

## **Laporan Praktikum CRUD Dan Login PHP Native**



Nama Anggota Kelompok :

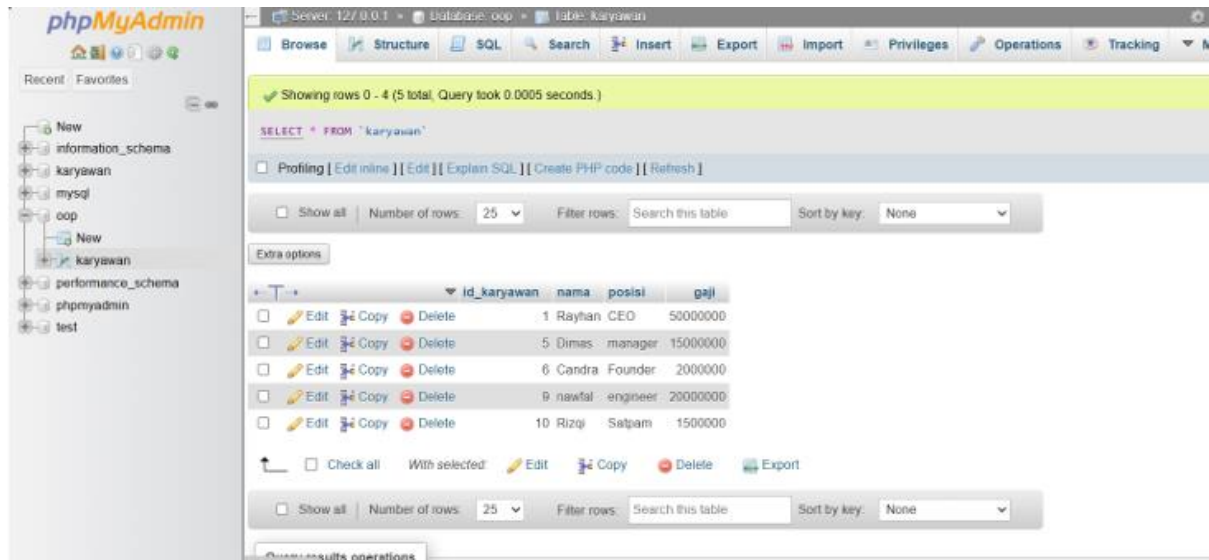
- Arya Adly Ilham P (5)
- Nichy Surya Pradana (23)
- Rayhan Ahmad Karnegi (28)
- Vallentino Marcel (33)

**REKAYASA PERANGKAT LUNAK**  
**SMK TEXMACO SEMARANG**

2024

## I. Tahapan

### 1. Membuat Database Karyawan



### 2. Membuat Koneksi dengan nama file connect.php

```
<?php
class Database {
    private $host = "localhost";
    private $db_name = "oop";
    private $username = "root";
    private $password = "";
    public $conn;

    public function getConnection() {
        $this->conn = null;
        try {
            $this->conn = new PDO("mysql:host=" . $this->host .
";port=3306;dbname=" . $this->db_name, $this->username, $this-
>password);
            $this->conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
            $this->conn->exec("set names utf8");
        } catch(PDOException $exception) {
            echo "Connection error: " . $exception->getMessage();
        }
        return $this->conn;
    }
}
```

```
}
```

```
?>
```

Penjelasan : Kode PHP di atas mendefinisikan kelas Database yang berfungsi untuk menghubungkan aplikasi dengan database MySQL menggunakan PDO (PHP Data Objects). Kelas ini memiliki atribut untuk menyimpan informasi koneksi seperti host, db\_name, username, dan password. Metode getConnection() berfungsi untuk membuat koneksi ke database dengan menggunakan informasi tersebut. Jika koneksi berhasil, PDO akan diatur dalam mode kesalahan (ERRMODE\_EXCEPTION) untuk menampilkan error secara eksplisit jika terjadi kegagalan. Metode ini mengembalikan objek koneksi, yang nantinya dapat digunakan oleh kelas atau fungsi lain untuk mengakses dan memanipulasi database. Jika terjadi kesalahan koneksi, pesan error akan ditampilkan.

