

# **LAPORAN PRAKTIKUM**

## **PRAKTIKUM DATABASE PERTEMUAN 5**

Disusun untuk Memenuhi Matakuliah Praktikum {Nama Matakuliah}

Dibimbing oleh : Sulaibatul Aslamiyah, M.Kom



Oleh:

**RAFLI RAHMAN.EFENDY**

**1124102162**

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

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM**

Matakuliah : Praktikum Database

Oleh : Rafli Rahman.Efendy

NIM : 1124102162

**Telah disahkan pada**

Hari : RABU

Tanggal : 30/04/2025

**Mengetahui/ Menyetujui :**

Dosen Pengampu Mata Kuliah

**Sulaibatul Aslamiah, M.Kom**

NIDN. 0712058304

## **KATA PENGANTAR**

Berisikan kata pengantar yang dituliskan oleh mahasiswa untuk laporan praktikum matakuliah yang diikuti.

## **MODUL PRAKTIKUM 1**

### **1.1 Capaian Praktikum Pertemuan 1**

Pencapaian pada praktikum ini adalah membuat sebuah table mysql pada database yang telah di buat

### **1.2 Indikator Capaian**

- Membuat Tabel 'Profil Kantor' dan 'user' di buat dalam table mysql tanpa eror.
- Membuat Sruktur table yang sesuai dengan yang di buat/study kasus nya(kolom tipe data, dan constraint).

### **1.3 Landasan Teori**

Jelaskan landasan teori yang digunakan dalam penyelesaian studi kasus.

### **1.4 Pelaksanaan Praktikum**

Landasan teori MySQL adalah Structured Query Language (SQL) sebagai bahasa interaktif untuk mengelola data. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL untuk menjalankan fungsinya.

### 1.4.1 Percobaan Pertama

Pada materi ini akan membahas mengenai perintah subSelect (perintah select dalam select), Trigger (perintah SQL secara otomatis), Function, serta Store Procedure (kumpulan perintah / statemen) yang masing-masing. Perintah query subSelect ini merupakan perintah yang dimana memungkinkan untuk melakukan query didalam query.

```
mysql> use db_toko;
Database changed
mysql> SELECT id_transaksi, jumlah FROM detail_transaksi WHERE jumlah = ( SELECT MAX(jumlah) FROM detail_transaksi);
+-----+-----+
| id_transaksi | jumlah |
+-----+-----+
|          2  |      3 |
+-----+-----+
1 row in set (0.02 sec)
```

#### Statement subSelect :

- SELECT id\_transaksi, jumlah FROM detail\_transaksi WHERE jumlah = ( SELECT MAX(jumlah) FROM detail\_transaksi);

#### Deskripsi :

sebuah pernyataan SELECT yang berada di dalam pernyataan SQL lain, biasanya digunakan untuk:

- Menyaring data berdasarkan hasil dari query lain.
- Menghitung nilai agregat untuk digunakan dalam query utama.
- Menyusun query yang kompleks dan fleksibel.

#### Hasil Uji Coba :

```
mysql>
mysql> DELIMITER ;
mysql> CALL jumlahItemBarang(2);
+-----+
| total_jumlah |
+-----+
|          3  |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

#### Penjelasan:

- SubSelect (SELECT MAX(jumlah) FROM detail\_transaksi) akan mencari nilai terbesar (maksimum) dari kolom jumlah pada tabel detail\_transaksi.
- Lalu query utama akan menampilkan semua id\_transaksi dan jumlah dari baris-baris yang jumlahnya sama dengan nilai maksimum tersebut.

### 1.4.2 Percobaan Dua

Trigger digunakan untuk memanggil satu atau beberapa perintah SQL secara otomatis sebelum atau sesudah terjadi proses INSERT, UPDATE atau DELETE dari suatu tabel. Sebagai contoh semisal ingin dilakukan penyimpanan id pelanggan secara otomatis ke tabel 'log' sebelum menghapus data di tabel pelanggan.

