

JOBSCHEET LEVEL DASAR: WEB CRUD DENGAN POSTGRESQL & PHP

TUJUAN PEMBELAJARAN

Setelah menyelesaikan jobsheet ini, mahasiswa mampu:

- Membuat koneksi PHP dengan PostgreSQL
- Mengimplementasikan operasi CRUD (Create, Read, Update, Delete)
- Menggunakan fungsi agregat PostgreSQL dalam query
- Membuat dan menggunakan View & Materialized View
- Membuat dashboard sederhana dengan data statistik

ALAT DAN BAHAN YANG DIPERLUKAN

- **PostgreSQL** 12 atau versi lebih baru
- **PHP** 7.4 atau versi lebih baru
- **Web Server** (XAMPP, WAMP, atau Apache)
- **Text Editor** (VS Code, Sublime Text, Notepad++)
- **Web Browser** (Chrome, Firefox)

LANGKAH 1: PERSIAPAN DATABASE

1.1 Membuat Database dan Tabel

```
-- Langkah 1: Buat database baru  
CREATE DATABASE company_db;
```

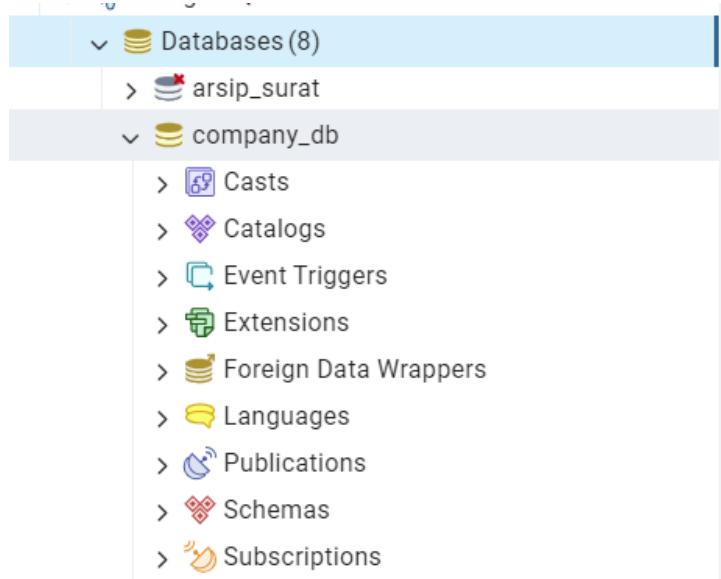
```
CREATE DATABASE
```

```
Query returned successfully in 967 msec.
```

```
-- Langkah 2: Masuk ke database
```

```
Pergi ke panel kiri(Object Browser)
```

```
Klik kanan pada databases lalu pilih refresh, kemudian klik nama database  
company_db yang sudah dibuat
```



```
-- Langkah 3: Buat tabel employees
```

```
CREATE TABLE employees (  
    id SERIAL PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    department VARCHAR(50),  
    position VARCHAR(50),  
    salary DECIMAL(10,2),  
    hire_date DATE,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
-- Langkah 4: Buat tabel projects
```

```
CREATE TABLE projects (  
    id SERIAL PRIMARY KEY,  
    project_name VARCHAR(100) NOT NULL,  
    department VARCHAR(50),  
    budget DECIMAL(12,2),  
    start_date DATE,  
    end_date DATE,  
    status VARCHAR(20) DEFAULT 'Active'  
);
```

```
CREATE TABLE
```

```
Query returned successfully in 67 msec.
```

1.2 Mengisi Data Sample (Minimal 20 Records)

```
-- Mengisi data employees (20 records)
INSERT INTO employees (first_name, last_name, email, department, position,
salary, hire_date) VALUES
('Ahmad', 'Hidayat', 'ahmad.hidayat@company.com', 'IT', 'Software Engineer',
8500000, '2022-01-15'),
('Siti', 'Rahayu', 'siti.rahayu@company.com', 'HR', 'HR Manager', 9500000,
'2021-03-20'),
('Budi', 'Santoso', 'budi.santoso@company.com', 'Finance', 'Accountant',
7500000, '2022-06-10'),
('Dewi', 'Anggraini', 'dewi.anggraini@company.com', 'Marketing', 'Marketing
Specialist', 7000000, '2023-02-01'),
('Rizki', 'Pratama', 'rizki.pratama@company.com', 'IT', 'System Administrator',
8000000, '2021-11-05'),
('Maya', 'Sari', 'maya.sari@company.com', 'Finance', 'Financial Analyst',
8200000, '2022-08-15'),
('Joko', 'Widodo', 'joko.widodo@company.com', 'Operations', 'Operations
Manager', 11000000, '2020-05-12'),
('Linda', 'Permata', 'linda.permata@company.com', 'Marketing', 'Digital
Marketer', 6800000, '2023-01-30'),
('Fajar', 'Nugroho', 'fajar.nugroho@company.com', 'IT', 'Web Developer',
7800000, '2022-09-22'),
('Rina', 'Wulandari', 'rina.wulandari@company.com', 'HR', 'Recruitment
Specialist', 6500000, '2023-03-10'),
('Hendra', 'Kurniawan', 'hendra.kurniawan@company.com', 'Finance', 'Tax
Officer', 7200000, '2022-04-18'),
('Dian', 'Puspita', 'dian.puspita@company.com', 'IT', 'Database Administrator',
9000000, '2021-07-25'),
('Eko', 'Supriyanto', 'eko.supriyanto@company.com', 'Operations', 'Logistics
Coordinator', 6000000, '2023-05-05'),
('Fitri', 'Handayani', 'fitri.handayani@company.com', 'Marketing', 'Content
Creator', 5800000, '2023-04-12'),
('Gita', 'Maharani', 'gita.maharani@company.com', 'HR', 'Training Officer',
6200000, '2022-12-01'),
('Irfan', 'Setiawan', 'irfan.setiawan@company.com', 'IT', 'Network Engineer',
8300000, '2021-09-14'),
('Kartika', 'Dewi', 'kartika.dewi@company.com', 'Finance', 'Auditor', 7700000,
'2022-03-08'),
('Lukman', 'Hakim', 'lukman.hakim@company.com', 'Operations', 'Warehouse
Manager', 7300000, '2021-12-20'),
('Nina', 'Zahra', 'nina.zahra@company.com', 'Marketing', 'Social Media
Specialist', 5900000, '2023-06-25'),
('Oscar', 'Fernando', 'oscar.fernando@company.com', 'IT', 'Mobile Developer',
8100000, '2022-11-11');
```

```
INSERT 0 20
```

```
Query returned successfully in 73 msec.
```

```
-- Mengisi data projects (10 records)
INSERT INTO projects (project_name, department, budget, start_date, end_date, status) VALUES
('Website Redesign', 'IT', 150000000, '2024-01-01', '2024-06-30', 'Active'),
('Employee Training Program', 'HR', 75000000, '2024-02-01', '2024-05-31', 'Active'),
('Financial System Upgrade', 'Finance', 200000000, '2024-03-01', '2024-08-31', 'Planning'),
('Digital Marketing Campaign', 'Marketing', 120000000, '2024-01-15', '2024-04-30', 'Active'),
('Warehouse Optimization', 'Operations', 90000000, '2024-02-20', '2024-07-15', 'Active'),
('Mobile App Development', 'IT', 180000000, '2024-03-10', '2024-09-30', 'Planning'),
('Recruitment Drive 2024', 'HR', 50000000, '2024-04-01', '2024-06-30', 'Active'),
('Customer Satisfaction Survey', 'Marketing', 40000000, '2024-03-15', '2024-05-31', 'Active'),
('Network Infrastructure', 'IT', 220000000, '2024-05-01', '2024-12-31', 'Planning'),
('Annual Audit Preparation', 'Finance', 60000000, '2024-04-15', '2024-07-31', 'Active');
```

```
INSERT 0 10
```

