A picture containing tableware, dishware

Description automatically generated

***JOBSHEET***

BASIS DATA LANJUT

A logo of a building

AI-generated content may be incorrect.Shape, rectangle

Description automatically generated

PERTEMUAN 3

Query Lanjut (SELECT, JOIN, CTE)

pada PostgreSQL

Jurusan Teknologi Informasi

POLITEKNIK NEGERI MALANG

TAHUN AJARAN 2025/2026

Team Teaching Basis Data Lanjut:

* Candra Bella Vista, S.Kom., MT.
* Moch Zawaruddin Abdullah, S.ST., M.Kom.
* Yan Watequlis Syaifudin, ST., MMT., PhD.
* Yoppy Yunhasnawa, S.ST., M.Sc.

|  |  |  |
| --- | --- | --- |
| Mata Kuliah | **:** | Basis Data Lanjut |
| Program Studi | **:** | D4 – Teknik Informatika **/** D4 – Sistem Informasi Bisnis |
| Semester | **:** | 3 (tiga) |
| Pertemuan ke- | **:** | 3 |

**MATERI 3: QUERY LANJUT PADA POSTGRESQL**

**Kuliah: Basisdata Lanjut**

**Tujuan Pembelajaran**

Setelah menyelesaikan materi ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar operasi SELECT dalam PostgreSQL;
2. Menerapkan berbagai jenis JOIN untuk menggabungkan data dari multiple tables;
3. Menggunakan fungsi agregasi untuk analisis data;
4. Memahami perbedaan sintaks PostgreSQL dengan database lainnya;
5. Mengimplementasikan CTE (Common Table Expression) untuk menyederhanakan query kompleks.

**Konsep Dasar**

Materi ini membahas inti dari operasi pembacaan data (query) dalam SQL, yang bertujuan untuk mengambil, menggabungkan, dan meringkas informasi dari database relasional seperti PostgreSQL. Konsep utamanya meliputi:

* **SELECT**: Perintah fundamental untuk memilih dan mengambil data dari satu atau lebih tabel.
* **JOIN**: Teknik untuk menggabungkan baris dari dua atau lebih tabel berdasarkan kolom yang saling terkait (relationship).
* **Agregasi**: Metode untuk melakukan perhitungan pada sekelompok data untuk menghasilkan nilai ringkasan tunggal (seperti jumlah, rata-rata, maksimum).
* **CTE (Common Table Expression)**: Ekspresi query sementara yang meningkatkan keterbacaan dan struktur query kompleks, membuatnya mirip dengan membuat "tabel virtual" sementara dalam satu perintah.

**Manfaat**

Menguasai materi ini memberikan manfaat yang sangat besar, yaitu:

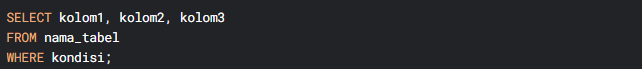
* **Efisiensi**: Dapat mengambil informasi yang tepat dan bermakna dari laut data dengan cepat dan akurat.
* **Kemampuan Analitis**: Mampu menjawab pertanyaan bisnis yang kompleks dengan menggabungkan data dari berbagai sumber dan meringkasnya.
* **Keterbacaan & Pemeliharaan**: Penggunaan CTE dan teknik query yang terstruktur membuat kode SQL lebih mudah dibaca, didebug, dan dipelihara oleh diri sendiri atau orang lain.
* **Portabilitas Skill**: Memahami perbedaan sintaks (seperti LIMIT/OFFSET) memudahkan adaptasi untuk bekerja dengan berbagai jenis database (PostgreSQL, MySQL, SQL Server, dll.).

**TEORI DASAR**

**1. Konsep Dasar SELECT**

Pengertian SELECT: SELECT adalah perintah fundamental SQL untuk mengambil data dari satu atau lebih tabel dalam database. Perintah ini memungkinkan kita untuk menyeleksi kolom tertentu, menerapkan filter, mengurutkan hasil, dan membatasi jumlah data yang ditampilkan.

