

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I**



Nama : Mitchel Christian Timothyus Singaraca

Nim : 193020503041

Kelas : A

Modul : III (PHP MySQL Database)

JURUSAN/PROGRAM STUDI 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).

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 1.1 Koneksi

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 Koneksi

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)) {
```

Gambar 1.3 Membuat Database

```

        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.4 Membuat Database

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 (

```

Gambar 1.5 Membuat Tabel

```

        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 1.6 Membuat Tabel

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 1.7 Memasukan Data

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){

```

Gambar 1.8 Memasukan Data

```

        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 1.9 Memasukan Data

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, $dbname);
// Periksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

```

Gambar 1.10 Mengambil Data

```

$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 1.11 Mengambil Data

Menghapus Data

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.

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);

```

Gambar 1.12 Menghapus


```

// 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 1.13 Menghapus

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){

```

Gambar 1.14 Update Data

```
        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 1.15 Update Data

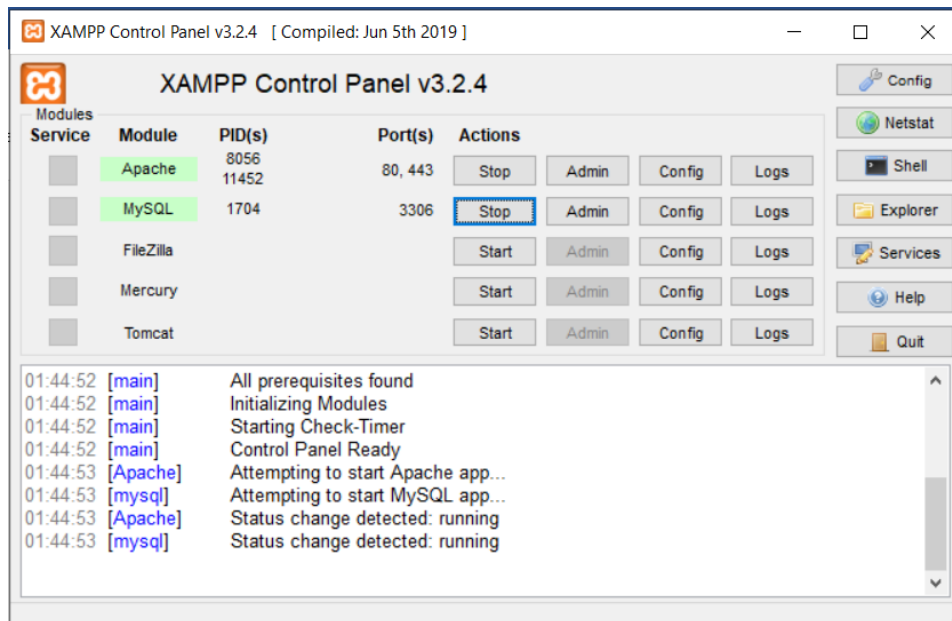
BAB II

PEMBAHASAN

2.1. Buatlah database pada MySql untuk menyimpan data-data pegawai beserta relasi tabelnya.

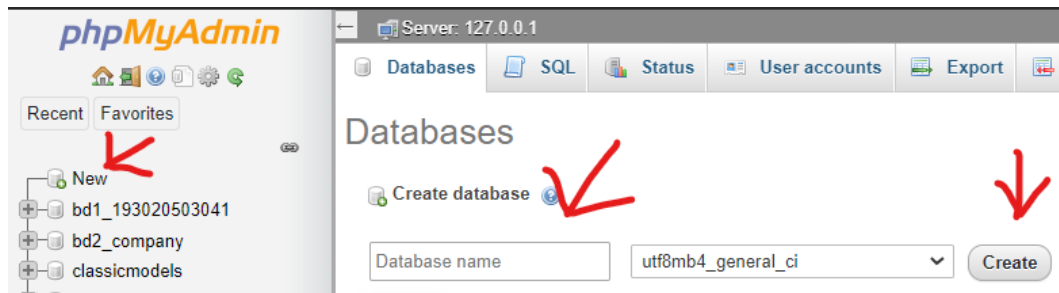
Pada bagian ini diminta untuk membuat database yang menyimpan data-data pada tabel pegawai dengan relasi terhadap tabel divisi. Untuk membuat sebuah database harus mempunyai aplikasi MySql dan SQLyog.

