

LAPORAN PRATIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Rio Geradi Muke
NIM : 193030503051
MODUL : III
KELAS : A

JURUSAN/PROGRAM STUDI 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);
?>

```

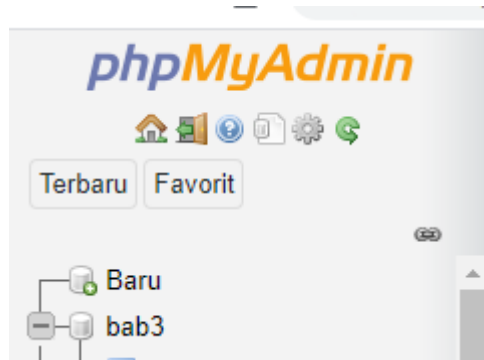
Gambar 1.5 Update Data

BAB II

PEMBAHASAN

Tugas Praktikum

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



Gambar 2.1 Membuat data base

Pertama masuk ke web <http://localhost/phpmyadmin>, lalu setelah itu akan memasuki sebuah website PHPMYADMIN untuk membuat data base setelah itu create data basenya dengan cara klik baru seperti gambar 2,1 tersebut,



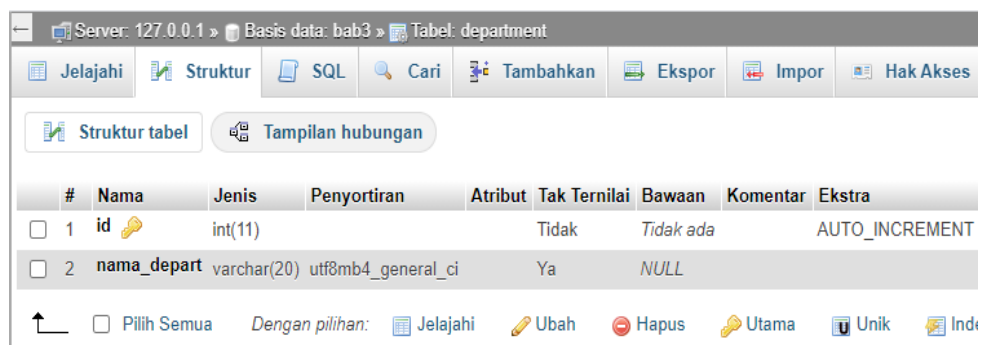
Gambar 2.2 Membuat data base.

Tinggal klik buat lalu setelah itu tinggal masukan nama basis data dan setelah itu buat data basenya tersebut.



Gambar 2.3 Data base SQL

Untuk gambar 2.3 adalah terdapat isi data id department dan nama_depatment lalu untuk memasukan data tersebut yaitu dengan cara insert into nama databasesnya adalah bab3 lalu isi data tersebut sesuai dengan tabelnya terdapat id department dan nama department adalah id 1, nama_department RGM dan seterusnya cara membuat databasenya. Lalu untuk id atau id_department adalah primary key. Apa bila membuat database di mysql menambah primary adalah create table (id_department primary key); jadi itu adalah query untuk membuat sebuah primary key.



Gambar 2.4 Data base SQL

Untuk gambar 2.4 adalah membuat tipe datanya yaitu id dan nama_departmentnya setelah itu tipe datanya adalah id int(11) maksudnya int adalah tipe data integer lalu 11 adalah panjang tipe data tersebut di dalam id lalu untuk nama_departmentnya adalah varchar(20) adalah untuk bisa semua huruf dan angka, dan 20 adalah tipe panjang data tersebut.

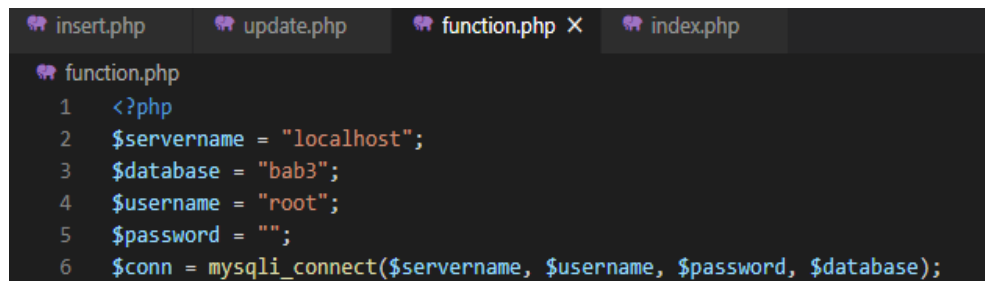
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tind
1	id_employe	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	
2	nama	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			
3	id_department	int(11)		Ya		NULL			

Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
Ubah Hapus	PRIMARY	BTREE	Ya	Tidak	id_employe	0	A	Tidak	
Ubah Hapus	id_department	BTREE	Tidak	Tidak	id_department	0	A	Ya	

Gambar 2.5 Employee

Pertama kita membuat sebuah tabel di employee yaitu employee adalah pegawai atau karyawan maka isi tabel tersebut adalah id_employe, nama dan id_department untuk id_employe saya masukan kedalam primary key dan id_department di dalam tabel employee adalah foreign karena ada relasi di dalam tabel department tersebut. Lalu id_employe int(11) adalah integer dan data panjang tipenya adalah 11 setelah itu untuk nama menggunakan varchar(20) yaitu agar bisa digunakan huruf dan angka jadi dan panjang datanya adalah 20. Lalu id_department int(11) tipe datanya integer dan panjang datanya adalah 11. Semuanya seperti itu.

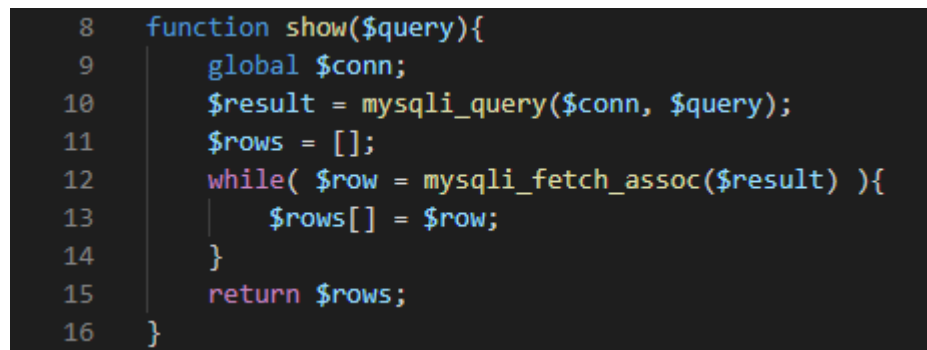
2. Buatlah program untuk menyimpan, menghapus dan mengubah data data pegawai tersebut.



