

**LAPORAN HASIL PRAKTIKUM  
PEMROGRAMAN WEB & MOBILE**



**NAMA : ELNATAN KENINGATKO  
NIM : 193020503038  
KELAS : A  
MODUL : III (PHP MySQL Database)**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKA RAYA  
2021**

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### 1.1. Tujuan

- 1.1.1. Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.
- 1.1.2. Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

### 1.2. Landasan Teori

Data dalam database MySQL disimpan dalam tabel-tabel. Sebuah tabel adalah koleksi dari data yang berelasi dan mengandung kolom dan baris. Database sangat bermanfaat untuk menyimpan informasi secara kategori. Contoh yang akan diberikan pada modul praktikum ini adalah tabel yang mengandung data Employees (Pekerja), Products (Produk), Customers (Pelanggan) dan Orders (Pesanan).

#### 1.2.1. Membuka Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL. Berikut adalah contoh kode program agar terhubung dengan server mySQL:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Jika kode diatas tidak berhasil, kemungkinan variabel \$connect\_error sudah tidak tersedia pada versi PHP yang kita gunakan, maka gunakan kode program berikut:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Membuat Hubungan
    $conn = mysqli_connect($servername, $username, $password);

    // Memeriksa
    Hubungan if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Connected successfully";
?>
```

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut:

**mysqli\_close(\$conn);**

#### 1.2.2. Membuat Database

Database pada MySQL bisa juga dibuat menggunakan kode program PHP. Program tersebut akan berisi statement SQL “CREATE DATABASE”. Berikut adalah contoh program yang digunakan untuk membuat database “myDB”:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }
}
```

```
// Membuat Database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

### 1.2.3. Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

```
CREATE TABLE MyGuests (id INT(6) UNSIGNED
AUTO_INCREMENT PRIMARY KEY, firstname
VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT
NULL, email VARCHAR(50), reg_date TIMESTAMP)
```

Berikut adalah contoh program untuk membuat tabel diatas:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password,
$dbname);
// Memeriksa koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// SQL untuk membuat tabel
$sql = "CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
    KEY, firstname VARCHAR(30) NOT NULL, lastname
    VARCHAR(30) NOT NULL,
    email VARCHAR(50),
```

```

        reg_date TIMESTAMP
    );

    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . mysqli_error($conn);
    }
    mysqli_close($conn);
?>

```

#### 1.2.4. Memasukkan Data ke Database

Untuk memasukkan data kedalam tabel di database, terdapat beberapa aturan sintaks yang harus diikuti:

1. Query SQL harus diberikan kutip dalam PHP.
2. Nilai string didalam query SQL harus diberikan kutip.
3. Nilai numeris tidak harus diberikan kutip.
4. Nilai NULL tidak harus diberikan kutip.

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:

```

INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)

```

adapun contoh kode program PHP untuk menambahkan data tersebut adalah sebagai berikut:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa Koneksi
    if (!$conn){

```

```

        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";

    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
    mysqli_close($conn);
?>

```

Jika kita melakukan perintah INSERT pada tabel yang menggunakan field dengan opsi AUTO\_INCREMENT, kita bisa mendapatkan ID dari baris yang terakhir diinputkan. Caranya menggunakan fungsi `mysqli_insert_id($conn)`; nilai kembalian dari fungsi ini akan memberikan id dari record atau baris terakhir yang diinputkan.

Untuk menambahkan record dengan jumlah yang banyak, kita bisa menggunakan fungsi `mysqli_multi_query($conn, $sql)`, berikut adalah contoh program untuk menambahkan record dengan jumlah yang banyak:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Mary', 'Moe', 'mary@example.com')";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Julie', 'Dooley', 'julie@example.com')";

```

```

        if (mysqli_multi_query($conn, $sql)) {
            echo "New records created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($conn);
        }
    }
    mysqli_close($conn);
?>

