Tutorial SQL

1. Perintah SQL

- SQL SELECT

Digunakan untuk memilih data dari database.

SELECT kolom1, kolom2 FROM nama_tabel;

- SQL SELECT DISTINCT

Mengembalikan nilai yang berbeda saja.

SELECT DISTINCT kolom1 FROM nama_tabel;

- SQL WHERE

Digunakan untuk memfilter data.

SELECT * FROM nama_tabel WHERE kondisi;

- SQL ORDER BY

Digunakan untuk mengurutkan hasil query.

SELECT * FROM nama_tabel ORDER BY kolom1 ASC|DESC;

- SQL AND / OR / NOT

SELECT * FROM nama_tabel WHERE kondisi1 AND/OR/NOT kondisi2;

- SQL INSERT INTO

Untuk memasukkan data baru.

INSERT INTO nama_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);

- SQL NULL Values

Mengecek nilai NULL.

SELECT * FROM nama_tabel WHERE nama_kolom IS NULL;

- SQL UPDATE

Untuk memperbarui data.

UPDATE nama_tabel SET kolom1 = nilai1 WHERE kondisi;

- SQL DELETE

Menghapus data dari tabel.

DELETE FROM nama_tabel WHERE kondisi;

- SQL SELECT TOP

Mengambil sejumlah data teratas.

SELECT TOP 10 * FROM nama_tabel;

- Fungsi Agregat SQL

- COUNT, SUM, AVG, MIN, MAX

SELECT COUNT(*), SUM(kolom), AVG(kolom), MIN(kolom), MAX(kolom) FROM nama_tabel;

- SQL LIKE dan Wildcards

```
SELECT * FROM nama_tabel WHERE kolom LIKE 'pola%';
```

- SQL IN / BETWEEN / ALIASES

```
SELECT * FROM nama_tabel WHERE kolom IN (nilai1, nilai2);
SELECT * FROM nama_tabel WHERE kolom BETWEEN nilai1 AND nilai2;
SELECT kolom AS alias_nama FROM nama_tabel;
```

SQL JOIN

- INNER JOIN

```
SELECT * FROM tabel1 INNER JOIN tabel2 ON tabel1.id = tabel2.id;
```

- LEFT JOIN / RIGHT JOIN / FULL JOIN / SELF JOIN

```
SELECT * FROM tabel1 LEFT JOIN tabel2 ON kondisi;
SELECT * FROM tabel1 RIGHT JOIN tabel2 ON kondisi;
SELECT * FROM tabel1 FULL OUTER JOIN tabel2 ON kondisi;
SELECT A.kolom, B.kolom FROM tabel A, tabel B WHERE A.id = B.parent_id;
```

- SQL UNION

```
SELECT kolom FROM tabel1 UNION SELECT kolom FROM tabel2;
```

- SQL GROUP BY / HAVING

```
SELECT kolom, COUNT(*) FROM tabel GROUP BY kolom;
SELECT kolom, COUNT(*) FROM tabel GROUP BY kolom HAVING COUNT(*) > 1;
```

- SQL EXISTS / ANY / ALL

```
SELECT kolom FROM tabel WHERE EXISTS (SELECT * FROM tabel_lain WHERE kondisi);
SELECT kolom FROM tabel WHERE kolom = ANY (SELECT kolom FROM tabel_lain);
SELECT kolom FROM tabel WHERE kolom > ALL (SELECT kolom FROM tabel_lain);
```

- SQL SELECT INTO / INSERT INTO SELECT

```
SELECT * INTO tabel_baru FROM tabel_lama;
INSERT INTO tabel1 SELECT * FROM tabel2;
```

- SQL CASE / Fungsi NULL

```
SELECT kolom, CASE WHEN kondisi THEN hasil ELSE default END FROM tabel;
SELECT ISNULL(kolom, 'default') FROM tabel;
```

- Prosedur Tersimpan SQL

```
CREATE PROCEDURE nama_prosedur AS perintah_sql;
EXEC nama_prosedur;
```

- Komentar dalam SQL

```
-- Ini komentar satu baris
/* Ini komentar beberapa baris */
```

- Operator SQL

- Aritmatika, Perbandingan, Logika (AND, OR, NOT), LIKE, IN, BETWEEN, dll.

2. Basis Data SQL

- SQL CREATE / DROP / BACKUP DATABASE

CREATE DATABASE nama_database;

DROP DATABASE nama_database;

BACKUP DATABASE nama_database TO DISK = 'lokasi_backup';

- SQL CREATE / DROP / ALTER TABLE

CREATE TABLE nama_tabel (id INT, nama VARCHAR(100));

DROP TABLE nama_tabel;

ALTER TABLE nama_tabel ADD nama_kolom tipe_data;

- SQL Constraints (Kendala)

- NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

CREATE TABLE tabel (id INT NOT NULL,email VARCHAR(100) UNIQUE,PRIMARY KEY (id), FOREIGN KEY (id) REFERENCES tabel_lain(id),CHECK (umur >= 18),DEFAULT GETDATE());

- SQL Index / Auto Increment

CREATE INDEX nama_index ON tabel (kolom);

CREATE TABLE tabel (id INT IDENTITY(1,1), nama VARCHAR(50));

- SQL Dates (Tanggal)

SELECT * FROM tabel WHERE kolom_tanggal = '2023-01-01';

- SQL Views

CREATE VIEW nama_view AS SELECT kolom1, kolom2 FROM tabel;

- SQL Injection (Masalah Keamanan)

- Hindari penggunaan query SQL dinamis. Gunakan query yang diparameterkan.

-- Rentan:

"SELECT * FROM users WHERE username = '" + masukanUser + "'"

-- Aman:

SqlCommand cmd = new SqlCommand("SELECT * FROM users WHERE username = @username");

cmd.Parameters.AddWithValue("@username", masukanUser);