

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN WEB & MOBILE I**



**NAMA : SATRIA SHANO**

**NIM : 11191068**

**KELAS : C**

**MODUL III**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

## BAB I

### TUJUAN DAN LANDASAN TEORI

#### 1. Tujuan Praktikum

**1.1.** Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.

**1.2.** Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

#### 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).

#### 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";
?>
```

**Gambar 1.2.1**

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";
?>
```

**Gambar 1.2.2**

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

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

### **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);
?>

```

**Gambar 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);
?>

```

**Gambar 1.2.4**

### Memasukkan Data ke Database

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

- Query SQL harus diberikan kutip dalam PHP.
- Nilai string didalam query SQL harus diberikan kutip.
- Nilai numeris tidak harus diberikan kutip.
- 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);
?>
```

**Gambar 1.2.5**

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);
?>

```

**Gambar 1.2.6**

### **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);
?>

```

**Gambar 1.2.7**

### **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 clausa WHERE, maka semua record yang ada pada tabel akan dihapus.

id	namadepan	namabelakan	email	reg_date
		g		
1	John	Doe	john@example.com	2014-10-22 14:26:15



2	Mary	Moe	mary@exampl e.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);
?>
```

**Gambar 1.2.8**

### **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);
?>
```

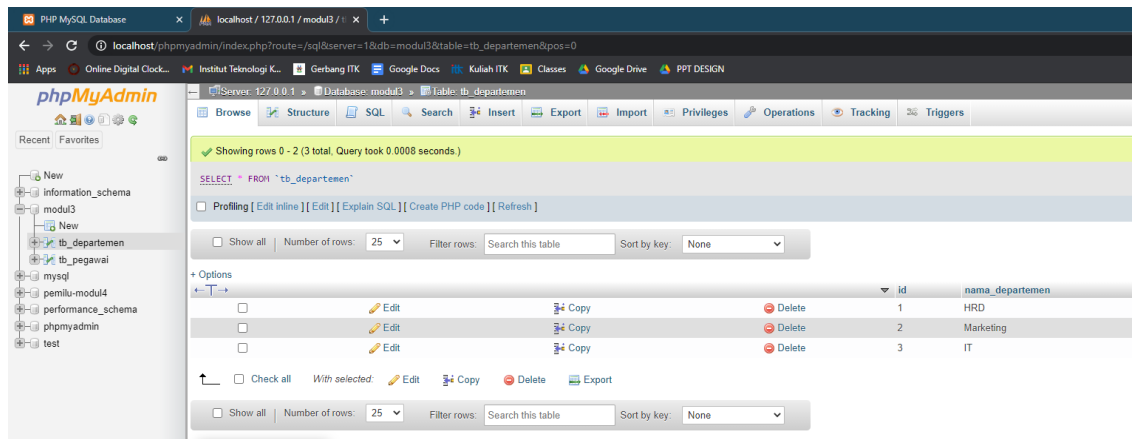
**Gambar 1.2.9**

## BAB II

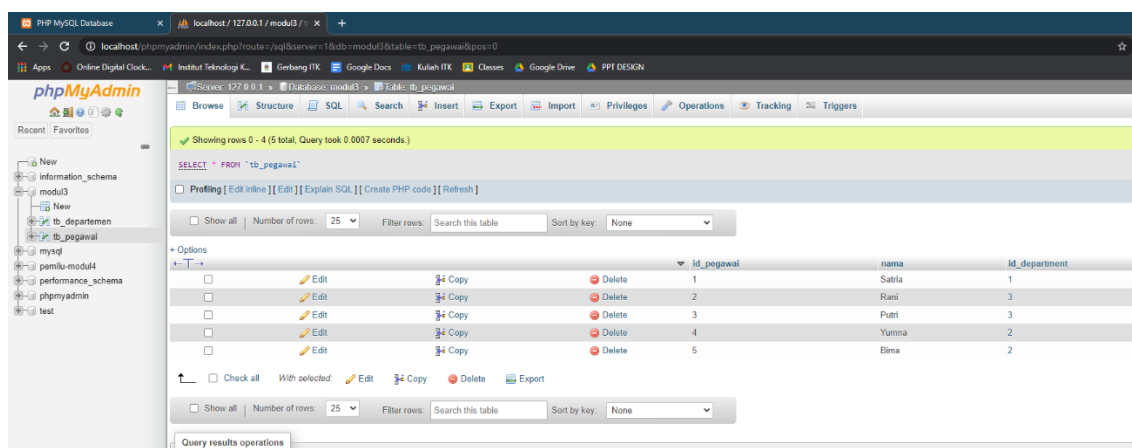
### PEMBAHASAN

1. Buatlah database pada MySql untuk menyimpan data-data pegawai beserta relasi tabelnya.
2. Buatlah program untuk menyimpan, menghapus, dan mengubah data-data pegawai tersebut.

Untuk soal nomor 1 kita diminta membuat database untuk menyimpan data pegawai, disini menggunakan phpmyadmin untuk pembuatan database nya. Disini menggunakan 2 tabel, yaitu tabel pegawai dan tabel departemen. Untuk relasinya, pegawai bekerja di suatu departemen.



Gambar 2.1 Tabel Departemen



Gambar 2.2 Tabel Pegawai

### Source Code function.php