### 3. Menampilkan Data/Output

```
<?php
include './config/connect.php';
include './classes/karyawan.php';

$db = new Database();
$karyawan = new Karyawan($db->getConnection());

session_start();
if (!isset($_SESSION['username'])) {
    echo "<script>alert('anda belum login.Silahkan login terlebih dahulu.');
```

```

        <div class="p-3">
            <div class="overflow-x-auto">
                <table class="table-auto w-full">
                    <thead class="text-xs font-semibold
uppercase text-gray-400 bg-gray-50">
                        <tr>
                            <th class="p-2 whitespace-nowrap">
                                <div class="font-semibold text-
left">ID</div>
                            </th>
                            <th class="p-2 whitespace-nowrap">
                                <div class="font-semibold text-
left">Name</div>
                            </th>
                            <th class="p-2 whitespace-nowrap">
                                <div class="font-semibold text-
center">Posisi</div>
                            </th>
                            <th class="p-2 whitespace-nowrap">
                                <div class="font-semibold text-
left">Gaji</div>
                            </th>
                        </tr>
                    </thead>
                    <tbody class="text-sm divide-y divide-gray-
100">
                        <?php
                        $no = 1;
                        foreach($karyawan->read() as $x){
                            $id = $x["id_karyawan"];
                            $nama = $x["nama"];
                            $posisi = $x["posisi"];
                            $gaji = $x["gaji"];
                            $f_gaji = number_format($x["gaji"]);
                            $dimas = ["IMG-20241004-WA0001.jpg",
"IMG-20241004-WA0004.jpg", "SmartSelect_20241018_080405_Photos.jpg",
"SmartSelect_20241018_080442_Photos.jpg",
"SmartSelect_20241018_080620_Photos.jpg"];
                            $farrel =
$dimas[array_rand($dimas)];

                            echo "
                            <tr>
                                <td class='p-2 whitespace-
nowrap'>
                                    <div class='text-
left'>$id</div>
                                </td>






```

```

                <td class='p-2 whitespace-
nowrap'>
                    <div class='flex items-
center'>
                        <div class='w-10 h-10
flex-shrink-0 mr-2 sm:mr-3'><img class='rounded-full'
src='https://raw.githubusercontent.com/Nopalaryan/A/main/$farrel'
width='40' height='40' alt='$nama'></div>
                        <div class='font-medium text-
gray-800'>$nama</div>
                    </div>
                </td>
                <td class='p-2 whitespace-
nowrap'>
                    <div class='text-
left'>$posisi</div>
                </td>
                <td class='p-2 whitespace-
nowrap'>
                    <div class='text-left font-
medium text-green-500'>Rp.$f_gaji</div>
                </td>
                <td class='p-2 whitespace-
nowrap'>
                    <div class='text-lg text-
center'>
                        </div>
                </td>
                <td class='p-3 px-5 flex
justify-end'><a href=\"./edit.php?id_karyawan=\" . $id . "&nama=\" .
$nama . "&posisi=\" . $posisi . "&gaji=\" . $gaji . "\"" type='button'
class='mr-3 text-sm bg-blue-500 hover:bg-blue-700 text-white py-1
px-2 rounded focus:outline-none focus:shadow-outline'>Edt</a><button
onclick=\"if (confirm('yakin hapus ini?')) window.location.href =
'./delete_action.php?id_karyawan=\" . $id . \"';\" type='button'
class='text-sm bg-red-500 hover:bg-red-700 text-white py-1 px-2
rounded focus:outline-none focus:shadow-
outline'>Delete</button></td>
            </tr>";
        }
    ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</section>

```

Kode di atas adalah halaman PHP untuk menampilkan data karyawan yang menggunakan kelas `Karyawan` untuk membaca data dari database, dengan tampilan yang dibuat menggunakan Tailwind CSS. Jika pengguna belum login, mereka diarahkan ke halaman login. Di dalam tabel, setiap karyawan memiliki kolom yang menampilkan ID, nama (dengan gambar profil yang dipilih secara acak dari beberapa pilihan), posisi, dan gaji (diformat dalam rupiah). Tabel ini juga memiliki tombol untuk menambah, mengedit, dan menghapus data karyawan; setiap baris karyawan menyertakan tombol "Edit" yang mengarahkan ke halaman edit sesuai data karyawan dan tombol "Delete" yang meminta konfirmasi sebelum menghapus data karyawan yang bersangkutan.

Data Karyawan				<button>Add Data</button>	
ID	NAME	POSISI	GAJI		
1	 Rayhan	CEO	Rp.50,000,000	<button>Edit</button>	<button>Delete</button>
5	 Dimas	manager	Rp.15,000,000	<button>Edit</button>	<button>Delete</button>
6	 Candra	Founder	Rp.200,000,000	<button>Edit</button>	<button>Delete</button>
9	 nawfal	engineer	Rp.20,000,000	<button>Edit</button>	<button>Delete</button>
12	 riskihibatulill	ob	Rp.500,000	<button>Edit</button>	<button>Delete</button>

## 5. Tambah Data

- create.php

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://cdn.tailwindcss.com"></script>
  <title>Tambah Karyawan</title>
</head>
<body style="background-color: #081E44;" >
  <!-- <form action="create_action.php" method="post">
    Nama:<input type="text" name="nama"><br>
    Posisi:<input type="text" name="posisi"><br>
    Gaji:<input type="text" name="gaji"><br>
    <input type="submit" value ="simpan">
  </form> -->

