

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



NAMA : EVAN ALPHARIO IMANUEL
NIM : 193030503059
KELAS : A
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 1.1 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, $dbname);
    // Memeriksa Koneksi
    if (!$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

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.

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 1.4 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);
    }

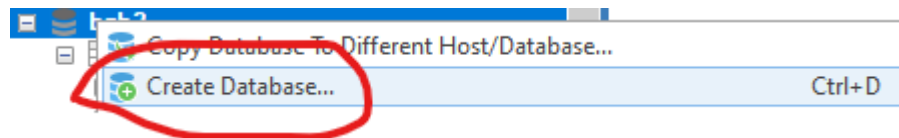
    mysqli_close($conn);
?>
```

Gambar 1.5 Update Data

BAB II

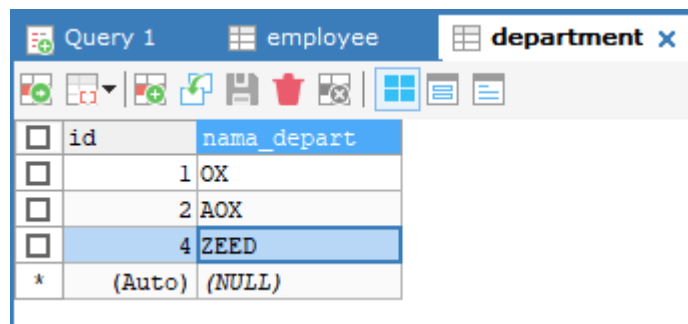
PEMBAHASAN

Pada tugas yang **pertama** kita diminta untuk membuat database pada MySQL untuk menyimpan data-data pegawai beserta relasi tabelnya. Untuk membuat database nya dengan cara klik kanan pada localhost database pada aplikasi SQLyog lalu pilih create database seperti berikut :



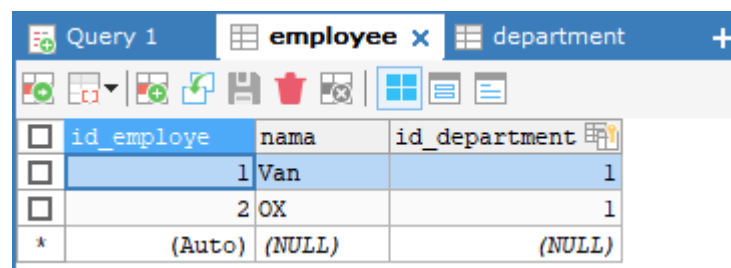
Gambar 2.1 Pembuatan Database.

Kemudian lanjutkan dengan membuat tabel yang bernama department dan employee yang dimana berisikan sebagai berikut :

A screenshot of the SQLyog application showing the 'department' table. The table has two columns: 'id' and 'nama_depart'. The data rows are: (1, 'OX'), (2, 'AOX'), (4, 'ZEED'), and a new row with '(Auto)' and '(NULL)'.

id	nama_depart
1	OX
2	AOX
4	ZEED
*	(Auto) (NULL)

Gambar 2.2 Tabel Department.

A screenshot of the SQLyog application showing the 'employee' table. The table has three columns: 'id_employe', 'nama', and 'id_department'. The data rows are: (1, 'Van', 1), (2, 'OX', 1), and a new row with '(Auto)', '(NULL)', and '(NULL)'.

id_employe	nama	id_department
1	Van	1
2	OX	1
*	(Auto)	(NULL)

Gambar 2.3 Tabel Employee atau pegawai.

Dari kedua tabel diatas relasi nya adalah dimana department memiliki pegawai dan pegawai bekerja di suatu department. Yang dimana tabel diatas telah dimasukan data.

Pada tugas yang **kedua** kita diminta untuk membuat program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut.

Program file yang pertama yaitu berupa function.php yang dimana merupakan tampilan dari seluruh perintah yang diminta pada tugas kedua yang berisikan data-data, insert, delete, update, yang dimana isi dari program tersebut adalah sebagai berikut :

