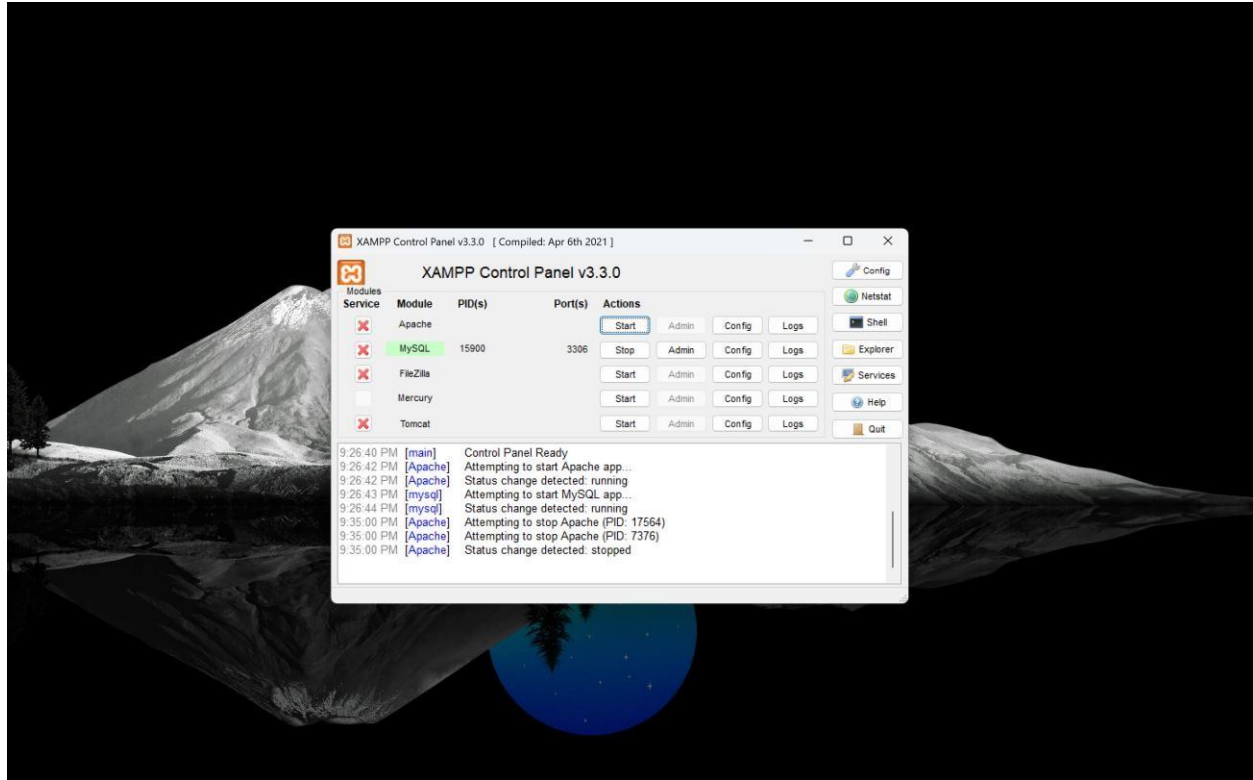


Nama : Firdaus Syafiq

Universitas : Universitas Amikom Purwokerto

## TUGAS PERTEMUAN 4

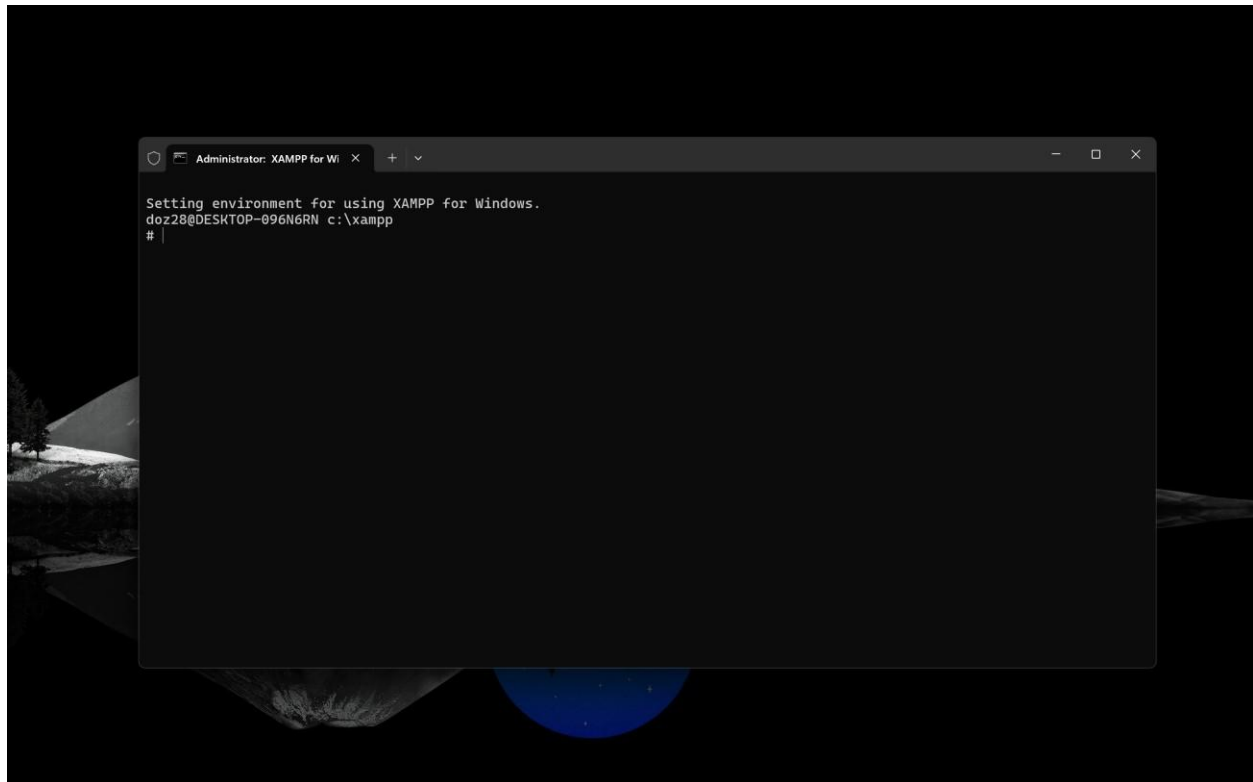
1. Buka XAMPP kemudian aktifkan MySQL



2. Kemudian tekan Shell, maka tampilannya sebagai berikut.

Nama : Firdaus Syafiq

Universitas : Universitas Amikom Purwokerto



3. Ketikkan `mysql -u root -p`

```
Setting environment for using XAMPP for Windows.
doz28@DESKTOP-096N6RN c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

4. Pilih database yang akan digunakan

```
MariaDB [(none)]> USE db_penjualan_buku
Database changed
```

5. Implementasi perintah SUB QUERY

Nama : Firdaus Syafiq

Universitas : Universitas Amikom Purwokerto

```
MariaDB [db_penjualan_buku]> SELECT title FROM books WHERE author_id = (  
-> SELECT id  
-> FROM authors  
-> WHERE name = 'Leo Tolstoy'  
-> );  
+-----+  
| title |  
+-----+  
| War and Peace |  
+-----+  
1 row in set (0.008 sec)
```

Menampilkan semua judul buku (title) dari tabel books yang ditulis oleh author bernama 'Leo Tolstoy'. Subquery mencari id dari penulis di tabel authors, lalu digunakan di kondisi WHERE utama.

#### 6. Implementasi perintah DISTINCT

```
MariaDB [db_penjualan_buku]> SELECT DISTINCT genre_id FROM books;  
+-----+  
| genre_id |  
+-----+  
| 1 |  
| 2 |  
| 4 |  
+-----+  
3 rows in set (0.001 sec)
```

Mengambil semua genre\_id yang unik dari tabel books. Artinya, menghilangkan duplikat genre sehingga setiap genre hanya muncul sekali di hasil.

#### 7. Implementasi perintah CASE

```
MariaDB [db_penjualan_buku]> SELECT id, total_amount,  
-> CASE  
-> WHEN total_amount <= 50000 THEN 'Kecil'  
-> WHEN total_amount BETWEEN 50000 AND 100000 THEN 'Sedang'  