```
mysql> DROP TRIGGER before_insert;
Query OK, 0 rows affected (0.03 sec)

mysql> |
```

- BEFORE | AFTER digunakan untuk menentukan kapan proses secara otomatis akan dieksekusi, sebelum atau sesudah proses.
- INSERT | UPDATE | DELETE digunakan untuk menentukan event yang dijadikan trigger untuk menjalankan perintah-perintah di dalam triggers.
- Statement Triggers
- CREATE TRIGGER name  
[BEFORE|AFTER] [INSERT|UPDATE|DELETE]  
ON tablename  
FOR EACH ROW statement

### 1.Membuat tabel Log

```
mysql> use db_toko;
Database changed
mysql> create TABLE log(
-> description TEXT,
-> datetime DATETIME,
-> user_id VARCHAR(50)
-> )
-> ;
Query OK, 0 rows affected (0.07 sec)
```

### 2. Triggers pencatatan ke tabel Log

```
mysql> use db_toko;
Database changed
mysql> DELIMITER $$
mysql> CREATE TRIGGER db_toko.before_insert BEFORE INSERT ON
-> db_toko.customer
-> FOR EACH ROW BEGIN
-> INSERT INTO `log` (description, `datetime`, user_id)
-> VALUES (CONCAT('Insert data ke tabel pelanggan id_plg
-> = ', NEW.id_customer), NOW(), USER());
-> END;
-> $$
Query OK, 0 rows affected (0.01 sec)
```

#### Deskripsi :

Statement atau perintah dalam trigger dapat berupa satu perintah saja, dan dapat juga beberapa perintah sekaligus. Jika terdapat beberapa perintah dalam trigger, maka gunakan perintah BEGIN dan END untuk mengawali dan mengakhiri perintah. Didalam statement trigger, pengguna dapat mengakses record tabel sebelum atau sesudah proses dengan menggunakan NEW dan OLD.

#### Hasil Uji Coba :

Penjelasan Kolom Membuat tabel Log :

- description: Menyimpan penjelasan atau isi dari log (misalnya "Hasil query dengan jumlah maksimal").
- datetime: Waktu saat log dimasukkan.
- user\_id: ID pengguna yang melakukan uji coba.

```
mysql> use db_toko;
Database changed
mysql> desc log;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| description | text      | YES  |     | NULL    |       |
| datetime   | datetime  | YES  |     | NULL    |       |
| user_id    | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)
```

```
CREATE TABLE log (
  description TEXT,
  datetime DATETIME,
  user_id VARCHAR(50)
);
```

Misalnya kamu ingin mencatat bahwa query subSelect berhasil dijalankan dan hasilnya jumlah maksimal adalah 7, bisa kamu lakukan seperti ini:

```
INSERT INTO log (description, datetime, user_id)
VALUES ('Query subSelect: jumlah maksimal adalah 7', NOW(), 'user123');
```

Penjelasan Triggers pencatatan ke tabel Log :

- trigger ini akan otomatis mencatat ke tabel log setiap kali ada INSERT ke tabel customer.
- NEW.id\_customer mengacu pada data yang akan dimasukkan.
- USER() mengambil nama user database yang sedang aktif.

```
mysql> use db_toko;
Database changed
mysql> desc log;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| description | text      | YES  |     | NULL    |       |
| datetime   | datetime  | YES  |     | NULL    |       |
| user_id    | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Versi Diperbaiki :

```
DELIMITER $$
```

```
CREATE TRIGGER before_insert_customer
```

```

BEFORE INSERT ON customer
FOR EACH ROW
BEGIN
    INSERT INTO log (description, datetime, user_id)
    VALUES (
        CONCAT('Insert data ke tabel pelanggan, id_customer = ', NEW.id_customer),
        NOW(),
        USER()
    );
END$$
DELIMITER ;

```

### 1.4.3 Percobaan Tiga

Stored Procedure dari segi bentuk dan sifatnya terbagi menjadi 2, yaitu FUNCTION dan PROCEDURE. Perbedaan utama antara function dan procedure adalah terletak pada nilai yang dikembalikannya (di-return). Function memiliki suatu nilai yang dikembalikan (di-return), sedangkan procedure tidak. Umumnya suatu procedure hanya berisi suatu kumpulan proses yang tidak menghasilkan value, biasanya hanya menampilkan saja.