Query returned successfully in 80 msec.

1.3 Membuat View dan Materialized View

```
-- VIEW 1: Employee Summary dengan informasi lengkap
CREATE VIEW employee_summary AS
SELECT
    id,
    CONCAT(first_name, ' ', last_name) AS full_name,
    department,
    position,
    salary,
    EXTRACT(YEAR FROM AGE(CURRENT_DATE, hire_date)) AS years_of_service,
    CASE
        WHEN EXTRACT(YEAR FROM AGE(CURRENT_DATE, hire_date)) >= 3 THEN 'Senior'
        WHEN EXTRACT(YEAR FROM AGE(CURRENT_DATE, hire_date)) >= 1 THEN
            'Intermediate'
        ELSE 'Junior'
    END AS experience_level
FROM employees
ORDER BY department, salary DESC;
```

CONTOH HASIL:

	id integer	full_name text	department character varying (50)	position character varying (50)	salary numeric (10,2)	years_of_service numeric	experience_level text
1	6	Maya Sari	Finance	Financial Analyst	8200000.00		3 Senior
2	17	Kartika Dewi	Finance	Auditor	7700000.00		3 Senior
3	3	Budi Santoso	Finance	Accountant	7500000.00		3 Senior
4	11	Hendra Kurniawan	Finance	Tax Officer	7200000.00		3 Senior
5	2	Siti Rahayu	HR	HR Manager	9500000.00		4 Senior
6	10	Rina Wulandari	HR	Recruitment Specialist	6500000.00		2 Intermediate
7	15	Gita Maharani	HR	Training Officer	6200000.00		2 Intermediate
8	12	Dian Puspita	IT	Database Administrator	9000000.00		4 Senior
9	1	Ahmad Hidayat	IT	Software Engineer	8500000.00		3 Senior
10	16	Irfan Setiawan	IT	Network Engineer	8300000.00		4 Senior
11	20	Oscar Fernando	IT	Mobile Developer	8100000.00		2 Intermediate
12	5	Rizki Pratama	IT	System Administrator	8000000.00		3 Senior
13	9	Fajar Nugroho	IT	Web Developer	7800000.00		3 Senior
14	4	Dewi Anggraini	Marketing	Marketing Specialist	7000000.00		2 Intermediate
15	8	Linda Permata	Marketing	Digital Marketer	6800000.00		2 Intermediate
16	19	Nina Zahra	Marketing	Social Media Specialist	5900000.00		2 Intermediate
17	14	Fitri Handayani	Marketing	Content Creator	5800000.00		2 Intermediate
18	7	Joko Widodo	Operations	Operations Manager	11000000.00		5 Senior
19	18	Lukman Hakim	Operations	Warehouse Manager	7300000.00		3 Senior
20	13	Eko Supriyanto	Operations	Logistics Coordinator	6000000.00		2 Intermediate

```
-- VIEW 2: Department Statistics
CREATE VIEW department_stats AS
SELECT
    department,
    COUNT(*) as total_employees,
    ROUND(AVG(salary), 2) as avg_salary,
    MIN(salary) as min_salary,
    MAX(salary) as max_salary,
    SUM(salary) as total_salary_budget
FROM employees
GROUP BY department
ORDER BY total_employees DESC;
```

