

LAPORAN RESMI
MODUL IV
SQL
SISTEM MANAJEMEN BASIS DATA



NAMA	: ALIYUL RIDHO
N.R.P	: 230441100135
DOSEN	: FITRI DAMAYANTI, S.Kom., M.Kom.
ASISTEN	: ABDUL JABBAR RAMADHANI
TGL PRAKTIKUM	: 19 April 2025

Disetujui : 22 April 2025
Asisten

Abdul Jabbar Ramadhani
21.04.411.00062



LABORATORIUM TEKNOLOGI INFORMASI
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengelolaan basis data, pemahaman terhadap jenis-jenis perintah SQL menjadi kunci utama dalam membangun sistem informasi yang efisien, aman, dan terstruktur. SQL (Structured Query Language) merupakan bahasa standar yang digunakan untuk mengelola dan memanipulasi data dalam sistem manajemen basis data (SMBD). Secara umum, perintah SQL terbagi menjadi beberapa kategori utama, yaitu Data Definition Language (DDL), Data Manipulation Language (DML), dan Data Control Language (DCL).

DDL (Data Definition Language) digunakan untuk mendefinisikan struktur basis data, seperti membuat, mengubah, atau menghapus tabel dan objek lainnya. Perintah DDL seperti `CREATE`, `ALTER`, dan `DROP` memungkinkan pengembang untuk membangun dan mengatur skema basis data secara sistematis.

DML (Data Manipulation Language) berfungsi untuk mengelola data yang tersimpan dalam tabel, termasuk menambahkan (`INSERT`), memperbarui (`UPDATE`), menghapus (`DELETE`), dan menampilkan data (`SELECT`). DML sangat penting dalam proses pengolahan data yang menjadi inti dari sistem informasi.

DCL (Data Control Language) digunakan untuk mengatur hak akses dan kontrol keamanan terhadap data dalam basis data. Perintah seperti `GRANT` dan `REVOKE` memungkinkan administrator basis data untuk mengatur siapa saja yang memiliki izin untuk mengakses atau memodifikasi data tertentu.

Pemahaman dan penerapan DDL, DML, dan DCL sangat penting dalam praktikum maupun pengembangan sistem informasi secara nyata. Dengan menguasai ketiga kategori perintah SQL ini, mahasiswa dapat membangun sistem basis data yang tidak hanya berfungsi dengan baik, tetapi juga terstruktur, aman, dan mudah dikelola dalam jangka panjang.

1.2 Tujuan

- Mampu memahami konsep dan membuat perintah SQL dalam berbagai kondisi
- Mampu memahami penerapan *JOIN* dalam tabel database

BAB II

DASAR TEORI

2.1 *SQL*

Secara umum perintah-perintah yang terdapat di dalam SQL, diklasifikasikan menjadi tiga bagian, antara lain yaitu :

1. DDL (*Data Definition Language*)

Merupakan perintah SQL yang berkaitan dengan pendefinisian suatu struktur database, dalam hal ini database dan *table*.

Perintah DDL adalah: *CREATE, ALTER, RENAME, DROP*

2. DML (*Data Manipulation Language*)

Merupakan perintah SQL yang berkaitan dengan manipulasi atau pengolahan data atau record dalam *table*.

Perintah DML antara lain: *SELECT, INSERT, UPDATE, DELETE*

3. DCL (*Data Control Language*)

Merupakan perintah SQL yang berkaitan dengan manipulasi user dan hak akses (*priviledges*).

Perintah SQL yang termasuk dalam DCL antara lain: *GRANT, REVOKE*.

2.2 *SELECT*

SELECT merupakan salah satu pondasi dalam SQL Programming. *SELECT* digunakan untuk menampilkan data, terlebih untuk mencari informasi dalam kumpulan data. Jenis – jenis stored Procedure

2.2.1 *Syntax*

SELECT dibagi kedalam 6 komponen, antara lain:

- a) *SELECT*. Diikuti oleh <*select_list*>, dapat berupa literal_value atau column_list atau asterisk (*).
- b) *FROM*. Diikuti oleh <*table_name*> sesuai dengan column_list. Jadi jika ada data yang diambil dari kolom tertentu, harus diketahui kolom tersebut diambil dari tabel mana. Tabel pada FROM dapat diikuti dengan alias untuk mempermudah penulisan khususnya ketika *join* dan subquery.
- c) *WHERE*. Diikuti oleh kondisi secara umum.