#### 1. Pemanggilan Procedure jumlah customer

```

mysql> use db_toko;
Database changed
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION hitung_total(harga INT, jumlah INT)
-> RETURNS INT
-> DETERMINISTIC
-> BEGIN
->     RETURN harga * jumlah;
-> END;
-> $$
Query OK, 0 rows affected (0.03 sec)

```

- jumlahCustomer2 adalah prosedur yang menerima satu parameter output (jumlah) bertipe INT.
- SELECT COUNT(\*) INTO jumlah akan menghitung total baris di tabel customer dan menyimpan nilainya ke variabel output.

```

DELIMITER $$

CREATE PROCEDURE jumlahCustomer2(OUT jumlah INT)
BEGIN
    SELECT COUNT(*) INTO jumlah FROM customer;

```



```
END$$
```

```
DELIMITER ;
```

### Hadil Uji Coba :

```
mysql> ALTER FUNCTION hitung_total COMMENT '...';  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> SHOW PROCEDURE STATUS WHERE Db = 'db_toko';  
Empty set (0.03 sec)
```

```
mysql> SHOW TRIGGERS FROM db_toko;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| Trigger | Event | Table | Statement | Timing | Created | Definer | sql_mode | character_set_client |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| collation_connection | Database Collation |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| before_insert | INSERT | customer | BEGIN  
INSERT INTO `log` (description, `datetime`, user_id)  
VALUES (CONCAT('Insert data ke tabel pelanggan id_plg  
= ', NEW.id_customer), NOW(), USER());  
END | BEFORE | 2025-04-23 11:48:01.72 | ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ERROR_FOR  
_DIVISION_BY_ZERO, NO_ENGINE_SUBSTITUTION | root@localhost | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.02 sec)
```

## 2. Procedure menghitung jumlah barang berdasarkan id transaksi

### Penjelasan . Procedure menghitung jumlah barang berdasarkan id transaksi :

- Menggunakan JOIN lebih jelas daripada koma (,), karena itu cara modern yang lebih readable.
- IN digunakan karena parameter cust hanya dipakai sebagai input.
- AS total\_barang memberi label agar hasil query lebih mudah dibaca.

```
mysql> use db_toko;  
Database changed  
mysql> DELIMITER $$  
mysql>  
mysql> CREATE FUNCTION jumlahStockBarang(prod VARCHAR(5))  
-> RETURNS INT  
-> DETERMINISTIC  
-> READS SQL DATA  
-> BEGIN  
-> DECLARE jumlah INT;  
-> SELECT COUNT(*) INTO jumlah  
-> FROM barang  
-> WHERE id_barang = prod;  
-> RETURN jumlah;  
-> END$$  
Query OK, 0 rows affected (0.01 sec)
```

### Hadil Uji Cobanya :

```
mysql>  
mysql> DELIMITER ;  
mysql> SELECT jumlahStockBarang('B002');  
+-----+  
| jumlahStockBarang('B002') |  
+-----+  
| 1 |  
+-----+  
1 row in set (0.01 sec)
```

```
DELIMITER $$
```

```
CREATE PROCEDURE jumlahItemBarang (IN cust VARCHAR(5))
```

```

BEGIN

SELECT SUM(detail_transaksi.jumlah) AS total_barang

FROM transaksi

JOIN detail_transaksi ON transaksi.id_transaksi = detail_transaksi.id_transaksi

WHERE transaksi.id_transaksi = cust;

END$$

DELIMITER ;

```

### 3. Function menghitung jumlah stok barang

Penjelasan Function menghitung jumlah stok barang :

- DETERMINISTIC artinya hasil fungsi ini selalu sama untuk input yang sama (dibutuhkan dalam beberapa versi MySQL).
- Jika stok bisa tersebar di beberapa baris (misalnya karena ada tabel mutasi barang), kamu bisa pakai SUM(stok).

```

mysql> use db_toko;
Database changed
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE jumlahItemBarang (
->     IN id_trans INT
-> )
-> BEGIN
->     SELECT SUM(detail_transaksi.jumlah) AS total_jumlah
->     FROM transaksi
->     JOIN detail_transaksi ON transaksi.id_transaksi = detail_transaksi.id_transaksi
->     WHERE transaksi.id_transaksi = id_trans;
-> END$$
Query OK, 0 rows affected (0.02 sec)