CONTOH HASIL

	department character varying (50)	total_employees bigint	avg_salary numeric	min_salary numeric	max_salary numeric	total_salary_budget numeric
1	IT	6	8283333.33	7800000.00	9000000.00	49700000.00
2	Marketing	4	6375000.00	5800000.00	7000000.00	25500000.00
3	Finance	4	7650000.00	7200000.00	8200000.00	30600000.00
4	Operations	3	8100000.00	6000000.00	11000000.00	24300000.00
5	HR	3	7400000.00	6200000.00	9500000.00	22200000.00

```
-- MATERIALIZED VIEW 1: Dashboard Summary
CREATE MATERIALIZED VIEW dashboard_summary AS
SELECT
    -- Total counts
    (SELECT COUNT(*) FROM employees) as total_employees,
    (SELECT COUNT(*) FROM projects) as total_projects,
    (SELECT COUNT(DISTINCT department) FROM employees) as total_departments,

    -- Salary statistics
    (SELECT ROUND(AVG(salary), 2) FROM employees) as company_avg_salary,
    (SELECT MAX(salary) FROM employees) as highest_salary,
    (SELECT MIN(salary) FROM employees) as lowest_salary,

    -- Project statistics
    (SELECT SUM(budget) FROM projects) as total_project_budget,
    (SELECT COUNT(*) FROM projects WHERE status = 'Active') as active_projects,
    (SELECT COUNT(*) FROM projects WHERE status = 'Planning') as planning_projects,

    -- Employee tenure
    (SELECT ROUND(AVG(EXTRACT(YEAR FROM AGE(CURRENT_DATE, hire_date))), 1) FROM
     employees) as avg_years_service;
```

	total_employees	total_projects	total_departments	company_avg_salary	highest_salary	lowest_salary	total_project_budget	active_projects	planning_projects	avg_years_service
1	20	10	5	7615000.00	11000000.00	5800000.00	118500000.00	7	3	2.9

```
-- Refresh materialized view
REFRESH MATERIALIZED VIEW dashboard_summary;
```

```
REFRESH MATERIALIZED VIEW
```

Query returned successfully in 81 msec.

LANGKAH 2: STRUKTUR PROJECT PHP

Struktur Folder yang Akan Dibuat:

```
employee-management/
├── config/
│   └── database.php
├── models/
│   └── EmployeeModel.php
├── views/
│   ├── header.php
│   ├── footer.php
│   ├── dashboard.php
│   ├── employee_list.php
│   ├── employee_form.php
│   └── department_stats.php
└── css/
    └── style.css
└── index.php
```

LANGKAH 3: IMPLEMENTASI KODE PHP

3.1 config/database.php - Koneksi Database

```
<?php
/**
 * FILE: config/database.php
 * FUNGSI: Membuat koneksi ke database PostgreSQL
 */

class Database {
    // Informasi koneksi database - GANTI dengan settingan lokal Anda
    private $host = "localhost";
    private $port = "5432";
    private $db_name = "company_db";
    private $username = "postgres";        // Default username PostgreSQL
    private $password = "password123";    // Ganti dengan password PostgreSQL Anda
    public $conn;

    // Method untuk mendapatkan koneksi
    public function getConnection() {
        $this->conn = null;

        try {
            // Membuat koneksi PDO ke PostgreSQL
            $dsn = "pgsql:host=" . $this->host . ";port=" . $this->port .
";dbname=" . $this->db_name;
            $this->conn = new PDO($dsn, $this->username, $this->password);

            // Set error mode ke exception untuk handling error yang lebih baik
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

            // echo "Koneksi database berhasil!";

        } catch(PDOException $exception) {
            echo "Error koneksi database: " . $exception->getMessage();
        }

        return $this->conn;
    }
}
?>
```

3.2 models/EmployeeModel.php - Model untuk Operasi Data

```
<?php
/**
 * FILE: models/EmployeeModel.php
 * FUNGSI: Berisi semua operasi database untuk tabel employees
 */

class EmployeeModel {
    private $conn;
```

```

private $table_name = "employees";

// Constructor
public function __construct($db) {
    $this->conn = $db;
}

// METHOD 1: Read semua employees
public function getAllEmployees() {
    $query = "SELECT * FROM " . $this->table_name . " ORDER BY id DESC";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt;
}

// METHOD 2: Create employee baru
public function createEmployee($data) {
    $query = "INSERT INTO " . $this->table_name . " (first_name, last_name,
email, department, position, salary, hire_date) VALUES (:first_name, :last_name,
:email, :department, :position, :salary, :hire_date)";

    $stmt = $this->conn->prepare($query);

    // Bind parameters untuk keamanan (mencegah SQL injection)
    $stmt->bindParam(":first_name", $data['first_name']);
    $stmt->bindParam(":last_name", $data['last_name']);
    $stmt->bindParam(":email", $data['email']);
    $stmt->bindParam(":department", $data['department']);
    $stmt->bindParam(":position", $data['position']);
    $stmt->bindParam(":salary", $data['salary']);
    $stmt->bindParam(":hire_date", $data['hire_date']);

    return $stmt->execute();
}

// METHOD 3: Update employee
public function updateEmployee($id, $data) {
    $query = "UPDATE " . $this->table_name . "
        SET first_name = :first_name, last_name = :last_name,
            email = :email, department = :department,
            position = :position, salary = :salary, hire_date = :hire_date
        WHERE id = :id";

    $stmt = $this->conn->prepare($query);

    $stmt->bindParam(":id", $id);
    $stmt->bindParam(":first_name", $data['first_name']);
    $stmt->bindParam(":last_name", $data['last_name']);
    $stmt->bindParam(":email", $data['email']);
    $stmt->bindParam(":department", $data['department']);
    $stmt->bindParam(":position", $data['position']);
    $stmt->bindParam(":salary", $data['salary']);
    $stmt->bindParam(":hire_date", $data['hire_date']);

    return $stmt->execute();
}

```

```

// METHOD 4: Delete employee
public function deleteEmployee($id) {
    $query = "DELETE FROM " . $this->table_name . " WHERE id = :id";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(":id", $id);
    return $stmt->execute();
}

// METHOD 5: Get single employee by ID
public function getEmployeeById($id) {
    $query = "SELECT * FROM " . $this->table_name . " WHERE id = :id";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(":id", $id);
    $stmt->execute();
    return $stmt->fetch(PDO::FETCH_ASSOC);
}

// METHOD 6: Get data dari VIEW employee_summary
public function getEmployeeSummary() {
    $query = "SELECT * FROM employee_summary";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt;
}

// METHOD 6: Get data dari VIEW employee_summary
public function getEmployeeSummary() {
    $query = "SELECT * FROM employee_summary";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt;
}

// METHOD 7: Get data dari VIEW department_stats
public function getDepartmentStats() {
    $query = "SELECT * FROM department_stats";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt;
}

// METHOD 8: Get data dari MATERIALIZED VIEW dashboard_summary
public function getDashboardSummary() {
    $query = "SELECT * FROM dashboard_summary";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt->fetch(PDO::FETCH_ASSOC);
}

// METHOD 9: Refresh materialized view
public function refreshDashboard() {
    $query = "REFRESH MATERIALIZED VIEW dashboard_summary";
    $stmt = $this->conn->prepare($query);
    return $stmt->execute();
}
}

?>

