

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE 1**



NAMA : AXEL BERKATI
NIM : 193010503007
KELAS : A
MODUL : III (PHP MYSQL DATABASE)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB 1

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

1.2.1. Cara Membuat Koneksi Database PHP ke MySQL

Basis data (atau database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut(Achmad Solichin. S.Kom, 2016).

Ada dua cara untuk membuat koneksi ke database MySQL dengan PHP, yaitu menggunakan MySQLi dan PDO(Ariata, 2020).

MySQLi merupakan singkatan dari MySQL Improved. Ekstensi MySQL ini menambahkan fitur baru di interface database MySQL. MySQLi bersifat prosedural dan juga berorientasi objek, di mana sifat prosedural adalah atribut yang diwariskan dari versi MySQL yang lebih lama.

MySQL versi awal membagi task ke dalam satu prosedur bertahap (langkah demi langkah) dan saling berhubungan. Tindakan modifikasi akan semakin sulit karena kode harus diubah dari bagian paling pertama. Di sisi lain, MySQLi melihat data sebagai satu set objek yang dapat ditukarkan dan memiliki fungsi serta memperbolehkan pengguna untuk menambah atau menghapus data.

PDO adalah singkatan dari PHP Data Object. Lain halnya dengan MySQLi, PDO justru berorientasi hanya pada objek dan mendukung

sejumlah tipe database yang menggunakan PHP, seperti MySQL, MSSQL, Informix, dan PostgreSQL.

Salah satu fungsi penting yang juga didukung oleh MySQLi dan PDO adalah prepared statements. Fungsi ini akan mempercepat waktu yang dibutuhkan oleh MySQL untuk menjalankan query yang sama berkali-kali. Prepared statement juga berfungsi untuk mencegah terjadinya serangan injeksi SQL ketika menjalankan perubahan di database.

1.2.2. Menggunakan MySQLi untuk Membuat Koneksi PHP ke MySQL

Empat langkah menggunakan MySQLi untuk membuat koneksi database PHP ke MySQL:

- a. Buka File Manager -> public_html.
- b. Buat File Baru dengan mengklik ikon tambah file pada menu di atas layar.
- c. Simpan dengan nama databaseconnect.php, atau nama lain yang Anda inginkan, tapi ekstensinya tetap .php.
- d. Salin dan letakkan barisan kode di bawah ini ke file tersebut. Klik dua kali untuk membukanya. Pastikan Anda sudah mengganti empat value pertama yang ada di bawah <?php dengan informasi (credential) yang dicatat sebelumnya.

```
<?php
$servername = "localhost";
$dbname = "databasename";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password,
$dbname);
// Check connection
```

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

1.2.3. Tentang Kode MySQLi

Fungsi utama yang digunakan di script ini adalah `mysqli_connect()`. Fungsi ini merupakan bagian internal PHP untuk membuat koneksi baru ke server MySQL.

Pada barisan kode di atas, di bagian awalnya terdapat deklarasi variabel serta value yang ditentukan untuk variabel ini. Biasanya ada empat variabel untuk pembuatan koneksi database: `$servername`, `$database`, `$username`, dan `$password`. Informasi database ditetapkan sebagai value untuk keempat variabel tersebut sehingga dapat dikirimkan ke fungsi.

Apabila koneksi tidak berhasil dibuat, fungsi `die()` akan dijalankan. Fungsi ini akan menghentikan jalannya script dan menampilkan pesan yang memberitahukan bahwa telah terjadi kesalahan koneksi. Secara default, notifikasi koneksi MySQL error akan memunculkan pemberitahuan singkat `Connection failed` yang kemudian diikuti dengan pesan yang mendeskripsikan kesalahan (error) tersebut.

Namun di sisi lain, apabila koneksi ke MySQL berhasil dibuat, kode akan menampilkan pesan `Connected successfully` (berhasil terhubung).

Bagian terakhir dari barisan kode di atas adalah `mysqli_close`. Fungsi ini akan menutup koneksi ke database secara manual. Jika tidak ditetapkan sebelumnya, koneksi MySQL akan berhenti sendiri setelah script selesai dijalankan.

1.2.4. Menggunakan PDO untuk Membuat Koneksi PHP ke MySQL

Cara kedua untuk membuat koneksi database PHP ke MySQL adalah dengan PDO. Langkahnya hampir sama dengan MySQLi meskipun ada sedikit perbedaan:

- a. Pada folder `public_html`, buat file yang akan diberi nama `pdoconfig.php`. Masukkan kode di bawah ini ke dalam file tersebut. Ganti value placeholder dengan informasi database yang sudah Anda buat dan catat sebelumnya. Klik `Save and Close` kalau sudah selesai.

```
<?php
    $host = 'localhost';
    $dbname = 'databasename';
    $username = 'username';
    $password = 'password';
```

- b. Pada direktori yang sama, buat file lain dan beri nama `databaseconnect.php`. Tambahkan kode di bawah ini. Jika file sebelumnya diberi nama lain (bukan `pdoconfig.php`), ganti value yang ada di `require_once`.

```
<?php
require_once 'pdoconfig.php';
try {
    $conn = new PDO("mysql:host=$host;dbname=$dbname",
    $username, $password);
    echo "Connected to $dbname at $host successfully.";
} catch (PDOException $pe) {
    die("Could not connect to the database $dbname :". $pe-
    >getMessage());
}
```

1.2.5. Tentang Kode PDO

Koneksi database PDO mengharuskan Anda untuk membuat objek PDO baru yang dilengkapi dengan Data Source Name (DSN), Username, dan Password.

DSN mendeskripsikan tipe database, nama database, dan, jika dibutuhkan, informasi lain yang berkaitan dengan database. Informasi tersebut merujuk pada variabel dan juga value yang telah ditentukan di file `dbconfig.php`, ditandai oleh baris `require_once` yang ada di `databaseconnect.php`.

Pada barisan kode di subbab sebelumnya, di bagian bawah, terdapat kode `try...catch`. Arti dari kode ini adalah script akan mencoba (`try`) membuat koneksi ke MySQL menggunakan kode yang tersedia. Namun, apabila ada masalah, kode di bagian `catch` yang akan dijalankan. Anda bisa menggunakan `catch` block untuk menampilkan pesan koneksi bermasalah atau menjalankan kode alternatif jika `try` block gagal.

Apabila koneksi berhasil dibuat, pesan yang muncul adalah “Connected to \$dbname at \$host successfully.” Namun, jika percobaannya gagal, kode `catch` akan menampilkan pesan error dan memberhentikan script.

1.2.6. Membuat Database MySQL di PHP

PHP menggunakan fungsi `mysql_query` untuk membuat database MySQL, fungsi ini membutuhkan dua parameter dan mengembalikan nilai `TRUE` jika berhasil atau `FALSE` jika gagal (Anonymous, 2015).

```
bool mysql_query( sql, connection );
```

Deskripsi parameter pada skrip diatas:

- a. `sql` – parameter ini diperlukan untuk menempatkan query SQL untuk membuat database

- b. connection – parameter ini bersifat opsional, jika tidak ditentukan, maka akan menggunakan koneksi dengan fungsi mysql_connect yang paling terakhir

Perhatikan contoh skrip membuat database MySQL di PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
echo 'Koneksi Berhasil';
$sql = 'CREATE Database test_db';
$buatdb = mysql_query( $sql, $koneksi );
if(! $buatdb )
{
    die('Pembuatan database, gagal: ' . mysql_error());
}
echo "Database test_db berhasil dibuat\n";
mysql_close($koneksi);
?>
```

1.2.7. Memilih Database MySQL di PHP

Setelah Anda membuat sambungan dengan database server maka langkah selanjutnya yaitu memilih database tertentu, hal ini diperlukan karena mungkin ada beberapa database yang berada pada server induk dan kita dapat melakukan pekerjaan dengan database yang telah dipilih ini pada suatu waktu.

PHP menyediakan fungsi `mysql_select_db` untuk memilih database. Fungsi ini mengembalikan nilai `TRUE` jika berhasil atau `FALSE` jika gagal.

```
bool mysql_select_db( db_name, connection )
```

Deskripsi parameter pada skrip diatas:

- `db_name` – parameter ini diperlukan untuk nama database yang akan dipilih
- `connection` – parameter ini bersifat opsional, jika tidak ditentukan, maka akan menggunakan koneksi dengan fungsi `mysql_connect` yang paling terakhir

Perhatikan contoh skrip memilih database MySQL di PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
echo 'Koneksi Berhasil';
mysql_select_db( 'test_db' );
mysql_close($koneksi );
?>
```

1.2.8. Membuat Tabel di Database MySQL dengan PHP

Untuk membuat tabel dalam database, yang perlu kita lakukan sama seperti kita membuat database. Pertama kita membuat query SQL untuk membuat tabel dan kemudian jalankan query tersebut menggunakan fungsi `mysql_query()`.

