

**BASIS DATA LANJUT**

**JOBSHEET 3**

*Query Lanjut (SELECT, JOIN, CTE)*



**Disusun Oleh:**

**Andini Tribuana**

**Kelas SIB - 1E**

**NIM. 244107060140**

**PROGRAM STUDI SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

# 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

```
psql -U username -d database_name
```

### 3. Buat Database Baru:

```
-- 2. Buat database baru  
CREATE DATABASE company_db;  
\c company_db;
```

### 4. Buat Tabel-tabel:

```
➥ create table departments (  
  id serial primary key,  
  name varchar(100) not null,  
  location varchar(100)  
);  
  
➥ create table employees (  
  id serial primary key,  
  name varchar(100) not null,  
  salary numeric(10,2),  
  department_id integer references departments(id),  
  hire_date date  
);  
  
➥ create table projects (  
  id serial primary key,  
  name varchar(100) not null,  
  budget numeric(15,2),  
  department_id integer references departments(id)  
);  
  
➥ create table employee_projects (  
  employee_id integer references employees(id),  
  project_id integer references projects(id),  
  hours_worked numeric(5,2),  
  assignment_date date default current_date,  
  primary key (employee_id, project_id)  
);|
```

## 5. Insert Data Sample:

```
INSERT INTO departments (name, location) VALUES
('IT', 'Jakarta'),
('HR', 'Bandung'),
('Finance', 'Surabaya'),
('Marketing', 'Medan'),
('Operations', 'Yogyakarta');

INSERT INTO employees (name, salary, department_id, hire_date) VALUES
('Budi Santoso', 800000, 1, '2022-01-15'),
('Siti Rahayu', 750000, 1, '2022-03-20'),
('Ahmad Fauzi', 900000, 2, '2021-11-10'),
('Dewi Anggraini', 850000, 2, '2022-02-28'),
('Rudi Hermawan', 950000, 3, '2021-08-05'),
('Maya Sari', 880000, 3, '2022-04-15'),
('Hendra Pratama', 820000, 4, '2022-05-20'),
('Lina Marlina', 780000, 4, '2022-06-10'),
('Joko Widodo', 920000, 1, '2021-12-01'),
('Ratna Dewi', 870000, 3, '2022-03-15'),
('Fajar Nugroho', 0, NULL, '2023-01-01'), -- Employee tanpa department
('Bambang Sutoyo', 650000, 5, '2022-07-01');

-- Data projects
INSERT INTO projects (name, budget, department_id) VALUES
('Website Development', 5000000, 1),
('Mobile App', 7500000, 1),
('Recruitment System', 3000000, 2),
('Payroll System', 4500000, 3),
('Marketing Campaign', 6000000, 4),
('Database Migration', 3500000, 1),
('Inventory System', 4000000, NULL), -- Project tanpa department
('CRM Implementation', 5500000, 5);

-- Data employee_projects
INSERT INTO employee_projects (employee_id, project_id, hours_worked, assignment_date) VALUES
(1, 1, 120.5, '2023-01-15'),
(1, 6, 80.0, '2023-02-01'),
(2, 1, 95.5, '2023-01-20'),
(2, 2, 150.0, '2023-02-15'),
(3, 3, 200.0, '2023-03-01'),
(4, 3, 180.5, '2023-03-05'),
(5, 4, 220.0, '2023-04-01'),
(6, 4, 190.5, '2023-04-05'),
(7, 5, 175.0, '2023-05-01'),
(8, 5, 160.5, '2023-05-10'),
(9, 7, 140.0, '2023-06-01'),
(9, 6, 90.5, '2023-06-15'),
(10, 4, 210.0, '2023-07-01'),
(11, 7, 100.0, '2023-07-15'),
(12, 8, 180.0, '2023-08-01');
```

## 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

```
SELECT * FROM employees LIMIT 5;
```

123 id	A-Z name	123 salary	123 department_id	hire_date
1	Budi Santoso	800,000	1	2022-01-15
2	Siti Rahayu	750,000	1	2022-03-20
3	Ahmad Fauzi	900,000	2	2021-11-10
4	Dewi Anggraini	850,000	2	2022-02-28
5	Rudi Hermawan	950,000	3	2021-08-05

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

```
SELECT name, salary FROM employees  
ORDER BY salary DESC  
LIMIT 3 OFFSET 2;
```