```
function.php
1  <?php
2  $servername = "localhost";
3  $database = "bab3";
4  $username = "root";
5  $password = "";
6  $conn = mysqli_connect($servername, $username, $password, $database);
```

Gambar 2.6 Function

Kegunaan dari gambar 2,6 adalah menjelaskan untuk menghubungkan database tersebut kedalam sebuah Bahasa pemrograman php jadi agar terdapat tampilan website tersebut. Lalu untuk `mysqli_connect` adalah untuk membuat koneksi atau hubungan. Dan semuanya harus menggunakan itu.



```
8  function show($query){
9      global $conn;
10     $result = mysqli_query($conn, $query);
11     $rows = [];
12     while( $row = mysqli_fetch_assoc($result) ){
13         $rows[] = $row;
14     }
15     return $rows;
16 }
```

Gambar 2.7 Function

Untuk mengambil sebuah data di dalam query atau database dan untuk pengambilan data tersebut bisa menggunakan sebuah kondisi WHERE.



```
17 function insert($data){
18     global $conn;
19     $id_employe = $_POST["id_employe"];
20     $nama = $_POST["nama"];
21     $id_department = $_POST["id_department"];
22
23     mysqli_query($conn, "INSERT INTO employee VALUES('$id_employe','$nama','$id_department')");
24     return mysqli_affected_rows($conn);
25 }
```

Gambar 2.8 Function

Untuk memasukan sebuah data base kedalam tabel yaitu menggunakan `insert into` nama atributnya ada 3 yaitu `id_employe`, `nama` dan

id department dan syntax global adalah untuk menghubungkan setiap file program atau juga bisa koneksi setiap file program.