```

#### 1.2.5. Mengambil Data dari Database

Untuk mengambil data, statement SQL yang digunakan adalah `SELECT nama_kolom() FROM nama_tabel`, atau kita bisa menggunakan karakter `*` untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan statement `WHERE` setelah `nama_tabel`. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat koneksi
$conn = mysqli_connect($servername, $username, $password,
$dbname);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " .
        $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
mysqli_close($conn);

```

?>

### 1.2.6. Menghapus Data dari Database

Statement DELETE digunakan untuk menghapus baris data atau records dari tabel. Clause WHERE digunakan untuk menspesifikasikan baris yang akan dihapus. Jika statement DELETE digunakan tanpa menggunakan klausa WHERE, maka semua record yang ada pada tabel akan dihapus.

| id | nama depan | nama belakang | email             | reg_date            |
|----|------------|---------------|-------------------|---------------------|
| 1  | John       | Doe           | john@example.com  | 2014-10-22 14:26:15 |
| 2  | Mary       | Moe           | mary@example.com  | 2014-10-23 10:22:30 |
| 3  | Julie      | Dooley        | julie@example.com | 2014-10-26 10:48:23 |

Jika kita memiliki data seperti tabel “MyGuest” seperti terlihat diatas, kemudian kita ingin menghapus data dengan nama depan Julie, maka kode program untuk menghapus data tersebut adalah sebagai berikut:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    // sql untuk menghapus record
    $sql = "DELETE FROM MyGuests WHERE id=3";

    if (mysqli_query($conn, $sql)) {
        echo "Record deleted successfully";
    } else {
        echo "Error deleting record: " . mysqli_error($conn);
    }
}
```



```
mysqli_close($conn);
?>
```

#### 1.2.7. Update Data dalam Database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:

```
UPDATE nama_tabel
SET kolom1=nilai1, kolom2=nilai2, ... WHERE
kolom_penentu=nilai_penentu ...
```

Jika statement ini digunakan untuk update data ke dua pada tabel “MyGuest” diatas untuk mengganti nama belakang dari “Moe” menjadi “Doe”, maka kode program untuk update tersebut adalah sebagai berikut:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE
    id=2";
    if (mysqli_query($conn, $sql)) {
        echo "Record updated successfully";
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
    mysqli_close($conn);
?>
```

## BAB II

### PEMBAHASAN

#### 2.1. Membuat Database pada MySQL untuk Menyimpan Data-Data Pegawai Beserta Relasi Tabelnya

Berikut ini adalah database dengan nama *pegawai* yang telah dibuat menggunakan software SQLyog. Database ini mempunyai tabel yaitu *data\_pegawai* yang mempunyai atribut *id* (tipe data INT dengan nilai tamping tiga karakter), *nama* (tipe data VARCHAR dengan nilai tamping dua puluh karakter) dan *jenis\_kelamin* (tipe data VARCHAR dengan nilai tamping dua puluh karakter). Tabel ini telah diisi data sebagai berikut.

| id | nama  | jenis_kelamin |
|----|-------|---------------|
| 1  | Yudha | Laki-Laki     |
| 2  | Sari  | Perempuan     |

**Gambar 2.1.** Data pada Tabel *data\_pegawai*.

#### 2.2. Program untuk Menyimpan, Menghapus dan Mengubah Data-Data Pegawai

Berikut ini merupakan penjelasan mengenai source code program yang telah dibuat.

```
<?php

$server = "localhost";
$user = "root";
$password = "";
$nama_database = "pegawai";

$db = mysqli_connect($server, $user, $password, $nama_database);

if( !$db ){
    die("Gagal terhubung dengan database: " . mysqli_connect_error());
}

?>
```

Blok code di atas merupakan file bernama *config.php* yang digunakan untuk menghubungkan koneksi dengan database *pegawai* yang telah dibuat sebelumnya. `$db = mysqli_connect($server, $user, $password, $nama_database);` digunakan untuk menjalankan argumen agar terkoneksi ke

server mysql dan atau database. Apabila database gagal terhubung, maka akan muncul tulisan “Gagal terhubung dengan database”.