- d) *GROUP BY*. Diikuti oleh `<select_list>`. Bagian ini muncul ketika ada fungsi-fungsi agregasi.
- e) *HAVING*. Diikuti oleh kondisi hanya untuk fungsi -fungsi agregasi.
- f) *ORDER BY*. Diikuti oleh `<select_list>`

2.2.2 Menyaring data

Tidak semua data yang ada pada tabel, ingin ditampilkan. Terlebih ketika tabel terbagi kedalam banyak kolom dengan jumlah data yang sangat besar.

Operator pembandingan :

operator	keterangan
=	Sama dengan
>	Lebih besar
>=	Lebih besar sama dengan
<	Lebih kecil
<=	Lebih kecil sama dengan
< > Atau !=	Tidak sama dengan
BETWEEN ... AND..	Diantara 2 nilai
IN (set)	Cocok dengan salah satu diantara daftar nilai
LIKE	Cocok dengan pola karakter
IS NULL	Sama dengan NULL

Syntax nya :

```
Select * From Nama_Table Where Nama_Field [Operator
Relasional] Ketentuan;
```

2.2.3 Pengurutan data (ACS , DESC, *ORDER BY*)

Untuk mengurutkan tampilan data dari suatu *table*, digunakan klausa OrderBy. Klausa *Order by*, dapat digunakan untuk mengurutkan data :

- ◆ Asc (Ascending) : Untuk mengurutkan data dari kecil ke besar
- ◆ Desc (Descending) : Untuk mengurutkan data dari besar ke kecil

Perintahnya :

```
Select * From Nama_Table Order by Nama_Field_Key  
Asc/Desc;
```

Contoh :

```
Select * From products Order by prod_name Asc;
```

2.3 OPERATOR BETWEEN, IN, LIKE

2.3.1 Operator *Between*

Operator Between merupakan operator yang digunakan untuk menangani operasi jangkauan.

Syntaxnya:

```
Select * From Nama_Table Where Nama_Field_ketentuan  
Between 'Ketentuan_1' And 'Ketentuan_2';
```

2.3.2 Operator *IN*

Operator In merupakan operator yang digunakan untuk mencocokkan suatu nilai.

Perintahnya :

```
Select Nama_Field From Nama_Table Where  
Nama_Field_Pencocok In  
( 'Isi_Field_1', 'Isi_Field_2' );
```

2.3.3 Operator *LIKE*

Operator Like merupakan operator yang digunakan untuk mencari suatu data (search).

Peletakan % :

- a. %kata : Mencari nilai yang berakhir dengan “kata”,
- b. kata% : Mencari nilai yang diawali dengan “kata”.
- c. %kata% : Mencari nilai yang mengandung “kata” di bagian manapun
- d. k_ta : Gunakan _ (underscore) jika ingin mencocokkan satu karakter tunggal.

Perintahnya :

```
Select * From Nama_Table Where Nama_Field_Dicari  
Like '%Key';
```

2.3.4 Agregasi

Dalam pengelolaan basis data, agregasi merupakan proses untuk merangkum atau menyajikan data dalam bentuk informasi yang lebih ringkas dan bermakna. Operasi agregasi dilakukan dengan menggunakan fungsi-fungsi agregat dalam SQL, seperti *COUNT*, *SUM*, *AVG*, *MAX*, dan *MIN*. Fungsi-fungsi ini digunakan untuk menghitung jumlah data, menjumlahkan nilai numerik, mencari nilai rata-rata, serta menentukan nilai maksimum dan minimum dari suatu kolom dalam tabel. Agregasi sangat berguna dalam analisis data karena memungkinkan pengguna untuk memperoleh gambaran umum dari data yang besar, seperti menghitung total penjualan, jumlah pelanggan, atau rata-rata pembelian. Dalam praktiknya, fungsi agregasi sering digabungkan dengan klausa *GROUP BY* untuk mengelompokkan data berdasarkan kategori tertentu, serta *HAVING* untuk menyaring hasil agregasi berdasarkan kondisi tertentu. Pemahaman konsep agregasi sangat penting dalam pengembangan sistem informasi karena membantu dalam menyajikan laporan dan ringkasan data yang mendukung pengambilan keputusan.