  <form class="max-w-sm mx-auto mt-10" action="create_action.php"
method="POST">
```

```

    <h1 class="block mb-2 text-lg text-center font-medium text-
gray-900 dark:text-white">Tambah Karyawan</h1>
    <div class="mb-5">
        <label for="nama" class="block mb-2 text-sm font-medium
text-gray-900 dark:text-white">Nama</label>
        <input type="text" id="nama" name="nama" class="shadow-
sm bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-
lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400
dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500
dark:shadow-sm-light" required />
    </div>
    <div class="mb-5">
        <label for="posisi" class="block mb-2 text-sm font-
medium text-gray-900 dark:text-white">Posisi</label>
        <input type="text" id="posisi" name="posisi"
class="shadow-sm bg-gray-50 border border-gray-300 text-gray-900
text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block
w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-
gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-
blue-500 dark:shadow-sm-light" required />
    </div>
    <div class="mb-5">
        <label for="gaji" class="block mb-2 text-sm font-medium
text-gray-900 dark:text-white">Gaji</label>
        <input type="number" id="gaji" name="gaji"
class="shadow-sm bg-gray-50 border border-gray-300 text-gray-900
text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block
w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-
gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-
blue-500 dark:shadow-sm-light" required />
    </div>
    <!-- <div class="flex items-start mb-5">
        <div class="flex items-center h-5">
            <input id="terms" type="checkbox" value="" class="w-4 h-
4 border border-gray-300 rounded bg-gray-50 focus:ring-3 focus:ring-
blue-300 dark:bg-gray-700 dark:border-gray-600 dark:focus:ring-blue-
600 dark:ring-offset-gray-800 dark:focus:ring-offset-gray-800"
required />
        </div>
        <label for="terms" class="ms-2 text-sm font-medium text-
gray-900 dark:text-gray-300">I agree with the <a href="#"
class="text-blue-600 hover:underline dark:text-blue-500">terms and
conditions</a></label>
    </div> -->
    <button type="submit" class="text-white bg-blue-700
hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-
300 font-medium rounded-lg text-sm px-5 py-2.5 text-center dark:bg-

```

```

blue-600 dark: hover: bg-blue-700 dark: focus: ring-blue-800">Register
Data</button>
</form>
</body>

```

Kode di atas adalah halaman HTML untuk menambah data karyawan ke dalam database menggunakan Tailwind CSS untuk tampilan yang responsif dan modern. Formulir ini berisi tiga input yang harus diisi pengguna: Nama, Posisi, dan Gaji, dengan validasi wajib di setiap kolom. Tombol "Register Data" di bagian akhir akan mengirim data ke file `create_action.php` melalui metode POST untuk diproses dan disimpan dalam basis data. Latar belakang halaman berwarna biru gelap (#081E44), memberikan tampilan yang kontras dengan komponen formulir berwarna terang.

-`create_action.php`

```

<?php
include_once 'connect.php';
include_once 'karyawan.php';

$databse = new Database();
$db = $databse->getConnection();

$karyawan = new Karyawan($db);
$karyawan->nama = $_POST['nama'];
$karyawan->posisi = $_POST['posisi'];
$karyawan->gaji = $_POST['gaji'];

$hasil = $karyawan->create();