A-Z name	123 salary
Ahmad Fauzi	900,000
Maya Sari	880,000
Ratna Dewi	870,000

##### 3. SELECT dengan WHERE dan LIMIT

```
SELECT * FROM employees  
WHERE salary > 8000000  
LIMIT 4;
```

123 id	A-Z name	123 salary	123 department_id	hire_date
3	Ahmad Fauzi	9,000,000	2	2021-11-10
4	Dewi Anggraini	8,500,000	2	2022-02-28
5	Rudi Hermawan	9,500,000	3	2021-08-05
6	Maya Sari	8,800,000	3	2022-04-15

## Soal Latihan Praktikum 1

1. Tampilkan 5 employee dengan gaji terendah

```
select * from employees
order by salary asc
limit 5;
```

123 id	AZ name	123 salary	123 department	hire_date
11	Fajar Nugroho	0	[NULL]	2023-01-01
12	Bambang Sutoyo	6,500,000	5	2022-07-01
2	Siti Rahayu	7,500,000	1	2022-03-20
8	Lina Marlina	7,800,000	4	2022-06-10
1	Budi Santoso	8,000,000	1	2022-01-15

2. Tampilkan halaman 2 data projects (3 data per halaman)

```
select * from projects
limit 3 offset 3;
```

123 id	AZ name	123 budget	123 department_id
4	Payroll System	4,500,000	3
5	Marketing Campaign	6,000,000	4
6	Database Migration	3,500,000	1

3. Tampilkan 3 employee yang dihire paling akhir

```
select * from employees
order by hire_date desc
limit 3;
```

123 id	AZ name	123 salary	123 department_id	hire_date
11	Fajar Nugroho	0	[NULL]	2023-01-01
12	Bambang Sutoyo	6,500,000	5	2022-07-01
8	Lina Marlina	7,800,000	4	2022-06-10

## 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

```
SELECT e.name, d.name as department, e.salary
FROM employees e
INNER JOIN departments d ON e.department_id = d.id;
```

A-Z name ▼	A-Z department ▼	123 salary ▼
Budi Santoso	IT	8,000,000
Siti Rahayu	IT	7,500,000
Ahmad Fauzi	HR	9,000,000
Dewi Anggraini	HR	8,500,000
Rudi Hermawan	Finance	9,500,000
Maya Sari	Finance	8,800,000
Hendra Pratama	Marketing	8,200,000
Lina Marlina	Marketing	7,800,000
Joko Widodo	IT	9,200,000
Ratna Dewi	Finance	8,700,000
Bambang Sutoyo	Operations	6,500,000

#### 2. LEFT JOIN

```
SELECT e.name, d.name as department
FROM employees e
LEFT JOIN departments d ON e.department_id = d.id;
```

A-Z name ▼	A-Z department ▼	
Budi Santoso	IT	
Siti Rahayu	IT	
Ahmad Fauzi	HR	
Dewi Anggraini	HR	
Rudi Hermawan	Finance	
Maya Sari	Finance	
Hendra Pratama	Marketing	
Lina Marlina	Marketing	
Joko Widodo	IT	
Ratna Dewi	Finance	
Fajar Nugroho	[NULL]	
Bambang Sutoyo	Operations	

### 3. RIGHT JOIN

```
SELECT d.name as department, COUNT(e.id) as employee_count
FROM departments d
RIGHT JOIN employees e ON d.id = e.department_id
GROUP BY d.name;
```

A-Z department ▼	123 employee_count ▼
[NULL]	1
hr	2
marketing	2
it	3
operations	1
finance	3

### 4. FULL OUTER JOIN

```
SELECT e.name as employee, d.name as department
FROM employees e
FULL OUTER JOIN departments d ON e.department_id = d.id;
```

A-Z employee ▼	A-Z department ▼
budi santoso	it
siti rahayu	it
ahmad fauzi	hr
dewi anggraini	hr
rudi hermawan	finance
maya sari	finance
hendra pratama	marketing
lina marlina	marketing
joko widodo	it
ratna dewi	finance
fajar nugroho	[NULL]
bambang sutoyo	operations

## 5. Multiple JOIN

```
SELECT e.name, p.name as project, ep.hours_worked
FROM employees e
INNER JOIN employee_projects ep ON e.id = ep.employee_id
INNER JOIN projects p ON ep.project_id = p.id;
```

A-Z name ▼	A-Z project ▼	123 hours_worked ▼
budi santoso	website development	120.5
budi santoso	database migration	80
siti rahayu	website development	95.5
siti rahayu	mobile app	150
ahmad fauzi	recruitment system	200
dewi anggraini	recruitment system	180.5
rudi hermawan	payroll system	220
maya sari	payroll system	190.5
hendra pratama	marketing campaign	175
lina marlina	marketing campaign	160.5
joko widodo	inventory system	140
joko widodo	database migration	90.5
ratna dewi	payroll system	210
fajar nugroho	inventory system	100
bambang sutoyo	crm implementation	180