```
<?php
$servername = "localhost";
$dbname = "bab3";
$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_employe = $_POST["id_employe"];
    $nama = $_POST["nama"];
    $id_department = $_POST["id_department$id_department"];

    mysqli_query($conn, "INSERT INTO employee VALUES('$id_employe', '$nama', '$id_department')");
    return mysqli_affected_rows($conn);
}

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

function update($data){
    global $conn;
    $id_employe = $_POST["id_employe"];
    $nama = $_POST["nama"];
```

```

$id_department = $_POST["id_department$id_department"];
mysqli_query($conn, "UPDATE employee SET nama = '$nama',
id_department = '$id_department'
WHERE id_employe = $id_employe");
return mysqli_affected_rows($conn);
}
?>

```

Pada baris function show dari program diatas merupakan fungsi untuk menampilkan data pada tampilan awal web tersebut yang dimana result nya ditampilkan.

Kemudian pada baris function insert merupakan fungsi untuk memasukan dan menyimpan data yang diinputkan kedalam database.

Kemudian pada baris function delete merupakan fungsi untuk menghapus data pegawai pada database yang dimana untuk menghapusnya menggunakan id_employee, dimana ketika button pada tampilan awal ditekan akan langsung menghapus data pegawai tersebut.

Pada baris function update merupakan fungsi untuk mengubah data pegawai pada database dan tabel employee. Yang dimana melakukan update atau perubahan data berdasarkan pada id_employee.

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.

Program file kedua yaitu membuat program untuk menyimpan data pegawai, berikut adalah isi dari program untuk menyimpan data pegawai :

```

<?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>";
    }
}

```

```

        </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_employe">id_employe</label>
                <input type="text" name="id_employe" id="id_employe"
>

            </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_depar
tment">

            </li>
            <li>
                <button type="submit" name="insert">insert</button>
            </li>
        </ul>
    </form>
</body>

</html>

```

Dari program diatas pada bagian awal program berikut :

```

<?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>";
    }
}

```

Program diatas fungsi require berfungsi untuk menyertakan file yaitu file function.php yang kemudian dilanjutkan dengan insert yang dimasukan kedalam 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. pada bagian body pada baris program di bawah ini berfungsi untuk mengatur dan menempatkan posisi tulisan serta membuat tombol pada web yang kita buat.

```

<body>
    <a href="index.php">home</a>
    <form action="" method="POST">
        <ul>
            <li>
                <label for="id_employe">id_employe</label>
                <input type="text" name="id_employe" id="id_employe"
            >
            </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_depar
tment">
            </li>
            <li>
                <button type="submit" name="insert">insert</button>
            </li>
        </ul>
    </form>
</body>
</html>

```

Pada file berikutnya adalah program update yang dimana isi dari program update data pegawai nya adalah sebagai berikut :

```

<?php
require("function.php");
$id_employe = $_GET["update"];

```

```

$data = show("SELECT * FROM employee WHERE id_employe = $id_employe"
);
if (isset($_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_employe">id_employe</label>
                        <input type="text" name="id_employe" id="id_employe" value=<?php echo $d["id_employe"]; ?> 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" value=<?php echo $d["id_department"]; ?>>
                    </li>
                    <li>
                        <button type="submit" name="ubah">update</button>
                    </li>
                </ul>
            }
        </?php>
    </form>

```



```

        </ul>
    <?php
    }
    ?>
</form>
</body>
</html>

```

Pada baris program diatas fungsi require berfungsi untuk menyertakan file yaitu file function.php yang dimana pada baris selanjutnya adalah pernyataan kondisi if yang berisikan 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 inser data untuk pengubahan data tersebut.

Pada file berikutnya adalah file index yang dimana file ini akan menjadi tampilan awal website. Berikut ini adalah isi dari file indeks.php :