```
<!DOCTYPE html>
<html>
<head>
  <title>Pendaftaran Pegawai Baru</title>
</head>

<body>
  <header>
    <h2>Pendaftaran Pegawai Baru</h2>
  </header>

  <h3>Menu</h3>
  <nav>
    <ul>
      <li><a href="form-daftar.php">Daftar Baru</a></li>
      <li><a href="list-pegawai.php">Pendaftar</a></li>
    </ul>
  </nav>

</body>
</html>
```

Blok code di atas merupakan file bernama *index.php* yang berfungsi sebagai halaman awal dari program web. `<nav></nav>` digunakan untuk mempresentasikan link navigasi. `<ul></ul>` digunakan untuk mengurutkan list dengan urutan berupa karakter special atau dengan symbol. `<li><a href="form-daftar.php">Daftar Baru</a></li>` digunakan untuk membuat kalimat “Daftar Baru” agar bisa menuju ke halaman berikutnya ketika diklik. Halaman yang dituju ini yaitu *form-daftar.php*. Sama halnya seperti `<li><a href="list-pegawai.php">Pendaftar</a></li>`. Perbedaannya yaitu kalimatnya adalah “Pendaftar” dan halaman yang dituju yaitu *list-pegawai.php*. Yang menampilkan hasil sebagai berikut.

← → ↻ 🏠 ⓘ localhost/pwm\_m3/index.php

## Pendaftaran Pegawai Baru

### Menu

- [Daftar Baru](#)
- [Pendaftar](#)

**Gambar 2.2.** Halaman Awal.

```
<?php include("config.php"); ?>

<!DOCTYPE html>
<html>
<head>
    <title>Pendaftaran Pegawai</title>
</head>

<body>
    <header>
        <h3>Pegawai yang sudah mendaftar</h3>
    </header>

    <table border="1">
    <thead>
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
        </tr>
    </thead>
    <tbody>

        <?php
        $sql = "SELECT * FROM data_pegawai";
        $query = mysqli_query($db, $sql);

        while($peg = mysqli_fetch_array($query)){
            echo "<tr>";

            echo "<td>".$peg['id'].</td>";
            echo "<td>".$peg['nama'].</td>";
            echo "<td>".$peg['jenis_kelamin'].</td>";

            echo "<td>";
            echo "<a href='form-edit.php?id=".$peg['id']."'>Edit</a> | ";
```

```

        echo "<a href='hapus.php?id=".$peg['id']."'>Hapus</a>";
        echo "</td>";

        echo "</tr>";
    }
    ?>

</tbody>
</table>

<p>Total: <?php echo mysqli_num_rows($query) ?></p>

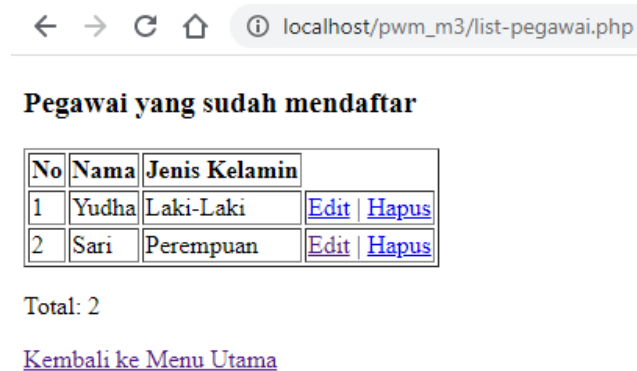
<a href="index.php">Kembali ke Menu Utama</a>

