

Laporan Praktikum 6

Konsep Basis Data

Operasi JOIN



Nama : Taufan Ali
NIM : 2215016135
Kelas Praktikum : C

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI TERAPAN
UNIVERSITAS AHMAD DAHLAN
YOGYAKARTA
2023

A. Kompetensi Dasar :

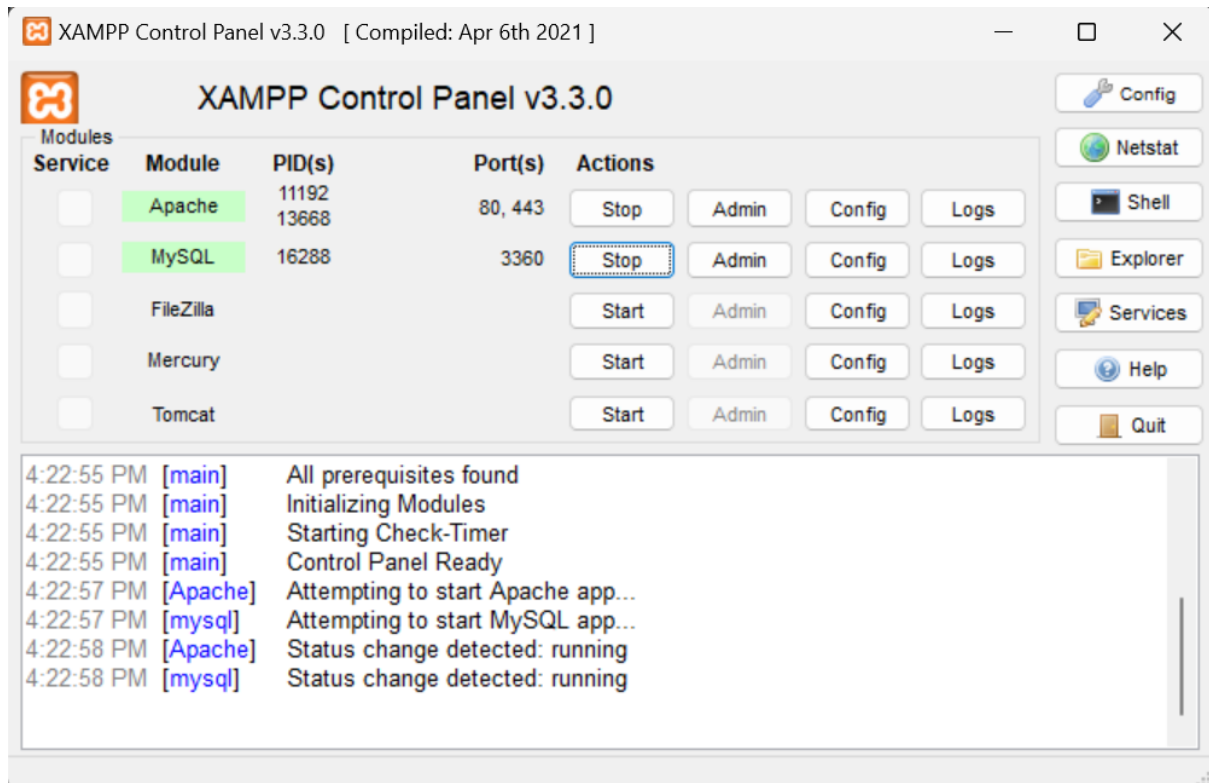
- a. Mahasiswa dapat membuat relasi antar table
- b. Mahasiswa dapat menggunakan Operasi JOIN

B. Teori Singkat :

Operator JOIN digunakan untuk merelasikan data yang ada pada beberapa table. Operator JOIN sebenarnya merupakan sebuah teknik mempersingkat query yang sangat panjang. Dengan menggunakan operator JOIN kita dapat menyederhanakan query yang ada menjadi relative lebih pendek dan mudah dipahami.