Perhatikan contoh skrip membuat tabel di database MySQL dengan PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
echo 'Koneksi Berhasil';
$sql = 'CREATE TABLE karyawan( '
    'id_karyawan INT NOT NULL AUTO_INCREMENT, '
    'nama_karyawan VARCHAR(20) NOT NULL, '
    'alm_karyawan VARCHAR(20) NOT NULL, '
    'gaji_karyawan INT NOT NULL, '
    'tgl_gabung timestamp(14) NOT NULL, '
    'primary key ( id_karyawan ))';
mysql_select_db('test_db');
$buattabel = mysql_query( $sql, $koneksi );
if(! $buattabel )
{
    die('Gagal Membuat Tabel: ' . mysql_error());
}
echo "Tabel Karyawan sukses dibuat\n";
mysql_close($koneksi);
?>
```

Perhatikan isi dari \$sql, pada variabel tersebut berisi query SQL untuk membuat tabel.

1.2.9. Langkah-Langkah Pembuatan CRUD

a. Membuat Database.sql

```
/* Create Database and Table */  
create database crud_db;  
  
use crud_db;  
  
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(100),  
  `email` varchar(100),  
  `mobile` varchar(15),  
  PRIMARY KEY (`id`)  
);
```

Pertama kita buat terlebih dahulu file database.sql kemudian kita impor file database.sql ini ke dalam database (menggunakan PHPMyAdmin).

b. Membuat File Config.php

```
<?php  
/**  
 * using mysqli_connect for database connection  
 */  
  
$databaseHost = 'localhost';  
$databaseName = 'crud_db';  
$databaseUsername = 'root';  
$databasePassword = '';
```

```
$mysqli = mysqli_connect($databaseHost, $databaseUsername,
    $databasePassword, $databaseName);

?>
```

File ****config.php**** menyimpan informasi tentang database host, username dan password. Sebagian besar server lokal bekerja dengan detail yang diberikan. Anda dapat mengubahnya sesuai dengan detail host dan database Anda.

Untuk menghubungkan PHP dengan MySQL, kita menggunakan fungsi `mysqli_connect()` dengan alamat server sebagai parameter pertama, user database sebagai parameter kedua, password user sebagai parameter ketiga, dan nama database sebagai parameter keempat.

Silahkan isi password sesuai dengan yang dibuat di servernya. Jika tidak menggunakan password, maka isikan dengan String kosong saja.

c. Membuat File **Index.php**

```
<?php
// Create database connection using config file
include_once("config.php");

// Fetch all users data from database
$result = mysqli_query($mysqli, "SELECT * FROM users
ORDER BY id DESC");

?>

<html>
<head>
    <title>Homepage</title>
```

```

</head>

<body>
<a href="add.php">Add New User</a><br/><br/>

<table width='80%' border=1>

<tr>
<th>Name</th>    <th>Mobile</th>    <th>Email</th>
<th>Update</th>
</tr>
<?php
while($user_data = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>".$user_data['name']. "</td>";
    echo "<td>".$user_data['mobile']. "</td>";
    echo "<td>".$user_data['email']. "</td>";
    echo "<td><a href='edit.php?id=$user_data[id]>Edit</a> | 
<a href='delete.php?id=$user_data[id]>Delete</a></td></tr>";
    }
?>
</table>
</body>
</html>

```

File index.php merupakan file utama yang menyertakan file konfigurasi untuk koneksi database. Kemudian menampilkan semua daftar pengguna menggunakan MySQL Select Query. Pengguna yang akan ditampilkan di dalam daftar perlu menambahkan terlebih dahulu menggunakan tautan 'Tambahkan Pengguna Baru'.

d. Membuat File add.php

```

<html>
<head>
    <title>Add Users</title>
</head>

<body>
    <a href="index.php">Go to Home</a>
    <br/><br/>

    <form action="add.php" method="post" name="form1">
        <table width="25%" border="0">
            <tr>
                <td>Name</td>
                <td><input
name="name"></td>
                <td><input
type="text"
            </tr>
            <tr>
                <td>Email</td>
                <td><input
name="email"></td>
                <td><input
type="text"
            </tr>
            <tr>
                <td>Mobile</td>
                <td><input
name="mobile"></td>
                <td><input
type="text"
            </tr>
            <tr>
                <td></td>
                <td><input
name="Submit" value="Add"></td>
                <td><input
type="submit"

```

```

        </tr>

    </table>

</form>

<?php

// Check If form submitted, insert form data into users table.
if(isset($_POST['Submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $mobile = $_POST['mobile'];

    // include database connection file
    include_once("config.php");

    // Insert user data into table
    $result = mysqli_query($mysqli, "INSERT INTO
users(name,email,mobile) VALUES('$name','$email','$mobile')");

    // Show message when user added
    echo "User added successfully. <a
href='index.php'>View Users</a>";
}
?>

</body>
</html>

```

File add.php berfungsi untuk menambahkan pengguna baru. Formulir HTML digunakan untuk menerima masukan data pengguna. Setelah data pengguna diserahkan, MySQL INSERT Query digunakan untuk memasukkan data pengguna ke dalam database.

e. Membuat File edit.php

```

<?php
// include database connection file
include_once("config.php");

// Check if form is submitted for user update, then redirect to
homepage after update
if(isset($_POST['update']))
{
    $id = $_POST['id'];

    $name=$_POST['name'];
    $mobile=$_POST['mobile'];
    $email=$_POST['email'];

    // update user data
    $result = mysqli_query($mysqli, "UPDATE users SET
name='$name',email='$email',mobile='$mobile'      WHERE
id=$id");

    // Redirect to homepage to display updated user in list
    header("Location: index.php");
}
?>
<?php
// Display selected user data based on id
// Getting id from url
$id = $_GET['id'];

// Fetch user data based on id

```

```

$result = mysqli_query($mysqli, "SELECT * FROM users
WHERE id=$id");

while($user_data = mysqli_fetch_array($result))
{
    $name = $user_data['name'];
    $email = $user_data['email'];
    $mobile = $user_data['mobile'];
}
?>
<html>
<head>
    <title>Edit User Data</title>
</head>

<body>
    <a href="index.php">Home</a>
    <br/><br/>

    <form        name="update_user"        method="post"
action="edit.php">
        <table border="0">
            <tr>
                <td>Name</td>
                <td><input                type="text"
name="name" value=<?php echo $name;?>></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input                type="text"
name="email" value=<?php echo $email;?>></td>

```



```

        </tr>
        <tr>
            <td>Mobile</td>
            <td><input                type="text"
name="mobile" value=<?php echo $mobile;?>></td>
        </tr>
        <tr>
            <td><input                type="hidden"
name="id" value=<?php echo $_GET['id'];?>></td>
            <td><input                type="submit"
name="update" value="Update"></td>
        </tr>
    </table>
</form>
</body>
</html>

```

Edit.php digunakan untuk mengedit / update data pengguna. Anda dapat mengubah data pengguna dan memperbaruinya. File ini akan mengarahkan pengguna kembali ke homepage, setelah update sukses.

f. Membuat File delete.php

```

<?php
// include database connection file
include_once("config.php");

// Get id from URL to delete that user
$id = $_GET['id'];

// Delete user row from table based on given id

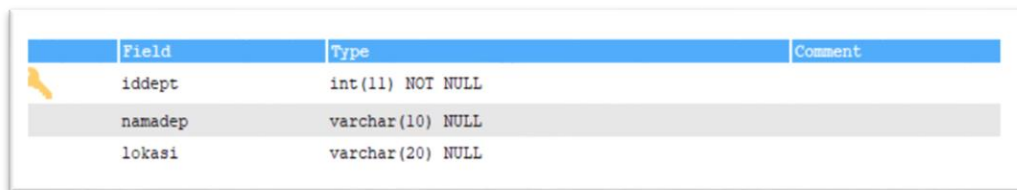
```

```
$result = mysqli_query($mysqli, "DELETE FROM users WHERE  
id=$id");  
  