```

3.3 views/header.php dan views/footer.php- Template Header

header.php

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sistem Manajemen Karyawan</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="container">
        <header class="header">
            <h1>Sistem Manajemen Karyawan</h1>
            <p>Aplikasi CRUD Sederhana dengan PostgreSQL & PHP</p>
        </header>

        <nav class="navbar">
            <a href="index.php?action=dashboard" class="nav-link">Dashboard</a>
            <a href="index.php?action=list" class="nav-link">Data Karyawan</a>
            <a href="index.php?action=create" class="nav-link">Tambah Karyawan</a>
            <a href="index.php?action=department_stats" class="nav-link">Statistik
Departemen</a>
            <a href="index.php?action=refresh" class="nav-link">Refresh
Dashboard</a>
        </nav>

        <main class="main-content">
```

footer.php

```
<?php
/**
 * FILE: views/footer.php
 * FUNGSI: Menutup tag HTML yang dibuka di header.php
 */
?>

    </main>

    <footer class="footer">
        <p>&copy; <?php echo date('Y'); ?> Sistem Manajemen Karyawan. All
rights reserved.</p>
    </footer>

    </div>
</body>
</html>
```

3.4 css/style.css - Styling Dasar

```
/* FILE: css/style.css */
/* FUNGSI: Styling untuk aplikasi web */

/* Reset dasar */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    line-height: 1.6;
    background-color: #f4f4f4;
    color: #333;
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
}

/* Header */
.header {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    padding: 2rem;
    border-radius: 10px;
    margin-bottom: 1rem;
    text-align: center;
}

.header h1 {
    margin-bottom: 0.5rem;
}

/* Navigation */
.navbar {
    background: white;
    padding: 1rem;
    border-radius: 8px;
    margin-bottom: 1rem;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
    display: flex;
    gap: 1rem;
    flex-wrap: wrap;
}

.nav-link {
    text-decoration: none;
    color: #667eea;
    padding: 0.5rem 1rem;
    border-radius: 5px;
```

```
    transition: background 0.3s;
}

.nav-link:hover {
  background: #667eea;
  color: white;
}

/* Main Content */
.main-content {
  background: white;
  padding: 2rem;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

/* Tables */
.data-table {
  width: 100%;
  border-collapse: collapse;
  margin: 1rem 0;
}

.data-table th,
.data-table td {
  padding: 12px;
  text-align: left;
  border-bottom: 1px solid #ddd;
}

.data-table th {
  background-color: #667eea;
  color: white;
}

.data-table tr:hover {
  background-color: #f5f5f5;
}

/* Forms */
.form-group {
  margin-bottom: 1rem;
}

.form-label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: bold;
}

.form-input {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
```

```
    font-size: 16px;
}

.form-input:focus {
    outline: none;
    border-color: #667eea;
}

/* Buttons */
.btn {
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    text-decoration: none;
    display: inline-block;
    font-size: 14px;
    transition: background 0.3s;
}

.btn-primary {
    background: #667eea;
    color: white;
}

.btn-primary:hover {
    background: #5a6fd8;
}

.btn-edit {
    background: #28a745;
    color: white;
}

.btn-edit:hover {
    background: #218838;
}

.btn-delete {
    background: #dc3545;
    color: white;
}

.btn-delete:hover {
    background: #c82333;
}

/* Dashboard Cards */
.dashboard-cards {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 1rem;
    margin: 2rem 0;
}
```

```

.card {
    background: white;
    padding: 1.5rem;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    text-align: center;
    border-left: 4px solid #667eea;
}

.card h3 {
    color: #666;
    margin-bottom: 0.5rem;
    font-size: 0.9rem;
}

.card .number {
    font-size: 2rem;
    font-weight: bold;
    color: #667eea;
}

/* Alert Messages */
.alert {
    padding: 1rem;
    border-radius: 5px;
    margin-bottom: 1rem;
}

.alert-success {
    background: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
}

.alert-error {
    background: #f8d7da;
    color: #721c24;
    border: 1px solid #f5c6cb;
}

/* Responsive */
@media (max-width: 768px) {
    .navbar {
        flex-direction: column;
    }

    .dashboard-cards {
        grid-template-columns: 1fr;
    }
}

```

3.5 views/dashboard.php - Halaman Dashboard

```
<?php
```

```

/***
 * FILE: views/dashboard.php
 * FUNGSI: Menampilkan dashboard dengan data statistik
 */
include 'views/header.php';
?>

<h2>Dashboard Overview</h2>

<!-- Dashboard Cards -->
<div class="dashboard-cards">
    <div class="card">
        <h3>Total Karyawan</h3>
        <div class="number"><?php echo $dashboard['total_employees']; ?></div>
    </div>
    <div class="card">
        <h3>Departemen</h3>
        <div class="number"><?php echo $dashboard['total_departments']; ?></div>
    </div>
    <div class="card">
        <h3>Proyek Aktif</h3>
        <div class="number"><?php echo $dashboard['active_projects']; ?></div>
    </div>
    <div class="card">
        <h3>Total Proyek</h3>
        <div class="number"><?php echo $dashboard['total_projects']; ?></div>
    </div>
</div>