</body>
</html>

```

Blok code di atas merupakan file bernama *list-pegawai.php* yang fungsinya untuk menampilkan jumlah pegawai yang telah terdaftar. `<table border="1"></table>` digunakan untuk membungkus tabel yang telah dibuat. `<thead></thead>` digunakan untuk membungkus bagian kepala dari tabel. Bagian kepala dari tabel ini berupa `<th>No</th>`, `<th>Nama</th>` dan `<th>Jenis Kelamin</th>`. `<th></th>` digunakan untuk menandakan bahwa merupakan bagian dari kepala tabel. `<tbody></tbody>` digunakan untuk membungkus bagian body dari tabel. `$sql = "SELECT * FROM data_pegawai";` merupakan perintah SQL yang digunakan untuk menampilkan data dari tabel *data\_pegawai* yang ditampung ke dalam variable `$sql`. `$query = mysqli_query($db, $sql);` digunakan untuk menampung berbagai jenis query dengan parameter berupa variable `$db` dan variable `$sql`. `while($peg = mysqli_fetch_array($query)){` digunakan untuk melakukan perulangan dan pengambilan data yang terdapat pada variable `$query` yang kemudian akan disimpan ke dalam variable `$peg`. `<tr></tr>` digunakan untuk membuat baris agar tabel teratur ke bawah. `echo "<td>".$peg['id']."</td>";` digunakan untuk menampilkan data dari variable `$peg` dengan parameter `id` yang dimasukkan ke dalam sel tabel secara teratur menggunakan tag `<td></td>`. Sama halnya seperti `echo "<td>".$peg['nama']."</td>";` dan `echo "<td>".$peg['jenis_kelamin']."</td>";` juga digunakan untuk menampilkan data dari variable `$peg`. Yang

membedakannya terletak di parameter yang menentukan isi dari data yang akan ditampilkan. `echo "<a href='form-edit.php?id=".$peg['id']."'>Edit</a> / ";` digunakan untuk membuat kalimat “Edit” agar bisa menuju ke halaman berikutnya ketika diklik. Halaman yang dituju ini yaitu `form-edit.php`. `echo "<a href='hapus.php?id=".$peg['id']."'>Hapus</a>";` juga digunakan untuk membuat kalimat “Hapus” agar bisa menuju ke halaman berikutnya ketika diklik. Tetapi halaman yang dituju yaitu `hapus.php`. `<p>Total: <?php echo mysqli_num_rows($query) ?></p>` digunakan untuk menampilkan total data yang telah tersimpan. `<a href="index.php">Kembali ke Menu Utama</a>` digunakan untuk membuat kalimat “Kembali ke Halaman Utama” dengan tujuan halaman yaitu `index.php`. Yang menampilkan hasil sebagai berikut.



**Gambar 2.3.** Halaman Daftar Pegawai.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulir Pendaftaran Pegawai</title>
</head>

<body>
  <header>
    <h3>Formulir Pendaftaran Pegawai Baru</h3>
  </header>

  <form action="proses-pendaftaran.php" method="POST">

    <fieldset>

      <p>
        <label for="nama">Nama: </label>
        <input type="text" name="nama" placeholder="nama
```

```

lengkap" />
</p>

<p>
    <label for="jenis_kelamin">Jenis Kelamin: </label>
    <label><input type="radio" name="jenis_kelamin"
value="laki-laki"> Laki-laki</label>
    <label><input type="radio" name="jenis_kelamin"
value="perempuan"> Perempuan</label>
</p>

<p>
    <input type="submit" value="Daftar" name="daftar" />
</p>

</fieldset>

</form>

</body>
</html>

```

Blok code di atas merupakan file bernama *form-daftar.php* yang fungsinya sebagai halaman untuk form input data. `<form action="proses-pendaftaran.php" method="POST">` digunakan untuk mengirimkan data form ke file *pendaftaran.php* dengan metode *post*. `<fieldset></fieldset>` digunakan untuk membungkus elemen-elemen input yang memiliki kategori sama. `<label for="nama">Nama: </label>` digunakan untuk melengkapi keterangan untuk objek form `<input type="text" name="nama" placeholder="nama lengkap" />`. Objek form ini berupa kolom yang diisi dengan text yang kemudian hasilnya akan disimpan ke dalam variabel *nama*. Sama seperti `<label for="jenis_kelamin">Jenis Kelamin: </label>` yang juga untuk melengkapi objek form `<label><input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki</label>` dan `<label><input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan</label>`. Yang membedakannya adalah tipe yang digunakan untuk memasukkan datanya yaitu berupa memilih salah satu opsi yang tersedia. `<input type="submit" value="Daftar" name="daftar" />` digunakan

untuk membuat tombol submit dengan nama “Daftar”. Yang menampilkan hasil sebagai berikut.