// After delete redirect to Home, so that latest user list will be  
displayed.  
header("Location:index.php");  
?>
```

File **** delete.php**** hanya dipanggil saat kita klik link ‘Delete’ untuk pengguna manapun yang dipilih.

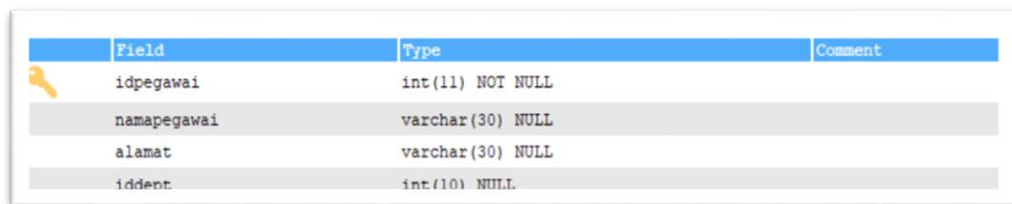
BAB II PEMBAHASAN

Untuk database yang saya gunakan, yaitu pegawai, yang memiliki 2 tabel, dengan nama tabel pegawai dan tabel departemen. Pada tabel pegawai, terdapat field idpegawai sebagai primary keynya, namapegawai, alamat dan iddept, iddept ini merupakan foreign key dari iddept pada tabel departemen. Pada tabel departemen terdapat field iddept sebagai primary keynya, namapegawai, dan lokasi.



Field	Type	Comment
iddept	int(11) NOT NULL	
namadep	varchar(10) NULL	
lokasi	varchar(20) NULL	

Gambar 2.1. Field pada tabel Departemen



Field	Type	Comment
idpegawai	int(11) NOT NULL	
namapegawai	varchar(30) NULL	
alamat	varchar(30) NULL	
iddept	int(10) NULL	

Gambar 2.2. Field pada tabel Pegawai

Pada tugas praktikum pada modul 3, saya membuat 6 file php, yaitu:

- a. function.php
- b. index.php
- c. insert.php
- d. insert.php
- e. insertdept.php
- f. update.php
- g. updatedept.php

Pertama-tama saya akan menjelaskan isi code pada function.php

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

Gambar 2.3. function.php(1)

Untuk membuat koneksi php ke sql, saya menggunakan pengkoneksian dengan MySQLi. Pada barisan kode di atas, di bagian awalnya terdapat deklarasi variabel serta value yang ditentukan untuk variabel ini. Biasanya ada empat variabel untuk pembuatan koneksi database: \$servername, \$database, \$username, dan \$password. Informasi database ditetapkan sebagai value untuk keempat variabel tersebut sehingga dapat dikirimkan ke fungsi. Pada \$servername didefinisikan hostnya (biasanya localhost).

Pada \$database didefinisikan nama database yang akan digunakan yaitu pegawai.

Pada \$username didefinisikan username nya(biasanya root). Pada \$password mendefinisikan password database (biasanya di xampp tidak ada passwordnya, jadi tidak usah di isi).

Untuk menghubungkan PHP dengan MySQL, kita menggunakan fungsi mysqli_connect() dengan alamat server sebagai parameter pertama, user database sebagai parameter kedua, password user sebagai parameter ketiga, dan nama database sebagai parameter keempat. Fungsi ini didefinisikan ke variabel \$conn.

```

9  function show($query){
10     global $conn;
11     $result = mysqli_query($conn, $query);
12     $rows = [];
13     while( $row = mysqli_fetch_assoc($result) ){
14         $rows[] = $row;
15     }
16     return $rows;
17 }

```

Gambar 2.4. function.php(2)

Pada code diatas dibuat fungsi dengan nama show yang memiliki parameter \$query. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian, terdapat fungsi mysqli_query yang digunakan untuk mengirimkan perintah SQL ke server MySQL untuk melakukan aktifitas CRUD. Kemudian pada parameternya terdapat \$conn yang merupakan variabel yang digunakan untuk menyimpan hasil dari eksekusi, dan \$query merupakan perintah SQL untuk melakukan manipulasi data seperti UPDATE, DELETE, INSERT, dan SELECT. Fungsi ini didefinisikan ke \$result. Kemudian, Perulangan while akan melakukan perulangan selama kondisi (syarat) terpenuhi, disini menggunakan mysqli_fetch_assoc, dimana tipe hasil nya adalah array asosiatif dimana setiap nama kolom dan nilai dari satu baris dikaitkan bersama sebagai nama, pasangan nilai. Selama masih ada baris di tabel tersebut maka akan ditampilkan datanya (kondisi while masih terpenuhi), jika tidak ada baris lagi maka while akan berhenti.

```

19 function insert($data){
20     global $conn;
21     $namapegawai = $_POST["namapegawai"];
22     $alamat = $_POST["alamat"];
23     $iddept = $_POST["iddept"];
24
25     mysqli_query($conn, "INSERT INTO pegawai VALUES('','$namapegawai','$alamat','$iddept')");
26     return mysqli_affected_rows($conn);
27 }

```

Gambar 2.5. function.php(3)

Pada code diatas dibuat fungsi dengan nama insert yang memiliki parameter \$data. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat

method post untuk menampung data pada namapegawai, didefinisikan kedalam variabel \$namapegawai, method post untuk menampung data pada alamat, didefinisikan kedalam variabel \$alamat, method post untuk menampung data pada iddept, didefinisikan kedalam variabel \$iddept. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk menginputkan data kevariabel \$namapegawai, \$alamat, \$iddept. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan untuk mengembalikan jumlah baris setelah dibuat query insert sebelumnya. Fungsi ini akan digunakan nantinya untuk insert data pada tabel pegawai.

```

28 function insert1($data2){
29     global $conn;
30     $namadep = $_POST["namadep"];
31     $lokasi = $_POST["lokasi"];
32
33     mysqli_query($conn, "INSERT INTO departemen VALUES('','$namadep','$lokasi')");
34     return mysqli_affected_rows($conn);
35 }

```

Gambar 2.6. function.php(4)

Pada code diatas dibuat fungsi dengan nama insert1 yang memiliki parameter \$data2. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat method post untuk menampung data pada namadep, didefinisikan kedalam variabel \$namadep, method post untuk menampung data pada lokasi, didefinisikan kedalam variabel \$lokasi. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk menginputkan data kevariabel \$namadep, \$lokasi. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan untuk mengembalikan jumlah baris setelah dibuat query insert sebelumnya. Fungsi ini hampir sama seperti sebelumnya, yang dimana fungsi ini akan digunakan nantinya untuk insert data pada tabel departemen.

```

37 function delete($data){
38     global $conn;
39     $idpegawai = $_GET["delete"];
40     mysqli_query($conn, "DELETE FROM pegawai WHERE idpegawai = $idpegawai");
41     return mysqli_affected_rows($conn);
42 }

```

Gambar 2.7. function.php(5)

Pada code diatas dibuat fungsi yang diberi nama delete dengan parameter \$data. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat method get menampung data delete, didefinisikan ke variabel \$idpegawai. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk menghapus data dengan kondisi idpgawai=\$idpegawai. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan untuk mengembalikan jumlah baris setelah dibuat query delete sebelumnya. Fungsi ini akan digunakan nantinya untuk delete data pada tabel pegawai.

```

function delete1($data2){
    global $conn;
    $iddept = $_GET["delete1"];
    mysqli_query($conn, "DELETE FROM departemen WHERE iddept = $iddept");
    return mysqli_affected_rows($conn);
}

```

Gambar 2.8. function.php(6)

Pada code diatas dibuat fungsi yang diberi nama delete1 dengan parameter \$data2. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat method get menampung data delete, didefinisikan ke variabel \$iddept. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk menghapus data dengan kondisi iddept=\$iddept. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan

untuk mengembalikan jumlah baris setelah dibuat query delete sebelumnya. Fungsi ini akan digunakan nantinya untuk delete data pada tabel departemen.

```

52  function update($data){
53      global $conn;
54      $idpegawai = $_POST["idpegawai"];
55      $namapegawai = $_POST["namapegawai"];
56      $alamat = $_POST["alamat"];
57      $iddept = $_POST["iddept"];
58      mysqli_query($conn, "UPDATE pegawai SET namapegawai = '$namapegawai',
59      alamat = '$alamat', iddept='$iddept' WHERE idpegawai = '$idpegawai'");
60      return mysqli_affected_rows($conn);
61  }

```

Gambar 2.9. function.php(7)

Pada code diatas dibuat fungsi yang diberi nama update dengan parameter \$data. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat method post untuk menampung data pada idpegawai, didefinisikan kedalam variabel \$idpegawai, method post untuk menampung data pada namapegawai, didefinisikan kedalam variabel \$namapegawai, method post untuk menampung data pada alamat, didefinisikan kedalam variabel \$alamat, method post untuk menampung data pada iddept, didefinisikan kedalam variabel \$iddept. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk update data pada tabel pegawai, namapegawai akan diset menjadi sesuai dengan inputan pada form pada variabel \$namapegawai nantinya, alamat akan diset menjadi sesuai dengan inputan pada form pada variabel \$alamat nantinya, iddept akan diset menjadi sesuai dengan inputan pada form pada variabel \$iddept nantinya, dengan kondisi idpegawai=\$idpegawai. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan untuk mengembalikan jumlah baris setelah dibuat query update sebelumnya.


