

Nama : Pasha Ari Oktavian Ramadhan - 2024071019

Nadia - 202471004

Mata Kuliah : Basis Data

Latihan Week 9

1. Pembuatan Database, Table & Insert Data Dummy

```
CREATE DATABASE praktikum_1;

USE praktikum_1;

CREATE TABLE jurusan (
    kode_jurusan VARCHAR(5) PRIMARY KEY NOT NULL,
    nama_jurusan VARCHAR(50) UNIQUE NOT NULL,
    ketua_jurusan VARCHAR(100) NOT NULL
);

CREATE TABLE biodata (
    no_mahasiswa INT PRIMARY KEY ,
    kode_jurusan VARCHAR(5),
    nama_mahasiswa VARCHAR(100) NOT NULL,
    alamat TEXT DEFAULT NULL,
    ipk DECIMAL (3,2) DEFAULT NULL,
    FOREIGN KEY (kode_jurusan) REFERENCES jurusan(kode_jurusan)
);

INSERT INTO jurusan (kode_jurusan, nama_jurusan, ketua_jurusan) VALUES
    ('KD01', 'Sistem Informasi', 'Hamaningrum,S.Si'),
    ('KD02', 'Teknik Informatika', 'EnnySela,S.Kom,M.Kom'),
    ('KD03', 'Teknik Komputer', 'Berta Bednar,S.Si,M.T.');

INSERT INTO biodata (no_mahasiswa, kode_jurusan, nama_mahasiswa, alamat, ipk)
VALUES
    ('210089', 'KD01', 'Rina Gunawan', 'Denpasar', '3'),
    ('210090', 'KD03', 'Gani Suprapto', 'Singaraja', '3.5'),
    ('210012', 'KD02', 'Alexandra', 'Nusa Dua', '3'),
    ('210099', 'KD02', 'Nadine', 'Ganyar', '3.2'),
    ('210002', 'KD01', 'Rizal Samurai', 'Denpasar', '3.7');
```

2. Insert Data Baru Dengan Kode KD04

```
INSERT INTO biodata (no_mahasiswa, kode_jurusan, nama_mahasiswa, alamat, ipk)
VALUES
('210077', 'KD04', 'Michael Yaps', 'Tangerang Selatan', '3.7');
```

Hasilnya Error Karena Tidak Ada Kode Jurusan KD04 Pada Tabel Jurusan

3. Update Table

```
UPDATE biodata SET nama_mahasiswa = 'Rina Gunawan Astuti'
WHERE nama_mahasiswa = 'Rina Gunawan';
```

```
UPDATE jurusan SET kode_jurusan = 'KM01'
WHERE kode_jurusan = 'KD01';
```

```
UPDATE biodata SET no_mahasiswa = '210098'
WHERE no_mahasiswa = '210089';
```

```
UPDATE biodata SET ipk = 3.3
WHERE ipk = 3.0;
```

```
UPDATE biodata SET kode_jurusan = 'KD05'
WHERE kode_jurusan = 'KD03';
```

4. Cross Join

```
SELECT biodata.no_mahasiswa, biodata.nama_mahasiswa, jurusan.kode_jurusan,
jurusan.nama_jurusan
FROM biodata CROSS JOIN jurusan;
```

Hasil Query :

no_mahasiswa	nama_mahasiswa	kode_jurusan	nama_jurusan
210002	Rizal Samurai	KD01	Sistem Informasi
210002	Rizal Samurai	KD02	Teknik Informatika
210002	Rizal Samurai	KD03	Teknik Komputer
210012	Alexandra	KD01	Sistem Informasi
210012	Alexandra	KD02	Teknik Informatika
210012	Alexandra	KD03	Teknik Komputer
210089	Rina Gunawan Astuti	KD01	Sistem Informasi
210089	Rina Gunawan Astuti	KD02	Teknik Informatika
210089	Rina Gunawan Astuti	KD03	Teknik Komputer
210090	Gani Suprapto	KD01	Sistem Informasi
210090	Gani Suprapto	KD02	Teknik Informatika
210090	Gani Suprapto	KD03	Teknik Komputer
210099	Nadine	KD01	Sistem Informasi
210099	Nadine	KD02	Teknik Informatika
210099	Nadine	KD03	Teknik Komputer

5. Inner Join

```
SELECT b.no_mahasiswa, b.nama_mahasiswa, j.kode_jurusan, j.nama_jurusan FROM biodata b INNER JOIN jurusan j ON b.kode_jurusan = j.kode_jurusan;
```

Hasil Query :

no_mahasiswa	nama_mahasiswa	kode_jurusan	nama_jurusan
210002	Rizal Samurai	KD01	Sistem Informasi
210012	Alexandra	KD02	Teknik Informatika
210089	Rina Gunawan Astuti	KD01	Sistem Informasi
210090	Gani Suprapto	KD03	Teknik Komputer
210099	Nadine	KD02	Teknik Informatika