```

Hasil Uji Cobanya :

```

mysql>
mysql> DELIMITER ;
mysql> CALL jumlahItemBarang(2);
+-----+
| total_jumlah |
+-----+
|          3 |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

```

#### 1.4.4 Percobaan Empat

##### Penseleksian Kondisi

Dengan adanya penyeleksian kondisi, dapat dilakukan pengaturan alur proses yang terjadi dalam database. Di MySQL, penyeleksian kondisi terdiri dari IF, IF...ELSE dan CASE. Berikut ini bentuk umum ketiga perintah.

##### 1. Perintah IF

```
mysql> use db_toko;
Database changed
mysql> SELECT COUNT(id_transaksi) AS Total,
-> IF(COUNT(id_transaksi) > 0,
->     CONCAT('Jumlah transaksi: ', COUNT(id_transaksi), ' kali'),
->     'Tidak ada transaksi') AS 'Status'
-> FROM transaksi
-> WHERE id_customer = 'P001'
-> ;
+-----+-----+
| Total | Status                |
+-----+-----+
|      2 | Jumlah transaksi: 2 kali |
+-----+-----+
1 row in set (0.02 sec)
```

##### Penjelasan Penjelasan IF :

- COUNT(id\_transaksi) menghitung jumlah transaksi dari customer 'P001'
- IF(..., ..., ...) akan menampilkan:
- "Jumlah transaksi: X kali" jika ada transaksi
- "Tidak ada transaksi" jika jumlahnya 0

##### • Perintah IF

```
SELECT COUNT(id_transaksi) AS Total,
      IF(COUNT(id_transaksi) > 0,
          CONCAT('Jumlah transaksi: ', COUNT(id_transaksi), ' kali'),
          'Tidak ada transaksi') AS 'Status'
FROM transaksi
WHERE id_customer = 'P001'
```

##### 2. Perintah Case

```
mysql> use db_toko;
Database changed
mysql> SELECT COUNT(id_transaksi) AS 'Total Transaksi', CASE
->     WHEN COUNT(id_transaksi) = 1 THEN 'Sedikit'
->     WHEN COUNT(id_transaksi) = 2 THEN 'Lumayan'
->     WHEN COUNT(id_transaksi) > 3 THEN 'Banyak'
->     ELSE 'Tidak ada transaksi' END AS 'Status'
-> FROM transaksi
-> WHERE id_customer='P001'
-> ;
+-----+-----+
| Total Transaksi | Status |
+-----+-----+
|                2 | Lumayan |
+-----+-----+
1 row in set (0.00 sec)
```

##### Penjelasan Perintah Case :

- Untuk memberi label data (kategori, status, level)
- Digunakan di SELECT, ORDER BY, bahkan WHERE

- **Perintah Case**

```
SELECT COUNT(id_transaksi) AS 'Total Transaksi', CASE
    WHEN COUNT(id_transaksi) = 1 THEN 'Sedikit'
    WHEN COUNT(id_transaksi) = 2 THEN 'Lumayan'
    WHEN COUNT(id_transaksi) > 3 THEN 'Banyak'
    ELSE 'Tidak ada transaksi' END AS 'Status'
FROM transaksi
WHERE id_customer='P001'
```

### 3. Function IF

#### Penjelasan Function IF :

Fungsi cekCustomer yang kamu buat adalah contoh penggunaan IF dalam sebuah FUNCTION di MySQL — dan kamu sudah menulisnya dengan sangat baik! Sekarang mari kita jelaskan langkah per langkah, termasuk struktur, logika IF, dan hasil uji cobanya.

- Fungsi ini sangat berguna untuk membuat respon personal terhadap data di database.
- IF di dalam FUNCTION bisa menggantikan CASE jika kondisi sederhana.
- Cocok untuk laporan, validasi data, atau feedback ke user.

