

# **PRAKTIKUM STORED PROCEDURE & FUNCTION**



**Disusun Oleh:**

Muhammad Farrel Ferdinand – XI RPL 1

**SEKOLAH MENENGAH KEJURUAN**

**NEGERI 11 BANDUNG**

**2024**

## LKPD 1

### 1. Buat sebuah function untuk menghitung rata-rata dari 3 bilangan

```
MySQL [procedure]> DELIMITER //
MySQL [procedure]> CREATE FUNCTION hitung_rata_rata(a INT,b INT,c INT)
-> RETURNS DECIMAL(10,2)
-> DETERMINISTIC
-> BEGIN
-> DECLARE rata_rata DECIMAL(10,2);
-> SET rata_rata = (a + b + c) / 3.0;
-> RETURN rata_rata;
-> END //
Query OK, 0 rows affected (0.003 sec)
```

```
MySQL [procedure]> SELECT hitung_rata_rata(10, 20, 30); //
+-----+
| hitung_rata_rata(10, 20, 30) |
+-----+
|                20.00 |
+-----+
1 row in set (0.001 sec)
```

Kode di atas adalah definisi fungsi MySQL `hitung\_rata\_rata`, yang mengambil tiga parameter bilangan bulat `a`, `b`, dan `c`, dan mengembalikan rata-rata dari ketiga bilangan tersebut sebagai nilai desimal. Fungsi ini diciptakan untuk memberikan solusi cepat dalam menghitung rata-rata dari sejumlah bilangan bulat yang diberikan sebagai argumen tanpa memerlukan proses manual yang rumit.

### 2. Buat sebuah function yang akan menampilkan jumlah dari seluruh Harga pada table produk

```
MySQL [procedure]> DELIMITER //
MySQL [procedure]> CREATE FUNCTION total_harga_produk()
-> RETURNS INT
-> DETERMINISTIC
-> BEGIN
-> DECLARE total INT;
-> SELECT SUM(harga) INTO total FROM produk;
-> RETURN total;
-> END //
Query OK, 0 rows affected (0.008 sec)

MySQL [procedure]> SELECT total_harga_produk();
-> //
+-----+
| total_harga_produk() |
+-----+
|                298000 |
+-----+
1 row in set (0.004 sec)
```

Kode di atas adalah definisi fungsi MySQL `total\_harga\_produk`, yang menghitung total harga dari semua produk dalam tabel `produk` dan mengembalikan nilai tersebut sebagai integer. Fungsi ini dibuat untuk memberikan solusi cepat dalam menghitung total harga produk secara otomatis tanpa perlu melakukan perhitungan manual.

3. buat sebuah prosedur untuk menambahkan data supplier dan buat 5 data baru supplier(screenshot juga tabelnya)

```
MySQL [procedure]> DELIMITER //
MySQL [procedure]> CREATE PROCEDURE tambah_supplier(
  -> IN nama_pemasok VARCHAR(50),
  -> IN alamat_pemasok VARCHAR(50),
  -> IN nama_barang VARCHAR(50),
  -> IN jumlah_barang INT
  -> )
  -> DETERMINISTIC
  -> BEGIN
  -> INSERT INTO supplier (nama_pemasok, alamat_pemasok, nama_barang, jumlah_barang)
  -> VALUES (nama_pemasok, alamat_pemasok, nama_barang, jumlah_barang);
  -> END //
Query OK, 0 rows affected (0.004 sec)
```

```
MySQL [procedure]> SELECT*FROM supplier //
+-----+-----+-----+-----+-----+
| ID_pemasok | nama_pemasok | alamat_pemasok | nama_barang | jumlah_barang |
+-----+-----+-----+-----+-----+
|          1 | Udin         | Makau          | Splash      |             20 |
|       165499 | Ucup         | Jakarta        | ChocoBar    |             50 |
+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)
```

Kode di atas adalah pembuatan sebuah prosedur MySQL bernama `tambah\_supplier`, yang bertujuan untuk menambahkan informasi pemasok baru ke dalam tabel `supplier`. Prosesnya dimulai dengan mendefinisikan empat parameter masukan: `nama\_pemasok`, `alamat\_pemasok`, `nama\_barang`, dan `jumlah\_barang`. Prosedur ini kemudian melakukan penambahan data ke dalam tabel `supplier` menggunakan pernyataan `INSERT INTO`, dengan nilai parameter yang diberikan. Tujuan utama dari prosedur ini adalah untuk menyederhanakan proses penambahan data pemasok baru ke dalam database, dengan menyediakan antarmuka yang jelas dan mudah digunakan bagi pengguna. Setelah prosedur dibuat, contoh penggunaannya ditunjukkan dengan memanggil prosedur tersebut dan menampilkan hasilnya melalui perintah `SELECT \* FROM supplier`.

4. Buat sebuah prosedur dimana hasilnya akan memunculkan sebuah tabel supplier yang berisikan data dari nama supplier yang dicari

```
MySQL [procedure]> DELIMITER //
MySQL [procedure]> CREATE PROCEDURE mencariSupplier(IN nama VARCHAR(100))
-> DETERMINISTIC
-> BEGIN
-> SELECT * FROM supplier WHERE nama_pemasok = nama COLLATE utf8mb4_general_ci;
-> END //
Query OK, 0 rows affected (0.021 sec)
```

```
MySQL [procedure]> call mencariSupplier('Udin') //
+-----+-----+-----+-----+-----+
| ID_pemasok | nama_pemasok | alamat_pemasok | nama_barang | jumlah_barang |
+-----+-----+-----+-----+-----+
|          1 | Udin         | Makau          | Splash      |             20 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.010 sec)
```

Program di atas adalah sebuah prosedur tersimpan (stored procedure) dalam MySQL yang disebut "mencariSupplier". Prosedur ini menerima satu parameter input berupa nama pemasok (nama), kemudian menjalankan sebuah query SELECT untuk mencari semua informasi pemasok dari tabel "supplier" yang memiliki nilai "nama\_pemasok" sama dengan nilai parameter input tersebut. Hasil pencarian tersebut kemudian ditampilkan dalam bentuk tabel yang menampilkan ID pemasok, nama pemasok, alamat pemasok, nama barang, dan jumlah barang yang dipasok oleh pemasok yang memiliki nama sesuai dengan parameter input. Penggunaan COLLATE utf8mb4\_general\_ci digunakan untuk menetapkan pengurutan dan perbandingan teks dalam pencarian sebagai case-insensitive.