The screenshot shows a web browser window with the address bar displaying 'localhost/pwm\_m3/form-daftar.php'. The page title is 'Formulir Pendaftaran Pegawai Baru'. The form contains a text input field labeled 'Nama: [nama lengkap]', two radio buttons for 'Jenis Kelamin: Laki-laki' and 'Perempuan', and a button labeled 'Daftar'.

**Gambar 2.4.** Halaman Form Daftar Pegawai.

```
<?php
include("config.php");

if(isset($_POST['daftar'])){

    $nama = $_POST['nama'];
    $jk = $_POST['jenis_kelamin'];

    $sql = "INSERT INTO data_pegawai (nama, jenis_kelamin) VALUE
('$nama', '$jk')";
    $query = mysqli_query($db, $sql);

    if( $query ) {
        header('Location: list-pegawai.php?status=sukses');
    } else {
        header('Location: list-pegawai.php?status=gagal');
    }
}
?>
```

Blok code di atas merupakan file bernama *proses-pendaftaran.php* yang fungsinya untuk melakukan query SQL. `$sql = "INSERT INTO data_pegawai (nama, jenis_kelamin) VALUE ('$nama', '$jk')";` merupakan query SQL untuk menambahkan data ke dalam tabel *data\_pegawai* dengan parameter berupa *nama* dan *jenis\_kelamin* yang kemudian akan disimpan ke dalam variable *\$sql*. Apabila data berhasil ditambahkan maka pada bagian atas halaman akan muncul kalimat “list-pegawai.php?status=sukses”, sedangkan apabila data gagal ditambahkan akan muncul kalimat “list-pegawai.php?status=gagal”. Yang menampilkan hasil sebagai berikut.





**Gambar 2.5.** Menambahkan Data Pegawai.

```
<?php
include("config.php");
if( isset($_GET['id']) ){
    $id = $_GET['id'];

    $sql = "DELETE FROM data_pegawai WHERE id=$id";
    $query = mysqli_query($db, $sql);

    if( $query ){
        header('Location: list-pegawai.php?status=sukses');
    } else {
        die("Gagal menghapus!");
    }
}
?>
```

Blok code di atas merupakan file bernama *hapus.php* yang fungsinya untuk melakukan query SQL. *\$sql = "DELETE FROM data\_pegawai WHERE id=\$id";* merupakan query SQL untuk menghapus data pada tabel *data\_pegawai* dengan acuan yang digunakan yaitu *id* yang kemudian akan disimpan ke dalam variable *\$sql*. Apabila data berhasil ditambahkan maka pada bagian atas halaman akan muncul kalimat “list-pegawai.php?status=sukses”, sedangkan apabila data gagal ditambahkan akan muncul kalimat “Gagal menghapus!”. Yang kemudian akan menampilkan hasil sebagai berikut.



**Gambar 2.6.** Menghapus Data Pegawai.

```
<?php
include("config.php");

if( !isset($_GET['id']) ){
    header('Location: list-pegawai.php');
}

$id = $_GET['id'];

$sql = "SELECT * FROM data_pegawai WHERE id=$id";
$query = mysqli_query($db, $sql);
$peg = mysqli_fetch_assoc($query);

if( mysqli_num_rows($query) < 1 ){
    die("data tidak ditemukan...");
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Formulir Edit Pegawai</title>
</head>

<body>
    <header>
        <h3>Formulir Edit Pegawai</h3>
    </header>

    <form action="proses-edit.php" method="POST">

        <fieldset>
```

```

        <input type="hidden" name="id" value="<?php echo
$peg['id'] ?>" />

        <p>
            <label for="nama">Nama: </label>
            <input type="text" name="nama" placeholder="nama
lengkap" value="<?php echo $peg['nama'] ?>" />
        </p>

        <p>
            <label for="jenis_kelamin">Jenis Kelamin: </label>
            <?php $jk = $peg['jenis_kelamin']; ?>
            <label><input type="radio" name="jenis_kelamin"
value="Laki-Laki" <?php echo ($jk == 'laki-laki') ? "checked": "" ?>>
Laki-laki</label>
            <label><input type="radio" name="jenis_kelamin"
value="Perempuan" <?php echo ($jk == 'perempuan') ? "checked": "" ?>>
Perempuan</label>
        </p>

        <p>
            <input type="submit" value="Simpan" name="simpan" />
        </p>

    </fieldset>
</form>
</body>
</html>