```

63  function update1($data2){
64      global $conn;
65      $iddept = $_POST["iddept"];
66      $namadep = $_POST["namadep"];
67      $lokasi = $_POST["lokasi"];
68      mysqli_query($conn, "UPDATE departemen SET namadep = '$namadep',
69      lokasi = '$lokasi' WHERE iddept = '$iddept'");
70      return mysqli_affected_rows($conn);
71  }
72
73
74
75  ?>

```

Gambar 2.10. function.php(8)

Pada code diatas dibuat fungsi yang diberi nama update1 dengan parameter \$data2. Kemudian, karena kita ingin menggunakan variabel \$conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian terdapat method post untuk menampung data pada iddept, didefinisikan kedalam variabel \$iddept, method post untuk menampung data pada namadep, didefinisikan kedalam variabel \$namadep, method post untuk menampung data pada lokasi, didefinisikan kedalam variabel \$lokasi. Kemudian, terdapat fungsi mysqli_query yang memiliki parameter \$conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli_connect sebelumnya, kemudian dibuat query untuk update data pada tabel departemen, namadep akan diset menjadi sesuai dengan inputan pada form pada variabel \$namadep nantinya, lokasi akan diset menjadi sesuai dengan inputan pada form pada variabel \$lokasi nantinya, dengan kondisi iddept=\$iddept. Kemudian, direturn fungsi mysqli_affected_rows dengan parameter \$conn, yang digunakan untuk mengembalikan jumlah baris setelah dibuat query update sebelumnya.

```

1  <?php
2  require("function.php");
3  $data = show("SELECT * FROM pegawai");
4  $data2 = show("SELECT * FROM departemen");
5  if (isset($_GET["delete"])) {
6      if (delete($_GET > 0)) {
7          echo "<script>
8              alert('data berhasil dihapus');
9              document.location.href = 'index.php';
10             </script>";
11      } else {
12          echo "<script>
13              alert('data gagal dihapus');
14              document.location.href = 'index.php';
15             </script>";
16      }
17  }
18  if (isset($_GET["delete1"])) {
19      if (delete1($_GET > 0)) {
20          echo "<script>
21              alert('data berhasil dihapus');
22              document.location.href = 'index.php';
23             </script>";
24      } else {
25          echo "<script>
26              alert('data gagal dihapus');
27              document.location.href = 'index.php';
28             </script>";
29      }
30  }
31  ?>

```

Gambar 2.11. index.php(1)

Pada code diatas pertama-tama dipanggil function.php kedalam code menggunakan fungsi require. Kemudian, dipanggil fungsi yang telah dibuat pada function.php yaitu show, yang pada parameternya terdapat query untuk menampilkan semua data pada tabel pegawai, didefinisikan ke variabel \$data. Kemudian, dipanggil juga fungsi show yang pada parameternya terdapat query untuk menampilkan semua data pada tabel departemen. Kemudian, terdapat validasi pada variabel \$_get(["delete"]) menggunakan fungsi isset, apakah tersedia untuk diproses atau tidak. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi delete yang telah dibuat pada function.php, dimana jika data pada pegawai lebih dari 0, maka akan ditampilkan pop up dari

javascript dalam bentuk alert yaitu data berhasil dihapus. Tetapi jika tidak, maka akan ditampilkan alert data gagal dihapus.

Kemudian, terdapat validasi pada variabel `$_get(["delete1"])` menggunakan fungsi `isset`, apakah tersedia untuk diproses atau tidak. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi `delete1` yang telah dibuat pada `function.php`, dimana jika data pada departemen lebih dari 0, maka akan ditampilkan pop up dari javascript dalam bentuk alert yaitu data berhasil dihapus. Tetapi jika tidak, maka akan ditampilkan alert data gagal dihapus.

```

32 <!DOCTYPE html>
33 <html lang="en">
34 <head>
35     <meta charset="UTF-8">
36     <meta name="viewport" content="width=device-width, initial-scale=1.0">
37     <title>Modul3</title>
38 </head>
39 <body>
40     <form action="insert.php">
41         <button>INSERT</button>
42     </form>
43     <table border="1">
44         <tr>
45             <td>id pegawai</td>
46             <td>nama pegawai</td>
47             <td>alamat</td>
48             <td>id dept</td>
49             <td>aksi</td>
50         </tr>
51     </table>
52 </body>
53 </html>
54 <?php

```

Gambar 2.12. index.php(2)

Karena pada code-code berikutnya terdapat tag-tag HTML, alangkah baiknya dibuat penjelasan masing-masing dari tag HTML yang mungkin nantinya akan digunakan, agar lebih mudah dalam menjelaskan code yang dibuat.

- `<!DOCTYPE>` Tag untuk menentukan tipe dokumen
- `<title>` Tag untuk membuat judul dari sebuah halaman
- `<body>` Tag untuk membuat tubuh dari sebuah halaman
- `<h1>` to `<h6>` Tag untuk membuat heading
- `
` Memasukan satu baris putus
- `<html>` Tag untuk membuat sebuah dokumen HTML

- g. <head>Tag untuk membuat informasi tentang dokumen
- h. <meta>Tag untuk membuat metadata tentang dokumen HTML
- i. <tr>Tag untuk membuat baris dalam sebuah tabel
- j. <th>Tag untuk membuat sebuah sel header tabel
- k. <td>Tag untuk membuat sel dalam sebuah tabel
- l. <form>Tag untuk membuat sebuah form HTML untuk input pengguna
- m. <input>Tag untuk membuat sebuah kontrol input
- n. <button>Tag untuk membuat sebuah tombol yang dapat diklik
- o. <link> Tag untuk membuat hubungan antara dokumen dan sumber daya eksternal (paling sering digunakan untuk link ke style sheet)
- p. Tag untuk membuat daftar dengan selain nomor
- q. Tag untuk membuat sebuah item daftar
- r. <table>Tag untuk membuat tabel

Masuk kedalam penjelasan code pada gambar 2.10. didalam tag <head> terdapat form, yang dimana aksi yang akan dilakukan pada form ini akan menggunakan insert.php. kemudian terdapat button dengan nama INSERT. Jadi, ketika button ini diklik, maka aksi pada insert.php akan dilakukan. Kemudian, dibuat tabel, yang dimana didalam tabel terdapat kolom idpegawai, namapegawai, alamat, dan id dept.

```

52     foreach ($data as $d) {
53     ?>
54         <tr>
55             <td><?php echo $d["idpegawai"]; ?></td>
56             <td><?php echo $d["namapegawai"]; ?></td>
57             <td><?php echo $d["alamat"]; ?></td>
58             <td><?php echo $d["iddept"]; ?></td>
59             <td>
60                 <form action="" method="GET">
61                     <button type="submit" name="delete" value=<?php echo $d["idpegawai"]; ?>>DELETE</button>
62                 </form>
63                 <form action="update.php" method="GET">
64                     <button type="submit" name="update" value=<?php echo $d["idpegawai"]; ?>>UPDATE</button>
65                 </form>
66             </td>
67         </tr>
68     <?php
69     }
70     ?>

```

Gambar 2.13. index.php(3)

Pada code diatas dibuat perulangan foreach, yang dimana akan menampilkan array pada \$data dan variabel perantaranya yaitu \$d. Yang akan ditampilkan yaitu data pada array \$d yaitu idpegawai pada kolom pertama, namapegawai pada kolom selanjutnya , alamat pada kolom selanjutnya, dan iddept pada kolom selanjutnya. Kemudian, terdapat form yang aksinya kosong yang berarti aksinya ada didalam index.php, dan metode pengiriman data menggunakan metode get. Kemudian, terdapat button dalam bentuk submit, dengan nama delete yang dimasukan kedalam kolom aksi, yang dimana delete ini akan dijalankan menurut idpegawainya. Jadi fungsi delete akan dipanggil ketika button delete diklik.

Kemudian terdapat juga form yang aksinya diambil dari update.php, dan metode pengiriman datanya menggunakan metode get. Kemudian, terdapat button dalam bentuk submit, dengan nama update yang dimasukan kedalam kolom aksi, yang dimana update ini akan dijalankan menurut idpegawainya. Jadi, fungsi update akan dipanggil ketika button update diklik.