Langkah pertama jalankan aplikasi Mysql dan nyalakan Apache dan MySQL. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. Sedangkan MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database. Berikut ini gambarnya.



Gambar 2.1 Aplikasi XAMPP

Langkah berikutnya untuk memudahkan dalam pembuatan database menggunakan localhost/phpmyadmin menggunakan browser. Buat databasenya dengan cara klik new, masukkan nama databasenya dan klik create. Berikut ini gambarnya.



Gambar 2.2 Tampilan pembuatan database

Setelah database dibuat, berikutnya membuat 2 buah tabel yaitu tabel divisi dan tabel pegawai. Pada tabel divisi memuat data-data id divisi dan nama divisinya, pada tabel tersebut id divisi dibuat menjadi foreign key. Lalu pada tabel pegawai memuat data-data id pegawai, nama dan id divisi, pada tabel tersebut id pegawai dibuat menjadi foreign key dan id divisi dijadikan primary key. Berikut gambar dari tampilan data tabel divisi dan tabel pegawai.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_divisi	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	nama_divisi	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 2.3 Tampilan data tabel divisi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_pegawai	int(2)			No	None			Change Drop More
<input type="checkbox"/> 2	nama	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	id_divisi	int(11)			Yes	NULL			Change Drop More

Gambar 2.4 Tampilan data tabel pegawai

Untuk memasukkan data pada tabel dapat menekan insert pada tabel yang ingin di inputkan datanya. Berikut ini gambar dari inputan data.

Gambar 2.5 Tampilan insert data

+ Options								
			id_divisi	nama_divisi				
<input type="checkbox"/>		Edit		Copy		Delete	1	Masyarakat
<input type="checkbox"/>		Edit		Copy		Delete	2	Komunikasi
<input type="checkbox"/>		Edit		Copy		Delete	3	IT
<input type="checkbox"/>		Edit		Copy		Delete	4	Keuangan

Gambar 2.6 Tampilan tabel divisi

+ Options									
			id_pegawai	nama	id_divisi				
<input type="checkbox"/>		Edit		Copy		Delete	1	Kejer	3
<input type="checkbox"/>		Edit		Copy		Delete	2	Kevin	3
<input type="checkbox"/>		Edit		Copy		Delete	3	Rio	4
<input type="checkbox"/>		Edit		Copy		Delete	4	Abi	3

Gambar 2.7 Tampilan tabel pegawai

2.2. Buatlah program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut

Pada bagian ini diminta untuk membuat sebuah program atau sintaks untuk melakukan penyimpanan atau input data, menghapus data, dan mengubah data.

Dalam program diatas terdapat 4 buah file php yang bernama function, index, insertdata, dan updatedata. Pada masing-masing file tersebut memiliki sintaks dan fungsi yang berbeda. Berikut ini pembahasan dari setiap file.

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL. Pada file function.php terdapat kode untuk menghubungkan php dengan MySQL, kita menggunakan fungsi `mysqli_connect()`.

Dengan parameter alamat server, user, password, dan nama database. Silahkan isi password sesuai dengan yang dibuat di servernya. Jika tidak menggunakan password, maka dikosongkan saja. Berikut adalah contoh kode program agar terhubung dengan server mySQL.

```
<?php
$servername = "localhost";
$dbname = "modul3";
$username = "root";
$password = "";
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

Gambar 2.8 Query koneksi database

Pada kode selanjutnya membuat beberapa fungsi seperti fungsi show yang mana berfungsi untuk menampilkan data yang terdapat pada database modul3 kedalam program web.

Fungsi insert berfungsi memasukkan data yang telah di inputkan dari program web kedalam 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 dan 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,..)

Fungsi delete berfungsi untuk menghapuskan data yang terdapat pada program web dan data pada 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.

Fungsi update berfungsi untuk mengubah data yang ingin dirubah pada kolom tertentu sesuai keinginan user. 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 ...

Berikut ini gambar dari fungsi-fungsi yang dijelaskan diatas.