### Soal Latihan Praktikum 2

1. Tampilkan semua projects beserta department penanggung jawabnya

```
select p.name as project, d.name as department
from projects p
left join departments d on p.department_id = d.id;
```

A-Z project ▼	A-Z department ▼
website development	it
mobile app	it
recruitment system	hr
payroll system	finance
marketing campaign	marketing
database migration	it
inventory system	[NULL]
crm implementation	operations



2. Tampilkan employee yang tidak memiliki department

```
select id, name, salary, hire_date
from employees
where department_id is null;
```

123 id ▼	A-Z name ▼	123 salary ▼	hire_date ▼
11	fajar nugroho	0	2023-01-01

3. Tampilkan department yang tidak memiliki employee

```
select d.id, d.name, d.location
from departments d
left join employees e on d.id = e.department_id
where e.id is null;
```

Kosong karena semua department memiliki karyawan

4. Tampilkan employee yang bekerja di project 'Website Development'

```
select e.name as employee, p.name as project
from employees e
join employee_projects ep on e.id = ep.employee_id
join projects p on ep.project_id = p.id
where p.name = 'website development';
```

A-Z employee ▼	A-Z project ▼
budi santoso	website development
siti rahayu	website development

### 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

```
SELECT
    COUNT(*) as total_employees,
    AVG(salary) as avg_salary,
    MAX(salary) as max_salary
FROM employees;
```

123 total_employees ▼	123 avg_salary ▼	123 max_salary ▼
12	7,641,666.6666666667	9,500,000

## 2. GROUP BY

```
SELECT
    d.name as department,
    COUNT(e.id) as employee_count,
    AVG(e.salary) as avg_salary
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
GROUP BY d.name;
```

A-Z department ▼	123 employee_count ▼	123 avg_salary ▼
hr	2	8,750,000
marketing	2	8,000,000
it	3	8,233,333.3333333333
operations	1	6,500,000
finance	3	9,000,000

## 3. HAVING

```
SELECT
    d.name as department,
    COUNT(e.id) as employee_count
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
GROUP BY d.name
HAVING COUNT(e.id) > 2;
```

A-Z department ▼	123 employee_count ▼
it	3
finance	3

#### 4. Multiple Agregasi

```
SELECT
  p.name as project,
  SUM(ep.hours_worked) as total_hours,
  COUNT(ep.employee_id) as total_employees,
  AVG(ep.hours_worked) as avg_hours_per_employee
FROM projects p
LEFT JOIN employee_projects ep ON p.id = ep.project_id
GROUP BY p.name;
```

A-Z project	123 total_hours	123 total_employees	123 avg_hours_per_employee
website development	216	2	108
payroll system	620.5	3	206.8333333333
marketing campaign	335.5	2	167.75
crm implementation	180	1	180
recruitment system	380.5	2	190.25
mobile app	150	1	150
database migration	170.5	2	85.25
inventory system	240	2	120

#### Soal Latihan Praktikum 3

1. Tampilkan total budget projects per department

```
select
  d.name as department,
  sum(p.budget) as total_budget
from departments d
left join projects p on d.id = p.department_id
group by d.name;
```

A-Z department	123 total_budget
hr	3,000,000
marketing	6,000,000
it	16,000,000
operations	5,500,000
finance	4,500,000

2. Tampilkan department dengan rata-rata gaji di atas 8.5 juta

```
select
  d.name as department,
  avg(e.salary) as avg_salary
from departments d
join employees e on d.id = e.department_id
group by d.name
having avg(e.salary) > 8500000;
```

A-Z department	123 avg_salary
hr	8,750,000
finance	9,000,000

3. Tampilkan project dengan total jam kerja lebih dari 200 jam

```
select
p.name as project,
sum(ep.hours_worked) as total_hours
from projects p
join employee_projects ep on p.id = ep.project_id
group by p.name
having sum(ep.hours_worked) > 200;
```

A-Z project ▼	123 total_hours ▼
website development	216
payroll system	620.5
marketing campaign	335.5
recruitment system	380.5
inventory system	240

4. Tampilkan employee dengan total jam kerja terbanyak

```
select
e.name as employee,
sum(ep.hours_worked) as total_hours
from employees e
join employee_projects ep on e.id = ep.employee_id
group by e.name
order by total_hours desc
limit 1;
```