if($hasil) {
    echo "Karyawan berhasil ditambahkan.";
} else {
    echo "Gagal menambahkan karyawan.";
}
?>

```

Kode PHP di atas digunakan untuk menambah data karyawan baru ke dalam database. Dengan menghubungkan ke file `connect.php` dan `karyawan.php`, kode ini menginisialisasi koneksi database melalui kelas `Database` dan memanfaatkan metode `create()` dari kelas `Karyawan`. Data yang dikirim dari formulir sebelumnya, yaitu nama, posisi, dan gaji, diambil dari `$_POST` dan disimpan dalam properti objek `Karyawan`. Jika eksekusi `create()` berhasil, sistem akan menampilkan pesan "Karyawan berhasil ditambahkan"; jika tidak, pesan "Gagal menambahkan karyawan" akan muncul, memberi tahu pengguna hasil operasi.

-Output Tambah Data



### Tambah Karyawan

Nama

Posisi

Gaji

Karyawan berhasil di tambahkan

- Hasil Output

**Data Karyawan**

ID	NAME	POSISI	GAJI	
1	Rayhan	CEO	Rp.50,000,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>
2	Dimas	manager	Rp.15,000,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>
3	Candra	Founder	Rp.200,000,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>
4	nawfal	engineer	Rp.20,000,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>
5	riskihibatulIII	ob	Rp.500,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>
13	Your name	Your Position	Rp.1,200,000	<input style="background-color: #4a90e2; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Edt"/> <input style="background-color: #e74c3c; color: white; padding: 2px 5px; border: none; border-radius: 3px;" type="button" value="Delete"/>

Data karyawan berhasil di tambah

6.Edit Data

- edit.php

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Edit</title>
  <style>
    /* Styling untuk body */
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f5;
      margin: 0;
      padding: 20px;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    /* Container form */
    form {
      background-color: #fff;
      border-radius: 10px;
      box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);
      padding: 30px;
      width: 350px;
    }

    /* Judul */
    h1 {
      text-align: center;
      color: #333;
      margin-bottom: 40px;
      width: 420px;
    }

    /* Style untuk label input */
    label {
      font-size: 16px;
      color: #555;
    }

    /* Style untuk input field */
    input[type="number"], input[type="text"] {
      width: 100%;
      padding: 8px;
      margin: 10px 0 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
  </style>
</head>
```

```

    }

    /* Style untuk tombol kirim */
    input[type="submit"] {
        width: 100%;
        background-color: #2068E4;
        color: white;
        padding: 10px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }

    /* Hover efek untuk tombol */
    input[type="submit"]:hover {
        background-color: #0A285A;
    }

</style>
</head>
<body>
    <h1>Edit Data Karyawan</h1>
    <form action="./edit_action.php" method="POST">
        Id: <input type="number" name="id_karyawan" id="id" min="0"
<?php $x = $_GET["id_karyawan"];if ($x !== null) echo "value='$x'";
?>><br>
        Nama: <input type="text" name="nama" id="nama" <?php $x =
$_GET["nama"];if ($x !== null) echo "value='$x'"; ?>><br>
        Posisi: <input type="text" name="posisi" id="posisi" <?php
$x = $_GET["posisi"];if ($x !== null) echo "value='$x'"; ?>><br>
        Gaji: <input type="number" name="gaji" id="gaji" min="1"
<?php $x = $_GET["gaji"];if ($x !== null) echo "value='$x'"; ?>><br>
        <input type="submit" value="Kirim">
    </form>
</body>
</html>

```

Kode HTML ini membuat halaman untuk mengedit data karyawan dengan desain form yang rapi. Menggunakan HTML dan CSS, halaman ini menampilkan form yang terdiri dari field input untuk ID, Nama, Posisi, dan Gaji karyawan. Form ini diatur agar data awal diisi otomatis dengan menggunakan parameter `$_GET`, sehingga pengguna dapat melihat informasi yang ada dan memperbaruinya jika diperlukan. Setelah data diubah, pengguna dapat mengirimkan form melalui tombol "Kirim", yang mengarahkan data ke file `edit_action.php` menggunakan metode POST untuk memproses perubahan data karyawan di backend. Desain halaman menggunakan warna latar abu-abu, serta bayangan dan efek hover pada tombol untuk mempercantik tampilan.

- edit\_action.php

```

<?php
include_once 'connect.php';
include_once 'karyawan.php';

$database = new Database();
$db = $database->getConnection();

$karyawan = new Karyawan($db);
$karyawan->id_karyawan = $_POST['id_karyawan'];
$karyawan->nama = $_POST['nama'];
$karyawan->posisi = $_POST['posisi'];
$karyawan->gaji = (int)$_POST['gaji'];

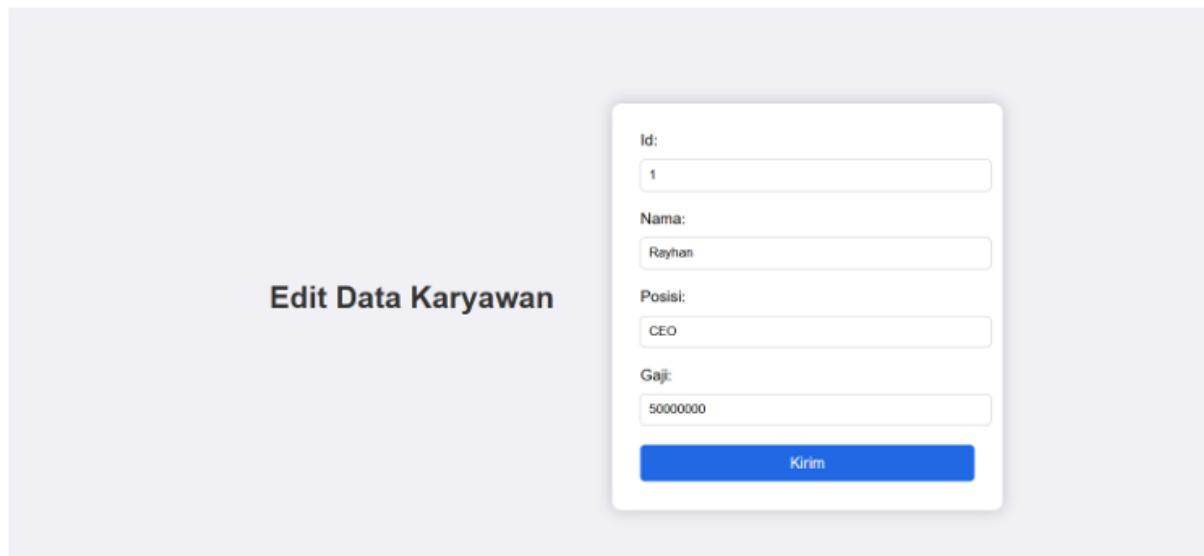
$hasil = $karyawan->edit();

echo "<h1>";
if($hasil) {
    echo "data berhasil di edit.";
} else {
    echo "data tidak bisa di edit.";
}
echo "</h1>";
echo " - <a href=\"../\">Back Home</a>";
?>