6. Left Join

```
SELECT b.no_mahasiswa, b.nama_mahasiswa, j.kode_jurusan, j.nama_jurusan FROM biodata b LEFT JOIN jurusan j ON b.kode_jurusan = j.kode_jurusan;
```

Hasil Query :

no_mahasiswa	nama_mahasiswa	kode_jurusan	nama_jurusan
210002	Rizal Samurai	KD01	Sistem Informasi
210012	Alexandra	KD02	Teknik Informatika
210089	Rina Gunawan Astuti	KD01	Sistem Informasi
210090	Gani Suprapto	KD03	Teknik Komputer
210099	Nadine	KD02	Teknik Informatika

7. Right Join

```
SELECT b.no_mahasiswa, b.nama_mahasiswa, j.kode_jurusan, j.nama_jurusan FROM biodata b RIGHT JOIN jurusan j ON b.kode_jurusan = j.kode_jurusan;
```

Hasil Query :

no_mahasiswa	nama_mahasiswa	kode_jurusan	nama_jurusan
210002	Rizal Samurai	KD01	Sistem Informasi
210012	Alexandra	KD02	Teknik Informatika
210089	Rina Gunawan Astuti	KD01	Sistem Informasi
210090	Gani Suprapto	KD03	Teknik Komputer
210099	Nadine	KD02	Teknik Informatika

8. Lower

```
SELECT nama_mahasiswa, LOWER(nama_mahasiswa) AS NAMA_KECIL FROM biodata;
```

Hasil Query :

nama_mahasiswa	NAMA_KECIL
Rizal Samurai	rizal samurai
Alexandra	alexandra
Rina Gunawan Astuti	rina gunawan astuti
Gani Suprapto	gani suprapto
Nadine	nadine

9. Upper

```
SELECT nama_mahasiswa, UPPER(nama_mahasiswa) AS NAMA_BESAR FROM biodata;
```

Hasil Query :

nama_mahasiswa	NAMA_BESAR
Rizal Samurai	RIZAL SAMURAI
Alexandra	ALEXANDRA
Rina Gunawan Astuti	RINA GUNAWAN ASTUTI
Gani Suprapto	GANI SUPRAPTO
Nadine	NADINE

10. SubString

```
SELECT nama_mahasiswa, SUBSTRING(nama_mahasiswa, 1, 4) AS NAMA_DEPAN FROM biodata;
```

Hasil Query :

nama_mahasiswa	NAMA_DEPAN
Rizal Samurai	Riza
Alexandra	Alex
Rina Gunawan Astuti	Rina
Gani Suprapto	Gani
Nadine	Nadi

11. Ltrim

```
SELECT LTRIM(' Informatika') AS TANPA_SPASI_KIRI;
```

Hasil Query :

TANPA_SPASI_KIRI
Informatika

12. Rtrim

```
SELECT RTRIM('Akuntansi ') AS TANPA_SPASI_KANAN;
```

Hasil Query :

TANPA_SPASI_KANAN
Akuntansi

13. Right

```
SELECT nama_mahasiswa, RIGHT(nama_mahasiswa, 5) AS HURUF_KANAN FROM biodata;
```

Hasil Query :

nama_mahasiswa	HURUF_KANAN
Rizal Samurai	murai
Alexandra	andra
Rina Gunawan Astuti	stuti
Gani Suprapto	raptō
Nadine	adine

14. Left

```
SELECT nama_mahasiswa, LEFT(nama_mahasiswa, 4) AS HURUF_KIRI FROM biodata;
```

Hasil Query :

nama_mahasiswa	HURUF_KIRI
Rizal Samurai	Riza
Alexandra	Alex
Rina Gunawan Astuti	Rina
Gani Suprapto	Gani
Nadine	Nadi

15. Length

```
SELECT nama_mahasiswa, LENGTH(nama_mahasiswa) AS PANJANG_NAMA FROM biodata;
```

Hasil Query :

nama_mahasiswa	PANJANG_NAMA
Rizal Samurai	13
Alexandra	9
Rina Gunawan Astuti	19
Gani Suprapto	13
Nadine	6

16. Reverse

```
SELECT nama_mahasiswa, REVERSE(nama_mahasiswa) AS TERBALIK FROM biodata;
```

Hasil Query :

nama_mahasiswa	TERBALIK
Rizal Samurai	iarumaS laziR
Alexandra	ardnaxelA
Rina Gunawan Astuti	itutsA nawanuG aniR
Gani Suprapto	otparpuS inaG
Nadine	enidaN

17. Space

```
SELECT CONCAT('Rina', SPACE(3), 'Gunawan') AS NAMA_DENGAN_SPASI;
```

Hasil Query :