```

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_divisi = $_POST["id_divisi$id_divisi"];
    mysqli_query($conn, "INSERT INTO pegawai VALUES('$id_pegawai','$nama','$id_divisi')");
    return mysqli_affected_rows($conn);
}

function delete($data){
    global $conn;
    $id_pegawai = $_GET["delete"];
    mysqli_query($conn, "DELETE FROM 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_divisi = $_POST["id_divisi$id_divisi"];
    mysqli_query($conn, "UPDATE pegawai SET nama = '$nama', id_divisi = '$id_divisi'
    WHERE id_pegawai = id_pegawai");
    return mysqli_affected_rows($conn);
}
}

```

Gambar 2.9 Query fungsi yang terdapat pada file function.php

File index.php merupakan file yang pertama kali dijalankan saat dihubungkan dengan localhost. Pada file ini terdapat kode untuk melakukan pengecekan data, yang mana apabila saat melakukan penghapusan data maka akan muncul sebuah pop up. Apabila dalam penghapusan berhasil maka akan muncul pop up penghapusan data berhasil dan apabila gagal maka akan muncul pop up penghapusan data gagal. Berikut ini kodenya.

```

<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
if (isset($_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>";
    }
}
?>

```

Gambar 2.10 Kode memunculkan pop up saat menghapus data

Dalam index terdapat juga kode untuk menampilkan sebuah tombol yang apabila ditekan akan menjalankan sebuah aksi yang membuat user menuju file insertdata.php. Dan pada bagian ini juga menampilkan sebuah tabel yang berisi id_pegawai, nama, id_divisi dan pilihan. Pada bagian kolom id_pegawai, nama dan id_divisi akan memanggil data dari dalam database yang telah dihubungkan sebelumnya, lalu dipanggil menggunakan perintah echo sehingga akan mencetak data yang terdapat pada database. Pada bagian kolom terdapat 2 pilihan yaitu delete dan update. Pada pilihan tersebut terdapat sebuah teks yang dapat ditekan yang mana akan mengeksekusi aksi yang dipilih. Pada saat delete dipilih maka akan menghapus data yang dipilih lalu akan memunculkan pop up data berhasil dihapus. Pada saat update dipilih maka program akan menjalankan sebuah aksi yang dimana aksi ini akan membuat user menuju file updatedata.php.

Method yang digunakan dalam file dibawah adalah GET merupakan sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. Berikut ini tampilan kodenya.

```
<body>
  <form action="insertdata.php">
    <button>INSERT DATA</button>
  </form>
  <table border="1">
    <tr>
      <td>id_pegawai</td>
      <td>nama</td>
      <td>id_divisi</td>
      <td>pilihan</td>
    </tr>
    <tr>
      <td><?php
        foreach ($data as $d) {
          <tr>
            <td><?php echo $d["id_pegawai"]; ?></td>
            <td><?php echo $d["nama"]; ?></td>
            <td><?php echo $d["id_divisi"]; ?></td>
            <td>
              <form action="" method="GET">
                <button type="submit" name="delete" value=<?php echo $d["id_pegawai"]; ?>>DELETE</button>
              </form>
              <form action="updatedata.php" method="GET">
                <button type="submit" name="update" value=<?php echo $d["id_pegawai"]; ?>>UPDATE</button>
              </form>
            </td>
          </tr>
        }
      </td>
    </tr>
  </table>
</body>
</html>
```

Gambar 2.11 Kode bagian body dari file index.php

Pada file insertdata.php merupakan file yang dapat menampilkan sebuah form inputan yang mana dapat menginputkan data ke dalam tabel. Method yang digunakan pada file ini adalah POST merupakan sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST. Menggunakan tag input type yang mana berfungsi untuk menginputkan nilai yang diketikkan kedalam labelnya.