```

Kode PHP ini adalah skrip backend untuk memperbarui data karyawan. File ini pertama-tama menyertakan file connect.php dan karyawan.php untuk menghubungkan database dan menggunakan kelas Karyawan. Skrip kemudian membuat objek Karyawan dan menetapkan nilai dari form (dikirim melalui metode POST) ke properti id\_karyawan, nama, posisi, dan gaji. Properti gaji dikonversi ke integer untuk memastikan tipe datanya sesuai. Setelah itu, metode edit() dari objek Karyawan dijalankan untuk memperbarui data di database. Jika pembaruan berhasil, pesan "data berhasil di edit" akan ditampilkan, sedangkan pesan "data tidak bisa di edit" akan muncul jika pembaruan gagal, dengan tautan kembali ke halaman utama untuk navigasi tambahan.

Hasil Output edit



The screenshot shows a web interface with a light purple background. In the center, there is a white rounded rectangle containing a form titled "Edit Data Karyawan". The form has four input fields: "Id:" with the value "1", "Nama:" with the value "Rayhan", "Posisi:" with the value "CEO", and "Gaji:" with the value "50000000". Below these fields is a blue button labeled "Kirim".

## 7.Delete Data

- delete.php

```
<?php
include_once 'connect.php';
include_once 'karyawan.php';

$databse = new Database();
$db = $databse->getConnection();

$karyawan = new Karyawan($db);
$karyawan->id = $_GET["id_karyawan"];

$hasil = $karyawan->delete();

if ($hasil) header("Location: ./");
else echo "<h1>Data Gagal Dihapus<h1><br> - <a href=\"./\">Back
Home</a>";
?>
```

Kode PHP ini adalah skrip untuk menghapus data karyawan berdasarkan id\_karyawan yang diperoleh dari URL menggunakan metode GET. Skrip pertama-tama menyertakan file connect.php dan karyawan.php untuk mengakses koneksi database dan kelas Karyawan. Setelah membuat objek Karyawan dengan koneksi database, skrip menetapkan nilai id dari id\_karyawan yang diterima. Kemudian, metode delete() dari objek Karyawan dipanggil untuk menghapus data karyawan dengan ID tersebut dari database. Jika penghapusan berhasil, pengguna akan dialihkan kembali ke halaman utama, sedangkan jika gagal, pesan "Data Gagal Dihapus" akan ditampilkan dengan tautan kembali ke halaman utama.