#### Hasil Uji Cobanya :

```
mysql> use db_toko;
Database changed
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION cekCustomer(customer VARCHAR(5))
-> RETURNS VARCHAR(100)
-> DETERMINISTIC
-> READS SQL DATA
-> BEGIN
-> DECLARE jumlah INT;
->
-> SELECT COUNT(id_transaksi) INTO jumlah
-> FROM transaksi
-> WHERE id_customer = customer;
->
-> IF jumlah > 0 THEN
-> RETURN CONCAT('Anda sudah bertransaksi sebanyak ', jumlah, ' kali');
-> ELSE
-> RETURN 'Anda belum pernah melakukan transaksi';
-> END IF;
-> END$$
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT cekCustomer('P001');
+-----+
| cekCustomer('P001') |
+-----+
| Anda sudah bertransaksi sebanyak 2 kali |
+-----+
```

- **Function IF**

DELIMITER \$\$

CREATE FUNCTION cekCustomer (customer VARCHAR(5))

```

RETURNS VARCHAR (100)

BEGIN
    DECLARE jumlah INT;
    SELECT COUNT(id_transaksi) INTO jumlah FROM transaksi
        WHERE id_customer=customer;
    IF (jumlah > 0) THEN
        RETURN CONCAT("Anda sudah bertransaksi sebanyak ",
            jumlah, " kali");
    ELSE
        RETURN "Anda belum pernah melakukan transaksi";
    END IF;
END$$
DELIMITER ;
SELECT cekCustomer('P001')

```

#### 4. Function Case

##### Penjelasan Function Case :

Function getDiskon yang kamu buat menggunakan struktur CASE untuk menentukan nilai diskon berdasarkan jumlah pembelian. Kodenya sudah benar dan jelas, dan ini contoh yang sangat bagus untuk pemakaian CASE dalam FUNCTION`.

- Fungsi ini sangat cocok untuk aplikasi diskon di e-commerce, kasir, atau sistem penjualan.
- CASE di dalam FUNCTION berguna saat logika bercabang lebih dari 2 kondisi.
- Bisa dengan mudah dikembangkan untuk lebih banyak tingkatan.

##### Hasil Uji Cobanya :

```
Database changed
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION getDiskon(jumlah INT)
mysql> -- RETURNS INT
mysql> -- DETERMINISTIC
mysql> -- NO SQL
mysql> -- BEGIN
mysql> -- DECLARE diskon INT;
mysql> --
mysql> -- CASE
mysql> -- WHEN jumlah >= 100 THEN
mysql> -- SET diskon = 10;
mysql> -- WHEN jumlah >= 50 AND jumlah < 100 THEN
mysql> -- SET diskon = 5;
mysql> -- WHEN jumlah >= 20 AND jumlah < 50 THEN
mysql> -- SET diskon = 3;
mysql> -- ELSE
mysql> -- SET diskon = 0;
mysql> -- END CASE;
mysql> -- RETURN diskon;
mysql> -- END$$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT getDiskon(170);
+-----+
| getDiskon(170) |
+-----+
|             10 |
+-----+
```

#### 1.4.3 Tugas Praktikum Database Pada Modul 5

Tampilkan id dan nama customer dari tabel customer yang pernah melakukan transaksi pada tabel transaksi dengan menggunakan perintah subSelect (Select didalam select).

Gunakan perintah seleksi CASE untuk mengubah field jumlah dari angka menjadi huruf dengan menampilkan id transaksi, nama barang, dan jumlah dari tabel detail transaksi dan tabel barang dengan mengurutkan data berdasarkan id transaksi secara ascending.

```
mysql> use db_toko;
Database changed
mysql> SELECT
-> dt.id_transaksi,
-> b.nama_barang,
-> CASE dt.jumlah
-> WHEN 1 THEN 'Satu'
-> WHEN 2 THEN 'Dua'
-> WHEN 3 THEN 'Tiga'
-> WHEN 4 THEN 'Empat'
-> WHEN 5 THEN 'Lima'
-> ELSE 'Banyak'
-> END AS jumlah_dalam_huruf
-> FROM detail_transaksi dt
-> JOIN barang b ON dt.id_barang = b.id_barang
-> ORDER BY dt.id_transaksi ASC;
+-----+-----+-----+
| id_transaksi | nama_barang | jumlah_dalam_huruf |
+-----+-----+-----+
| 1 | Pulpen | Satu |
| 1 | Penggaris | Satu |
| 2 | Pensil | Tiga |
| 3 | Buku Tulis | Satu |
| 4 | Pulpen | Dua |
| 5 | Pensil | Satu |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

**Penjelasan :**

#### Bagian 1: SubSelect (Select di dalam Select).

- Tampilkan id dan nama customer dari tabel customer
- yang pernah melakukan transaksi di tabel transaksi.

**Query:**