```
26 function delete($data){
27     global $conn;
28     $id_employe = $_GET["delete"];
29     mysqli_query($conn, "DELETE FROM employee WHERE id_employe = $id_employe");
30     return mysqli_affected_rows($conn);
31 }
32 function update($data){
33     global $conn;
34     $id_employe = $_POST["id_employe"];
35     $nama = $_POST["nama"];
36     $id_department = $_POST["id_department$id_department"];
37     mysqli_query($conn, "UPDATE employee SET nama = '$nama',
38     id_department = '$id_department'
39     WHERE id_employe = $id_employe");
40     return mysqli_affected_rows($conn);
41 }
42 ?>
43
```

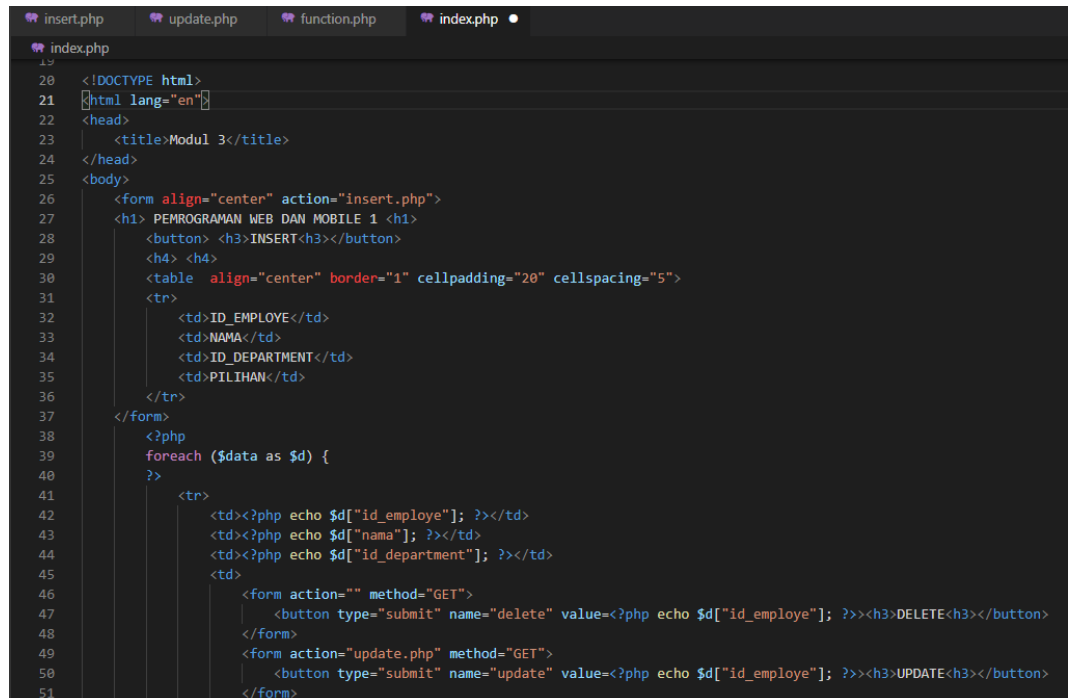
Gambar 2.9 Function

Untuk codingan gambar 2.9 ini adalah digunakan untuk fungsi delete di dalam tabel employee di kondisi dalam id_employe dan ada juga untuk program mengupdate sebuah id_employe, nama dan id_department semua itu di dalam kondisi where di tabel tabel yang ingin di update. Dan fungsi dari syntax global adalah untuk mengkoneksikan semua program tersebut dan bisa di panggil.

```
insert.php  update.php  function.php  index.php X
index.php
1  <?php
2  require("function.php");
3  $data = show("SELECT * FROM employee");
4  if (isset($_GET["delete"])) {
5      if (delete($_GET > 0)) {
6          echo "<script>
7              alert('data berhasil dihapus');
8              document.location.href = 'index.php';
9          </script>";
10     } else {
11         echo "<script>
12             alert('data gagal dihapus');
13             document.location.href = 'index.php';
14         </script>";
15     }
16 }
17
18 ?>
```

Gambar 2.10 Function

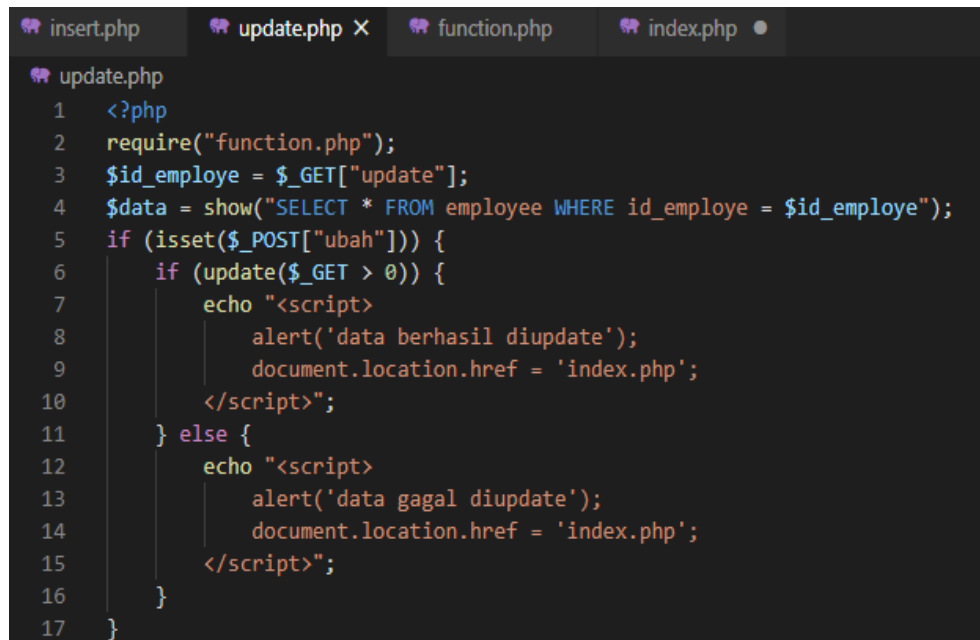
Untuk penjelasan gambar 2.10 adalah fungsi dari program delete yaitu `select*from employee` di jelaskan akan menampilkan data di dalam employee setelah itu saya menggunakan kondisi if dan elsa apa bila data tersebut terdisi atau terdelete maka akan bertulisan data berhasil di hapus lalu apa bila tidak ada data atau maka tulisannya adalah data gagal di hapus.