- delete\_action.php

```
<?php
include_once 'connect.php';
include_once 'karyawan.php';

$database = new Database();
$db = $database->getConnection();

$karyawan = new Karyawan($db);
$karyawan->id_karyawan = $_GET['id_karyawan'];

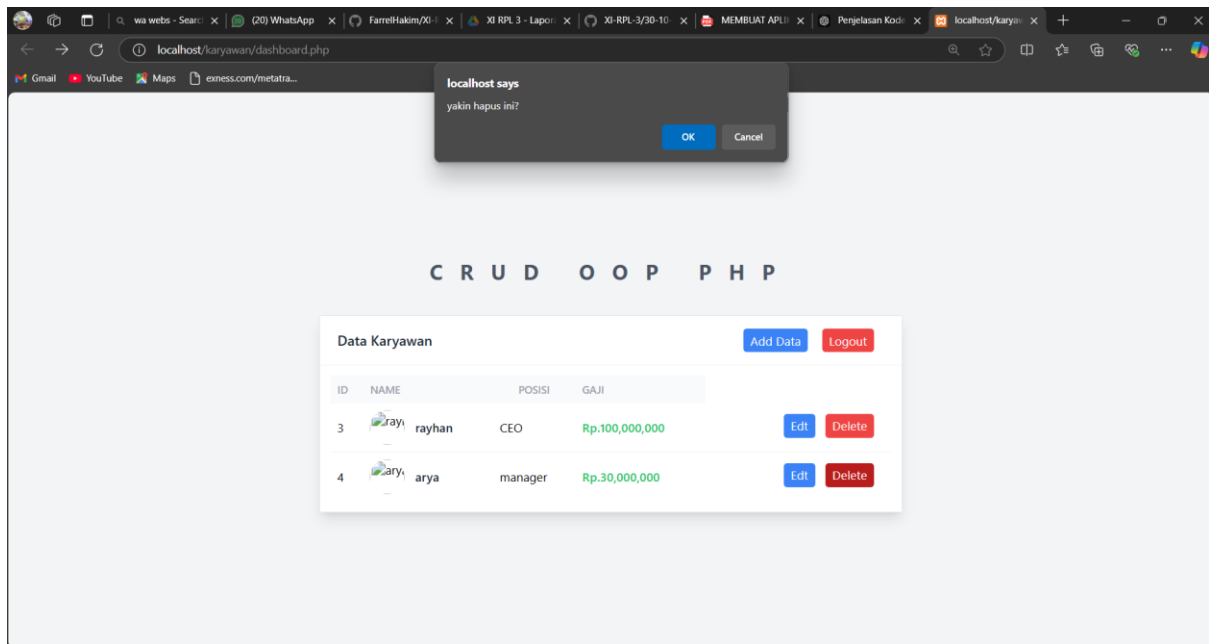
$hasil = $karyawan->delete();

if($hasil) {
    header('location: ./');
} else {
    echo "<h1>";
    echo "data tidak bisa di hapus.";
    echo "</h1>";
    echo " - <a href=\"./\">Back Home</a>";
}

?>
```

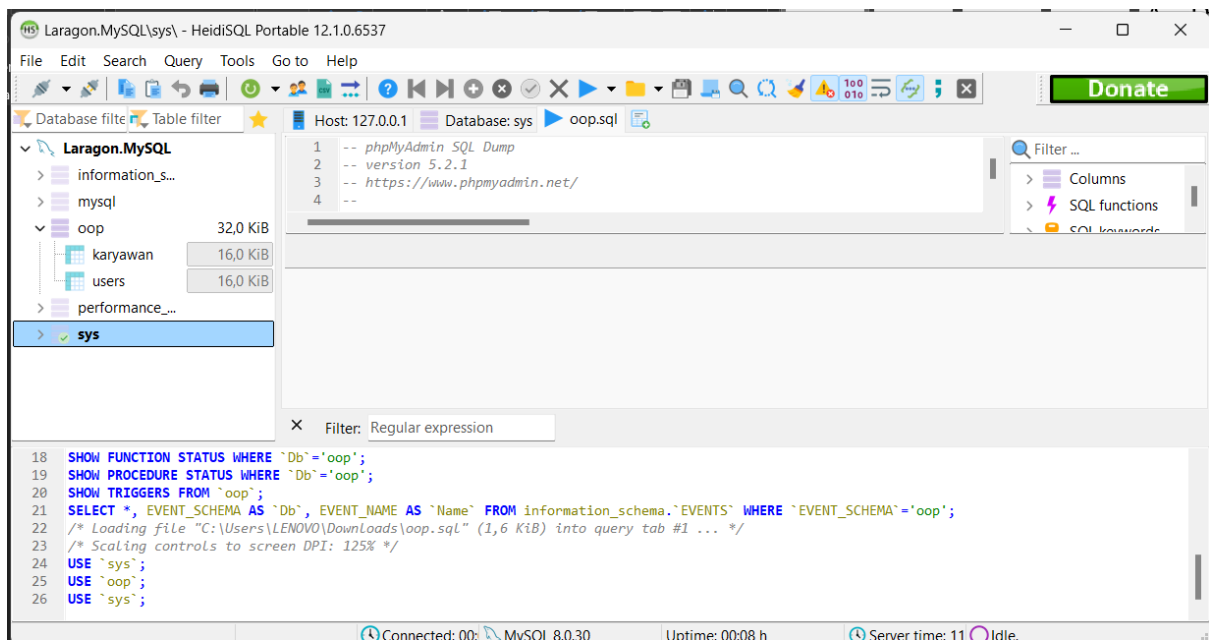
Penjelasan : Kode PHP ini menghapus data karyawan dari database berdasarkan id\_karyawan yang diperoleh dari parameter URL dengan metode GET. Pertama, skrip menyertakan file connect.php dan karyawan.php untuk mendapatkan koneksi database dan mendefinisikan kelas Karyawan. Setelah membuat objek Karyawan dengan koneksi database, skrip menetapkan nilai id\_karyawan berdasarkan parameter id\_karyawan yang diterima. Kemudian, metode delete() dipanggil untuk menghapus data karyawan dengan ID yang ditentukan. Jika penghapusan berhasil, pengguna dialihkan ke halaman utama (./). Jika gagal, pesan "data tidak bisa di hapus" ditampilkan bersama dengan tautan untuk kembali ke halaman utama.