2.3.5 JOIN

Dalam basis data relasional, *JOIN* merupakan operasi yang digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan kolom yang memiliki relasi. Konsep *JOIN* sangat penting karena dalam perancangan basis data yang baik, data biasanya disimpan secara terpisah dalam beberapa tabel yang saling terhubung melalui kunci (key). Dengan *JOIN*, data yang tersebar di berbagai tabel dapat ditampilkan secara menyatu untuk menghasilkan informasi yang utuh. Terdapat beberapa jenis *JOIN* yang umum digunakan, yaitu *INNER JOIN*, *LEFT JOIN*, *RIGHT JOIN*, dan *FULL OUTER JOIN*. *INNER JOIN* menampilkan hanya data yang cocok di kedua tabel, sedangkan *LEFT JOIN* dan *RIGHT JOIN* akan menampilkan seluruh data dari tabel kiri atau kanan, walaupun tidak ada pasangan yang cocok di tabel lainnya. *FULL OUTER JOIN* menampilkan semua data dari kedua tabel, baik yang memiliki pasangan maupun tidak. *JOIN* biasanya dilakukan

berdasarkan primary key dan foreign key untuk menjaga integritas data. Pemahaman terhadap *JOIN* sangat penting dalam praktik pengolahan data karena memungkinkan penggabungan informasi kompleks untuk keperluan analisis maupun laporan.

2.3.6 SubQuery

Subquery merupakan query didalam query. Umumnya, subquery ini dipakai untuk mencari data yang belum diketahui. Penggunaan query didalam query ini umumnya menjadi bagian dari kondisi.

Sintak subquery adalah sebagai berikut:

```
SELECT <select_list>
FROM   <tabel>
WHERE  <column> =
        (SELECT <single_column>
         FROM   <tabel>
         WHERE  <kondisi>);
```


BAB II TUGAS PERBATALUAN.

3.1 Soal

1. Jelaskan fungsi dari DDL, DML dan DCL. beserta dengan contoh penggunaan kode di dalam program!
2. Sebutkan operator perbandingan beserta contoh penggunaannya di dalam kode program! (minimal 3)
3. Jelaskan perbedaan klawisa pengurutan (ORDER BY, DESC, dan ASC) beserta contoh penggunaan dan hasilnya!
4. Jelaskan perbedaan left join, right join, dan self join dengan contoh penerapannya pada program!
5. Jelaskan apa yang dimaksud dengan subquery dan bagaimana cara penggunaannya?
6. Jelaskan prinsip join 3 tabel beserta dengan contohnya dalam program!

3.2 Jawaban.

1. - DDL (Data Definition Language) : mengatur struktur database
 contoh : `CREATE TABLE mahasiswa (Nim INT, Nama VARCHAR(50));`
 - DML (Data Manipulation Language) : mengelola data dalam tabel
 contoh : `INSERT INTO mahasiswa values (112, "Atidi");`
 - DCL (Data Control Language) : mengatur hak akses data
 contoh : `GRANT SELECT ON mahasiswa TO user1;`
2. • = (sama dengan) : `SELECT * FROM mahasiswa WHERE Nim = 112;`
 • > (lebih besar) = `SELECT * FROM nilai WHERE skor > 70;`
 • < (lebih kecil) = `SELECT * FROM barang WHERE harga < 1000;`
3. ORDER BY untuk mengurutkan data berdasarkan kolom, ASC mengurutkan secara naik (default), DESC mengurutkan secara menurun. contoh :
 • `SELECT * FROM hama ORDER BY nama ASC;` hasilnya urut nama A-Z
 • `SELECT * FROM hama ORDER BY berat DESC;` hasilnya urut berat dari berat ke ringan.
4. • Left join mengambil semua data dari tabel kiri meski tidak cocok dengan tabel kanan. contoh : `SELECT * FROM a LEFT JOIN B ON a.id = B.id;`
 • Right join mengambil semua data tabel kanan contoh : `SELECT * FROM a RIGHT JOIN B ON a.id = B.id;`
 • Self join, join tabel dengan dirinya sendiri, contoh :
`SELECT A.nama, B.nama FROM Gamer a, Gamer b where a.coach_id = B.id;`
5. subquery adalah query di dalam query lain, biasanya digunakan dalam klawisa WHERE, FROM atau SELECT. contoh
`SELECT nama FROM mahasiswa where nim in (select nim from nilai where skor > 80);`

6. Join 3 tabel menggabungkan dari tiga tabel yang saling berhubungan melalui key. contoh:

```
select a.nama, b.matakuliah, c.nilai from mahasiswa a  
join krs b on a.nim = b.nim join nilai c  
on b.id_krs = c.id_krs;
```

penjelasan: Query tersebut akan mengambil data nama mahasiswa, nama matakuliah, dan nilai dari tiga tabel yang saling berhubungan.