**Contoh dasar:**

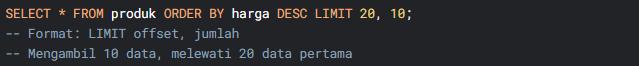


**2. Perbedaan LIMIT/OFFSET: PostgreSQL vs MySQL**

**PostgreSQL:**

-- Mengambil 10 data, melewati 20 data pertama

**MySQL:**

-- Format: LIMIT offset, jumlah

-- Mengambil 10 data, melewati 20 data pertama

**Perbedaan utama:**

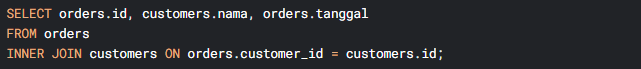
* PostgreSQL menggunakan kata kunci terpisah: LIMIT dan OFFSET
* MySQL menggunakan sintaks gabungan: LIMIT offset, jumlah
* PostgreSQL lebih eksplisit dan mudah dibaca untuk pagination kompleks

**3. Konsep JOIN**

JOIN digunakan untuk menggabungkan baris dari dua atau lebih tabel berdasarkan kolom yang terkait antara tabel-tabel tersebut.

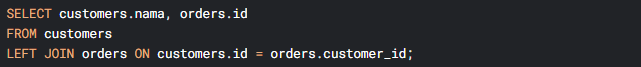
**a. INNER JOIN**

Menampilkan hanya baris yang memiliki nilai yang cocok di kedua tabel.



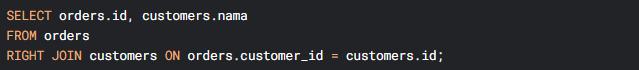
**b. LEFT JOIN**

Menampilkan semua baris dari tabel kiri (pertama), dan baris yang cocok dari tabel kanan. Jika tidak ada kecocokan, hasilnya NULL dari sisi kanan.



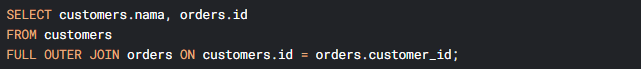
**c. RIGHT JOIN**

Kebalikan dari LEFT JOIN. Menampilkan semua baris dari tabel kanan, dan baris yang cocok dari tabel kiri.



**d. FULL OUTER JOIN**

Menampilkan semua baris ketika ada kecocokan di salah satu tabel. Jika tidak ada kecocokan, kolom dari tabel tanpa kecocokan akan berisi NULL.



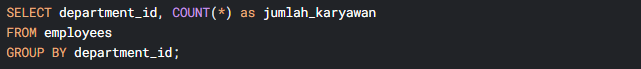
**4. Fungsi Agregasi**

Fungsi agregasi melakukan perhitungan pada sekumpulan nilai dan mengembalikan nilai tunggal. Sering digunakan dengan klausa GROUP BY.

Fungsi Agregasi Umum:

* COUNT(): Menghitung jumlah baris
* SUM(): Menjumlahkan nilai
* AVG(): Menghitung rata-rata
* MAX(): Mencari nilai tertinggi
* MIN(): Mencari nilai terendah

**GROUP BY**: Mengelompokkan baris yang memiliki nilai yang sama ke dalam summary rows.



**HAVING**: Memfilter hasil agregasi (setelah GROUP BY), berbeda dengan WHERE yang memfilter sebelum agregasi.

A black background with white text

AI-generated content may be incorrect.

**Perbedaan WHERE vs HAVING:**

* WHERE memfilter baris sebelum pengelompokan
* HAVING memfilter hasil setelah pengelompokan

**5. CTE (Common Table Expression)**

**Konsep WITH Clause**

CTE memungkinkan kita membuat query sementara yang dapat dirujuk dalam query utama, membuat query kompleks menjadi lebih modular dan mudah dibaca.

**Struktur dasar:**

**A black rectangle with white text

AI-generated content may be incorrect.**

**Contoh Implementasi CTE**

**A computer screen shot of a code

AI-generated content may be incorrect.**

**Perbandingan CTE vs Subquery di MySQL/PostgreSQL**

**CTE:**

-- Lebih mudah dibaca dan dimaintenance

**A black background with white text

AI-generated content may be incorrect.**

**Subquery:**

-- Lebih kompleks untuk query bertingkat

A black background with white text

AI-generated content may be incorrect.

**Keunggulan CTE:**

1. **Keterbacaan**: Struktur lebih jelas dan terorganisir
2. **Reusability**: Dapat dirujuk multiple times dalam query yang sama
3. **Maintainability**: Lebih mudah di-debug dan dimodifikasi
4. **Recursive Queries**: Mendukung query rekursif (hanya di CTE)

Menggunakan CTE untuk Query Kompleks

A screenshot of a computer program

AI-generated content may be incorrect.