<!-- Statistics Grid -->
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 1rem; margin: 2rem 0;">
    <div class="card">
        <h3>Gaji Rata-rata</h3>
        <div class="number">Rp <?php echo number_format($dashboard['company_avg_salary'], 0, ',', '.'); ?></div>
    </div>
    <div class="card">
        <h3>Gaji Tertinggi</h3>
        <div class="number">Rp <?php echo number_format($dashboard['highest_salary'], 0, ',', '.'); ?></div>
    </div>
    <div class="card">
        <h3>Gaji Terendah</h3>
        <div class="number">Rp <?php echo number_format($dashboard['lowest_salary'], 0, ',', '.'); ?></div>
    </div>
    <div class="card">
        <h3>Masa Kerja Rata-rata</h3>
        <div class="number"><?php echo $dashboard['avg_years_service']; ?>tahun</div>
    </div>
</div>

<!-- Budget Information -->

```

```

<div class="card">
    <h3>Total Budget Proyek</h3>
    <div class="number" style="font-size: 1.5rem;">
        Rp <?php echo number_format($dashboard['total_project_budget'], 0, ',', '.') ; ?>
    </div>
    <p style="margin-top: 0.5rem; color: #666;">
        <?php echo $dashboard['active_projects']; ?> proyek aktif | 
        <?php echo $dashboard['planning_projects']; ?> proyek dalam perencanaan
    </p>
</div>

<?php include 'views/footer.php'; ?>

```

3.6 views/employee_list.php – Menampilkan semua karyawan

```

<?php
/**
 * FILE: views/employee_list.php
 * FUNGSI: Menampilkan daftar semua karyawan dalam bentuk tabel
 */
include 'views/header.php';
?>

<h2>Daftar Karyawan</h2>

<!-- Pesan Sukses -->
<?php if (isset($_GET['message'])): ?>
    <div class="alert alert-success">
        <?php
            $messages = [
                'created' => 'Karyawan berhasil ditambahkan!',
                'updated' => 'Data karyawan berhasil diupdate!',
                'deleted' => 'Karyawan berhasil dihapus!'
            ];
            echo $messages[$_GET['message']] ?? 'Operasi berhasil!';
        ?>
    </div>
<?php endif; ?>

<!-- Tombol Tambah Karyawan -->
<div style="margin-bottom: 1rem;">
    <a href="index.php?action=create" class="btn btn-primary">Tambah Karyawan Baru</a>
</div>

<!-- Tabel Karyawan -->
<table class="data-table">
    <thead>
        <tr>
            <th>ID</th>
            <th>Nama Lengkap</th>
            <th>Email</th>
            <th>Departemen</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>John Doe</td>
            <td>john.doe@example.com</td>
            <td>IT</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Jane Smith</td>
            <td>jane.smith@example.com</td>
            <td>HR</td>
        </tr>
        <tr>
            <td>3</td>
            <td>Mike Johnson</td>
            <td>mike.johnson@example.com</td>
            <td>Sales</td>
        </tr>
    </tbody>
</table>

```

```

<th>Posisi</th>
<th>Gaji</th>
<th>Tanggal Mulai</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
    <?php if ($employees->rowCount() > 0): ?>
        <?php while ($row = $employees->fetch(PDO::FETCH_ASSOC)): ?>
            <tr>
                <td><?php echo htmlspecialchars($row['id']); ?></td>
                <td>
                    <strong><?php echo htmlspecialchars($row['first_name'] . ' ' .
$row['last_name']); ?></strong>
                </td>
                <td><?php echo htmlspecialchars($row['email']); ?></td>
                <td>
                    <span style="padding: 0.25rem 0.5rem; background: #e9ecef;
border-radius: 4px;">
                        <?php echo htmlspecialchars($row['department']); ?>
                    </span>
                </td>
                <td><?php echo htmlspecialchars($row['position']); ?></td>
                <td>
                    <strong>Rp <?php echo number_format($row['salary'], 0, ',', ,
'.'); ?></strong>
                </td>
                <td><?php echo date('d/m/Y', strtotime($row['hire_date'])); ?></td>
                <td>
                    <div style="display: flex; gap: 0.5rem;">
                        <a href="index.php?action=edit&id=<?php echo $row['id'];
?>">
                            class="btn btn-edit"
                            title="Edit karyawan">
                                Edit
                            </a>
                        <a href="index.php?action=delete&id=<?php echo $row['id'];
?>">
                            class="btn btn-delete"
                            title="Hapus karyawan"
                            onclick="return confirm('Apakah Anda yakin ingin
menghapus karyawan <?php echo htmlspecialchars($row['first_name'] . ' ' .
$row['last_name']); ?>?')">
                                Hapus
                            </a>
                        </div>
                    </td>
                </tr>
            <?php endwhile; ?>
        <?php else: ?>
            <tr>
                <td colspan="8" style="text-align: center; padding: 2rem;">
                    <p>Tidak ada data karyawan.</p>
                    <a href="index.php?action=create" class="btn btn-
primary">Tambah Karyawan Pertama</a>
                </td>
            </tr>
        <?php endif; ?>
    </tbody>
</table>

```

```

        </td>
    </tr>
    <?php endif; ?>
</tbody>
</table>

<!-- Informasi Jumlah Data --&gt;
&lt;div style="margin-top: 1rem; padding: 1rem; background: #f8f9fa; border-radius: 5px;"&gt;
    &lt;strong&gt;Total Data:&lt;/strong&gt; &lt;?php echo $employees-&gt;rowCount(); ?&gt; karyawan
&lt;/div&gt;

&lt;?php include 'views/footer.php'; ?&gt;
</pre>

```

3.7 views/employee_form.php – Mengedit data karyawan

```

<?php
/**
 * FILE: views/employee_form.php
 * FUNGSI: Form untuk menambah atau mengedit data karyawan
 */
include 'views/header.php';

// Tentukan mode (edit atau create)
$is_edit = isset($employee) && $employee;
$form_title = $is_edit ? 'Edit Data Karyawan' : 'Tambah Karyawan Baru';
$button_text = $is_edit ? 'Update Data' : 'Simpan Data';
$form_action = $is_edit ? "index.php?action=edit&id={$employee['id']}":
"index.php?action=create";
?>

<h2><?php echo $form_title; ?></h2>

<!-- Pesan Error --&gt;
&lt;?php if (isset($error)): ?&gt;
    &lt;div class="alert alert-error"&gt;
        &lt;?php echo $error; ?&gt;
    &lt;/div&gt;
&lt;?php endif; ?&gt;