BAB IV

IMPLEMENTASI

4.1 Tugas Praktikum

4.1.1 Soal

Berdasarkan database yang telah kalian buat sebelumnya, buatlah :

1. Kolom keterangan di salah satu tabel pada bagian akhir tabel tersebut!
2. Gabungan 2 tabel yang memungkinkan dan memiliki fungsi pada penerapannya!
3. Urutan kolom pada setiap tabel menggunakan perintah *Order by*, DESC, dan ASC (minimal 1 untuk setiap perintah)
4. Perubahan pada salah satu tipe data yang dapat berguna
5. Kode *Left Join*, *Right Join* dan *Self Join* beserta dengan alur prosesnya
6. Kode yang mengandung operator perbandingan (Minimal 5)

4.2 Source Code

4.2.1 Tugas Praktikum

- a) Menambahkan kolom 'keterangan' di tabel *orders*

```
ALTER TABLE orders
ADD keterangan TEXT AFTER STATUS;
```

- b) Gabungan tabel *orders* dan *customers*

```
SELECT
    a.order_id,
    b.name AS nama_pelanggan,
    a.total_price,
    a.STATUS
FROM orders a
JOIN customers b ON a.customer_id = b.customer_id;
```

- c) Urutan data menggunakan *ORDER BY* DESC dan ASC

```
-- 3a. Urutkan produk berdasarkan harga dari
tertinggi (DESC)
SELECT * FROM products a
```

```

ORDER BY a.harga DESC;

-- 3b. Urutkan pelanggan berdasarkan nama (ASC)
SELECT * FROM customers a
ORDER BY a.NAME ASC;

-- 3c. Urutkan pesanan berdasarkan tanggal terbaru
(DESC)
SELECT * FROM orders a
ORDER BY a.order_date DESC;

```

d) Mengubah tipe data kolom *phone* di tabel *customers*

```

ALTER TABLE customers
MODIFY phone VARCHAR(20);
SELECT * FROM customers;

```

e) *JOIN* (LEFT *JOIN*, RIGHT *JOIN*, SELF *JOIN*)

```

-- 5a. LEFT JOIN: semua produk dan pemasok
SELECT
    a.NAME AS nama_produk,
    c.nama_pemasok
FROM products a
LEFT JOIN product_suppliers b ON a.id_produk =
b.id_produk
LEFT JOIN suppliers c ON b.id_pemasok =
c.id_pemasok;

-- 5b. RIGHT JOIN: semua pemasok dan produk yang
mereka suplai (jika ada)
SELECT
    c.nama_pemasok,
    a.NAME AS nama_produk

```

```

FROM products a
RIGHT JOIN product_suppliers b ON a.id_produk =
b.id_produk
RIGHT JOIN suppliers c ON b.id_pemasok =
c.id_pemasok;

SELECT * FROM suppliers;
SELECT * FROM product_suppliers;
INSERT INTO product_suppliers VALUES (1,4);
-- 5c. SELF JOIN: pelanggan dengan alamat yang
sama
SELECT
    a.NAME AS pelanggan_1,
    b.NAME AS pelanggan_2,
    a.address
FROM customers a
JOIN customers b ON a.address = b.address AND
a.customer_id <> b.customer_id;
SELECT * FROM customers;
INSERT INTO customers (NAME,address) VALUES ('ridho
slebew','Desa Konoha, Rumah Uchiha');

```

f) Operator perbandingan (>, <, <=, >=, <>)

```

-- 6a. Produk dengan harga lebih dari 100
SELECT * FROM products a WHERE a.harga > 100;

-- 6b. Produk dengan harga kurang dari atau sama
dengan 50
SELECT * FROM products a WHERE a.harga <= 50;

-- 6c. Pesanan dengan total harga lebih dari atau
sama dengan 100

```

```
SELECT * FROM orders a WHERE a.total_price >= 100;
```

-- 6d. Produk yang stok-nya tidak sama dengan 0

```
SELECT * FROM products a WHERE a.stok <> 0;
```

-- 6e. Pelanggan dengan ID kurang dari 5

```
SELECT * FROM customers a WHERE a.customer_id < 5;
```

4.3 Hasil

4.3.1 Tugas Praktikum

- Menambahkan kolom 'keterangan' di tabel *orders*

order_id	customer_id	order_date	total_price	status	keterangan
1	1	2025-03-31 19:03:45	99.99	Pending	pesanan sedang dikemas
2	2	2025-03-31 19:03:45	77.77	Shipped	pesanan sedang dalam tujuan
3	3	2025-03-31 19:03:45	200.00	Shipped	pesanan sudah samapi di alamat tujuan
4	4	2025-03-31 19:03:45	88.88	Canceled	(NULL)
5	5	2025-03-31 19:03:45	55.55	Pending	pesanan sedang dikemas
6	6	2025-03-31 19:03:45	10.10	Completed	pesanan sudah samapi di alamat tujuan
7	7	2025-03-31 19:03:45	150.00	Pending	pesanan sedang dikemas
8	8	2025-03-31 19:03:45	49.49	Shipped	pesanan sedang dalam tujuan
9	9	2025-03-31 19:03:45	120.00	Pending	pesanan sedang dikemas
10	10	2025-03-31 19:03:45	200.00	Completed	pesanan sudah samapi di alamat tujuan