Best Practices

1. **SELECT Spesifik**: Hindari SELECT \*, sebutkan kolom explicitly
2. **Indexing**: Pastikan kolom yang di-JOIN dan di-WHERE ter-index
3. **LIMIT untuk Testing**: Gunakan LIMIT saat mengembangkan query kompleks
4. **CTE untuk Kompleksitas**: Gunakan CTE untuk query yang memiliki multiple subqueries
5. **EXPLAIN ANALYZE**: Gunakan untuk menganalisis performance query

Dengan memahami konsep-konsep dasar ini, mahasiswa akan memiliki fondasi yang kuat untuk melakukan query data yang efektif dan efisien dalam PostgreSQL.

Konsep-konsep ini adalah tulang punggung dari hampir semua sistem yang berurusan dengan data:

1. **Laporan dan Dashboard**: Membuat laporan penjualan per bulan, rata-rata nilai pelanggan, atau jumlah produk yang terjual per kategori (menggunakan **Agregasi** dan **GROUP BY**).
2. **Sistem E-commerce**: Menampilkan daftar pesanan beserta detail informasi pelanggan dan produknya (menggunakan **JOIN** antara tabel orders, customers, dan products).
3. **Media Sosial**: Menampilkan feed berita yang berisi postingan dari orang yang diikuti (menggunakan **JOIN** pada tabel users, posts, dan follows).
4. **Analisis Data Kompleks**: Membersihkan data, melakukan transformasi multi-tahap, dan menghitung metrik bisnis yang rumit sebelum ditampilkan (menggunakan **CTE** untuk menyederhanakan setiap tahapan).

**JOBSHEET PRAKTIKUM**

**SETUP AWAL**

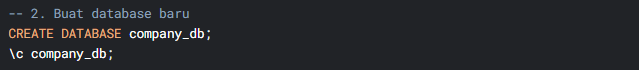
**1. Install PostgreSQL**

* Download dan install PostgreSQL dari <https://www.postgresql.org/download/>
* Install pgAdmin (GUI tool) atau gunakan psql (command line tool)

**2. Login ke Database**



**3. Buat Database Baru**:



**4. Buat Tabel-tabel**:

A screenshot of a computer program

AI-generated content may be incorrect.

**5. Insert Data Sample**:

A computer screen shot of a program code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Studi Kasus: Sistem Manajemen Perusahaan XYZ**

Database perusahaan dengan berbagai entitas: departments, employees, projects, dan hubungan many-to-many antara employees dan projects.

**Praktikum 1: SELECT dengan LIMIT/OFFSET**

**Konsep Dasar**

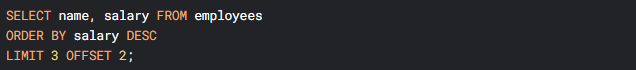
* **LIMIT**: Membatasi jumlah baris yang ditampilkan
* **OFFSET**: Melewati sejumlah baris tertentu
* **Perbedaan dengan MySQL**: PostgreSQL menggunakan LIMIT x OFFSET y vs MySQL LIMIT y, x

**Latihan Praktis**

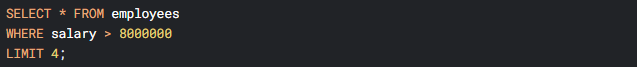
**1. Basic SELECT dengan LIMIT**



**2. SELECT dengan ORDER BY dan LIMIT/OFFSET**



**3. SELECT dengan WHERE dan LIMIT**



**Soal Latihan Praktikum 1**

1. Tampilkan 5 employee dengan gaji terendah
2. Tampilkan halaman 2 data projects (3 data per halaman)
3. Tampilkan 3 employee yang dihire paling akhir

**Query Mahasiswa:**

|  |
| --- |
|  |

**Hasil Eksekusi:**

|  |
| --- |
|  |

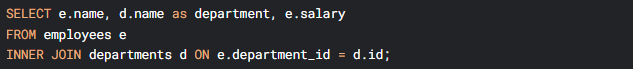
**Praktikum 2: JOIN Operations**

**Jenis-Jenis JOIN**

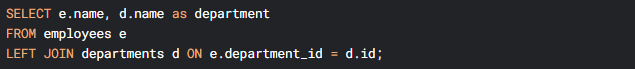
* **INNER JOIN**: Data yang match di kedua tabel
* **LEFT JOIN**: Semua data dari tabel kiri + match dari kanan
* **RIGHT JOIN**: Semua data dari tabel kanan + match dari kiri
* **FULL OUTER JOIN**: Semua data dari kedua tabel

**Latihan Praktis**

**1. INNER JOIN**



**2. LEFT JOIN**

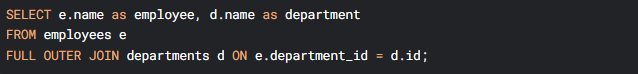


**3. RIGHT JOIN**

A screen shot of a computer

AI-generated content may be incorrect.