```
<?php
$servername="localhost";
$dbname="modul3";
$username="root";
$password="";

$conn=mysqli_connect($servername,$username,$password,$dbname);
function show($query){
    global $conn;
    $result=mysqli_query($conn,$query);
    $rows=[];
    while($row=mysqli_fetch_assoc($result)){
        $rows[]=$row;
    }
    return $rows;
}
function insert($data){
    global $conn;
    $id_pegawai=$_POST["id_pegawai"];
    $nama=$_POST["nama"];
    $id_department=$_POST["id_department$id_department"];

    mysqli_query($conn,"INSERT INTO tb_pegawai VALUES('$id_pegawai','$nama'
,$id_department')");
    return mysqli_affected_rows($conn);
}
function delete($data){
    global $conn;
    $id_pegawai=$_GET["delete"];
    mysqli_query($conn,"DELETE FROM tb_pegawai WHERE id_pegawai=$id_pegawai
");
    return mysqli_affected_rows($conn);
}
function update($data){
    global $conn;
    $id_pegawai=$_POST["id_pegawai"];
    $nama=$_POST["nama"];
    $id_department=$_POST["id_department $id_department"];
    mysqli_query($conn,"UPDATE tb_pegawai SET nama='$nama',
id_department='$id_department'
WHERE id_pegawai= $id_pegawai");
    return mysqli_affected_rows($conn);
}
```

```
}  
?>
```

## Penjelasan

Pada file pertama, dibuat file dengan nama function.php yang isinya berupa fungsi-fungsi untuk menambahkan data, menghapus data, dan mengubah data.

```
$servername="localhost";  
$database="modul3";  
$username="root";  
$password="";
```

Baris kode di atas untuk mengatur server, database, username, dan password yang digunakan untuk program yang dibuat. Database yang digunakan adalah database bernama “modul3” yang sudah dibuat pada nomor 1 tadi.

```
function show($query){  
global $conn;  
    $result=mysqli_query($conn,$query);  
    $rows=[];  
while($row=mysqli_fetch_assoc($result)){  
    $rows[]=$row;  
    }  
return $rows;  
}
```

Function show merupakan fungsi untuk menampilkan data pada tampilan awal web tersebut yang dimana hasil nya ditampilkan.

```
function insert($data){  
global $conn;  
    $id_pegawai=$_POST["id_pegawai"];  
    $nama=$_POST["nama"];  
    $id_department=$_POST["id_department$id_department"];  
    mysqli_query($conn,"INSERT INTO tb_pegawai  
VALUES('$id_pegawai',  
'$nama','$id_department')");  
return mysqli_affected_rows($conn);  
}
```

Function insert merupakan fungsi untuk memasukan dan menyimpan data yang diinputkan melalui web ke dalam database.

```
function delete($data){
global $conn;
    $id_pegawai=$_GET["delete"];    mysqli_query($conn,"DELETE
FROM tb_pegawai WHERE id_pegawai=$id_pegawai");
    return mysqli_affected_rows($conn);
}
```

Function delete merupakan fungsi untuk menghapus data pegawai pada database yang dimana untuk menghapusnya menggunakan id\_pegawai, dimana ketika button pada tampilan awal ditekan akan langsung menghapus data pegawai tersebut.