```

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

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

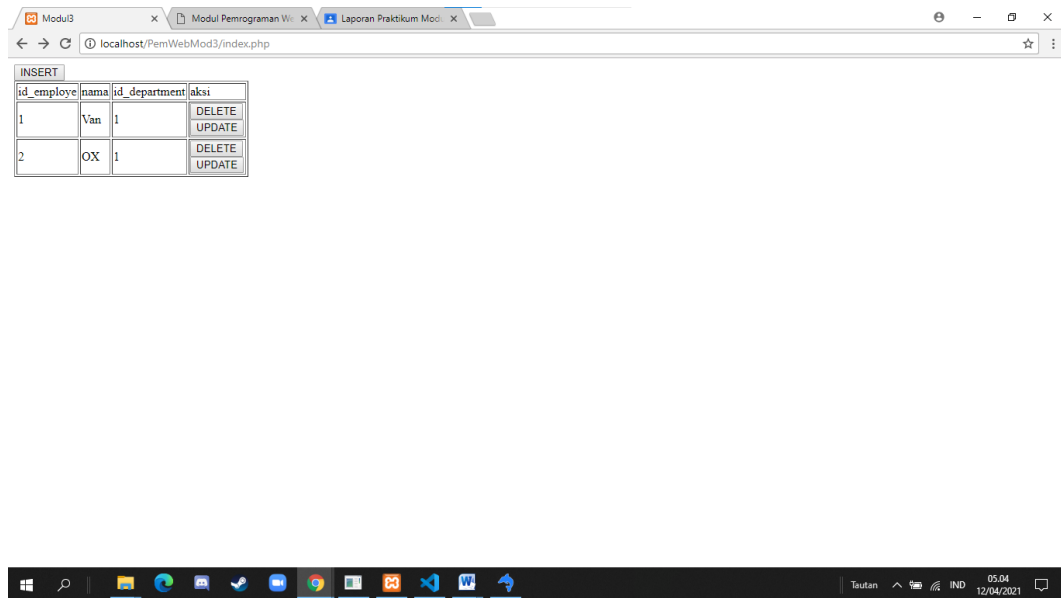
```

```

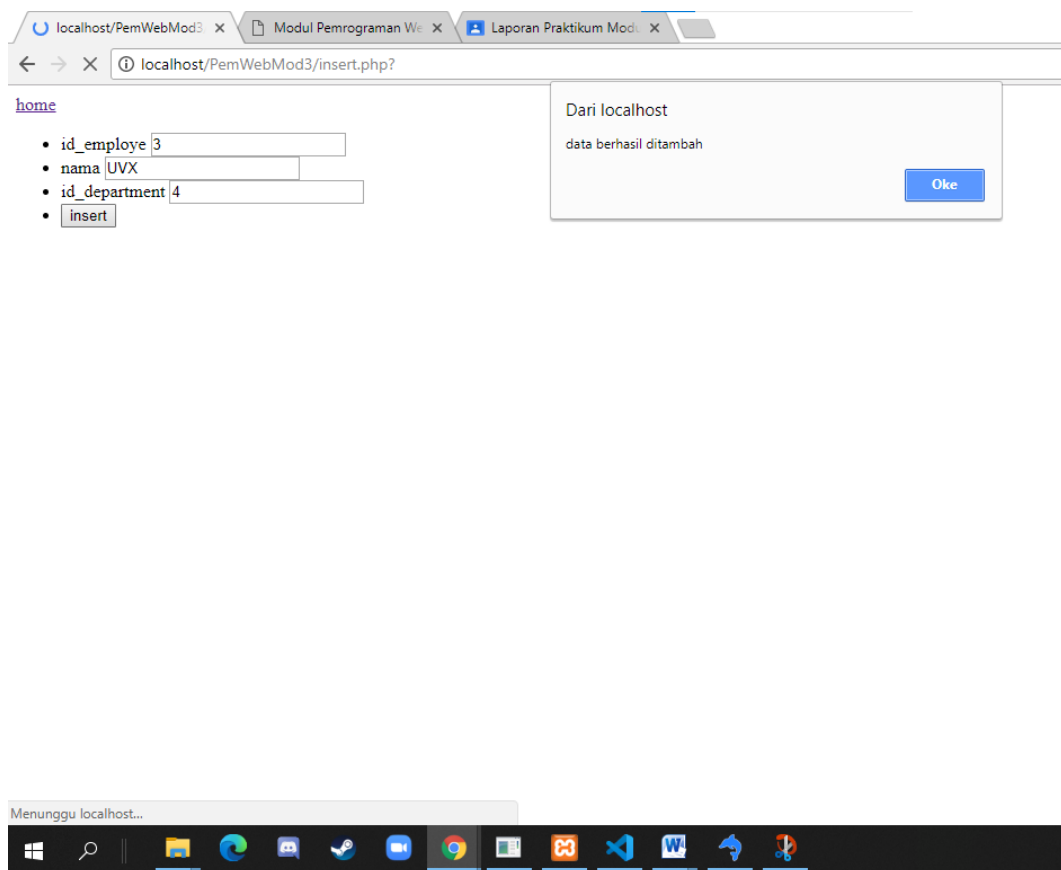
<title>Modul3</title>
</head>
<body>
  <form action="insert.php">
    <button>INSERT</button>
  </form>
  <table border="1">
    <tr>
      <td>id_employe</td>
      <td>nama</td>
      <td>id_department</td>
      <td>aksi</td>
    </tr>
    <?php
    foreach ($data as $d) {
      ?>
      <tr>
        <td><?php echo $d["id_employe"]; ?></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_employe"]; ?>>DELETE</button>
          </form>
          <form action="update.php" method="GET">
            <button type="submit" name="update" value=<?
php echo $d["id_employe"]; ?>>UPDATE</button>
          </form>
        </td>
      </tr>
    <?php
    }
    ?>
  </table>
</body>
</html>

```

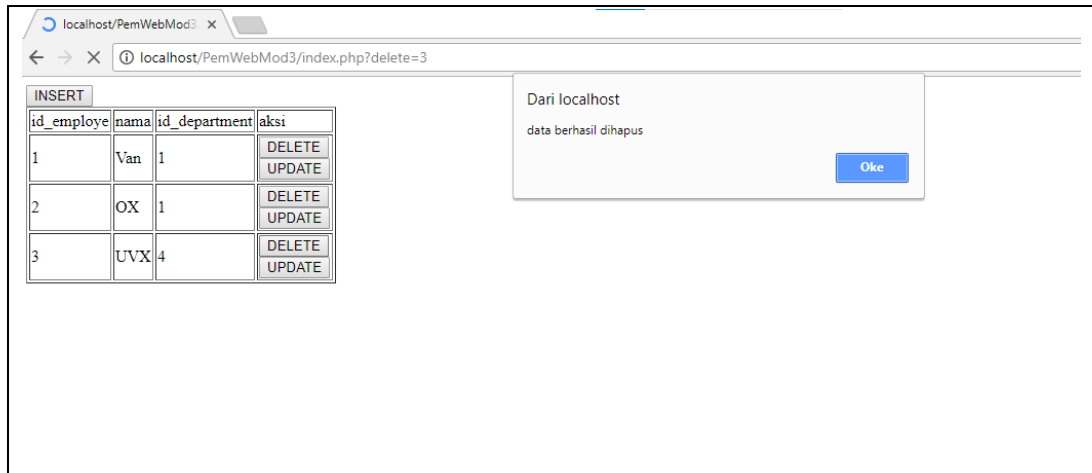