```
20 <!DOCTYPE html>
21 <html lang="en">
22 <head>
23 <title>Modul 3</title>
24 </head>
25 <body>
26 <form align="center" action="insert.php">
27 <h1> PEMROGRAMAN WEB DAN MOBILE 1 </h1>
28 <button> <h3>INSERT</h3></button>
29 <h4> <h4>
30 <table align="center" border="1" cellpadding="20" cellspacing="5">
31 <tr>
32 <td>ID_EMPLOYE</td>
33 <td>NAMA</td>
34 <td>ID_DEPARTMENT</td>
35 <td>PILIHAN</td>
36 </tr>
37 </form>
38 <?php
39 foreach ($data as $d) {
40 <?
41 <tr>
42 <td><?php echo $d["id_employe"]; ?></td>
43 <td><?php echo $d["nama"]; ?></td>
44 <td><?php echo $d["id_department"]; ?></td>
45 <td>
46 <form action="" method="GET">
47 <button type="submit" name="delete" value=<?php echo $d["id_employe"]; ?>><h3>DELETE</h3></button>
48 </form>
49 <form action="update.php" method="GET">
50 <button type="submit" name="update" value=<?php echo $d["id_employe"]; ?>><h3>UPDATE</h3></button>
51 </form>
```

Gambar 2.11 Insert

Penjelasannya untuk gambar 2.11 adalah terdapat program untuk insert.php adalah agar tabel di insert memiliki isi tabel yaitu ID_EMPLOYE, NAMA, ID_DEPARTMENT, PILIHAN. Lalu untuk kegunaannya foreach adalah berfungsi pada array, dan digunakan untuk loop melalui setiap pasangan kunci/nilai dalam sebuah array. Di dalam data dan di dalam data insert tersebut untuk memasukkan id_employe, nama dan id_department setelah itu ada juga Atribut metode menentukan bagaimana mengirim formulir-data lalu data formulir dikirim ke halaman yang dinyatakan dalam atribut action. Seperti delete dan update di dalam gambar 2,11 tersebut.



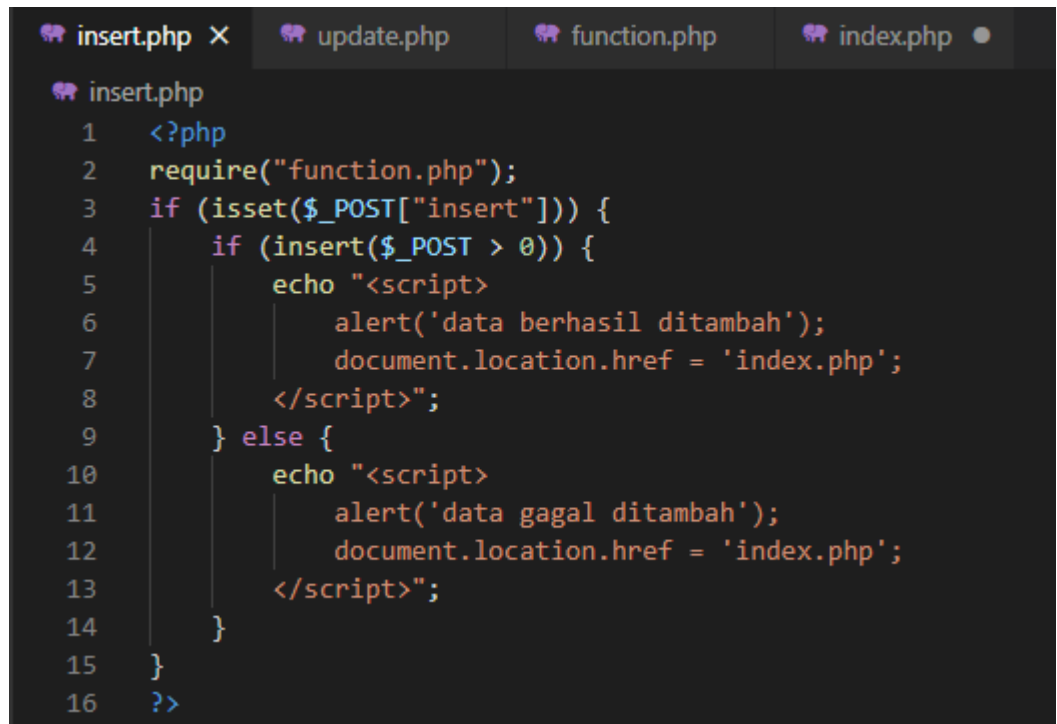
```