<!-- Form --&gt;
&lt;form method="POST" action="&lt;?php echo $form_action; ?&gt;"&gt;

    <!-- Hidden ID untuk mode edit --&gt;
    &lt;?php if ($is_edit): ?&gt;
        &lt;input type="hidden" name="product_id" value="&lt;?php echo $employee['id'];
?&gt;"&gt;
    &lt;?php endif; ?&gt;

    &lt;div class="form-group"&gt;
        &lt;label for="first_name" class="form-label"&gt;Nama Depan *&lt;/label&gt;
        &lt;input type="text"
            id="first_name"
            name="first_name"
</pre>

```

```

        class="form-input"
        value=<?php echo $is_edit ?
htmlspecialchars($employee['first_name']) : ''; ?>
    required
    placeholder="Masukkan nama depan">
</div>

<div class="form-group">
    <label for="last_name" class="form-label">Nama Belakang *</label>
    <input type="text"
        id="last_name"
        name="last_name"
        class="form-input"
        value=<?php echo $is_edit ?
htmlspecialchars($employee['last_name']) : ''; ?>
    required
    placeholder="Masukkan nama belakang">
</div>

<div class="form-group">
    <label for="email" class="form-label">Email *</label>
    <input type="email"
        id="email"
        name="email"
        class="form-input"
        value=<?php echo $is_edit ? htmlspecialchars($employee['email']) :
'; ?>
    required
    placeholder="contoh@company.com">
</div>

<div class="form-group">
    <label for="department" class="form-label">Departemen *</label>
    <select id="department" name="department" class="form-input" required>
        <option value="">Pilih Departemen</option>
        <option value="IT" <?php echo ($is_edit && $employee['department'] ==
'IT') ? 'selected' : ''; ?>>IT</option>
        <option value="HR" <?php echo ($is_edit && $employee['department'] ==
'HR') ? 'selected' : ''; ?>>HR</option>
        <option value="Finance" <?php echo ($is_edit && $employee['department'] ==
'Finance') ? 'selected' : ''; ?>>Finance</option>
        <option value="Marketing" <?php echo ($is_edit &&
$employee['department'] == 'Marketing') ? 'selected' : ''; ?>>Marketing</option>
        <option value="Operations" <?php echo ($is_edit &&
$employee['department'] == 'Operations') ? 'selected' : ''; ?>>Operations</option>
        <option value="Sales" <?php echo ($is_edit && $employee['department'] ==
'Sales') ? 'selected' : ''; ?>>Sales</option>
    </select>
</div>

<div class="form-group">
    <label for="position" class="form-label">Posisi *</label>
    <input type="text"
        id="position"
        name="position"

```

```

        class="form-input"
        value="php echo $is_edit ? htmlspecialchars($employee['position']) : ''; ?&gt;"
        required
        placeholder="Masukkan posisi/jabatan"
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label for="salary" class="form-label"&gt;Gaji (Rp) *&lt;/label&gt;
        &lt;input type="number"
            id="salary"
            name="salary"
            class="form-input"
            value="<?php echo $is_edit ? $employee['salary'] : ''; ?&gt;"
            required
            min="0"
            step="100000"
            placeholder="Contoh: 5000000"&gt;
        &lt;small style="color: #666;"&gt;Isikan angka tanpa titik (contoh: 5000000 untuk
5 juta)&lt;/small&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label for="hire_date" class="form-label"&gt;Tanggal Mulai Bekerja *&lt;/label&gt;
        &lt;input type="date"
            id="hire_date"
            name="hire_date"
            class="form-input"
            value="<?php echo $is_edit ? $employee['hire_date'] : ''; ?&gt;"
            required&gt;
    &lt;/div&gt;

    &lt;div class="form-group" style="display: flex; gap: 1rem; margin-top: 2rem;"&gt;
        &lt;button type="submit" class="btn btn-primary"&gt;
            &lt;?php echo $button_text; ?&gt;
        &lt;/button&gt;
        &lt;a href="index.php?action=list" class="btn" style="background: #6c757d;
color: white;"&gt;
            Kembali ke Daftar
        &lt;/a&gt;
        &lt;?php if ($is_edit): ?&gt;
            &lt;a href="index.php?action=delete&amp;id=&lt;?php echo $employee['id']; ?&gt;"&gt;
                class="btn btn-delete"
                onclick="return confirm('Yakin ingin menghapus karyawan &lt;?php echo
htmlspecialchars($employee['first_name']) . ' ' . $employee['last_name']); ?&gt;?')"&gt;
                    Hapus Karyawan
                &lt;/a&gt;
            &lt;?php endif; ?&gt;
    &lt;/div&gt;

&lt;/form&gt;

&lt;!-- JavaScript untuk validasi tambahan --&gt;
&lt;script&gt;
document.getElementById('salary').addEventListener('input', function(e) {
</pre

```

```

// Format tampilan salary dengan separator
let value = e.target.value.replace(/\D/g, '');
if (value) {
    e.target.value = parseInt(value).toLocaleString('id-ID');
}
});

// Validasi form sebelum submit
document.querySelector('form').addEventListener('submit', function(e) {
    const salary = document.getElementById('salary').value;
    const hireDate = document.getElementById('hire_date').value;
    const today = new Date().toISOString().split('T')[0];

    // Validasi hire date tidak boleh lebih dari hari ini
    if (hireDate > today) {
        alert('Tanggal mulai bekerja tidak boleh lebih dari hari ini!');
        e.preventDefault();
        return false;
    }

    // Validasi salary minimal
    if (parseInt(salary.replace(/\D/g, '')) < 1000000) {
        alert('Gaji minimal Rp 1.000.000!');
        e.preventDefault();
        return false;
    }
});
</script>

<?php include 'views/footer.php'; ?>

```

3.8 views/department_stats.php – Mengedit data karyawan