```

71      </table>
72      <br></br>
73      <form action="insertdept.php">
74          <button>INSERT</button>
75      </form>
76      <table border="1">
77          <tr>
78              <td>iddept</td>
79              <td>namadept</td>
80              <td>lokasi</td>
81              <td>aksi</td>
82          </tr>
83          <?php

```

Gambar 2.14. index.php(4)

Pada code diatas terdapat form, yang dimana aksi yang akan dilakukan pada form ini akan menggunakan insertdept.php. Kemudian terdapat button dengan nama INSERT. Jadi, ketika button ini diklik, maka aksi pada insertdept.php akan

dilakukan. Kemudian, dibuat tabel, yang dimana didalam tabel terdapat kolom iddept, namadept, lokasi, dan aksi.

```

84         foreach ($data2 as $d) {
85             <?php
86             <tr>
87                 <td><?php echo $d["iddept"]; ?></td>
88                 <td><?php echo $d["namadept"]; ?></td>
89                 <td><?php echo $d["lokasi"]; ?></td>
90                 <td>
91                     <form action="" method="GET">
92                         <button type="submit" name="delete1" value=<?php echo $d["iddept"]; ?>>DELETE</button>
93                     </form>
94                     <form action="updatedept.php" method="GET">
95                         <button type="submit" name="update1" value=<?php echo $d["iddept"]; ?>>UPDATE</button>
96                     </form>
97                 </td>
98             </tr>
99             <?php
100         }
101     <?php
102 </table>
103 </body>
104 </html>

```

Gambar 2.15. index.php(5)

Pada code diatas dibuat perulangan foreach, yang dimana akan menampilkan array pada \$data2 dan variabel perantaranya yaitu \$d. Yang akan ditampilkan yaitu data pada array \$d yaitu iddept pada kolom pertama, namadept pada kolom selanjutnya, lokasi pada kolom selanjutnya. Kemudian, terdapat form yang aksinya kosong yang berarti aksinya ada didalam index.php, dan metode pengiriman data menggunakan metode get. Kemudian, terdapat button dalam bentuk submit, dengan nama delete1 yang dimasukan kedalam kolom aksi, yang dimana delete ini akan dijalankan menurut iddeptnya. Jadi fungsi delete1 akan dipanggil ketika button delete diklik.

Kemudian terdapat juga form yang aksinya diambil dari updatedept.php, dan metode pengiriman datanya menggunakan metode get. Kemudian, terdapat button dalam bentuk submit, dengan nama update yang dimasukan kedalam kolom aksi, yang dimana update ini akan dijalankan menurut iddeptnya. Jadi, fungsi update1 akan dipanggil ketika button update diklik.

```
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 2.16. insert.php(1)

Pada code diatas, dimulai dengan pembuka program php. Kemudian, dipanggil function.php kedalam code menggunakan fungsi require. Kemudian, terdapat validasi pada variabel `$_get(["insert"])` menggunakan fungsi `isset`, apakah tersedia untuk diproses atau tidak. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi `insert` yang telah dibuat pada `function.php`, dimana jika data pada pegawai lebih dari 0, maka akan ditampilkan pop up dari javascript dalam bentuk alert yaitu data berhasil ditambah. Tetapi jika tidak, maka akan ditampilkan alert data gagal ditambah.

```

17 <!DOCTYPE html>
18 <html lang="en">
19
20 <head>
21   <meta charset="UTF-8">
22   <meta name="viewport" content="width=device-width, initial-scale=1.0">
23   <title>Insert</title>
24 </head>
25
26 <body>
27   <a href="index.php">home</a>
28   <form action="" method="POST">
29     <ul>
30       <li>
31         <label for="namapegawai">namapegawai</label>
32         <input type="text" name="namapegawai" id="namapegawai">
33       </li>
34       <li>
35         <label for="alamat">alamat</label>
36         <input type="text" name="alamat" id="alamat">
37       </li>
38       <li>
39         <label for="iddept">iddept</label>
40         <input type="text" name="iddept" id="iddept">
41       </li>
42       <li>
43         <button type="submit" name="insert">insert</button>
44       </li>
45     </ul>
46   </form>
47 </body>
48
49 </html>
50

```

Gambar 2.17. insert.php(2)

Pada awal code diatas, dibuat sintaks untuk HTML. Masuk ke dalam tag <body>, didalam tag ini terdapat tag <a> untuk membuat link yang dimana alamat yang dituju yaitu pada index.php, dengan nama link nya yaitu home. Kemudian terdapat form, yang dimana aksi yang akan dilakukan pada form ini di isi kosong, yang berarti aksi nya berada pada insert.php itu sendiri, dan method pengiriman data nya menggunakan method POST. Kemudian dibuat Unordered list yang dimana unordered list adalah list yang tak terurut yang menggunakan simbol-simbol pada item-nya. Unordered list dibuat dengan tag dan untuk item-nya dibuat juga dengan tag . Kemudian, dibuat label untuk menginput pada

namapegawai, alamat, dan iddept. Kemudian terdapat tombol submit yang diberi nama insert. Jadi, hasil penginputan pada insert.php ini akan dipost ke fungsi pada insert yang dibuat pada function.php.

```
1  <?php
2  require("function.php");
3  if (isset($_POST["insert1"])) {
4      if (insert1($_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 2.18. insertdept.php(1)

Pada code diatas, dimulai dengan pembuka program php. Kemudian, dipanggil function.php kedalam code menggunakan fungsi require. Kemudian, terdapat validasi pada variabel `$_get(["insert1"])` menggunakan fungsi `isset`, apakah tersedia untuk diproses atau tidak. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi insert yang telah dibuat pada function.php, dimana jika data pada departemen lebih dari 0, maka akan ditampilkan pop up dari javascript dalam bentuk alert yaitu data berhasil ditambah. Tetapi jika tidak, maka akan ditampilkan alert data gagal dihapus.

```

17 <!DOCTYPE html>
18 <html lang="en">
19
20 <head>
21   <meta charset="UTF-8">
22   <meta name="viewport" content="width=device-width, initial-scale=1.0">
23   <title>Insert</title>
24 </head>
25
26 <body>
27   <a href="index.php">home</a>
28   <form action="" method="POST">
29     <ul>
30
31     <li>
32       <label for="namadep">namadep</label>
33       <input type="text" name="namadep" id="namadep">
34     </li>
35     <li>
36       <label for="lokasi">lokasi</label>
37       <input type="text" name="lokasi" id="lokasi">
38     </li>
39     <li>
40       <button type="submit" name="insert1">insert</button>
41     </li>
42   </ul>
43 </form>
44 </body>
45
46 </html>

```

Gambar 2.19. insertdept.php(2)

Pada awal code diatas, dibuat sintaks untuk HTML. Masuk ke dalam tag <body>, didalam tag ini terdapat tag <a> untuk membuat link yang dimana alamat yang dituju yaitu pada index.php, dengan nama link nya yaitu home. Kemudian terdapat form, yang dimana aksi yang akan dilakukan pada form ini di isi kosong, yang berarti aksi nya berada pada insert.php itu sendiri, dan method pengiriman data nya menggunakan method POST. Kemudian dibuat Unordered list yang dimana unordered list adalah list yang tak terurut yang menggunakan simbol-simbol pada item-nya. Unordered list dibuat dengan tag dan untuk item-nya dibuat juga dengan tag . Kemudian, dibuat label untuk menginput pada namadep dan lokasi. Kemudian terdapat tombol submit dengan nama insert1. Jadi, hasil penginputan pada insertdept.php ini akan dipost ke fungsi pada insert1 yang dibuat pada function.php.

```

1  <?php
2  require("function.php");
3  $idpegawai = $_GET["update"];
4  $data = show("SELECT * FROM pegawai WHERE idpegawai = $idpegawai");
5  if (isset($_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  }
18
19  ?>

```

Gambar 2.20. update.php(1)

Pada code diatas, dimulai dengan pembuka program php. Kemudian, dipanggil function.php kedalam code menggunakan fungsi require. Kemudian, terdapat validasi pada variabel `$_get(["update"])` menggunakan fungsi `isset`, apakah tersedia untuk diproses atau tidak. Kemudian, terdapat fungsi `show` yang memiliki query untuk menampilkan semua data ketika `idpegawai=$idpegawai`, didefinisikan ke variabel `$data`. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi `update` yang telah dibuat pada `function.php`, dimana jika data pada departemen lebih dari 0, maka akan ditampilkan pop up dari javascript dalam bentuk alert yaitu data berhasil diupdate. Tetapi jika tidak, maka akan ditampilkan alert data gagal diupdate.