C. Percobaan :

1. Sebelum membuat database, pastikan terlebih dahulu bahwa mysql dan apache pada xampp sudah menyala



2. Masuk ke folder bin yang ada pada folder mysql yang berada pada folder xampp

```
Lenovo Ideapad@AlienzWindows C: > cd xampp/mysql/bin
Lenovo Ideapad@AlienzWindows C: > xampp > mysql > bin
```

3. Login ke mysql dengan menggunakan perintah **mysql -u username_anda**(default=root) atau **mysql -u root -p**(jika menggunakan password)

```
Lenovo Ideapad@AlienzWindows C: > xampp > mysql > bin mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> |
```

4. Gunakan tabel digilabs yang telah dibuat pada saat praktikum 2

```
MariaDB [(none)]> use digilabs;
Database changed
MariaDB [digilabs]> |
```

5. Pola sederhana menggunakan klausa ON : Digunakan untuk menggabungkan 2 tabel yang memiliki isi column yang sama pada 1 database yang sama.

```
MariaDB [digilabs]> select * from mahasiswa join prodi on mahasiswa.kode_prodi = prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P04	Sastra Inggris	F04

```
6 rows in set (0.001 sec)
```

6. JOIN dengan klausa USING, sebagai alternatif pengganti klausa ON : Sama seperti join on, cuman bedanya di syntax..

```
MariaDB [digilabs]> select * from mahasiswa join prodi using(kode_prodi);
```

kode_prodi	nim	Nama	Nama_prodi	kode_fak
P06	01006001	Andrian	Psikologi	F06
P03	02003001	Fajar Yuda	Teknik Industri	F03
P05	03005001	Tanakung	Hukum	F05
P01	04001001	Andi Herwanto	Ilmu Komputer	F01
P02	04002001	Didik Cahyono	Ekonomi	F02
P04	05004001	Yanti	Sastra Inggris	F04

```
6 rows in set (0.001 sec)
```

7. Menggunakan CROSS JOIN : Mirip seperti join on, bedanya pada cross join menggunakan WHERE sebagai kunci untuk penyatuan 2 tabel yang memiliki column dengan isi yang sama.

```
MariaDB [digilabs]> SELECT * FROM mahasiswa CROSS JOIN prodi WHERE
→ mahasiswa.kode_prodi = prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P04	Sastra Inggris	F04

```
6 rows in set (0.001 sec)
```

8. Menggunakan INNER JOIN : Mengembalikan baris yang memiliki nilai yang sesuai di kedua tabel yang digunakan

```
MariaDB [digilabs]> SELECT * FROM mahasiswa INNER JOIN prodi ON mahasiswa.kode_prodi =  
→ prodi.kode_prodi ;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P04	Sastra Inggris	F04

6 rows in set (0.001 sec)

9. Menggunakan RIGHT [OUTER] JOIN : Mengembalikan semua baris dari tabel yang ada di sebelah kanan, dan baris yang sesuai dari tabel di sebelah kanan, masing-masing.

```
MariaDB [digilabs]> SELECT * FROM mahasiswa RIGHT JOIN prodi ON mahasiswa.kode_prodi =  
→ prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
05004001	Yanti	P04	P04	Sastra Inggris	F04
03005001	Tanakung	P05	P05	Hukum	F05
01006001	Andrian	P06	P06	Psikologi	F06

6 rows in set (0.007 sec)

```
MariaDB [digilabs]> SELECT * FROM mahasiswa RIGHT OUTER JOIN prodi ON  
→ mahasiswa.kode_prodi = prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
05004001	Yanti	P04	P04	Sastra Inggris	F04
03005001	Tanakung	P05	P05	Hukum	F05
01006001	Andrian	P06	P06	Psikologi	F06

6 rows in set (0.001 sec)

10. Menggunakan LEFT [OUTER] JOIN : mengembalikan semua baris dari tabel yang ada di sebelah kiri, dan baris yang sesuai dari tabel di sebelah kiri, masing-masing.