-> ELSE 'Besar'  
-> END AS kategori_transaksi  
-> FROM transactions;  
+-----+-----+-----+  
| id | total_amount | kategori_transaksi |  
+-----+-----+-----+  
| 1 | 100000.00 | Sedang |  
| 2 | 75000.00 | Sedang |  
| 3 | 120000.00 | Besar |  
| 4 | 150000.00 | Besar |  
| 5 | 50000.00 | Kecil |  
+-----+-----+-----+  
5 rows in set (0.000 sec)
```

Menampilkan data dari tabel transactions dengan tambahan kolom baru kategori\_transaksi. kategori\_transaksi akan berisi:

Nama : Firdaus Syafiq

Universitas : Universitas Amikom Purwokerto

- "Kecil" jika total\_amount kurang dari atau sama dengan 50.000
- "Sedang" jika antara 50.000 sampai 100.000
- "Besar" jika lebih dari 100.000

#### 8. Implementasi perintah GROUP BY

```
MariaDB [db_penjualan_buku]> SELECT genre_id, COUNT(*) AS total_books FROM books GROUP BY genre_id;
```

| genre_id | total_books |
|----------|-------------|
| 1        | 3           |
| 2        | 1           |
| 4        | 1           |

```
3 rows in set (0.008 sec)
```

Mengelompokkan data buku berdasarkan genre\_id lalu menghitung jumlah buku di setiap genre. Hasilnya: genre apa saja beserta berapa banyak buku yang termasuk genre itu.

#### 9. Implementasi perintah HAVING

```
MariaDB [db_penjualan_buku]> SELECT genre_id, COUNT(*) AS total_books  
-> FROM books  
-> GROUP BY genre_id  
-> HAVING COUNT(*) > 2;
```

| genre_id | total_books |
|----------|-------------|
| 1        | 3           |

```
1 row in set (0.001 sec)
```

Hampir sama seperti perintah Group By sebelumnya, tetapi sekarang hanya menampilkan genre yang memiliki lebih dari 2 buku. HAVING digunakan untuk memfilter hasil setelah pengelompokan (GROUP BY).