```

21 <!DOCTYPE html>
22 <html lang="en">
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Update</title>
27 </head>
28 <body>
29     <a href="index.php">home</a>
30     <form action="" method="POST">
31         <?php
32         foreach ($data as $d) {
33             ?>
34             <ul>
35                 <li>
36                     <label for="idpegawai">idpegawai</label>
37                     <input type="text" name="idpegawai" id="idpegawai" value=<?php echo $d["idpegawai"]; ?>>
38                 </li>
39                 <li>
40                     <label for="namapegawai">namapegawai</label>
41                     <input type="text" name="namapegawai" id="namapegawai" value=<?php echo $d["namapegawai"]; ?>>
42                 </li>
43                 <li>
44                     <label for="alamat">alamat</label>
45                     <input type="text" name="alamat" id="alamat" value=<?php echo $d["alamat"]; ?>>
46                 </li>
47                 <li>
48                     <label for="iddept">iddept</label>
49                     <input type="text" name="iddept" id="iddept" value=<?php echo $d["iddept"]; ?>>
50                 </li>
51                 <li>
52                     <button type="submit" name="ubah">update</button>
53                 </li>
54             </ul>
55             <?php
56             ?>
57         </form>
58 </body>
59 </html>

```

Gambar 2.21. update.php(2)

Pada awal code diatas, dibuat sintaks untuk HTML. Masuk ke dalam tag <body>, didalam tag ini terdapat tag <a> untuk membuat link yang dimana alamat yang dituju yaitu pada index.php, dengan nama link nya yaitu home. Kemudian terdapat form, yang dimana aksi yang akan dilakukan pada form ini di isi kosong, yang berarti aksi nya berada pada update.php itu sendiri, dan method pengiriman data nya menggunakan method POST. Kemudian dibuat pembuka program php, yang didalamnya terdapat perulangan foreach dengan parameter \$data sebagai \$d. Didalam perulangan ini akan, menampilkan idpegawai, namapegawai, alamat, dan iddept, sesuai dengan idpegawai yang dipilih sebelumnya pada home. Kemudian, hasil data tersebut dapat diganti atau diupdate, dan proses update akan berlangsung ketika kita mengklik tombol submit dengan nama update.

```

1  <?php
2  require("function.php");
3  $iddept = $_GET["update1"];
4  $data2 = show("SELECT * FROM departemen WHERE iddept = $iddept");
5  if (isset($_POST["ubah1"])) {
6      if (update1($_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  }
18
19  ?>

```

Gambar 2.22. updatedept.php(1)

Pada code diatas, dimulai dengan pembuka program php. Kemudian, dipanggil function.php kedalam code menggunakan fungsi require. Kemudian, terdapat validasi pada variabel \$_get(["update"]) menggunakan fungsi isset, apakah tersedia untuk diproses atau tidak. Kemudian, terdapat fungsi show yang memiliki query untuk menampilkan semua data ketika iddept=\$iddept, didefinisikan ke \$data2. Jika tersedia, kemudian terdapat pemeriksaan kondisi, yang didalamnya terdapat fungsi update yang telah dibuat pada function.php, dimana jika data pada departemen lebih dari 0, maka akan ditampilkan pop up dari javascript dalam bentuk alert yaitu data berhasil diupdate. Tetapi jika tidak, maka akan ditampilkan alert data gagal diupdate.

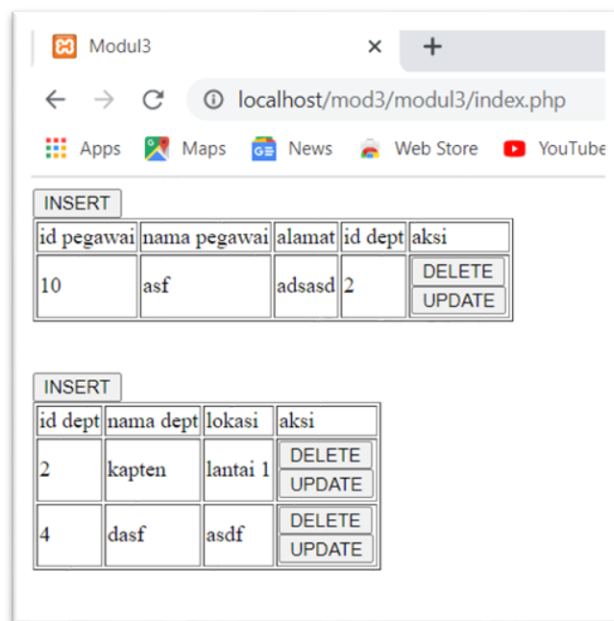
```

21 <!DOCTYPE html>
22 <html lang="en">
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Update</title>
27 </head>
28 <body>
29     <a href="index.php">home</a>
30     <form action="" method="POST">
31         <?php
32         foreach ($data2 as $d) {
33             <?
34             <ul>
35                 <li>
36                     <label for="iddept">iddept</label>
37                     <input type="text" name="iddept" id="iddept" value=<?php echo $d["iddept"]; ?>>
38                 </li>
39                 <li>
40                     <label for="namadept">namadept</label>
41                     <input type="text" name="namadept" id="namadept" value=<?php echo $d["namadept"]; ?>>
42                 </li>
43                 <li>
44                     <label for="lokasi">lokasi</label>
45                     <input type="text" name="lokasi" id="lokasi" value=<?php echo $d["lokasi"]; ?>>
46                 </li>
47                 <li>
48                     <button type="submit" name="ubah1">update</button>
49                 </li>
50             </ul>
51             <?php}
52             <?>
53         </form>
54 </body>
55 </html>