```

<?php
/**
 * FILE: views/department_stats.php
 * FUNGSI: Menampilkan statistik karyawan per departemen menggunakan VIEW database
 */
include 'views/header.php';
?>

<h2>Statistik Departemen</h2>

<p style="margin-bottom: 2rem; color: #666;">
    Data statistik berikut diambil dari VIEW <code>department_stats</code> di
    database PostgreSQL.
</p>

<?php if ($stats->rowCount() > 0): ?>
    <!-- Cards Summary -->
    <div class="dashboard-cards">
        <div class="card">
            <h3>Total Departemen</h3>
            <div class="number"><?php echo $stats->rowCount(); ?></div>
        </div>
    </div>

```

```

<?php
// Hitung total statistics
$stats->execute();
$all_stats = $stats->fetchAll(PDO::FETCH_ASSOC);

$total_employees = array_sum(array_column($all_stats, 'total_employees'));
$total_salary_budget = array_sum(array_column($all_stats,
'total_salary_budget'));
$avg_salary_all = $total_salary_budget / $total_employees;
?>
<div class="card">
    <h3>Total Karyawan</h3>
    <div class="number"><?php echo $total_employees; ?></div>
</div>
<div class="card">
    <h3>Total Budget Gaji</h3>
    <div class="number">Rp <?php echo number_format($total_salary_budget,
0, ',', '.'); ?></div>
</div>
<div class="card">
    <h3>Rata-rata Gaji</h3>
    <div class="number">Rp <?php echo number_format($avg_salary_all, 0,
',', '.'); ?></div>
</div>
</div>

<!-- Tabel Statistik Detail -->
<table class="data-table">
    <thead>
        <tr>
            <th>Departemen</th>
            <th>Jumlah Karyawan</th>
            <th>Gaji Rata-rata</th>
            <th>Gaji Terendah</th>
            <th>Gaji Tertinggi</th>
            <th>Total Budget</th>
        </tr>
    </thead>
    <tbody>
        <?php foreach ($all_stats as $dept): ?>
        <tr>
            <td>
                <strong><?php echo htmlspecialchars($dept['department']); ?></strong>
            </td>
            <td style="text-align: center;">
                <span style="padding: 0.25rem 0.75rem; background: #667eea;
color: white; border-radius: 20px;">
                    <?php echo $dept['total_employees']; ?>
                </span>
            </td>
            <td>
                <strong>Rp <?php echo number_format($dept['avg_salary'], 0,
',', '.'); ?></strong>
            </td>
        </tr>
    </tbody>
</table>

```

```

        <td>Rp <?php echo number_format($dept['min_salary'], 0, ',', '.');
?></td>
        <td>Rp <?php echo number_format($dept['max_salary'], 0, ',', '.');
?></td>
        <td>
            <strong style="color: #27ae60;">
                Rp <?php echo number_format($dept['total_salary_budget'], 0, ',', '.'); ?>
            </strong>
        </td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>


<div style="margin-top: 3rem;">
    <h3>Visualisasi Data</h3>

    
    <div style="background: white; padding: 1.5rem; border-radius: 8px; margin: 1rem 0; border-left: 4px solid #667eea;">
        <h4>Gaji Rata-rata per Departemen</h4>
        <?php foreach ($all_stats as $dept): ?>
            <div style="margin: 0.5rem 0;">
                <div style="display: flex; justify-content: space-between; margin-bottom: 0.25rem;">
                    <span><?php echo htmlspecialchars($dept['department']); ?></span>
                    <span>Rp <?php echo number_format($dept['avg_salary'], 0, ',', '.'); ?></span>
                </div>
                <div style="background: #f0f0f0; border-radius: 4px; height: 20px;">
                    <div style="background: #667eea; height: 100%; border-radius: 4px; width: <?php echo ($dept['avg_salary'] / max(array_column($all_stats, 'avg_salary')) * 100); ?>%;"></div>
                </div>
            </div>
        <?php endforeach; ?>
    </div>

    
    <div style="background: white; padding: 1.5rem; border-radius: 8px; margin: 1rem 0; border-left: 4px solid #27ae60;">
        <h4>Jumlah Karyawan per Departemen</h4>
        <?php foreach ($all_stats as $dept): ?>
            <div style="margin: 0.5rem 0;">
                <div style="display: flex; justify-content: space-between; margin-bottom: 0.25rem;">
                    <span><?php echo htmlspecialchars($dept['department']); ?></span>
                    <span><?php echo $dept['total_employees']; ?> orang</span>
                </div>
            </div>
        <?php endforeach; ?>
    </div>

```

```

        <div style="background: #f0f0f0; border-radius: 4px; height: 20px;">
            <div style="background: #27ae60; height: 100%; border-radius: 4px; width: <?php echo ($dept['total_employees'] / max(array_column($all_stats, 'total_employees')) * 100); ?%;>"></div>
        </div>
    </div>
    <?php endforeach; ?>
</div>

<?php else: ?>
    <div style="text-align: center; padding: 3rem; background: #f8f9fa; border-radius: 8px;">
        <p style="font-size: 1.2rem; color: #666;"> X Tidak ada data statistik departemen.</p>
        <p style="color: #999;">Pastikan sudah ada data karyawan dan VIEW <code>department_stats</code> sudah dibuat di database.</p>
        <a href="index.php?action=create" class="btn btn-primary" style="margin-top: 1rem;">Tambah Data Karyawan</a>
    </div>
<?php endif; ?>

<div style="margin-top: 2rem; padding: 1rem; background: #e7f3ff; border-radius: 5px;">
    <strong>💡 Informasi:</strong>
    Data ini di-generate secara real-time dari VIEW PostgreSQL yang menggunakan fungsi agregat
    <code>COUNT()</code>, <code>AVG()</code>, <code>MIN()</code>, <code>MAX()</code>, dan <code>SUM()</code>.
</div>

<?php include 'views/footer.php'; ?>

```

3.9 index.php - Main Controller

```

<?php
/**
 * FILE: index.php
 * FUNGSI: Controller utama yang menangani semua request
 */