```

Blok code di atas merupakan file bernama *form-edit.php* yang fungsinya sebagai halaman form untuk mengedit data pegawai. Pada bagian PHP merupakan pemanggilan database yang akan digunakan untuk form mengedit data pegawai. `<input type="hidden" name="id" value="<?php echo $peg['id'] ?>" />` merupakan objek form tersembunyi yang digunakan untuk menyisipkan data *id* untuk diproses. `<input type="text" name="nama" placeholder="nama lengkap" value="<?php echo $peg['nama'] ?>" />` digunakan untuk menginputkan text pada bagian nama yang akan edit tetapi akan memunculkan data sebelum yang tersimpan sebelum diedit. Begitu pula pada bagian jenis kelamin. Hanya yang membedakannya terdapat atribut `checked` yang digunakan untuk mengindikasikan apakah elemen jenis kelamin

tersebut harus diperiksa ketika laman melakukan loading. Yang menampilkan hasil sebagai berikut.



The screenshot shows a web browser window with the address bar displaying 'localhost/pwm\_m3/form-edit.php?id=1'. The page title is 'Formulir Edit Pegawai'. The form contains a text input field labeled 'Nama' with the value 'Yudha', a radio button group labeled 'Jenis Kelamin' with 'Laki-laki' selected, and a 'Simpan' button.

**Gambar 2.7.** Halaman Form Edit Data Pegawai.

```
<?php
include("config.php");

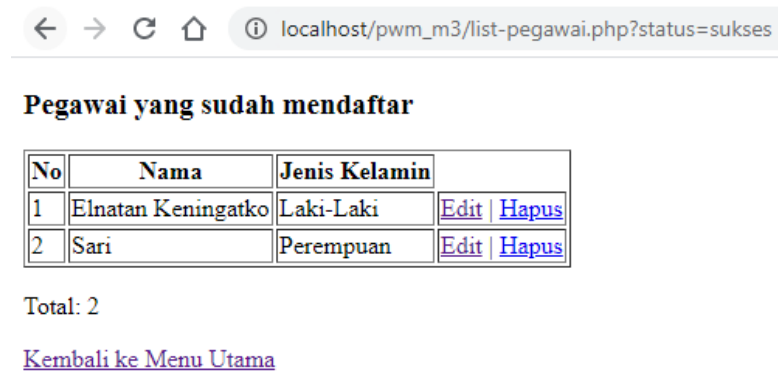
if(isset($_POST['simpan'])){

    $id = $_POST['id'];
    $nama = $_POST['nama'];
    $jk = $_POST['jenis_kelamin'];

    $sql = "UPDATE data_pegawai SET nama='$nama',
jenis_kelamin='$jk' WHERE id=$id";
    $query = mysqli_query($db, $sql);

    if( $query ) {
        header('Location: list-pegawai.php?status=sukses');
    } else {
        die("Gagal menyimpan perubahan!");
    }
}
?>
```

Blok code di atas merupakan file bernama *proses-edit.php* yang fungsinya untuk melakukan query SQL. `$sql = "UPDATE data_pegawai SET nama='$nama', jenis_kelamin='$jk' WHERE id=$id";` merupakan query SQL untuk mengedit data pada tabel *data\_pegawai* dengan acuan yang digunakan yaitu *id* yang kemudian akan disimpan ke dalam variable *\$sql*. Apabila data berhasil ditambahkan maka pada bagian atas halaman akan muncul kalimat "list-pegawai.php?status=sukses", sedangkan apabila data gagal ditambahkan akan muncul kalimat "Gagal menyimpan perubahan!". Yang menampilkan hasil sebagai berikut.