```
<!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_divisi">id_divisi</label>
        <input type="text" name="id_divisi" id="id_divisi">
      </li>
      <li>
        <button type="submit" name="insert">insert</button>
      </li>
    </ul>
  </form>
</body>
</html>
```

Gambar 2.12 Kode bagian body dari file insertdata.php

Pada saat data telah di inputkan tekan insert. Saat insert ditekan maka akan muncul pop up yaitu Data Berhasil Ditambah, dikarenakan kode berikut ini.

```
<?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>";
  }
}
?>
```

Gambar 2.13 Kode memunculkan pop up insert

Pada file `updatedata.php` merupakan file yang dapat menampilkan sebuah form inputan yang mana dapat merubah nilai dalam kolom tanpa merubah nilai pada kolom lainnya. Method yang digunakan pada file ini adalah POST merupakan sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST. Menggunakan tag input type yang mana berfungsi untuk menginputkan nilai yang diketikkan kedalam labelnya.

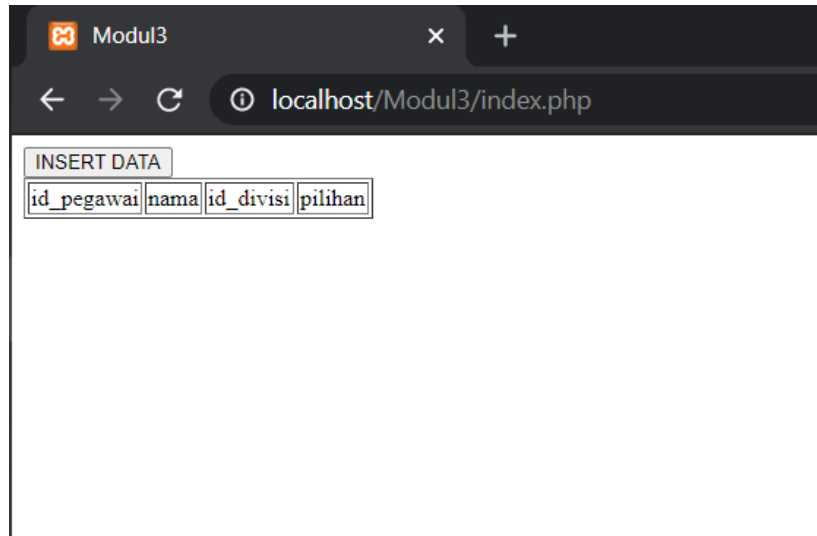
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Update</title>
</head>
<body>
  <a href="index.php">home</a>
  <form action="" method="POST">
    <?php
    foreach ($data as $d) {
    ?>
      <ul>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

Gambar 2.14 Kode bagian body dari file `updatedata.php`

Pada saat data telah di inputkan tekan update. Saat update ditekan maka akan muncul pop up yaitu Data Berhasil Diupdate, dikarenakan kode berikut ini.

```
<?php
require("function.php");
$id_pegawai = $_GET["update"];
$data = show("SELECT * FROM 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>";
  }
}
?>
```

Gambar 2.15 Kode memunculkan pop up update



Gambar 2.16 Tampilan index

INSERT DATA			
id_pegawai	nama	id_divisi	pilihan
1	Kevin	4	DELETE UPDATE
2	Rio	4	DELETE UPDATE
3	Cakra	4	DELETE UPDATE

Gambar 2.17 Tampilan tabel data

[home](#)

- id_pegawai
- nama
- id_divisi
-

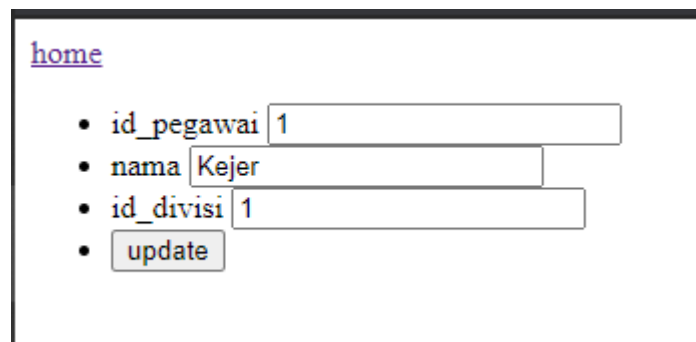
Gambar 2.18 Tampilan insert data



Gambar 2.19 Tampilan pop up insert data



Gambar 2.20 Tampilan pop up hapus data



Gambar 2.21 Tampilan update data



Gambar 2.22 Tampilan pop up update data

BAB III

KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa dalam pembuatan website yang menggunakan database dapat dilakukan menggunakan php dan mysql, dengan adanya mysql ini membuat programmer dapat menyimpan banyak data kedalam sebuah database dan dapat ditampilkan kedalam web dengan menggunakan php. Cara pengaplikasiannya yang sederhana dapat membantu para pemula belajar dalam pembuatan sebuah web yang di koneksikan dengan database sehingga data yang di inputkan kedalam web tersebut tidak hilang melainkan ikut masuk kedalam database.

DAFTAR PUSTAKA

Dosen Teknik Informatika. Pemrograman Web dan Mobile 1. 2021.

MODUL PRAKTIKUM PEMROGRAMAN WEB I.

Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika

Muhardian, A. (2017). *Tutorial PHP dan MySQL: Membuat Aplikasi CRUD [Studi Kasus Pendaftaran Siswa Baru]*.

<https://www.petanikode.com/tutorial-php-mysql/>

Peranginangin, K. (2010). *PHP dan MySQL*. 1.

LAMPIRAN