```
function update($data){
global $conn;
    $id_pegawai=$_POST["id_pegawai"];
    $nama=$_POST["nama"];
    $id_department=$_POST["id_department $id_department"];
mysqli_query($conn,"UPDATE tb_pegawai SET nama='$nama',
id_department='$id_department' WHERE id_pegawai=
$id_pegawai");    return mysqli_affected_rows($conn);
}
```

Function update merupakan fungsi untuk mengubah data pegawai pada database dan tabel pegawai. Yang dimana melakukan update atau perubahan data berdasarkan pada id\_pegawai.

Dari keempat perintah function diatas dimana masing-masing function tersebut perintah \$\_POST yang dimana berfungsi untuk menampung data sehingga penggunaan nya dapat dicontohkan seperti \$nama = \$\_POST yang dapat diartikan nama = post yang telah ditampung.

### Source Code insert.php

```
<?php
require("function.php");
if(isset($_POST["insert"])){
    if(insert($_POST)>0){
        echo "
        <script>alert('data berhasil ditambah');
        document.location.href='index.php';
```

```

        </script>";
    }
    else{
        echo"<script>
        alert('data gagal ditambah');
        document.location.href='index.php';
        </script>";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<title>Insert</title>
</head>
<body>
    <a href="index.php">home</a>
    <form action="" method="POST">
        <ul>
            <li>
                <label for="id_pegawai">id_pegawai</label>
                <input type="text" name="id_pegawai" id="id_pegawai">
            </li>
            <li>
                <label for="nama">nama</label>
                <input type="text" name="nama" id="nama">
            </li>
            <li>
                <label for="id_department">id_department</label>
                <input type="text" name="id_department" id="id_department">
            </li>
            <li>
                <button type="submit" name="insert">insert</button>
            </li>
        </ul>
    </form>
</body>
</html>

```

## Penjelasan

```
require("function.php");
if(isset($_POST["insert"])){
    if(insert($_POST>0)){
        echo "
        <script>alert('data berhasil ditambah');
        document.location.href='index.php';
        </script>";
    }
    else{
        echo"<script>
        alert('data gagal ditambah');
        document.location.href='index.php';
        </script>";
    }
}
```

Kode diatas menggunakan fungsi require yang berfungsi untuk menyertakan file yaitu file function.php yang kemudian dilanjutkan dengan insert yang dimasukan ke dalam fungsi \$\_POST kemudian dilanjutkan dengan fungsi \$\_POST yang berfungsi untuk menampung nilai atau data yang akan diinputkan dan lalu dilanjutkan dengan perintah echo yang berfungsi untuk menampilkan data.

## Source Code update.php

```
<?php
require("function.php");
$id_pegawai=$_GET["update"];
$data=show("SELECT * FROM tb_pegawai WHERE id_pegawai =$id_pegawai");
if(isset($_POST["ubah"])){
    if(update($_GET>0)){
        echo "<script>
        alert('data berhasil diupdate');
        document.location.href='index.php';
        </script>";
    }
    else{
        echo"<script>
        alert('data gagal diupdate');
        document.location.href='index.php';</script>";
    }
}
}??>
<!DOCTYPE html>
```



```

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<title>Update</title>
</head>
<body>
<a href="index.php">home</a>
<form action="" method="POST">
    <?php
    foreach($data as $d){
    ?>
        <ul>
            <li>
                <label for="id_pegawai">id_pegawai</label>
                <input type="text" name="id_pegawai" id="id_pegawai" value=<?p
hp echo $d["id_pegawai"];?>readonly>
            </li>
            <li>
                <label for="nama">nama</label>
                <input type="text" name="nama" id="nama" value=<?php echo $d["
nama"];?>>
            </li>
            <li>
                <label for="id_department">id_department</label>
                <input type="text" name="id_department" id="id_department" val
ue=<?php echo $d["id_department"];?>>
            </li>
            <li>
                <button type="submit" name="ubah">update</button>
            </li>
        </ul>
    <?php
    }?>
</form>
</body>
</html>

```

## Penjelasan