```
MariaDB [digilabs]> SELECT * FROM mahasiswa LEFT JOIN prodi ON mahasiswa.kode_prodi =  
→ prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P04	Sastra Inggris	F04

6 rows in set (0.001 sec)

```
MariaDB [digilabs]> SELECT * FROM mahasiswa LEFT OUTER JOIN prodi ON mahasiswa.kode_prodi = prodi.kode_prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P04	Sastra Inggris	F04

6 rows in set (0.001 sec)

11. Menggunakan STRAIGHT JOIN : Dengan menggunakan straight join maka akan memaksa data pada tabel mahasiswa menyatu dengan data pada tabel prodi. Sehingga setiap data pada tabel prodi akan memiliki data dari tabel mahasiswa. Bisa dilihat dari contoh di bawah, setiap data pada column kode_prodi memiliki data yang sama sejumlah dengan row pada tabel mahasiswa.

```
MariaDB [digilabs]> SELECT * FROM mahasiswa STRAIGHT JOIN prodi;
```

nim	Nama	kode_prodi	kode_prodi	Nama_prodi	kode_fak
01006001	Andrian	P06	P01	Ilmu Komputer	F01
02003001	Fajar Yuda	P03	P01	Ilmu Komputer	F01
03005001	Tanakung	P05	P01	Ilmu Komputer	F01
04001001	Andi Herwanto	P01	P01	Ilmu Komputer	F01
04002001	Didik Cahyono	P02	P01	Ilmu Komputer	F01
05004001	Yanti	P04	P01	Ilmu Komputer	F01
01006001	Andrian	P06	P02	Ekonomi	F02
02003001	Fajar Yuda	P03	P02	Ekonomi	F02
03005001	Tanakung	P05	P02	Ekonomi	F02
04001001	Andi Herwanto	P01	P02	Ekonomi	F02
04002001	Didik Cahyono	P02	P02	Ekonomi	F02
05004001	Yanti	P04	P02	Ekonomi	F02
01006001	Andrian	P06	P03	Teknik Industri	F03
02003001	Fajar Yuda	P03	P03	Teknik Industri	F03
03005001	Tanakung	P05	P03	Teknik Industri	F03
04001001	Andi Herwanto	P01	P03	Teknik Industri	F03
04002001	Didik Cahyono	P02	P03	Teknik Industri	F03
05004001	Yanti	P04	P03	Teknik Industri	F03
01006001	Andrian	P06	P04	Sastra Inggris	F04
02003001	Fajar Yuda	P03	P04	Sastra Inggris	F04
03005001	Tanakung	P05	P04	Sastra Inggris	F04
04001001	Andi Herwanto	P01	P04	Sastra Inggris	F04
04002001	Didik Cahyono	P02	P04	Sastra Inggris	F04
05004001	Yanti	P04	P04	Sastra Inggris	F04
01006001	Andrian	P06	P05	Hukum	F05
02003001	Fajar Yuda	P03	P05	Hukum	F05
03005001	Tanakung	P05	P05	Hukum	F05
04001001	Andi Herwanto	P01	P05	Hukum	F05
04002001	Didik Cahyono	P02	P05	Hukum	F05
05004001	Yanti	P04	P05	Hukum	F05
01006001	Andrian	P06	P06	Psikologi	F06
02003001	Fajar Yuda	P03	P06	Psikologi	F06
03005001	Tanakung	P05	P06	Psikologi	F06
04001001	Andi Herwanto	P01	P06	Psikologi	F06
04002001	Didik Cahyono	P02	P06	Psikologi	F06
05004001	Yanti	P04	P06	Psikologi	F06

36 rows in set (0.001 sec)

12. Menggunakan NATURAL JOIN adalah jenis operasi JOIN di SQL yang otomatis menggabungkan baris berdasarkan kolom-kolom yang memiliki nama yang sama di kedua tabel yang dihubungkan.

```
MariaDB [digilabs]> SELECT * FROM mahasiswa NATURAL JOIN prodi ;
```

kode_prodi	nim	Nama	Nama_prodi	kode_fak
P06	01006001	Andrian	Psikologi	F06
P03	02003001	Fajar Yuda	Teknik Industri	F03
P05	03005001	Tanakung	Hukum	F05
P01	04001001	Andi Herwanto	Ilmu Komputer	F01
P02	04002001	Didik Cahyono	Ekonomi	F02
P04	05004001	Yanti	Sastra Inggris	F04

6 rows in set (0.001 sec)