```
<?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 1.1 Koneksi

```
<?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 Koneksi

```
<?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)) {
```

Gambar 1.3 Membuat Database

```

        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.4 Membuat Database

```

<?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 (

```

Gambar 1.5 Membuat Tabel

```

        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 1.6 Membuat Tabel

```

<?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 1.7 Memasukan Data

```

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

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

```

Gambar 1.8 Memasukan Data

```

    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 1.9 Memasukan Data

```

<?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());
    }

```

Gambar 1.10 Mengambil Data

```

$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 1.11 Mengambil Data

```

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

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

```

Gambar 1.12 Menghapus

```

// 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 1.13 Menghapus

```

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

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

```

Gambar 1.14 Update Data

```

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

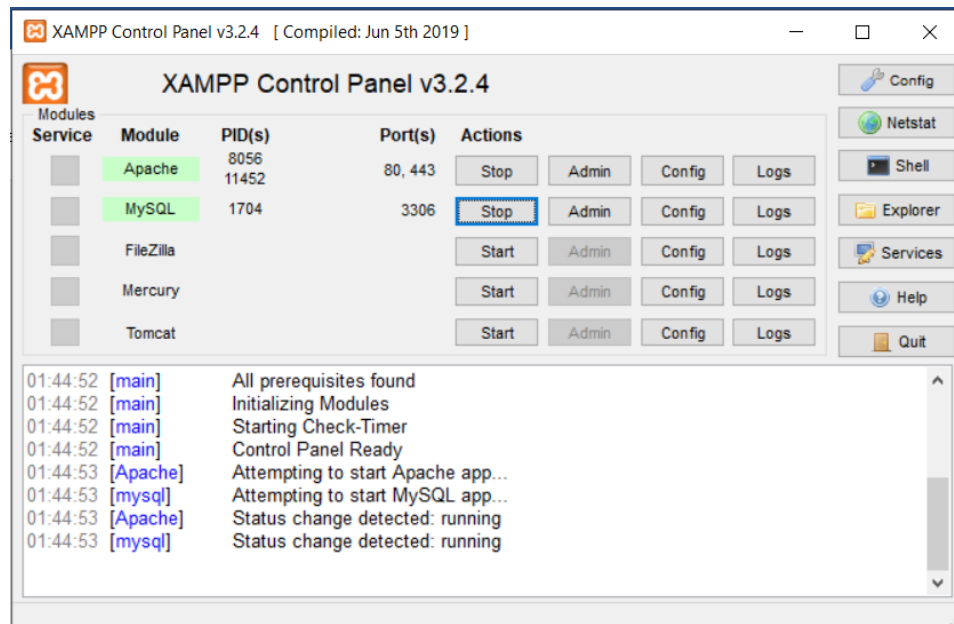
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