- Tampilan menghapus data



## II. Login

1. pertama tama buatlah sebuah database di phpMyAdmin/laragon dengan nama OOP lalu buatlah sebuah table bernama kan karyawan dan users seperti di Gambar di bawah ini



2. Membuat koneksi php ke localhost.

disini saya menamai file saya dengan nama connect.php code bisa di lihat di bawah ini

```
<?php
class Database {
    private $host = "localhost";
    private $db_name = "oop";
    private $username = "root";
    private $password = "";
    public $conn;

    public function getConnection() {
        $this->conn = null;
        try {
            $this->conn = new PDO("mysql:host=" . $this->host .
";port=3306;dbname=" . $this->db_name, $this->username, $this->password);
            $this->conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
            $this->conn->exec("set names utf8");
        } catch(PDOException $exception) {
            echo "Connection error: " . $exception->getMessage();
        }
        return $this->conn;
    }
}

?>
```

Penjelasan : Class `Database` dalam kode di atas bertugas untuk mengatur koneksi ke database MySQL menggunakan PDO. Class ini menyimpan konfigurasi dasar database seperti host, nama database, username, dan password sebagai properti privat, sedangkan properti publik `\$conn` digunakan untuk menyimpan objek koneksi. Melalui metode `getConnection()`, class ini mencoba membuat koneksi ke database, menangani error dengan blok `try-catch` untuk menampilkan pesan jika koneksi gagal. Jika koneksi berhasil, properti `\$conn` akan menyimpan objek PDO dengan pengaturan mode error exception dan karakter UTF-8, sehingga memudahkan operasi database yang stabil dan terstruktur.

### 3. Membuat file dashboard.php

```
<?php
include './config/connect.php';
include './classes/karyawan.php';
```



```

if ($_SERVER["REQUEST_METHOD"] == "GET") {
    if (isset($_GET["logout"])) {
        session_start();
        session_unset();
        session_destroy();
        header("Location: ./index.php");
        exit();
    }
}

$db = new Database();
$karyawan = new Karyawan($db->getConnection());

session_start();
if (!isset($_SESSION['username'])) {
    echo "<script>alert('anda belum login.Silahkan login terlebih dahulu.');

```

```

        <table class="table-auto w-full">
            <thead class="text-xs font-semibold
uppercase text-gray-400 bg-gray-50">
                <tr>
                    <th class="p-2 whitespace-
nowrap">
                        <div class="font-semibold
text-left">ID</div>
                    </th>
                    <th class="p-2 whitespace-
nowrap">
                        <div class="font-semibold
text-left">Name</div>
                    </th>
                    <th class="p-2 whitespace-
nowrap">
                        <div class="font-semibold
text-center">Posisi</div>
                    </th>
                    <th class="p-2 whitespace-
nowrap">
                        <div class="font-semibold
text-left">Gaji</div>
                    </th>
                </tr>
            </thead>
            <tbody class="text-sm divide-y divide-
gray-100">
                <?php
$no = 1;
foreach($karyawan->read() as $x){
    $id = $x["id_karyawan"];
    $nama = $x["nama"];
    $posisi = $x["posisi"];
    $gaji = $x["gaji"];
    $f_gaji =
number_format($x["gaji"]);
    $dimas = ["IMG-20241004-
WA0001.jpg", "IMG-20241004-WA0004.jpg",
"SmartSelect_20241018_080405_Photos.jpg",
"SmartSelect_20241018_080442_Photos.jpg",
"SmartSelect_20241018_080620_Photos.jpg"];
    $farrel =
echo "
<tr>

```

```
  |
```

```

        </div>
    </div>
</div>
</div>
</section>

```

Penjelasan : Kode di atas adalah bagian dari aplikasi web berbasis PHP yang mengimplementasikan fungsi CRUD (Create, Read, Update, Delete) untuk data karyawan. Diawali dengan menginklusikan file koneksi database dan class Karyawan, kode ini memeriksa apakah permintaan adalah metode GET dan menangani logout jika parameter logout ada. Selanjutnya, aplikasi memeriksa sesi pengguna; jika pengguna belum login, akan muncul peringatan dan pengguna diarahkan ke halaman login. Setelah itu, aplikasi membuat objek Database dan Karyawan, lalu menampilkan data karyawan dalam bentuk tabel yang rapi dengan menggunakan Tailwind CSS. Tabel ini mencakup ID, nama, posisi, dan gaji karyawan, di mana setiap baris data juga menyediakan tautan untuk mengedit dan menghapus data karyawan, serta menampilkan foto karyawan secara acak dari kumpulan gambar.

