+
| id | nama_barang |
+----+-----+
| 1  | Pulpen      |
| 2  | Buku Tulis  |
| 3  | Penghapus   |
+----+-----+
3 rows in set (0.00 sec)
```

Uji coba :

- Data barang berhasil dimasukkan, dengan masing-masing barang memiliki id yang unik dan sesuai dengan yang diinginkan. Hasil Uji coba sebagai Berikut:

5. Membuat table detail transaksi

Tabel detail_transaksi digunakan untuk mencatat rincian transaksi, seperti barang yang dibeli dan jumlahnya. Pada tahap ini, uji coba dilakukan dengan memastikan bahwa data transaksi dapat dimasukkan dengan benar dan relasi antar tabel bekerja sebagaimana mestinya.

```
mysql> -- Tabel detail_transaksi
mysql> CREATE TABLE detail_transaksi (
-> id INT PRIMARY KEY,
-> id_transaksi INT,
-> id_barang INT,
-> jumlah INT,
-> FOREIGN KEY (id_barang) REFERENCES barang(id)
-> );
Query OK, 0 rows affected (0.02 sec)
```


Melihat struktur table detail_transaksi

Untuk melihat struktur table customer dengan_desc detail_barang;

```
mysql> desc detail_transaksi;
+-----+-----+-----+-----+-----+-----+
| Field          | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | int  | NO   | PRI | NULL    |       |
| id_transaksi  | int  | YES  |     | NULL    |       |
| id_barang     | int  | YES  | MUL | NULL    |       |
| jumlah        | int  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Query untuk Pemasukan Data detail transaksi:

Setelah struktur tabel dibuat, langkah berikutnya adalah memasukkan data ke dalam tabel.

Pemasukan data dapat dilakukan menggunakan perintah INSERT INTO yang digunakan untuk menambahkan baris data ke dalam table

```
mysql> -- Data detail_transaksi
mysql> INSERT INTO detail_transaksi (id, id_transaksi, id_barang,
jumlah) VALUES
-> (1, 101, 1, 1),
-> (2, 102, 2, 2),
-> (3, 103, 3, 3),
-> (4, 104, 1, 4);
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Melihat struktur table detail_transaksi

Untuk memastikan bahwa data telah berhasil dimasukkan ke dalam tabel, Anda dapat menggunakan perintah **SELECT** untuk menampilkan semua data yang ada dalam tabel.

```
mysql> select * from detail_transaksi;
+-----+-----+-----+-----+
| id | id_transaksi | id_barang | jumlah |
+-----+-----+-----+-----+
| 1  | 101         | 1         | 1      |
| 2  | 102         | 2         | 2      |
| 3  | 103         | 3         | 3      |
| 4  | 104         | 1         | 4      |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Uji coba :

- Data detail transaksi berhasil dimasukkan.
- Data pada tabel detail_transaksi berhasil berelasi dengan baik dengan tabel barang dan transaksi berdasarkan id_barang dan id_transaksi

a. 1.1 Langkah Uji Coba

Menampilkan **id** dan **nama** customer yang **pernah melakukan transaksi**, dengan memanfaatkan **subquery** (SELECT di dalam SELECT).

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetCustomerPernahTransaksi()
-> BEGIN
->     SELECT id, nama
->     FROM customer
->     WHERE id IN (
->         SELECT DISTINCT customer_id
->         FROM transaksi
->     );
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql> -- Jalankan stored procedure
mysql> CALL GetCustomerPernahTransaksi();
+----+-----+
| id | nama |
+----+-----+
| 1  | Andi |
| 2  | Budi |
| 3  | Citra|
+----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

- **Subquery:**

SELECT DISTINCT customer_id FROM transaksi

→ Mengambil semua customer_id unik dari tabel transaksi.

- **QueryUtama:**

SELECT id, nama FROM customer WHERE id IN (...)