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

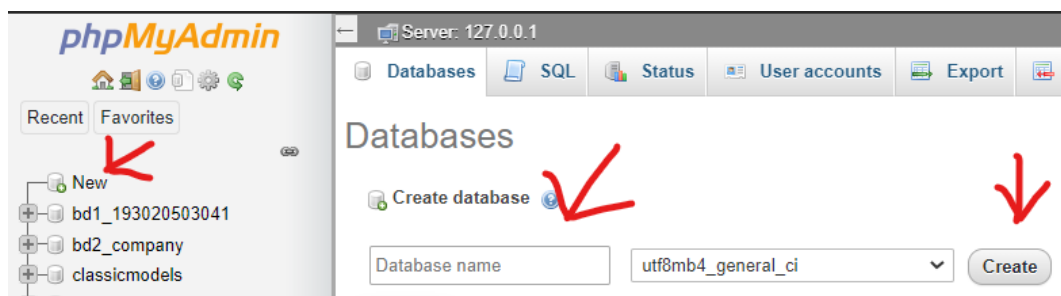
mysqli_close($conn);
?>

```

Gambar 1.15 Update Data



Gambar 2.1 Aplikasi XAMPP



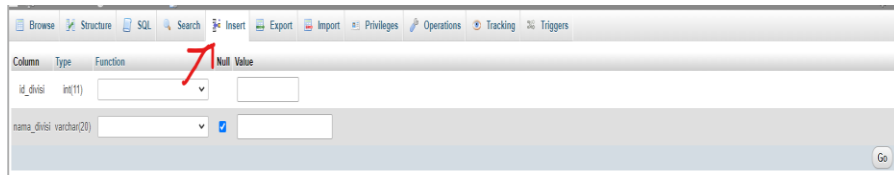
Gambar 2.2 Tampilan pembuatan database

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_divisi	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	nama_divisi	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 2.3 Tampilan data tabel divisi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_pegawai	int(2)			No	None			Change Drop More
<input type="checkbox"/> 2	nama	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	id_divisi	int(11)			Yes	NULL			Change Drop More

Gambar 2.4 Tampilan data tabel pegawai



Gambar 2.5 Tampilan insert data

+ Options

					id_divisi	nama_divisi
<input type="checkbox"/>		Edit		Copy		Delete
					1	Masyarakat
<input type="checkbox"/>		Edit		Copy		Delete
					2	Komunikasi
<input type="checkbox"/>		Edit		Copy		Delete
					3	IT
<input type="checkbox"/>		Edit		Copy		Delete
					4	Keuangan

Gambar 2.6 Tampilan tabel divisi

+ Options

					id_pegawai	nama	id_divisi
<input type="checkbox"/>		Edit		Copy		Delete	
					1	Kejer	3
<input type="checkbox"/>		Edit		Copy		Delete	
					2	Kevin	3
<input type="checkbox"/>		Edit		Copy		Delete	
					3	Rio	4
<input type="checkbox"/>		Edit		Copy		Delete	
					4	Abi	3

Gambar 2.7 Tampilan tabel pegawai

```
<?php
$servername = "localhost";
$dbname = "modul3";
$username = "root";
$password = "";
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

Gambar 2.8 Query koneksi database

```

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_divisi = $_POST["id_divisi$id_divisi"];
    mysqli_query($conn, "INSERT INTO pegawai VALUES('$id_pegawai','$nama','$id_divisi')");
    return mysqli_affected_rows($conn);
}

function delete($data){
    global $conn;
    $id_pegawai = $_GET["delete"];
    mysqli_query($conn, "DELETE FROM 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_divisi = $_POST["id_divisi$id_divisi"];
    mysqli_query($conn, "UPDATE pegawai SET nama = '$nama', id_divisi = '$id_divisi'
    WHERE id_pegawai = id_pegawai");
    return mysqli_affected_rows($conn);
}
?>

```

Gambar 2.9 Query fungsi yang terdapat pada file function.php

```

<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
if (isset($_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>";
    }
}
?>