```
require("function.php");
$id_pegawai=$_GET["update"];
$data=show("SELECT * FROM tb_pegawai WHERE id_pegawai =$id_pegawai")
;
if(isset($_POST["ubah"])){
if(update($_GET>0)){          echo
"<script>          alert('data berhasil
diupdate');
document.location.href='index.php';
          </script>";
    }
else{
        echo"<script>
alert('data gagal diupdate');
document.location.href='index.
php';</script>";
    } }
}
```

Kode diatas menggunakan fungsi require yang berfungsi untuk memanggil file yaitu file function.php yang dimana pada baris selanjutnya adalah pernyataan kondisi yang dimana jika tombol update pada tampilan awal web ditekan maka akan mengambil data sebelumnya untuk ditampilkan dan dirubah, kemudian pada bagian selanjutnya yaitu pada bagian body berisikan function untuk tempat insert file yang akan diubah dan juga format pengubahan data pegawai serta tombol untuk submit data yang telah diubah pada form insert data untuk pengubahan data tersebut.

## Source Code index.php

```
<?php
require("function.php");
$data=show("SELECT * FROM tb_pegawai");
if(isset($_GET["delete"])){
    if(delete($_GET>0)){
        echo"<script>
        alert('data berhasil dihapus');
        document.location.href='index.php';
        </script>";
    }
    else{
        echo"<script>
        alert('data gagal dihapus');
        document.location.href='index.php';
    }
}
```

```

        </script>";}
    }
?>
<!DOCTYPEhtml>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>PHP MySQL Database</title>
</head>
<body>
<table border="1">
    <tr>
        <td>id_pegawai</td>
        <td>nama</td>
        <td>id_department</td>
        <td>aksi</td>
    </tr>
    <?php foreach($data as $d){
    ?>
    <tr>
        <td><?php echo $d["id_pegawai"];?></td>
        <td><?php echo $d["nama"];?></td>
        <td><?php echo $d["id_department"];?></td>
        <td>
            <form action=""method="GET">
                <button type="submit"name="delete"value=<?php echo $d["id_p
egawai"];?>>Hapus</button>
            </form>
            <form action="update.php"method="GET">
                <button type="submit"name="update"value=<?php echo $d["id_p
egawai"];?>>Ubah</button>
            </form>
        </td>
    </tr>
    <?php
    }?>
</table>
<form action="insert.php">
    <button>Tambahkan Data</button>
</form>
</body>
</html>

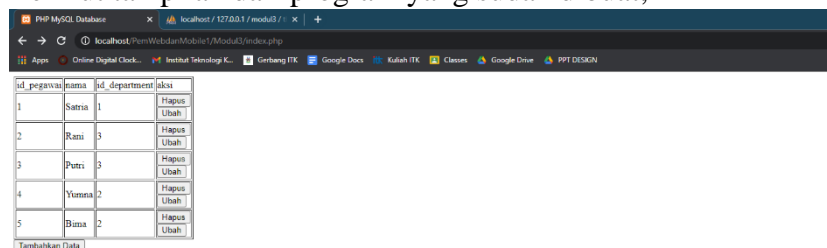
```

## Penjelasan

Kode program diatas merupakan tampilan untuk tampilan awal website nya. Dan kode di bawah adalah kode untuk menghapus data pegawai yang dijadikan satu di dalam index.php untuk menghemat jumlah file. Pada kode di bawah juga menggunakan fungsi require yang berfungsi untuk memanggil file yaitu file function.php yang dimana pada baris selanjutnya adalah pernyataan kondisi yang dimana jika tombol delete pada tampilan awal web ditekan maka akan menghapus data pegawai tersebut serta menampilkan alert bahwa data berhasil dihapus atau data gagal dihapus.