**4. FULL OUTER JOIN**



**5. Multiple JOIN**

**A computer screen shot of a black background

AI-generated content may be incorrect.**

**Soal Latihan Praktikum 2**

1. Tampilkan semua projects beserta department penanggung jawabnya
2. Tampilkan employee yang tidak memiliki department
3. Tampilkan department yang tidak memiliki employee
4. Tampilkan employee yang bekerja di project 'Website Development'

**Query Mahasiswa:**

|  |
| --- |
|  |

**Hasil Eksekusi:**

|  |
| --- |
|  |

**Praktikum 3: Fungsi Agregasi**

**Konsep Dasar**

* **GROUP BY**: Mengelompokkan data berdasarkan kolom tertentu
* **HAVING**: Filter hasil agregasi (setelah GROUP BY)
* **Fungsi Agregasi**: COUNT, SUM, AVG, MAX, MIN

**Latihan Praktis**

**1. Basic Agregasi**

A black rectangle with white text

AI-generated content may be incorrect.

**2. GROUP BY**

A computer screen shot of a computer screen

AI-generated content may be incorrect.

**3. HAVING**

A computer screen shot of a computer screen

AI-generated content may be incorrect.

**4. Multiple Agregasi**

A computer screen shot of white text

AI-generated content may be incorrect.

**Soal Latihan Praktikum 3**

1. Tampilkan total budget projects per department
2. Tampilkan department dengan rata-rata gaji di atas 8.5 juta
3. Tampilkan project dengan total jam kerja lebih dari 200 jam
4. Tampilkan employee dengan total jam kerja terbanyak

**Query Mahasiswa:**

|  |
| --- |
|  |

**Hasil Eksekusi:**

|  |
| --- |
|  |

**Praktikum 4: Common Table Expression (CTE)**

**Konsep Dasar**

* **CTE**: Tabel sementara yang hanya ada selama eksekusi query
* **Keunggulan**: Lebih mudah dibaca dan dimaintain dibanding subquery
* **Syntax**: WITH cte\_name AS (SELECT ...) SELECT \* FROM cte\_name

**Latihan Praktis**

**1. Basic CTE**

A screenshot of a computer screen

AI-generated content may be incorrect.

**2. Multiple CTEs**

A screen shot of a computer program

AI-generated content may be incorrect.

**3. CTE vs Subquery**

A computer screen shot of text

AI-generated content may be incorrect.

**4. CTE Complex Query**

A screen shot of a computer program

AI-generated content may be incorrect.

**Soal Latihan Praktikum 4**

1. Buat CTE untuk menampilkan top 3 employee dengan gaji tertinggi
2. Buat CTE untuk menghitung rata-rata jam kerja per department
3. Buat CTE untuk menampilkan project dengan budget di atas rata-rata
4. Buat CTE berjenjang (multiple CTEs) untuk analisis department lengkap

**Query Mahasiswa:**

|  |
| --- |
|  |

**Hasil Eksekusi:**

|  |
| --- |
|  |

**Tugas Akhir Integrasi**

Buat query lengkap yang menggabungkan semua konsep untuk menjawab pertanyaan berikut:

1. Tampilkan 5 department dengan total gaji tertinggi
2. Tampilkan project dengan rata-rata jam kerja per employee tertinggi
3. Tampilkan employee yang bekerja di lebih dari 1 project
4. Buat laporan performance department: nama department, jumlah employee, jumlah project, total budget projects, rata-rata gaji, total jam kerja

**Query Mahasiswa:**

|  |
| --- |
|  |

**Hasil Eksekusi:**

|  |
| --- |
|  |

**Refleksi Pembelajaran**

**Nama Mahasiswa**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NIM**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Hal yang sudah dipahami:**

* SELECT dengan LIMIT/OFFSET
* Berbagai jenis JOIN
* Fungsi agregasi dengan GROUP BY dan HAVING
* Penggunaan CTE untuk query kompleks

**Kesulitan yang dialami:**