A-Z employee ▼	123 total_hours ▼
siti rahayu	245.5

## 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

```
WITH high_salary_employees AS (
    SELECT * FROM employees WHERE salary > 8500000
)
SELECT * FROM high_salary_employees ORDER BY salary DESC;
```

123 id ▼	A-Z name ▼	123 salary ▼	123 department_id ▼	hire_date ▼
5	rudi hermawan	9,500,000	3	2021-08-05
9	joko widodo	9,200,000	1	2021-12-01
3	ahmad fauzi	9,000,000	2	2021-11-10
6	maya sari	8,800,000	3	2022-04-15
10	ratna dewi	8,700,000	3	2022-03-15

## 2. Multiple CTEs

```
WITH dept_stats AS (  
    SELECT  
        department_id,  
        COUNT(*) as emp_count,  
        AVG(salary) as avg_salary  
    FROM employees  
    GROUP BY department_id  
)  
project_stats AS (  
    SELECT  
        department_id,  
        COUNT(*) as project_count  
    FROM projects  
    GROUP BY department_id  
)  
SELECT  
    d.name as department,  
    ds.emp_count,  
    ds.avg_salary,  
    ps.project_count  
FROM departments d  
LEFT JOIN dept_stats ds ON d.id = ds.department_id  
LEFT JOIN project_stats ps ON d.id = ps.department_id;
```

A-Z department ▼	123 emp_count ▼	123 avg_salary ▼	123 project_count ▼
it	3	8,233,333.333333333	3
hr	2	8,750,000	1
finance	3	9,000,000	1
marketing	2	8,000,000	1
operations	1	6,500,000	1

## 3. CTE vs Subquery

```
-- Subquery (lebih rumit)  
SELECT name, salary  
FROM employees  
WHERE salary > (SELECT AVG(salary) FROM employees);  
  
-- CTE (lebih mudah dibaca)  
WITH avg_salary AS (  
    SELECT AVG(salary) as avg_sal FROM employees  
)  
SELECT name, salary  
FROM employees, avg_salary  
WHERE salary > avg_sal;
```

A-Z name ▼	123 salary ▼
budi santoso	8,000,000
ahmad fauzi	9,000,000
dewi anggraini	8,500,000
rudi hermawan	9,500,000
maya sari	8,800,000
hendra pratama	8,200,000
lina marlina	7,800,000
joko widodo	9,200,000
ratna dewi	8,700,000

#### 4. CTE Complex Query

```

WITH employee_project_stats AS (
    SELECT
        e.name as employee_name,
        COUNT(ep.project_id) as total_projects,
        SUM(ep.hours_worked) as total_hours
    FROM employees e
    LEFT JOIN employee_projects ep ON e.id = ep.employee_id
    GROUP BY e.name
)
SELECT
    employee_name,
    total_projects,
    total_hours,
    CASE
        WHEN total_hours > 200 THEN 'High'
        WHEN total_hours > 100 THEN 'Medium'
        ELSE 'Low'
    END as workload_level
FROM employee_project_stats
ORDER BY total_hours DESC;

```

A-Z employee_name ▼	123 total_projects ▼	123 total_hours ▼	A-Z workload_level ▼
siti rahayu	2	245.5	high
joko widodo	2	230.5	high
rudi hermawan	1	220	high
ratna dewi	1	210	high
budi santoso	2	200.5	high
ahmad fauzi	1	200	medium
maya sari	1	190.5	medium
dewi anggraini	1	180.5	medium
bambang sutoyo	1	180	medium
hendra pratama	1	175	medium
lina marlina	1	160.5	medium
fajar nugroho	1	100	low

#### Soal Latihan Praktikum 4

1. Buat CTE untuk menampilkan top 3 employee dengan gaji tertinggi

```
with top_salary as (
select id, name, salary
from employees
order by salary desc
limit 3)
select * from top_salary;
```

123 id ▼	A-Z name ▼	123 salary ▼
5	rudi hermawan	9,500,000
9	joko widodo	9,200,000
3	ahmad fauzi	9,000,000

2. Buat CTE untuk menghitung rata-rata jam kerja per department

```
with dept_hours as (
select d.id as dept_id, d.name as department,
avg(ep.hours_worked) as avg_hours
from departments d
join employees e on d.id = e.department_id
join employee_projects ep on e.id = ep.employee_id
group by d.id, d.name)
select * from dept_hours;
```