NAMA_DENGAN_SPASI

Rina Gunawan

18. Aggregate Function

```
SELECT AVG(ipk) AS RATA_RATA_IPK FROM biodata;  
SELECT COUNT(*) AS JUMLAH_MAHASISWA FROM biodata;  
SELECT SUM(ipk) AS TOTAL_IPK FROM biodata;  
SELECT MAX(ipk) AS IPK_TERTINGGI FROM biodata;  
SELECT MIN(ipk) AS IPK_TERENDAH FROM biodata;
```

19. Klausa Order By, Group By & Having

```
SELECT * FROM biodata ORDER BY ipk DESC;  
SELECT kode_jurusan, COUNT(*) AS JUMLAH_MHS FROM biodata GROUP BY  
kode_jurusan;  
SELECT kode_jurusan, COUNT(*) AS JUMLAH_MHS FROM biodata GROUP BY  
kode_jurusan HAVING COUNT(*) > 1;
```

20. Sub Query

```
SELECT nama_mahasiswa, ipk FROM biodata WHERE ipk > (SELECT AVG(ipk) FROM  
biodata);  
SELECT kode_jurusan, rata_ipk FROM ( SELECT kode_jurusan, AVG(ipk) AS rata_ipk FROM  
biodata GROUP BY kode_jurusan ) AS hasil ORDER BY rata_ipk DESC;  
SELECT nama_mahasiswa, ipk, (SELECT MAX(ipk) FROM biodata) AS IPK_TERTINGGI  
FROM biodata;
```

21. Set Operation

```
SELECT nama_mahasiswa, kode_jurusan FROM biodata WHERE kode_jurusan = 'KD01'  
UNION SELECT nama_mahasiswa, kode_jurusan FROM biodata WHERE kode_jurusan =  
'KD02';  
SELECT nama_mahasiswa, kode_jurusan FROM biodata WHERE kode_jurusan = 'KD01'  
AND nama_mahasiswa IN ( SELECT nama_mahasiswa FROM biodata WHERE  
kode_jurusan = 'KD02' );  
SELECT nama_mahasiswa, kode_jurusan FROM biodata WHERE kode_jurusan = 'KD01'  
AND nama_mahasiswa NOT IN ( SELECT nama_mahasiswa FROM biodata WHERE  
kode_jurusan = 'KD02' );
```

22. View

```

CREATE VIEW v_mahasiswa_jurusan AS SELECT b.no_mahasiswa, b.nama_mahasiswa,
j.nama_jurusan, b.ipk FROM biodata b JOIN jurusan j ON b.kode_jurusan =
j.kode_jurusan;

SELECT * FROM v_mahasiswa_jurusan;

```

23. Control Flow & Temporary Table

```

SELECT nama_mahasiswa, ipk, IF(ipk >= 3.5, 'Lulus dengan Pujian', 'Lulus') AS STATUS
FROM biodata;

SELECT nama_mahasiswa, ipk, CASE WHEN ipk >= 3.5 THEN 'Sangat Baik' WHEN ipk >=
3.0 THEN 'Baik' WHEN ipk >= 2.5 THEN 'Cukup' ELSE 'Kurang' END AS KATEGORI_IPK
FROM biodata;

SELECT nama_mahasiswa, IFNULL(ipk, 0) AS IPK_FIX FROM biodata;

// Temporary Table

SELECT nama_mahasiswa, ipk INTO OUTFILE 'C:/laragon/www/tmp_mahasiswa.csv'
FROM biodata WHERE ipk >= 3.0;

```

24. Stored Procedure (Ketua Jurusan)

```

DELIMITER $$ CREATE PROCEDURE tampilanMahasiswaByKetua( IN nama_ketua
VARCHAR(50) ) BEGIN SELECT b.no_mahasiswa, b.nama_mahasiswa, j.nama_jurusan,
j.ketua_jurusan, b.ipk FROM biodata b JOIN jurusan j ON b.kode_jurusan =
j.kode_jurusan WHERE j.ketua_jurusan = nama_ketua; END$$ DELIMITER ;

```

25. Stored Procedure (Nama)

```

DELIMITER $$ CREATE PROCEDURE cariKetuaJurusan( IN nama_cari VARCHAR(50) )
BEGIN SELECT * FROM jurusan WHERE ketua_jurusan LIKE CONCAT('%', nama_cari, '%');
END$$ DELIMITER ;

```

26. Trigger

```

CREATE TABLE log_aktivitas ( id_log INT AUTO_INCREMENT PRIMARY KEY, aksi
VARCHAR(50), nama_mahasiswa VARCHAR(50), waktu DATETIME );

DELIMITER $$ CREATE TRIGGER after_insert_biodata AFTER INSERT ON biodata FOR
EACH ROW BEGIN INSERT INTO log_aktivitas (aksi, nama_mahasiswa, waktu) VALUES
('TAMBAH DATA', NEW.nama_mahasiswa, NOW()); END$$ DELIMITER ;

```