```
SELECT id_customer, nama_customer
FROM customer
WHERE id_customer IN (
    SELECT DISTINCT id_customer
    FROM transaksi
);
```

**Penjelasan:**

- SubSelect SELECT DISTINCT id\_customer FROM transaksi akan mengambil semua id\_customer yang pernah bertransaksi.
- Query utama akan mengambil id\_customer dan nama\_customer dari tabel customer yang ID-nya terdapat dalam hasil subquery.

#### Bagian 2: Seleksi CASE untuk Konversi Angka ke Huruf.

Tampilkan id\_transaksi, nama\_barang, dan jumlah dalam bentuk huruf (bukan angka) dari tabel detail\_transaksi dan barang, menggunakan CASE danurut berdasarkan id\_transaksi ASC.

**Query:**

```
SELECT
    dt.id_transaksi,
    b.nama_barang,
```

```

CASE dt.jumlah
  WHEN 1 THEN 'Satu'
  WHEN 2 THEN 'Dua'
  WHEN 3 THEN 'Tiga'
  WHEN 4 THEN 'Empat'
  WHEN 5 THEN 'Lima'
END AS jumlah_huruf
FROM detail_transaksi dt
JOIN barang b ON dt.id_barang = b.id_barang
ORDER BY dt.id_transaksi ASC;

```

**Query 2:**

```

SELECT
dt.id_transaksi,
b.nama_barang,
CASE dt.jumlah
  WHEN 1 THEN 'Satu'
  WHEN 2 THEN 'Dua'
  WHEN 3 THEN 'Tiga'
  WHEN 4 THEN 'Empat'
  WHEN 5 THEN 'Lima'
  ELSE 'Banyak'
END AS jumlah_dalam_huruf
FROM detail_transaksi dt
JOIN barang b ON dt.id_barang = b.id_barang
ORDER BY dt.id_transaksi ASC;

```

**Deskripsi :**

**Menampilkan tiga kolom:**

- id\_transaksi → ID transaksi dari tabel detail\_transaksi
  - nama\_barang → Nama barang dari tabel barang
  - jumlah\_dalam\_huruf → Jumlah barang dalam bentuk huruf ("Satu", "Dua", dst), menggunakan CASE
1. Jika jumlah 1–5, akan dikonversi menjadi teks (Satu, Dua, dll).



2. Jika jumlah lebih dari 5, akan ditampilkan sebagai 'Banyak'.
3. Data akan diurutkan berdasarkan id\_transaksi secara ascending (A-Z).

## **BAB 1**

### **PENUTUP**

#### **Kesimpulan**

teori MySQL adalah Structured Query Language (SQL) sebagai bahasa interaktif untuk mengelola data. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL untuk menjalankan fungsinya..

#### **Saran**

Saran saya tidak ada perubahan pembelajaran sama seperti di Semester 1 dan masih masuk di praktikum kali kecuali pada pengisian laporan praktikum database ke word itu yang perlu di tanyakan dan cara pengisian laporan untuk kedepannya.

## DAFTAR PUSTAKA

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

1. Tim Asisten Dosen. 2014. Modul 1 Pengenalan Sistem Operasi, Ide Visual C++, Dan Algoritma Pemrograman. Malang: Universitas Negeri Malang.
2. Program Konversi Suhu (online)  
<http://bondanoky.blogspot.com/2012/10/program-konversi-suhu-c.html>. Di akses 8 September.

## **LAMPIRAN**

Berisikan syntax atau gambar yang dibutuhkan dalam tiap pertemuan praktikum.