#### 4. Membuat file user.php

```

<?php
//classes/User.php

require_once './config/connect.php';

class User {
    private $conn;
    private $table_name = "users";

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

    public function hasUsername($username) {
        $query = "SELECT * FROM " . $this->table_name . " WHERE username
= :username LIMIT 1";
        //memepersiapkan statement untuk query di atas
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":username", $username);
        $stmt->execute();

        // Jika ada data user yang sesuai dengan username yang di input
        if($stmt->rowCount() > 0) return true;
        return false;
    }

    public function login($username, $password) {

```

```

    //query untuk mengambil data user yang sesuai dengan username yang
    di input
    $query = "SELECT * FROM " . $this->table_name . " WHERE username =
:username LIMIT 1";
    //memepersiapkan statement untuk query di atas
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(":username", $username);
    $stmt->execute();

    // Jika ada data user yang sesuai dengan username yang di input
    if($stmt->rowCount() > 0) {
        // ambil data user yang diinput sesuai username yang di input
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        // jika password yang diinput sesuai dengan password yang
        if(password_verify($password, $row['PASSWORD'])) {
            // kembalikan true
            session_start();
            $_SESSION['username'] = $username;
            return true;
        }
    }

    // Kembalikan false jika tidak ada data user yang sesuai dengan
    username yang diinput atau password yang di input tidak sesuai
    dengan password yang tersimpan di database
    return false;
}

//fungsi untuk menambahkan user baru
public function register($username, $password) {
    if ($this->hasUsername($username)) return false;
    $query = "INSERT INTO " . $this->table_name . " (username,
password) VALUES (:username, :password)";
    $stmt = $this->conn->prepare($query);

    //enkripsi password sebelum di simpan ke database
    $hashed_password = password_hash($password, PASSWORD_DEFAULT);

    $stmt->bindParam(":username", $username);
    $stmt->bindParam(":password", $hashed_password);

    if($stmt->execute()) {
        return true;
    }
    return false;
}
}

```

Penjelasan : Kelas User dalam kode PHP ini berfungsi untuk mengelola operasi terkait pengguna dalam aplikasi, seperti pendaftaran dan autentikasi. Kelas ini terhubung ke database melalui koneksi yang diberikan dalam konstruktor dan memiliki tabel users sebagai tempat penyimpanan data pengguna. Metode hasUsername() memeriksa apakah username yang diberikan sudah ada di database, sedangkan metode login() mengambil data pengguna berdasarkan username dan memverifikasi password yang dimasukkan dengan yang tersimpan di database. Jika autentikasi berhasil, username disimpan dalam sesi. Metode register() digunakan untuk mendaftarkan pengguna baru, dengan terlebih dahulu memeriksa apakah username sudah ada; jika tidak, username dan password yang telah dienkripsi disimpan dalam tabel users.

## 5. Membuat file register.php

```
<?php
//register.php
require_once 'config/connect.php';
require_once 'classes/user.php';

$databse = new Database();
$db = $databse->getConnection();
$user = new User($db);

if($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    if($user->register($username, $password)) {
        $success_message = "Registrasi berhasil silahkan login";
    } else {
        $error_message = "Registrasi gagal. Username Mungkin sudah
di gunakan.";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
```

```

        <title>Register</title>
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-4">
                <h3 class="text-center">Register</h3>
                <?php if (isset($error_message)): ?>
                    <div class="alert alert-danger">
                        <?php echo htmlspecialchars($error_message);
                    ?>
                </div>
                <?php endif; ?>
                <form action="" method="POST">
                    <div class="form-group mb-3">
                        <label>Username</label>
                        <input type="text" name="username"
class="form-control" required>
                    </div>
                    <div class="form-group mb-3">
                        <label>Password</label>
                        <input type="password" name="password"
class="form-control" required>
                    </div>
                    <button type="submit" class="btn btn-primary w-
100">Daftar</button>
                    <p class="text-center mt-3">sudah punya akun?
                    <a href="index.php">login di sini</a>
                </form>
            </div>
        </div>
    </div>
</body>
</html>

```

Penjelasan : Kode di atas adalah file `register.php` yang berfungsi sebagai halaman registrasi pengguna baru. Ketika formulir registrasi dikirim dengan metode POST, sistem memproses data yang diinputkan untuk membuat akun baru. File ini menggunakan kelas `User` untuk memanggil metode `register()`, yang memeriksa apakah username sudah ada. Jika pendaftaran berhasil, pesan sukses muncul, namun jika gagal (misalnya, karena username sudah digunakan), pesan error akan ditampilkan. Tampilan halaman ini menggunakan Bootstrap untuk styling, menyediakan formulir input untuk username dan password, serta tombol "Daftar" yang dikaitkan dengan aksi pendaftaran. Di bagian bawah, pengguna diarahkan ke halaman login jika sudah memiliki akun.