Dari program diatas merupakan tempat untuk mengatur dan mengubah font tulisan serta tempat pembuatan tabel dan tata letak tabel serta ukuran font tabel yang dimana pada baris program diatas adalah tempat berjalan nya fungsi untuk menghapus data pegawai. Yang dimana pada program diatas merupakan tempat untuk menempatkan tombol button insert, delete, dan update pada website tampilan awal nya. Berikut ini adalah tampilan awal dari website dan juga tampilan dari setiap function yang telah dijalankan :



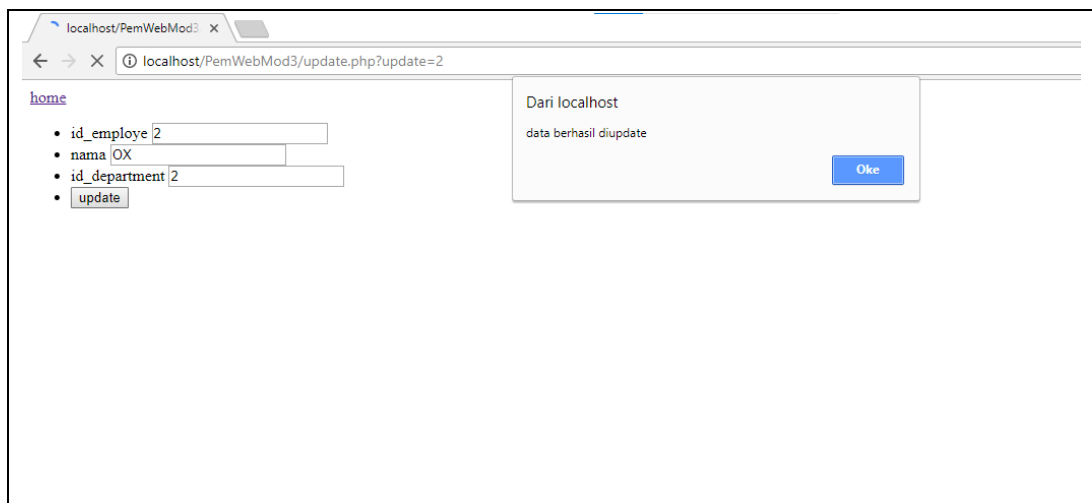
Gambar 2.4 Tampilan awal website.



Gambar 2.5 Insert data pegawai.



Gambar 2.6 Delete Data pegawai.



Gambar 2.7 Update Data Pegawai.

BAB III

KESIMPULAN

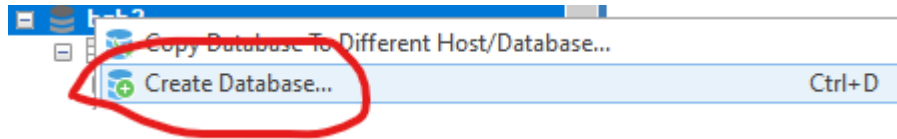
Jadi dari praktikum kali ini dapat disimpulkan bahwa dengan adanya mysql serta php membuat pembuatan website yang menggunakan database dapat dilakukan, penggunaannya yang dapat dilakukan secara sederhana dapat membantu dalam pembelajaran pembuatan web bagi pemula, sehingga mudah dipahami.

DAFTAR PUSTAKA

Fungsi Mysql_connect di PHP - MySQLi connect. (n.d.).

Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan
Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

LAMPIRAN



Query 1 employee department x

id	nama_depart
1	OX
2	AOX
4	ZEED
*	(Auto) (NULL)

Query 1 employee x department +

id_employee	nama	id_department
1	Van	1
2	OX	1
*	(Auto) (NULL)	(NULL)

Modu3 Modul Pemrograman W... Laporan Praktikum Mod... localhost/PemWebMod3/index.php

id_employee	nama	id_department	aksi
1	Van	1	DELETE UPDATE
2	OX	1	DELETE UPDATE

localhost/PemWebMod3 x Modul Pemrograman W x Laporan Praktikum Mod x

localhost/PemWebMod3/insert.php?

[home](#)

- id_employe 3
- nama UVX
- id_department 4
-

Dari localhost

data berhasil ditambah



localhost/PemWebMod3 x

localhost/PemWebMod3/index.php?delete=3

id_employe	nama	id_department	aksi
1	Van	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	OX	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	UVX	4	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Dari localhost

data berhasil dihapus

localhost/PemWebMod3 x

localhost/PemWebMod3/update.php?update=2

[home](#)

- id_employe 2
- nama OX
- id_department 2
-

Dari localhost

data berhasil diupdate