- Gabungan tabel *orders* dan *customers*

order_id	nama_pelanggan	total_price	STATUS
1	Sasuke Uchiha	99.99	Pending
2	Megumin	77.77	Shipped
3	Levi Ackerman	200.00	Shipped
4	Gintoki Sakata	88.88	Canceled
5	Aqua	55.55	Pending
6	Zoro Roronoa	10.10	Completed
7	Senku Ishigami	150.00	Pending
8	Anya Forger	49.49	Shipped
9	Kaguya Shinomiya	120.00	Pending
10	Tanjiro Kamado	200.00	Completed

- Urutan data menggunakan *ORDER BY* DESC dan ASC

a) Urutkan produk berdasarkan harga dari tertinggi (DESC)

id_produk	name	kategori	harga	stok	deskripsi
1	Sharingan Bait	Fishing Lures	250000.00	10	Menarik perhatian ikan dengan Genjutsu!
9	Love Is War Net	Nets	250000.00	12	Strategi cerdas menangkap ikan, tapi hati-hati jangan jat...
10	Sun Breathing Line	Fishing Line	200.00	8	Matahari terbit! Jaring yang tak tertandingi.
3	Ackerman Hook	Hooks	129.99	15	Secepat refleksi Levi!
2	Explosion Bait	Fishing Lures	77.77	20	Bait ini hanya bisa digunakan sekali. EXPLOSION!
4	Zura Rod	Fishing Rods	70.00	25	Bukan Katsura, tapi pancing Zura!
5	Divine Water Bait	Fishing Lures	55.55	30	Diberkahi oleh Aqua, tapi jangan percaya airnya.
8	Elegant Hook	Hooks	49.49	35	Bait ini memiliki ELEGANCE, cocok untuk Anya.
6	Lost Navigator Compass	Accessories	10.10	50	Dijamin bikin kamu kehilangan arah seperti Zoro.

b) Urutkan pelanggan berdasarkan nama (ASC)

customer_id	name	email	phone	address	created_at
0	Anyo Forger	anya@elegant.com	08990123456	Rumah Keluarga Forger, Kota Berlint	2025-03-31 19:03:12
5	Aqua	aqua@goddess.com	08567890123	Dimensi Dewa, Tempat Reinkarnasi	2025-03-31 19:03:12
4	Gintoki Sakata	gintoki@yorozuya.com	08456789012	Kabukicho, Edo, Jepang	2025-03-31 19:03:12
9	Kaguya Shinomiya	kaguya@shuchiin.com	08901234567	Akademi Shuchiin, Ruang OSIS	2025-03-31 19:03:12
3	Levi Ackerman	levi@scoutinglegion.com	08345678901	Dinding Maria, Kamar Kapten Levi	2025-03-31 19:03:12
2	Megumin	megumin@explosion.com	08234567890	Belakang Guild Adventurer, Axel Town	2025-03-31 19:03:12
11	Ridho Slebew		(NULL)	Desa Konoha, Rumah Uchiha	2025-04-22 17:24:24
1	Sasuke Uchiha	sasuke@konoha.com	08123456789	Desa Konoha, Rumah Uchiha	2025-03-31 19:03:12
7	Senku Ishigami	senku@100billionpercent.com	08789012345	Stone World, Jepang Masa Batu	2025-03-31 19:03:12
10	Tanjiro Kamado	tanjiro@demoncorp.com	08012345678	Gunung Sagiri, Jepang	2025-03-31 19:03:12
6	Zoro Roronoa	zoro@strawhats.com	08678901234	??? (Dia sendiri juga tidak tahu)	2025-03-31 19:03:12

c) Urutkan pesanan berdasarkan tanggal terbaru (DESC)