1  <?php
2  require("function.php");
3  $id_employe = $_GET["update"];
4  $data = show("SELECT * FROM employee WHERE id_employe = $id_employe");
5  if (isset($_POST["ubah"])) {
6      if (update($_GET > 0)) {
7          echo "<script>
8              alert('data berhasil diupdate');
9              document.location.href = 'index.php';
10             </script>";
11      } else {
12          echo "<script>
13              alert('data gagal diupdate');
14              document.location.href = 'index.php';
15             </script>";
16      }
17  }

```

Gambar 2.12 Update

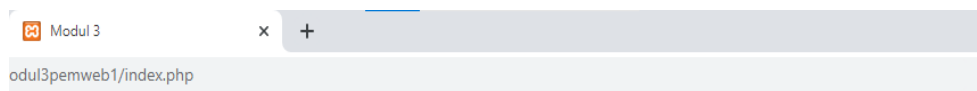
Untuk gambar 2.12 ini adalah fungsi untuk melakukan update pada tabel employe lalu memiliki querynya adalah select*from employee WHERE id_employe lalu setelah itu akan menampilkan untuk mengubah data tabel tersebut yaitu menggunakan sebuah perintah if dan else apa bila gagal di update maka akan bertulisan gagal diupdate lalu sebaliknya apa bila berhasil di update maka akan bertulisan data berhasil diupdate. Dan insert tersebut berrelasi dengan function.php.