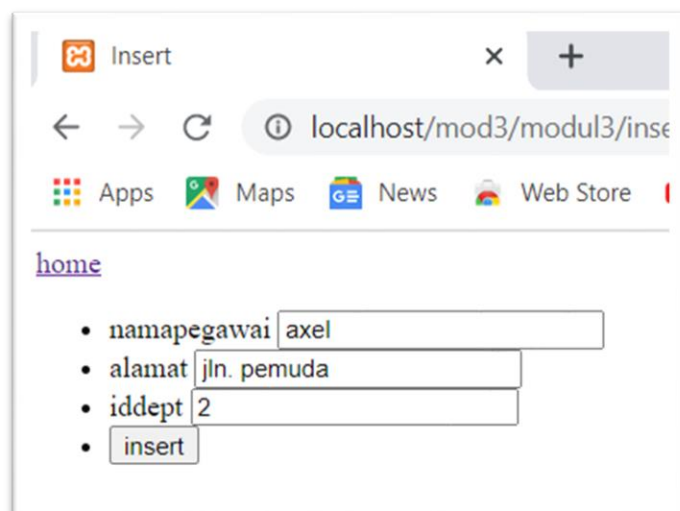
```

Gambar 2.23. updatedept.php (2)

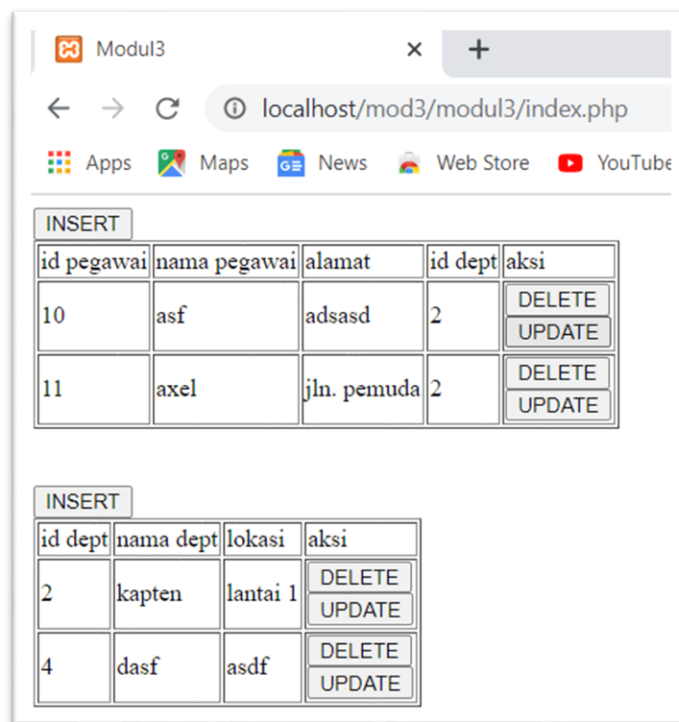
Pada awal code diatas, dibuat sintaks untuk HTML. Masuk ke dalam tag <body>, didalam tag ini terdapat tag <a> untuk membuat link yang dimana alamat yang dituju yaitu pada index.php, dengan nama link nya yaitu home. Kemudian terdapat form, yang dimana aksi yang akan dilakukan pada form ini di isi kosong, yang berarti aksi nya berada pada updatedept.php itu sendiri, dan method pengiriman data nya menggunakan method POST. Kemudian dibuat pembuka program php, yang didalamnya terdapat perulangan foreach dengan parameter \$data sebagai \$d. Didalam perulangan ini akan, menampilkan idpegawai, namapegawai, alamat, dan iddept, sesuai dengan idpegawai yang dipilih sebelumnya pada home. Kemudian, hasil data tersebut dapat diganti atau diupdate, dan proses update akan berlangsung ketika kita mengklik tombol submit dengan nama update.



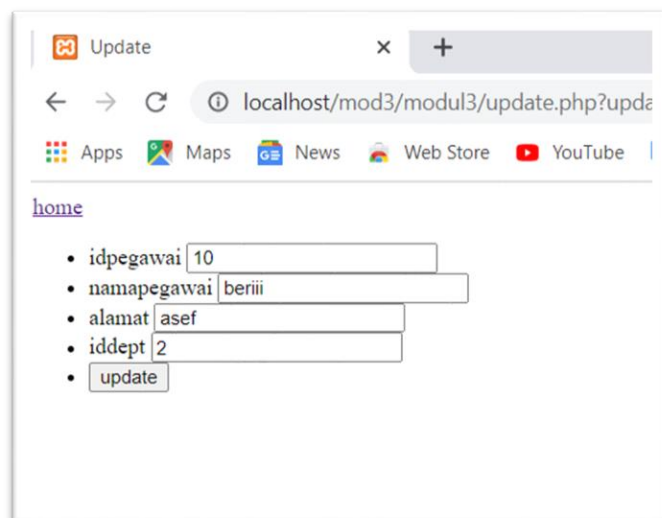
Gambar 2.24 Tampilan awal web



Gambar 2.25 Insert data pegawai



Gambar 2.26. Hasil insert data pegawai



Gambar 2.27. Update data pegawai dengan id pegawai 10

INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
10	beriii	asef	2	DELETE UPDATE
11	axel	jln. pemuda	2	DELETE UPDATE

INSERT

id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE

Gambar 2.28. Hasil update data pegawai

Modul3

localhost/mod3/modul3/index.php

Apps Maps News Web Store YouTube

INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
11	axel	jln. pemuda	2	DELETE UPDATE

INSERT

id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE

Gambar 2.29. Hasil Delete data pegawai dengan id pegawai 10

Insert

localhost/mod3/modul3/i

Apps Maps News Web Store

[home](#)

- namadep
- lokasi
-

Gambar 2.30. Insert data departemen

INSERT			
id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE
5	administra	kalimantan tengah	DELETE UPDATE

Gambar 2.31. Hasil insert data departemen

Update

localhost/mod3/modul3/

Apps Maps News Web Store

[home](#)

- iddept
- namadep
- lokasi
-

Gambar 2.32. Update data departemen dengan iddept 2

Modul3

localhost/mod3/modul3/index.php

Apps Maps News Web Store YouTube

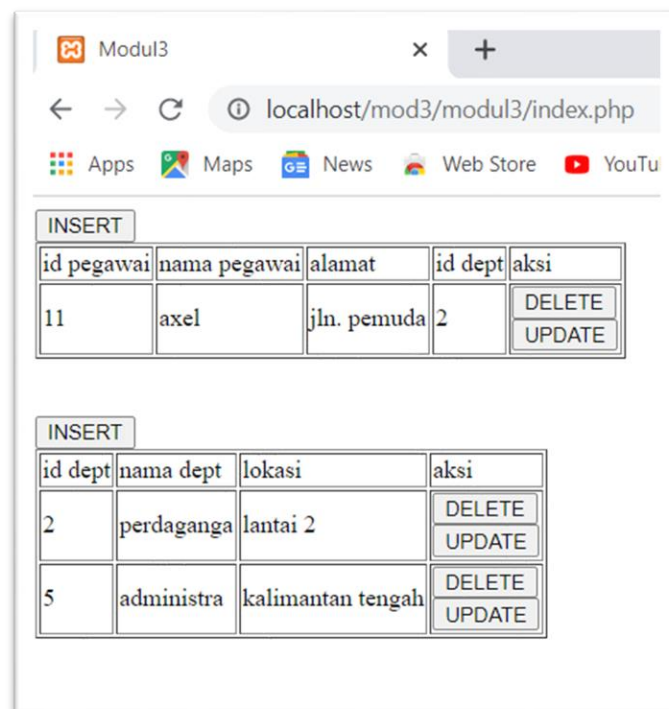
INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
11	axel	jln. pemuda	2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

INSERT

id dept	nama dept	lokasi	aksi
2	perdaganga	lantai 2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
4	dasf	asdf	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
5	administra	kalimantan tengah	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.33. Hasil update data departemen



Gambar 2.34. Hasil delete data pegawai dengan iddept 4

BAB III

KESIMPULAN


Dari praktikum yang telah dilaksanakan dapat disimpulkan bahwa, Ada dua cara untuk membuat koneksi ke database MySQL dengan PHP, yaitu menggunakan MySQLi dan PDO. MySQLi merupakan singkatan dari MySQL Improved. Ekstensi MySQL ini menambahkan fitur baru di interface database MySQL. MySQLi bersifat prosedural dan juga berorientasi objek, di mana sifat prosedural adalah atribut yang diwariskan dari versi MySQL yang lebih lama. PDO adalah singkatan dari PHP Data Object. Lain halnya dengan MySQLi, PDO justru berorientasi hanya pada objek dan mendukung sejumlah tipe database yang menggunakan PHP, seperti MySQL, MSSQL, Informix, dan PostgreSQL.

Dalam praktikum kali ini, praktikan juga membuat CRUD dalam php. CRUD adalah akronim untuk Create, Read, Update, dan Delete. Operasi CRUD adalah manipulasi data dasar untuk database.


DAFTAR PUSTAKA

- Achmad Solichin. S.Kom. (2016). Pemrograman Web dengan PHP dan MySQL - Achmad Solichin - Google Buku. *Universitas Budi Luhur, June*, 85. <https://books.google.co.id/books?id=kcD4BQAAQBAJ&printsec=frontcover&dq=aplikasi+berbasis+web+dengan&hl=id&sa=X&ved=0ahUKEwib-fT80ITYAhVBrI8KHT9GD6QQ6AEIJzAA#v=onepage&q=aplikasi+berbasis+web+dengan&f=false>
- Informatika, D. T. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.
- Anonymous. (2015). *Cara Membuat Tabel dan Database MySQL di PHP*. Diakses pada tanggal 10 April 2021 di: <http://www.nyekrip.com/skrip/cara-membuat-tabel-database-mysql-php/>
- Ariata. (2020). *Cara Membuat Koneksi PHP ke Database MySQL*. Diakses pada tanggal 10 April 2021 di: <https://www.hostinger.co.id/tutorial/koneksi-database-php#Membuat-Database-MySQL-Optional>
- Yuni97. (2019). *Tutorial Membuat CRUD PHP Dengan MySQL*. Diakses pada tanggal 10 April 2021 di: codepolitan.com/tutorial-membuat-crud-php-dengan-mysql-59897c72d8470

LAMPIRAN

	Field	Type	Comment
	iddept	int(11) NOT NULL	
	namadep	varchar(10) NULL	
	lokasi	varchar(20) NULL	

Gambar 2.1. Field pada tabel Departemen

	Field	Type	Comment
	idpegawai	int(11) NOT NULL	
	namapegawai	varchar(30) NULL	
	alamat	varchar(30) NULL	
	iddent	int(10) NULL	

Gambar 2.2. Field pada tabel Pegawai

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

Gambar 2.3. function.php(1)

```
9  ▼ function show($query){
10     global $conn;
11     $result = mysqli_query($conn, $query);
12     $rows = [];
13     ▼ while( $row = mysqli_fetch_assoc($result) ){
14         $rows[] = $row;
15     }
16     return $rows;
17 }
```

Gambar 2.4. function.php(2)

```

19 function insert($data){
20     global $conn;
21     $namapegawai = $_POST["namapegawai"];
22     $alamat = $_POST["alamat"];
23     $iddept = $_POST["iddept"];
24
25     mysqli_query($conn, "INSERT INTO pegawai VALUES('','$namapegawai','$alamat','$iddept')");
26     return mysqli_affected_rows($conn);
27 }

```

Gambar 2.5. function.php(3)

```

28 function insert1($data2){
29     global $conn;
30     $namadep = $_POST["namadep"];
31     $lokasi = $_POST["lokasi"];
32
33     mysqli_query($conn, "INSERT INTO departemen VALUES('','$namadep','$lokasi')");
34     return mysqli_affected_rows($conn);
35 }

```

Gambar 2.6. function.php(4)

```

37 function delete($data){
38     global $conn;
39     $idpegawai = $_GET["delete"];
40     mysqli_query($conn, "DELETE FROM pegawai WHERE idpegawai = $idpegawai");
41     return mysqli_affected_rows($conn);
42 }

```

Gambar 2.7. function.php(5)