order_id	customer_id	order_date	total_price	status	keterangan	
1		1 2025-03-31 19:03:45	99.99	Pending	pesanan sedang dikemas	22B
2		2 2025-03-31 19:03:45	77.77	Shipped	pesanan sedang dalam tujuan	27B
3		3 2025-03-31 19:03:45	200.00	Shipped	pesanan sudah samapi di alamat tujuan	37B
4		4 2025-03-31 19:03:45	88.88	Canceled	(NULL)	0K
5		5 2025-03-31 19:03:45	55.55	Pending	pesanan sedang dikemas	22B
6		6 2025-03-31 19:03:45	10.10	Completed	pesanan sudah samapi di alamat tujuan	37B
7		7 2025-03-31 19:03:45	150.00	Pending	pesanan sedang dikemas	22B
8		8 2025-03-31 19:03:45	49.49	Shipped	pesanan sedang dalam tujuan	27B
9		9 2025-03-31 19:03:45	120.00	Pending	pesanan sedang dikemas	22B
10		10 2025-03-31 19:03:45	200.00	Completed	pesanan sudah samapi di alamat tujuan	37B

- Mengubah tipe data kolom *phone* di tabel *customers*

customer_id	name	email	phone	address	created_at
1	Sasuke Uchiha	sasuke@konoha.com	08123456789	Desa Konoha, Rumah Uchiha	2025-03-31 19:03:12
2	Megumin	megumin@explosion.com	08234567890	Belakang Guild Adventurer, Axel Town	2025-03-31 19:03:12
3	Levi Ackerman	levi@scoutinglegion.com	08345678901	Dinding Maria, Kamar Kapten Levi	2025-03-31 19:03:12
4	Gintoki Sakata	gintoki@yorozuya.com	08456789012	Kabukicho, Edo, Jepang	2025-03-31 19:03:12
5	Aqua	aqua@goddess.com	08567890123	Dimensi Dewa, Tempat Reinkarnasi	2025-03-31 19:03:12
6	Zoro Roronoa	zoro@strawhats.com	08678901234	??? (Dia sendiri juga tidak tahu)	2025-03-31 19:03:12
7	Senku Ishigami	senku@100billionpercent.com	08789012345	Stone World, Jepang Masa Batu	2025-03-31 19:03:12
8	Anyo Forger	anya@elegant.com	08890123456	Rumah Keluarga Forger, Kota Berlint	2025-03-31 19:03:12
9	Kaguya Shinomiya	kaguya@shuchiin.com	08901234567	Akademi Shuchiin, Ruang OSIS	2025-03-31 19:03:12
10	Tanjiro Kamado	tanjiro@demoncorp.com	08012345678	Gunung Sagiri, Jepang	2025-03-31 19:03:12
11	Ridho Slebew		(NULL)	Desa Konoha, Rumah Uchiha	2025-04-22 17:24:24

- *JOIN* (*LEFT JOIN*, *RIGHT JOIN*, *SELF JOIN*)

a) *LEFT JOIN*: semua produk dan pemasok

nama_produk	nama_pemasok
Sharingan Bait	Uchiha Fishing Corp
Sharingan Bait	Yorozuya Fishing Tools
Explosion Bait	Explosion Supply Co.
Ackerman Hook	Ackerman Gear Works
Zura Rod	Yorozuya Fishing Tools
Divine Water Bait	Aqua Holy Water Inc.
Lost Navigator Compass	Lost Navigator Supplies
Elegant Hook	Elegant Gear
Love Is War Net	Shuchiin Fishing Tech
Sun Breathing Line	Demon Slayer Corp

b) *RIGHT JOIN*: semua pemasok dan produk yang mereka suplai (jika ada)

nama_pemasok	nama_produk
Uchiha Fishing Corp	Sharingan Bait
Explosion Supply Co.	Explosion Bait
Ackerman Gear Works	Ackerman Hook
Yorozuya Fishing Tools	Sharingan Bait
Yorozuya Fishing Tools	Zura Rod
Aqua Holy Water Inc.	Divine Water Bait
Lost Navigator Supplies	Lost Navigator Compass
Ishigami Science Lab	(NULL)
Elegant Gear	Elegant Hook
Shuchiin Fishing Tech	Love Is War Net
Demon Slayer Corp	Sun Breathing Line

c) SELF JOIN: pelanggan dengan alamat yang sama

pelanggan_1	pelanggan_2	address
ridho slebew	Sasuke Uchiha	Desa Konoha, Rumah Uchiha
Sasuke Uchiha	ridho slebew	Desa Konoha, Rumah Uchiha