```
1  <?php
2  require("function.php");
3  if (isset($_POST["insert"])) {
4      if (insert($_POST) > 0) {
5          echo "<script>
6              alert('data berhasil ditambah');
7              document.location.href = 'index.php';
8          </script>";
9      } else {
10         echo "<script>
11             alert('data gagal ditambah');
12             document.location.href = 'index.php';
13         </script>";
14     }
15 }
16 ?>
```

Gambar 2.13 Insert

Lalu untuk gambar 2.13 kegunaan dari require untuk memerlukan sebuah data di function.php setelah itu isset adalah untuk memeriksa apakah memiliki variabel kosong. Lalu memiliki kondisi if dan else dan apa bila data tersebut sudah di masukan atau di tambahkan maka akan bertuliskan data berhasil ditambah dan apa bila tidak ada data yang ditambahkan maka akan bertuliskan data gagal di tambah setelah itu akan terhubung kedalam file index.php karena saling berelasi setiap program atau sintax tersebut.



PEMROGRAMAN WEB DAN MOBILE 1

INSERT			
ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN

Gambar 2.14 Tampilan

PEMROGRAMAN WEB DAN MOBILE 1

INSERT

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
1	AMAT	1	<div>DELETE</div> <div>UPDATE</div>
2	UDON	9	<div>DELETE</div> <div>UPDATE</div>

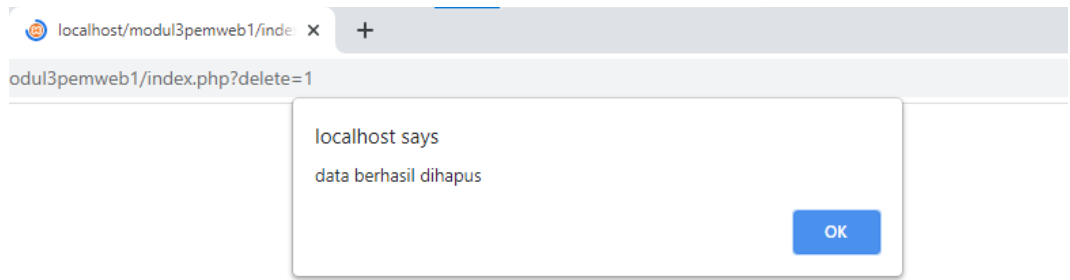
Gambar 2.15 Tampilan Insert Data

PEMROGRAMAN WEB DAN MOBILE 1

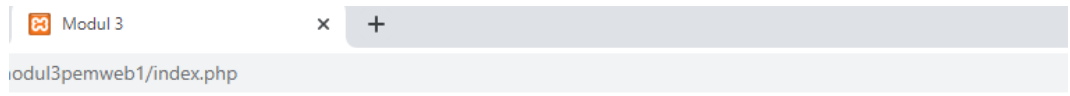
INSERT

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
2	UDON	9	<div>DELETE</div> <div>UPDATE</div>

Gambar 2.16 Tampilan Delete Data



Gambar 2.17 Tampilan Delete Berhasil



PEMROGRAMAN WEB DAN MOBILE 1

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
2	UDON1	1	<div style="text-align: center;"> <input type="button" value="DELETE"/> <input type="button" value="UPDATE"/> </div>

Gambar 2.18 Tampilan Update Berhasil

A screenshot of a web browser window. The address bar shows 'localhost/modul3pemweb1/insert.php?'. Below the address bar, there is a 'KEMBALI' (Back) link. The form contains three input fields: 'ID_EMPLOYE' with value '2', 'NAMA' with value 'UDON1', and 'ID_DEPARTMENT' with value '1'. There is an 'INSERT' button below the fields.

Gambar 2.19 Tampilan Insert Memasukan Data

A screenshot of a web browser window. The address bar shows 'localhost/modul3pemweb1/update.php?update=2'. Below the address bar, there is a 'KEMBALI' (Back) link. The form contains three input fields: 'ID_EMPLOYE' with value '2', 'NAMA' with value 'UDON1', and 'ID_DEPARTMENT' with value '1'. There is an 'UPDATE' button below the fields.

Gambar 2.20 Tampilan Update Data

BAB III

KESIMPULAN

Kesimpulan yang saya dapat dari modul III ini adalah Fungsi `Mysql_query` dalam PHP adalah untuk menjalankan perintah atau instruksi query ke database MySQL dimana `mysql_query()` merupakan sarana dasar yang akan menampung jenis query apapun di dalam nya. antara lain adalah `INSERT`, `SELECT`, `UPDATE`, dan `DELETE`. Fungsi `mysql_query` dapat juga dikatakan sebagai fungsi untuk menampung sebuah perintah dalam atau intruksi ke database MySQL untuk menyabungkan ke Bahasa pemrograman PHP.

DAFTAR PUSTAKA

- Aris, S. (2019). *Tutorial PHP Part 40 – Membuat Koneksi Database PHP dan Mysql dengan konsep OOP _ Warung Belajar.*
- Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*
- Raja, P. (2013). *Fungsi Mysql_query dalam PHP _ MySQLi query.*

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

```
// Periksa 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 1.4 Update Data

```

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

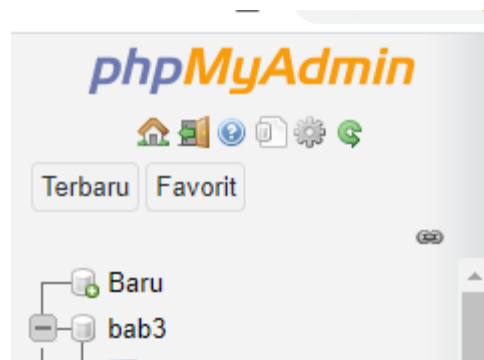
    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

    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



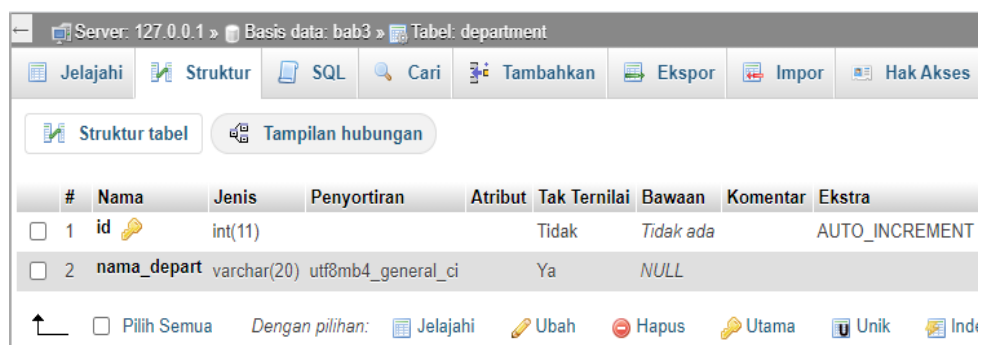
Gambar 2.1 Membuat data base



Gambar 2.2 Membuat data base.



Gambar 2.3 Data base SQL



Gambar 2.4 Data base SQL

Server: 127.0.0.1 » Basis data: bab3 » Tabel: employee

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Ope

Struktur tabel Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tind
<input type="checkbox"/> 1	id_employe	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	U
<input type="checkbox"/> 2	nama	varchar(20) utf8mb4_general_ci			Tidak	Tidak ada			U
<input type="checkbox"/> 3	id_department	int(11)			Ya	NULL			U

☐ Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks

Remove from central columns

Cetak Usulkan struktur tabel Lacak tabel Move columns Normalisasi

Tambahkan kolom setelah id_department

Indeks

Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
Ubah Hapus	PRIMARY	BTREE	Ya	Tidak	id_employe	0	A	Tidak	
Ubah Hapus	id_department	BTREE	Tidak	Tidak	id_department	0	A	Ya	

Gambar 2.5 Employee

insert.php update.php function.php X index.php