**Gambar 2.8.** Sukses Mengedit Data Pegawai.

### **BAB III**

### **KESIMPULAN**

Pada MySQL, database disimpan dalam tabel. Tabel adalah kumpulan data terkait yang berisi kolom dan baris. Database sangat berguna untuk menyimpan informasi dalam kategori. PHP (PHP: Hypertext Preprocessor) adalah bahasa pemrograman skrip sisi server open source. Sebagai bahasa skrip, PHP menjalankan instruksi pemrograman pada saat runtime. Hasil instruksi pasti akan berbeda-beda sesuai dengan data yang diolah. HTML adalah singkatan dari Hypertext Markup Language, yaitu bahasa pemrograman standar yang digunakan untuk membuat halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di Internet Web browser (browser). HTML juga dapat digunakan sebagai link antar situs atau file menggunakan localhost di komputer, atau untuk menghubungkan link antar situs di Internet.

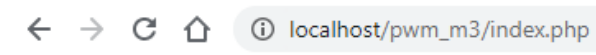
## **DAFTAR PUSTAKA**

MODUL PRAKTIKUM PEMROGRAMAN WEB I. Palangka Raya:, Universitas  
Palangka Raya.

## LAMPIRAN

| id | nama  | jenis_kelamin |
|----|-------|---------------|
| 1  | Yudha | Laki-Laki     |
| 2  | Sari  | Perempuan     |

Gambar 2.1. Data pada Tabel data\_pegawai.

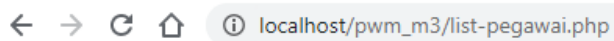


### Pendaftaran Pegawai Baru

#### Menu

- [Daftar Baru](#)
- [Pendaftar](#)

Gambar 2.2. Halaman Awal.



#### Pegawai yang sudah mendaftar

| No | Nama  | Jenis Kelamin |  |
|----|-------|---------------|--|
| 1  | Yudha | Laki-Laki     | <a href="#">Edit</a>   <a href="#">Hapus</a> |
| 2  | Sari  | Perempuan     | <a href="#">Edit</a>   <a href="#">Hapus</a> |

Total: 2

[Kembali ke Menu Utama](#)

Gambar 2.3. Halaman Daftar Pegawai.

A screenshot of a web browser showing a registration form. The form has a title 'Formulir Pendaftaran Pegawai Baru'. It contains a text input field for 'Nama' with a placeholder 'nama lengkap', a radio button group for 'Jenis Kelamin' with options 'Laki-laki' and 'Perempuan', and a 'Daftar' button.

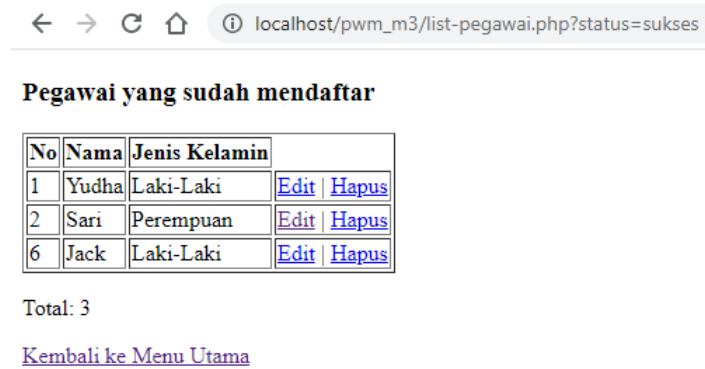
Formulir Pendaftaran Pegawai Baru

Nama:

Jenis Kelamin: ☐ Laki-laki ☐ Perempuan

Gambar 2.4. Halaman Form Daftar Pegawai.





**Gambar 2.5.** Menambahkan Data Pegawai.



**Gambar 2.6.** Menghapus Data Pegawai.



**Gambar 2.7.** Halaman Form Edit Data Pegawai.



**Gambar 2.8.** Sukses Mengedit Data Pegawai.