→ Mengambil id dan nama customer yang id-nya terdapat dalam hasil subquery di atas.

b. 1.2 Langkah Uji Coba

- Menggunakan **CASE** untuk menampilkan nilai **jumlah** dalam bentuk **huruf/text** (**satuan 1–10**).
- Menampilkan:
 - id_transaksi
 - nama_barang
 - jumlah dalam bentuk kata (satu, dua, tiga, dst)
- Mengurutkan berdasarkan id_transaksi ascending.
- Menggunakan **Stored Procedure**

```
mysql> DELIMITER //
```

```
mysql>
```

```
mysql> CREATE PROCEDURE GetDetailTransaksiHuruf()
```

```
mysql> BEGIN
```

```
mysql>     SELECT
```

```
mysql>         dt.id_transaksi,
```

```
mysql>         b.nama_barang,
```

```
mysql>         CASE dt.jumlah
```

```
mysql>             WHEN 1 THEN 'Satu'
```

```
mysql>             WHEN 2 THEN 'Dua'
```

```
mysql>             WHEN 3 THEN 'Tiga'
```

```
mysql>             WHEN 4 THEN 'Empat'
```

```
mysql>             WHEN 5 THEN 'Lima'
```

```
mysql>             ELSE 'Jumlah Besar'
```

```
mysql>         END AS jumlah_dalam_huruf
```

```
mysql>     FROM
```

```
mysql>         detail_transaksi dt
```

```
mysql>     JOIN
```

```
mysql>         barang b ON dt.id_barang = b.id
```

```
mysql>     ORDER BY
```

```
mysql>         dt.id_transaksi ASC;
```

```
mysql> END //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
```

```
mysql> DELIMITER ;
```

```
mysql> CALL GetDetailTransaksiHuruf();
```

id_transaksi	nama_barang	jumlah_dalam_huruf
101	Pulpen	Satu
102	Buku Tulis	Dua
103	Penghapus	Tiga
104	Pulpen	Empat

4 rows in set (0.00 sec)

c. Analisa Hasil

Stored Procedure berhasil dibuat dan berjalan sesuai spesifikasi. Perintah CASE mampu mengubah nilai angka menjadi bentuk teks dengan baik, dan data disajikan dengan cara yang lebih mudah dipahami. Struktur prosedur ini sangat cocok digunakan dalam aplikasi laporan transaksi toko atau sistem penjualan.

1.5 Kesimpulan

- Praktikum ini berhasil mengimplementasikan **konsep subquery (subSelect)** dan **perintah seleksi CASE** dalam pengelolaan basis data MySQL.
- Subquery berhasil digunakan untuk menampilkan **id dan nama customer yang pernah melakukan transaksi**, dengan memanfaatkan relasi antar tabel secara efisien.
- Penggunaan **Stored Procedure** terbukti mempermudah otomatisasi query kompleks dan meningkatkan efisiensi pengolahan data, terutama dalam menampilkan data yang membutuhkan logika tambahan seperti konversi angka ke teks.
- Penerapan perintah **CASE** dalam Stored Procedure berhasil mengubah nilai *jumlah* dari angka menjadi **kata dalam bahasa Indonesia**, tanpa mengganggu integritas data numerik di tabel utama

BAB **

PENUTUP

****.1 Kesimpulan**

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa untuk pembuatan dan pengelolaan database merupakan sebuah keterampilan penting dalam pengelolaan data. Mahasiswa dapat memahami perintah dasar SQL untuk membuat, mengubah, dan menghapus data dalam tabel di database. Praktikum ini membantu mahasiswa memahami konsep dasar sebuah pengolahan data. Melalui latihan langsung, mahasiswa dapat lebih memahami struktur dan hubungan antara tabel dalam sebuah database.

****.2 Saran**

Diharapkan mahasiswa lebih aktif dalam mengeksplorasi perintah SQL lainnya untuk memperdalam pemahaman. Sebaiknya sebelum memulai praktikum, mahasiswa belajar memahami teori terlebih dahulu agar lebih efektif dalam mengerjakan tugas. Juga saya ucapkan terimakasih kepada Sulaibatul Aslamiyah, M.Kom selaku dosen yang telah mengajarkan ilmu ini kepada saya. Dan mohon maaf jika ada salah kata yang kurang berkenan

DAFTAR PUSTAKA

Tuliskan rujukan yang anda gunakan baik website maupun buku seperti contoh dibawah.

1. <https://www.youtube.com/@farukalfiyan/playlists>
2. <https://www.youtube.com/@suhutv>
3. <https://ngodingdata.com/tutorial-mysql-untuk-pemula/>

LAMPIRAN