```

function.php
1  <?php
2  $servername = "localhost";
3  $database = "bab3";
4  $username = "root";
5  $password = "";
6  $conn = mysqli_connect($servername, $username, $password, $database);
  
```

Gambar 2.6 Function

```

8  function show($query){
9      global $conn;
10     $result = mysqli_query($conn, $query);
11     $rows = [];
12     while( $row = mysqli_fetch_assoc($result) ){
13         $rows[] = $row;
14     }
15     return $rows;
16 }
  
```

Gambar 2.7 Function

```

17 function insert($data){
18     global $conn;
19     $id_employe = $_POST["id_employe"];
20     $nama = $_POST["nama"];
21     $id_department = $_POST["id_department$id_department"];
22
23     mysqli_query($conn, "INSERT INTO employee VALUES('$id_employe','$nama','$id_department')");
24     return mysqli_affected_rows($conn);
25 }

```

Gambar 2.8 Function

```

26 function delete($data){
27     global $conn;
28     $id_employe = $_GET["delete"];
29     mysqli_query($conn, "DELETE FROM employee WHERE id_employe = $id_employe");
30     return mysqli_affected_rows($conn);
31 }
32 function update($data){
33     global $conn;
34     $id_employe = $_POST["id_employe"];
35     $nama = $_POST["nama"];
36     $id_department = $_POST["id_department$id_department"];
37     mysqli_query($conn, "UPDATE employee SET nama = '$nama',
38         id_department = '$id_department'
39         WHERE id_employe = $id_employe");
40     return mysqli_affected_rows($conn);
41 }
42 ?>
43

```

Gambar 2.9 Function

```

insert.php  update.php  function.php  index.php X
index.php
1  <?php
2  require("function.php");
3  $data = show("SELECT * FROM employee");
4  if (isset($_GET["delete"])) {
5      if (delete($_GET > 0)) {
6          echo "<script>
7              alert('data berhasil dihapus');
8              document.location.href = 'index.php';
9          </script>";
10     } else {
11         echo "<script>
12             alert('data gagal dihapus');
13             document.location.href = 'index.php';
14         </script>";
15     }
16 }
17
18 ?>

```

Gambar 2.10 Function

```
insert.php update.php function.php index.php ●
index.php
19
20 <!DOCTYPE html>
21 <html lang="en">
22 <head>
23 <title>Modul 3</title>
24 </head>
25 <body>
26 <form align="center" action="insert.php">
27 <h1> PEMROGRAMAN WEB DAN MOBILE 1 </h1>
28 <button> <h3>INSERT</h3></button>
29 <h4> <h4>
30 <table align="center" border="1" cellpadding="20" cellspacing="5">
31 <tr>
32 <td>ID_EMPLOYE</td>
33 <td>NAMA</td>
34 <td>ID_DEPARTMENT</td>
35 <td>PILIHAN</td>
36 </tr>
37 </form>
38 <?php
39 foreach ($data as $d) {
40 ?>
41 <tr>
42 <td><?php echo $d["id_employe"]; ?></td>
43 <td><?php echo $d["nama"]; ?></td>
44 <td><?php echo $d["id_department"]; ?></td>
45 <td>
46 <form action="" method="GET">
47 <button type="submit" name="delete" value=<?php echo $d["id_employe"]; ?>><h3>DELETE</h3></button>
48 </form>
49 <form action="update.php" method="GET">
50 <button type="submit" name="update" value=<?php echo $d["id_employe"]; ?>><h3>UPDATE</h3></button>
51 </form>
```

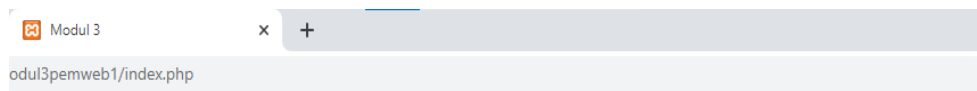
Gambar 2.11 Insert