- Operator perbandingan (>, <, <=, >=, <>)

a) Produk dengan harga lebih dari 100

id_produk	name	kategori	harga	stok	deskripsi	created_at
1	Sharingan Bait	Fishing Lures	250000.00	10	Menarik perhatian ikan dengan Genjutsu!	39B 2025-03-31
3	Ackerman Hook	Hooks	129.99	15	Secepat refleksi Levi!	21B 2025-03-31
9	Love Is War Net	Nets	250000.00	12	Strategi cerdas menangkap ikan, tapi hati-hati jangan jat...	66B 2025-03-31
10	Sun Breathing Line	Fishing Line	200.00	8	Matahari terbit! Jaring yang tak tertandingi.	45B 2025-03-31

b) Produk dengan harga kurang dari atau sama dengan 50

id_produk	name	kategori	harga	stok	deskripsi	created_at
6	Lost Navigator Compass	Accessories	10.10	50	Dijamin bikin kamu kehilangan arah seper...	48B 2025-03-31 19:03:07
8	Elegant Hook	Hooks	49.49	35	Bait ini memiliki ELEGANCE, cocok untuk ...	45B 2025-03-31 19:03:07

c) Pesanan dengan total harga lebih dari atau sama dengan 100

order_id	customer_id	order_date	total_price	status	keterangan	
3	3	2025-03-31 19:03:45	200.00	Shipped	pesanan sudah samapi di alamat tujuan	37B
7	7	2025-03-31 19:03:45	150.00	Pending	pesanan sedang dikemas	22B
9	9	2025-03-31 19:03:45	120.00	Pending	pesanan sedang dikemas	22B
10	10	2025-03-31 19:03:45	200.00	Completed	pesanan sudah samapi di alamat tujuan	37B

d) Produk yang stok-nya tidak sama dengan 0

id_produk	name	kategori	harga	stok	deskripsi	creat
1	Sharingan Bait	Fishing Lures	250000.00	10	Menarik perhatian ikan dengan Genjutsu!	39B 2025-
2	Explosion Bait	Fishing Lures	77.77	20	Bait ini hanya bisa digunakan sekali. EXPLOSION!	48B 2025-
3	Ackerman Hook	Hooks	129.99	15	Secepat refleksi Levi!	21B 2025-
4	Eura Rod	Fishing Rods	70.00	25	Bukan Katsura, tapi pancing Eura!	33B 2025-
5	Divine Water Bait	Fishing Lures	55.55	30	Diberkahi oleh Aqua, tapi jangan percaya airnya.	48B 2025-
6	Lost Navigator Compass	Accessories	10.10	50	Dijamin bikin kamu kehilangan arah seperti Zoro.	48B 2025-
8	Elegant Hook	Hooks	49.49	35	Bait ini memiliki ELEGANCE, cocok untuk Anya.	45B 2025-
9	Love Is War Net	Nets	250000.00	12	Strategi cerdas menangkap ikan, tapi hati-hati jangan jat...	66B 2025-
10	Sun Breathing Line	Fishing Line	200.00	8	Matahari terbit! Jaring yang tak tertandingi.	45B 2025-

e) Pelanggan dengan ID kurang dari 5

customer_id	name	email	phone	address	created_at
1	Sasuke Uchiha	sasuke@konoha.com	08123456789	Desa Konoha, Rumah Uchiha	2025-03-31 19:03:12
2	Megumin	megumin@explosion.com	08234567890	Belakang Guild Adventurer, Axel Town	2025-03-31 19:03:12
3	Levi Ackerman	levi@scoutinglegion.com	08345678901	Dinding Maria, Kamar Kapten Levi	2025-03-31 19:03:12
4	Sintoki Sakata	gintoki@yorozuya.com	08456789012	Nabukicho, Edo, Jepang	2025-03-31 19:03:12

4.4 Penjelasan

4.4.1 Tugas Praktikum

Pada praktikum kali ini, dilakukan beberapa perintah SQL yang berkaitan dengan manipulasi data dan struktur tabel dalam basis data. Pertama, ditambahkan kolom baru bernama keterangan pada tabel *orders* menggunakan perintah *ALTER TABLE*, yang berfungsi untuk memberikan informasi tambahan terkait status pesanan. Setelah kolom ditambahkan, data pada kolom tersebut diisi menggunakan perintah *UPDATE* berdasarkan kondisi tertentu, seperti jika status pesanan adalah 'Shipped' maka keterangan diisi dengan “pesanan sedang dalam tujuan”, dan jika

statusnya 'Completed' maka diisi dengan “pesanan sudah sampai di alamat tujuan”. Selanjutnya, dilakukan penggabungan data antara tabel *orders* dan *customers* menggunakan *JOIN* untuk menampilkan data pesanan lengkap dengan nama pelanggan. Kemudian, data ditampilkan secara lebih terstruktur dengan menerapkan perintah *ORDER BY*, seperti mengurutkan data produk berdasarkan harga dari yang tertinggi, mengurutkan nama pelanggan secara alfabet, dan mengurutkan pesanan berdasarkan tanggal terbaru. Terakhir, dilakukan perubahan tipe data pada kolom *phone* di tabel *customers* menjadi *VARCHAR(20)* agar format nomor telepon lebih fleksibel dan dapat menampung berbagai variasi format penulisan. Seluruh perintah tersebut digunakan untuk memanipulasi dan menampilkan data dengan lebih efisien sesuai kebutuhan pengolahan basis data dalam aplikasi.