123 dept_id ▼	A-Z department ▼	123 avg_hours ▼
3	finance	206.8333333333
5	operations	180
4	marketing	167.75
2	hr	190.25
1	it	112.75

3. Buat CTE untuk menampilkan project dengan budget di atas rata-rata

```
with avg_budget as (
select avg(budget) as avg_budget from projects)
select p.id, p.name, p.budget
from projects p, avg_budget ab
where p.budget > ab.avg_budget;
```

123 id ▼	A-Z name ▼	123 budget ▼
1	website development	5,000,000
2	mobile app	7,500,000
5	marketing campaign	6,000,000
8	crm implementation	5,500,000

4. Buat CTE berjenjang (multiple CTEs) untuk analisis department lengkap

```
with emp_count as (
select department_id, count(*) as total_employees
from employees
group by department_id
),
avg_salary as (
select department_id, avg(salary) as avg_salary
from employees
group by department_id
),
hours_per_dept as (
select d.id as department_id, avg(ep.hours_worked) as avg_hours
from departments d
join employees e on d.id = e.department_id
join employee_projects ep on e.id = ep.employee_id
group by d.id
)
select d.name as department,
coalesce(ec.total_employees,0) as total_employees,
coalesce(a.avg_salary,0) as avg_salary,
coalesce(h.avg_hours,0) as avg_hours
from departments d
left join emp_count ec on d.id = ec.department_id
left join avg_salary a on d.id = a.department_id
left join hours_per_dept h on d.id = h.department_id;
```

A-Z department ▼	123 total_employees ▼	123 avg_salary ▼	123 avg_hours ▼
it	3	8,233,333.333333333	112.75
hr	2	8,750,000	190.25
finance	3	9,000,000	206.833333333
marketing	2	8,000,000	167.75
operations	1	6,500,000	180



## Tugas Akhir Integrasi

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

1. Tampilkan 5 department dengan total gaji tertinggi

```
select d.name as department,  
sum(e.salary) as total_salary  
from departments d  
left join employees e on d.id = e.department_id  
group by d.name  
order by total_salary desc  
limit 5;
```

A-Z department ▼	123 total_salary ▼
finance	27,000,000
it	24,700,000
hr	17,500,000
marketing	16,000,000
operations	6,500,000

2. Tampilkan project dengan rata-rata jam kerja per employee tertinggi

```
select p.name as project,  
avg(ep.hours_worked) as avg_hours_per_employee  
from projects p  
join employee_projects ep on p.id = ep.project_id  
group by p.name  
order by avg_hours_per_employee desc;
```

A-Z project ▼	123 avg_hours_per_employee ▼
payroll system	206.8333333333
recruitment system	190.25
crm implementation	180
marketing campaign	167.75
mobile app	150
inventory system	120
website development	108
database migration	85.25

3. Tampilkan employee yang bekerja di lebih dari 1 project

```
select e.name as employee,
count(ep.project_id) as project_count
from employees e
join employee_projects ep on e.id = ep.employee_id
group by e.id, e.name
having count(ep.project_id) > 1;
```

AZ employee ▼	123 project_count ▼
siti rahayu	2
budi santoso	2
joko widodo	2

4. Buat laporan performance department: nama department, jumlah employee, jumlah project, total budget projects, rata-rata gaji, total jam kerja

```
select d.name as department,
count(distinct e.id) as employee_count,
count(distinct p.id) as project_count,
coalesce(sum(p.budget),0) as total_project_budget,
coalesce(avg(e.salary),0) as avg_salary,
coalesce(sum(ep.hours_worked),0) as total_hours_worked
from departments d
left join employees e on d.id = e.department_id
left join projects p on d.id = p.department_id
left join employee_projects ep on e.id = ep.employee_id
group by d.id, d.name
order by department;
```

AZ department ▼	123 employee_count ▼	123 project_count ▼	123 total_project_budget ▼	123 avg_salary ▼	123 total_hours_worked ▼
finance	3	1	13,500,000	9,000,000	620.5
hr	2	1	6,000,000	8,750,000	380.5
it	3	3	96,000,000	3,333.3333333333	2,029.5
marketing	2	1	12,000,000	8,000,000	335.5
operations	1	1	5,500,000	6,500,000	180

## **Refleksi Pembelajaran**

**Nama Mahasiswa:** Andini Tribuana

**NIM:** 244107060140

### **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:**

Materi bagian CTE dan Sub Query masihlah bingung untuk saya pahami. Bagian Join saya cukup paham meskipun masih sering terbolak-balik penggunaannya.