```
insert.php update.php X function.php index.php ●
update.php
1 <?php
2 require("function.php");
3 $id_employe = $_GET["update"];
4 $data = show("SELECT * FROM employee WHERE id_employe = $id_employe");
5 if (isset($_POST["ubah"])) {
6     if (update($_GET > 0)) {
7         echo "<script>
8             alert('data berhasil diupdate');
9             document.location.href = 'index.php';
10        </script>";
11     } else {
12         echo "<script>
13             alert('data gagal diupdate');
14             document.location.href = 'index.php';
15        </script>";
16     }
17 }
```

Gambar 2.12 Update


```
insert.php X update.php function.php index.php
insert.php
1  <?php
2  require("function.php");
3  if (isset($_POST["insert"])) {
4      if (insert($_POST) > 0) {
5          echo "<script>
6              alert('data berhasil ditambah');
7              document.location.href = 'index.php';
8          </script>";
9      } else {
10         echo "<script>
11             alert('data gagal ditambah');
12             document.location.href = 'index.php';
13         </script>";
14     }
15 }
16 ?>
```

Gambar 2.13 Insert



PEMROGRAMAN WEB DAN MOBILE 1

INSERT			
ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN

Gambar 2.14 Tampilan

PEMROGRAMAN WEB DAN MOBILE 1

INSERT

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
1	AMAT	1	<div>DELETE</div> <div>UPDATE</div>
2	UDON	9	<div>DELETE</div> <div>UPDATE</div>

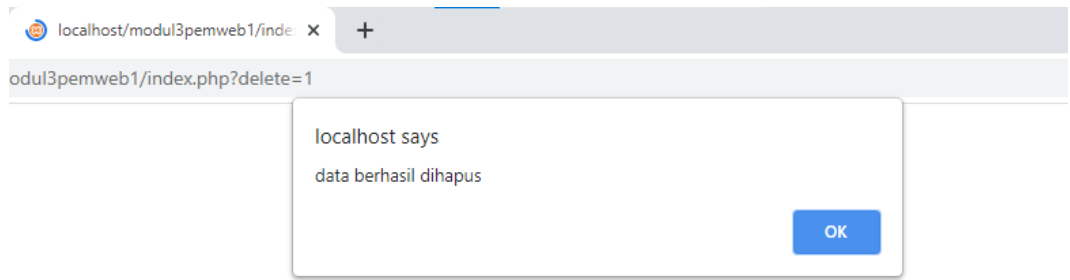
Gambar 2.15 Tampilan Insert Data

PEMROGRAMAN WEB DAN MOBILE 1

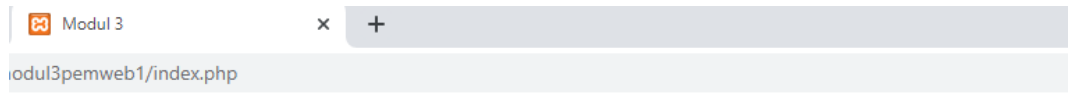
INSERT

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
2	UDON	9	<div>DELETE</div> <div>UPDATE</div>

Gambar 2.16 Tampilan Delete Data



Gambar 2.17 Tampilan Delete Berhasil



PEMROGRAMAN WEB DAN MOBILE 1

ID_EMPLOYE	NAMA	ID_DEPARTMENT	PILIHAN
2	UDON1	1	<div style="text-align: center; margin-bottom: 5px;"> <input type="button" value="DELETE"/> </div> <div style="text-align: center;"> <input type="button" value="UPDATE"/> </div>

Gambar 2.18 Tampilan Update Berhasil

A screenshot of a web browser window. The address bar shows 'localhost/modul3pemweb1/insert.php?'. Below the address bar, there is a 'KEMBALI' (Back) link. The form contains three input fields: 'ID_EMPLOYE' with value '2', 'NAMA' with value 'UDON1', and 'ID_DEPARTMENT' with value '1'. There is an 'INSERT' button below the fields.

Gambar 2.19 Tampilan Insert Memasukan Data

A screenshot of a web browser window. The address bar shows 'localhost/modul3pemweb1/update.php?update=2'. Below the address bar, there is a 'KEMBALI' (Back) link. The form contains three input fields: 'ID_EMPLOYE' with value '2', 'NAMA' with value 'UDON1', and 'ID_DEPARTMENT' with value '1'. There is an 'UPDATE' button below the fields.

Gambar 2.20 Tampilan Update Data