```
require("function.php");
$data=show("SELECT * FROM tb_pegawai");
if(isset($_GET["delete"])){
    if(delete($_GET>0)){
        echo"<script>
        alert('data berhasil dihapus');
        document.location.href='index.php';
        </script>";
    }
    else{
        echo"<script>
        alert('data gagal dihapus');
        document.location.href='index.php';
        </script>";}
    }
}
```

Berikut tampilan dari program yang sudah dibuat;

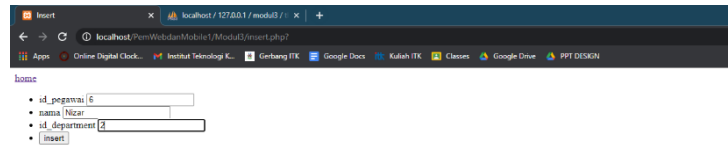


The screenshot shows a web browser window with a table of employee data. The table has columns for id\_pegawai, nama, id\_department, and aksi. There are 5 rows of data. Each row has a 'Hapus' button and an 'Ubah' button in the 'aksi' column. Below the table is a 'Tambahkan Data' button.

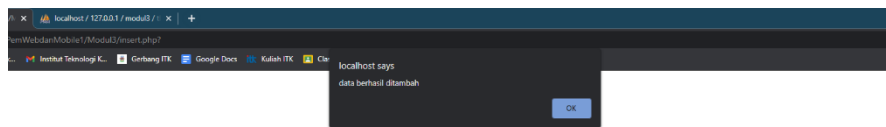
id_pegawai	nama	id_department	aksi
1	Satrio	1	Hapus Ubah
2	Rani	3	Hapus Ubah
3	Putri	3	Hapus Ubah
4	Yusma	2	Hapus Ubah
5	Bima	2	Hapus Ubah

Tambahkan Data

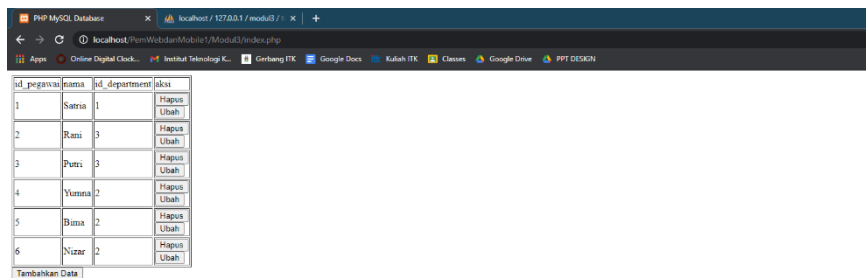
**Gambar 2.3 Tampilan awal**



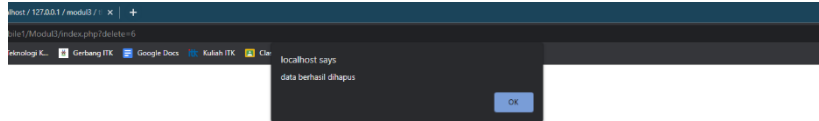
**Gambar 2.4 Tampilan Untuk Menambahkan/Insert Data**



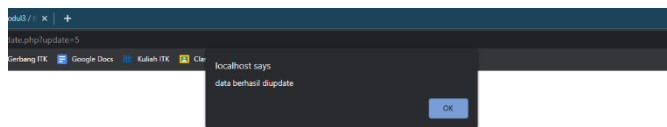
**Gambar 2.5 Tampilan jika data berhasil ditambah.**



**Gambar 2.6 Tampilan data yang telah ditambahkan.**



**Gambar 2.7 Tampilan jika data berhasil dihapus.**



**Gambar 2.10 Tampilan jika data berhasil diubah.**

### **BAB III**

#### **KESIMPULAN**

PHP dan MySQL merupakan kolaborasi antara bahasa pemrograman dan layanan database yang populer saat ini. Jadi dengan menggabungkan PHP dan MySQL kita bisa membuat program sederhana seperti diatas untuk memudahkan pengguna awam untuk mengedit data yang ada didalam database.

## **DAFTAR PUSTAKA**

Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan*

*Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

# LAMPIRAN

The image displays two screenshots of the phpMyAdmin web interface, showing database management operations.

**Top Screenshot:** The interface shows the 'tb\_departemen' table. The SQL query executed is `SELECT * FROM 'tb_departemen'`. The table contains 3 rows of data:

id	nama_departemen
1	HRD
2	Marketing
3	IT

**Bottom Screenshot:** The interface shows the 'tb\_pegawai' table. The SQL query executed is `SELECT * FROM 'tb_pegawai'`. The table contains 5 rows of data:

id_pegawai	nama	id_department
1	Sahla	1
2	Rani	3
3	Puti	3
4	Yuma	2
5	Bima	2



PHP MySQL Database | localhost / 127.0.0.1 / modu3 / X

localhost / 127.0.0.1 / modu3 / index.php

id_pegawai	nama	id_department	aksi
1	Satria	1	Hapus Ubah
2	Rani	3	Hapus Ubah
3	Putri	3	Hapus Ubah
4	Yumna	2	Hapus Ubah
5	Bima	2	Hapus Ubah

Tambahkan Data

Insert | localhost / 127.0.0.1 / modu3 / X

localhost / 127.0.0.1 / modu3 / insert.php

home

- id\_pegawai:
- nama:
- id\_department:
-