```

Gambar 2.10 Kode memunculkan pop up saat menghapus data

```

<body>
  <form action="insertdata.php">
    <button>INSERT DATA</button>
  </form>
  <table border="1">
    <tr>
      <td>id_pegawai</td>
      <td>nama</td>
      <td>id_divisi</td>
      <td>pilihan</td>
    </tr>
    <tr>
      <td>
        <?php
        foreach ($data as $d) {
          <tr>
            <td><?php echo $d["id_pegawai"]; ?></td>
            <td><?php echo $d["nama"]; ?></td>
            <td><?php echo $d["id_divisi"]; ?></td>
            <td>
              <form action="" method="GET">
                <button type="submit" name="delete" value=<?php echo $d["id_pegawai"]; ?>>DELETE</button>
              </form>
              <form action="updatedata.php" method="GET">
                <button type="submit" name="update" value=<?php echo $d["id_pegawai"]; ?>>UPDATE</button>
              </form>
            </td>
          </tr>
        </td>
      </tr>
    </table>
  </body>
</html>

```

Gambar 2.11 Kode bagian body dari file index.php

```

<!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_divisi">id_divisi</label>
        <input type="text" name="id_divisi" id="id_divisi">
      </li>
      <li>
        <button type="submit" name="insert">insert</button>
      </li>
    </ul>
  </form>
</body>
</html>

```

Gambar 2.12 Kode bagian body dari file insertdata.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>";
    }
}
?>

```

Gambar 2.13 Kode memunculkan pop up insert

```

<!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=?php 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_divisi">id_divisi</label>
                    <input type="text" name="id_divisi" id="id_divisi" value=?php echo $d["id_divisi"]; ?>>
                </li>
                <li>
                    <button type="submit" name="ubah">update</button>
                </li>
            </ul>
            <?php
            }
            ?>
        </form>
    </body>
</html>

```

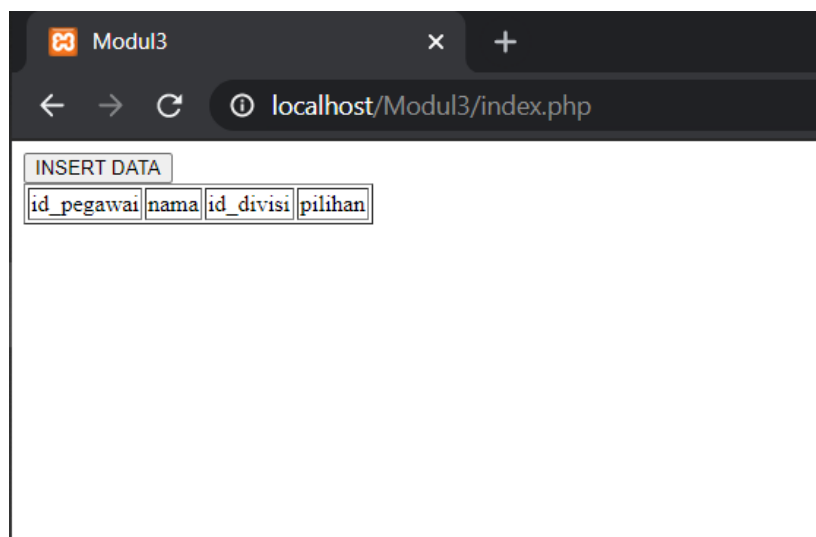
Gambar 2.14 Kode bagian body dari file updatedata.php


```

<?php
require("function.php");
$id_pegawai = $_GET["update"];
$data = show("SELECT * FROM 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>";
    }
}
?>

```

Gambar 2.15 Kode memunculkan pop up update



Gambar 2.16 Tampilan index

INSERT DATA			
id_pegawai	nama	id_divisi	pilihan
1	Kevin	4	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	Rio	4	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	Cakra	4	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.17 Tampilan tabel data

[home](#)

- id_pegawai
- nama
- id_divisi
-

Gambar 2.18 Tampilan insert data



Gambar 2.19 Tampilan pop up insert data



Gambar 2.20 Tampilan pop up hapus data

[home](#)

- id_pegawai
- nama
- id_divisi
-

Gambar 2.21 Tampilan update data



Gambar 2.22 Tampilan pop up update data