BAB V

PENUTUP

5.1 Analisa

Pada praktikum kali ini, saya mempelajari tiga kelompok utama dalam perintah SQL, yaitu Data Definition Language (DDL), Data Manipulation Language (DML), dan Data Control Language (DCL). Ketiga jenis perintah ini memiliki fungsi yang berbeda namun saling berkaitan dalam pengelolaan basis data. DDL digunakan untuk membuat dan memodifikasi struktur tabel. Contoh perintah yang saya gunakan adalah *CREATE TABLE* untuk membuat tabel baru dan *ALTER TABLE* untuk menambahkan atau mengubah kolom dalam tabel yang sudah ada.

Selanjutnya, saya mempraktikkan perintah DML yang digunakan untuk mengolah isi data dalam tabel. Perintah seperti *INSERT*, *UPDATE*, *DELETE*, dan *SELECT* saya gunakan untuk menambahkan data UMKM, memperbarui informasi produk, menghapus data yang tidak diperlukan, serta menampilkan data berdasarkan kondisi tertentu. DML menjadi bagian penting karena berhubungan langsung dengan pengolahan data yang dibutuhkan aplikasi dalam kehidupan nyata.

Meskipun tidak digunakan secara langsung dalam praktikum ini, saya juga mempelajari DCL, yang berfungsi untuk mengatur hak akses pengguna terhadap data. Perintah seperti *GRANT* dan *REVOKE* sangat penting dalam menjaga keamanan dan privasi data, terutama ketika sistem digunakan oleh banyak pengguna dengan peran yang berbeda-beda. Dengan memahami DDL, DML, dan DCL, saya dapat mengelola basis data secara lebih terstruktur, aman, dan efisien sesuai dengan kebutuhan aplikasi yang sedang dikembangkan.

5.2 Kesimpulan

Berdasarkan praktikum yang telah saya lakukan, dapat disimpulkan bahwa stored procedure adalah fitur penting dalam basis data yang memungkinkan eksekusi otomatis sekumpulan perintah SQL secara efisien dan terstruktur. Dengan stored procedure, proses seperti *INSERT*, *UPDATE*, *DELETE*, dan *SELECT* dapat dilakukan berulang tanpa harus menulis ulang query, sehingga lebih praktis dan konsisten. Penggunaan parameter *IN*, *OUT*, dan *INOUT* juga memberikan fleksibilitas dalam pengolahan data. *Stored procedure* membantu menjaga validasi

logika bisnis langsung di sisi *database*, meningkatkan performa karena dijalankan oleh server, dan memberikan keamanan tambahan dengan membatasi akses langsung ke tabel. Selain itu, praktikum ini memperkuat pemahaman saya tentang logika kondisi, manipulasi data dinamis, serta integrasi antar tabel dalam pengembangan sistem basis data. Oleh karena itu berikut ini beberapa poin yang didapatkan oleh praktikan :

- a) Stored procedure adalah fitur penting dalam basis data yang digunakan untuk menyimpan dan menjalankan sekumpulan perintah SQL secara otomatis dan berulang.
- b) Stored procedure mempermudah proses manipulasi data seperti *INSERT*, *UPDATE*, *DELETE*, dan *SELECT* tanpa perlu menulis ulang query.
- c) Stored procedure mendukung parameter *IN*, *OUT*, dan *INOUT* yang memungkinkan fleksibilitas dalam pengolahan data.
- d) Logika bisnis dapat ditanam langsung di sisi database sehingga menjaga konsistensi dan validasi data lebih baik.
- e) Eksekusi *procedure* langsung oleh server database membuat sistem menjadi lebih efisien dan berperforma tinggi.
- f) Stored procedure meningkatkan keamanan karena pengguna hanya diberi akses melalui prosedur tertentu, bukan langsung ke tabel.
- g) Praktikum ini memperkuat pemahaman tentang penggunaan kondisi logika, pengolahan data dinamis, serta relasi antar tabel dalam SQL.