// Include file yang diperlukan
require_once 'config/database.php';
require_once 'models/EmployeeModel.php';

// Inisialisasi database dan model
$database = new Database();
$db = $database->getConnection();
$employeeModel = new EmployeeModel($db);

// Tangani action dari URL
$action = isset($_GET['action']) ? $_GET['action'] : 'dashboard';

```

```

switch ($action) {
    case 'dashboard':
        $dashboard = $employeeModel->getDashboardSummary();
        include 'views/dashboard.php';
        break;

    case 'list':
        $employees = $employeeModel->getAllEmployees();
        include 'views/employee_list.php';
        break;

    case 'create':
        if ($_POST) {
            // Proses form submission
            $data = [
                'first_name' => $_POST['first_name'],
                'last_name' => $_POST['last_name'],
                'email' => $_POST['email'],
                'department' => $_POST['department'],
                'position' => $_POST['position'],
                'salary' => $_POST['salary'],
                'hire_date' => $_POST['hire_date']
            ];

            if ($employeeModel->createEmployee($data)) {
                header("Location: index.php?action=list&message=created");
            } else {
                $error = "Gagal menambah karyawan";
            }
        }
        include 'views/employee_form.php';
        break;

    case 'edit':
        $id = $_GET['id'];

        if ($_POST) {
            $data = [
                'first_name' => $_POST['first_name'],
                'last_name' => $_POST['last_name'],
                'email' => $_POST['email'],
                'department' => $_POST['department'],
                'position' => $_POST['position'],
                'salary' => $_POST['salary'],
                'hire_date' => $_POST['hire_date']
            ];

            if ($employeeModel->updateEmployee($id, $data)) {
                header("Location: index.php?action=list&message=updated");
            } else {
                $error = "Gagal mengupdate karyawan";
            }
        }
}

```

```
$employee = $employeeModel->getEmployeeById($id);
include 'views/employee_form.php';
break;

case 'delete':
    $id = $_GET['id'];
    if ($employeeModel->deleteEmployee($id)) {
        header("Location: index.php?action=list&message=deleted");
    } else {
        header("Location: index.php?action=list&message=delete_error");
    }
break;

case 'department_stats':
    $stats = $employeeModel->getDepartmentStats();
    include 'views/department_stats.php';
break;

case 'refresh':
    $employeeModel->refreshDashboard();
    header("Location: index.php?action=dashboard&message=refreshed");
break;

default:
    $dashboard = $employeeModel->getDashboardSummary();
    include 'views/dashboard.php';
break;
}
?>
```

CONTOH HASIL:

Sistem Manajemen Karyawan

Aplikasi CRUD Sederhana dengan PostgreSQL & PHP

Dashboard Data Karyawan Tambah Karyawan Statistik Departemen Refresh Dashboard

Dashboard Overview

Total Karyawan 20	Departemen 5	Proyek Aktif 7	Total Proyek 10
Gaji Rata-rata Rp 7.615.000	Gaji Tertinggi Rp 11.000.000	Gaji Terendah Rp 5.800.000	
Masa Kerja Rata-rata 2.9 tahun			

Total Budget Proyek
Rp 1.185.000.000

7 proyek aktif | 3 proyek dalam perencanaan

LANGKAH 4: TESTING APLIKASI

4.1 Checklist Testing

- **Koneksi Database:** Buka index.php, pastikan tidak ada error

Screenshoot disini

- **Dashboard:** Akses index.php?action=dashboard, lihat statistic

Screenshoot disini

- **List Karyawan:** Akses index.php?action=list, pastikan 20 data muncul

Screenshot disini

- **Tambah Karyawan:** Coba tambah karyawan baru

Screenshot disini

- **Edit Karyawan:** Edit data karyawan yang sudah ada

Screenshot disini

- **Hapus Karyawan:** Hapus satu karyawan (pastikan konfirmasi)

Screenshot disini

- **Statistik Departemen:** Lihat halaman statistic

Screenshot disini

- **Refresh Dashboard:** Test fungsi refresh materialized view

Screenshot disini

4.2 Expected Results

- Semua operasi CRUD bekerja dengan baik
- Data dari View dan Materialized View tampil benar
- Dashboard menampilkan statistik yang akurat
- Tidak ada error PHP atau database

TIPS DAN CATATAN

1. **Pastikan** extension pdo_pgsql aktif di PHP
2. **Ganti** credential database di config/database.php sesuai environment
3. **Test** secara bertahap setiap kali menambah fitur baru
4. **Backup** database secara berkala selama development
5. **Gunakan** PostgreSQL 12+ untuk compatibility terbaik

TUGAS YANG HARUS DISELESAIKAN

TUGAS 1: IMPLEMENTASI FUNGSI AGREGAT PADA WEB

Buat **3 halaman statistik baru** yang menggunakan fungsi agregat PostgreSQL:

1. Halaman Statistik Gaji (salary_stats.php)

- Tampilkan: Rata-rata gaji, gaji tertinggi, gaji terendah per departemen
- Gunakan fungsi: AVG(), MAX(), MIN(), GROUP BY

2. Halaman Masa Kerja (tenure_stats.php)

- Tampilkan: Karyawan berdasarkan masa kerja (Junior: <1 tahun, Middle: 1-3 tahun, Senior: >3 tahun)
- Gunakan fungsi: COUNT(), CASE WHEN, GROUP BY

3. Halaman Ringkasan Karyawan (employee_overview.php)

- Tampilkan: Total karyawan, total gaji per bulan, rata-rata masa kerja
- Gunakan fungsi: COUNT(), SUM(), AVG()

TUGAS 2: BUAT TABLE FUNCTION POSTGRESQL

Buat **2 table function** di PostgreSQL:

1. Function: get_employees_by_salary_range(min_salary DECIMAL, max_salary DECIMAL)

- Mengembalikan: karyawan dengan gaji dalam range tertentu
- Kolom: id, full_name, department, position, salary

2. Function: get_department_summary()

- Mengembalikan: ringkasan statistik per departemen
- Kolom: department, employee_count, avg_salary, total_budget