```

✓ function delete1($data2){
    global $conn;
    $iddept = $_GET["delete1"];
    mysqli_query($conn, "DELETE FROM departemen WHERE iddept = $iddept");
    return mysqli_affected_rows($conn);
}

```

Gambar 2.68. function.php(6)

```

52 ✓ function update($data){
53     global $conn;
54     $idpegawai = $_POST["idpegawai"];
55     $namapegawai = $_POST["namapegawai"];
56     $alamat = $_POST["alamat"];
57     $iddept = $_POST["iddept"];
58     mysqli_query($conn, "UPDATE pegawai SET namapegawai = '$namapegawai',
59     alamat = '$alamat', iddept='$iddept' WHERE idpegawai = '$idpegawai'");
60     return mysqli_affected_rows($conn);
61 }

```

Gambar 2.9. function.php(7)


```

63 function update1($data2){
64     global $conn;
65     $iddept = $_POST["iddept"];
66     $namadep = $_POST["namadep"];
67     $lokasi = $_POST["lokasi"];
68     mysqli_query($conn, "UPDATE departemen SET namadep = '$namadep',
69     lokasi = '$lokasi' WHERE iddept = '$iddept'");
70     return mysqli_affected_rows($conn);
71 }
72
73
74
75 ?>

```

Gambar 2.10. function.php(7)

```

1  <?php
2  require("function.php");
3  $data = show("SELECT * FROM pegawai");
4  $data2 = show("SELECT * FROM departemen");
5  if (isset($_GET["delete"])) {
6      if (delete($_GET > 0)) {
7          echo "<script>
8              alert('data berhasil dihapus');
9              document.location.href = 'index.php';
10             </script>";
11      } else {
12          echo "<script>
13              alert('data gagal dihapus');
14              document.location.href = 'index.php';
15             </script>";
16      }
17  }
18  if (isset($_GET["delete1"])) {
19      if (delete1($_GET > 0)) {
20          echo "<script>
21              alert('data berhasil dihapus');
22              document.location.href = 'index.php';
23             </script>";
24      } else {
25          echo "<script>
26              alert('data gagal dihapus');
27              document.location.href = 'index.php';
28             </script>";
29      }
30  }
31  ?>

```

Gambar 2.11. index.php(1)

```

32 <!DOCTYPE html>
33 <html lang="en">
34 <head>
35     <meta charset="UTF-8">
36     <meta name="viewport" content="width=device-width, initial-scale=1.0">
37     <title>Modul3</title>
38 </head>
39 <body>
40     <form action="insert.php">
41         <button>INSERT</button>
42     </form>
43     <table border="1">
44         <tr>
45             <td>idpegawai</td>
46             <td>jeniskelamin</td>
47             <td>alamat</td>
48             <td>iddept</td>
49             <td>aksi</td>
50         </tr>
51     </table>
</body>
</html>

```

Gambar 2.12. index.php(2)

```

52 foreach ($data as $d) {
53     <tr>
54         <td><?php echo $d["idpegawai"]; ?></td>
55         <td><?php echo $d["namapegawai"]; ?></td>
56         <td><?php echo $d["alamat"]; ?></td>
57         <td><?php echo $d["iddept"]; ?></td>
58         <td>
59             <form action="" method="GET">
60                 <button type="submit" name="delete" value=<?php echo $d["idpegawai"]; ?>>DELETE</button>
61             </form>
62             <form action="update.php" method="GET">
63                 <button type="submit" name="update" value=<?php echo $d["idpegawai"]; ?>>UPDATE</button>
64             </form>
65         </td>
66     </tr>
67 }
68 </table>
69 </body>
70 </html>

```

Gambar 2.13. index.php(3)

```

71     </table>
72     <br></br>
73     <form action="insertdept.php">
74         <button>INSERT</button>
75     </form>
76     <table border="1">
77         <tr>
78             <td>iddept</td>
79             <td>namadept</td>
80             <td>lokasi</td>
81             <td>aksi</td>
82         </tr>
83         <?php

```

Gambar 2.14. index.php(4)

```

84     foreach ($data2 as $d) {
85     >
86         <tr>
87             <td><?php echo $d["iddept"]; ?></td>
88             <td><?php echo $d["namadep"]; ?></td>
89             <td><?php echo $d["lokasi"]; ?></td>
90             <td>
91                 <form action="" method="GET">
92                     <button type="submit" name="delete1" value=<?php echo $d["iddept"]; ?>>DELETE</button>
93                 </form>
94                 <form action="updatedept.php" method="GET">
95                     <button type="submit" name="update1" value=<?php echo $d["iddept"]; ?>>UPDATE</button>
96                 </form>
97             </td>
98         </tr>
99     <?php
100     }
101     >
102 </table>
103 </body>
104 </html>

```

Gambar 2.15. index.php(5)

```

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 2.16. insert.php(1)

```

17 <!DOCTYPE html>
18 <html lang="en">
19
20 <head>
21     <meta charset="UTF-8">
22     <meta name="viewport" content="width=device-width, initial-scale=1.0">
23     <title>Insert</title>
24 </head>
25
26 <body>
27     <a href="index.php">home</a>
28     <form action="" method="POST">
29         <ul>
30
31             <li>
32                 <label for="namapegawai">namapegawai</label>
33                 <input type="text" name="namapegawai" id="namapegawai">
34             </li>
35             <li>
36                 <label for="alamat">alamat</label>
37                 <input type="text" name="alamat" id="alamat">
38             </li>
39             <li>
40                 <label for="iddept">iddept</label>
41                 <input type="text" name="iddept" id="iddept">
42             </li>
43             <li>
44                 <button type="submit" name="insert">insert</button>
45             </li>
46         </ul>
47     </form>
48 </body>
49
50 </html>

```

Gambar 2.17. insert.php(2)

```

1  <?php
2  require("function.php");
3  if (isset($_POST["insert1"])) {
4      if (insert1($_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 2.18. insertdept.php(1)

```

17 <!DOCTYPE html>
18 <html lang="en">
19
20 <head>
21     <meta charset="UTF-8">
22     <meta name="viewport" content="width=device-width, initial-scale=1.0">
23     <title>Insert</title>
24 </head>
25
26 <body>
27     <a href="index.php">home</a>
28     <form action="" method="POST">
29         <ul>
30
31             <li>
32                 <label for="namadep">namadep</label>
33                 <input type="text" name="namadep" id="namadep">
34             </li>
35             <li>
36                 <label for="lokasi">lokasi</label>
37                 <input type="text" name="lokasi" id="lokasi">
38             </li>
39             <li>
40                 <button type="submit" name="insert1">insert</button>
41             </li>
42         </ul>
43     </form>
44 </body>
45
46 </html>

```

Gambar 2.19. insertdept.php(2)

```

1  <?php
2  require("function.php");
3  $idpegawai = $_GET["update"];
4  $data = show("SELECT * FROM pegawai WHERE idpegawai = $idpegawai");
5  if (isset($_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 }
18
19 ?>

```

Gambar 2.20. update.php(1)

```

21 <!DOCTYPE html>
22 <html lang="en">
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Update</title>
27 </head>
28 <body>
29     <a href="index.php">home</a>
30     <form action="" method="POST">
31         <?php
32         foreach ($data as $d) {
33             ?>
34             <ul>
35                 <li>
36                     <label for="idpegawai">idpegawai</label>
37                     <input type="text" name="idpegawai" id="idpegawai" value=<?php echo $d["idpegawai"]; ?>>
38                 </li>
39                 <li>
40                     <label for="namapegawai">namapegawai</label>
41                     <input type="text" name="namapegawai" id="namapegawai" value=<?php echo $d["namapegawai"]; ?>>
42                 </li>
43                 <li>
44                     <label for="alamat">alamat</label>
45                     <input type="text" name="alamat" id="alamat" value=<?php echo $d["alamat"]; ?>>
46                 </li>
47                 <li>
48                     <label for="iddept">iddept</label>
49                     <input type="text" name="iddept" id="iddept" value=<?php echo $d["iddept"]; ?>>
50                 </li>
51                 <li>
52                     <button type="submit" name="ubah">update</button>
53                 </li>
54             </ul>
55         <?php>
56         ?>
57     </form>
58 </body>
59 </html>

```

Gambar 2.21. update.php(2)

```

1  <?php
2  require("function.php");
3  $iddept = $_GET["update1"];
4  $data2 = show("SELECT * FROM departemen WHERE iddept = $iddept");
5  if (isset($_POST["ubah1"])) {
6      if (update1($_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  }
18
19  ?>

```

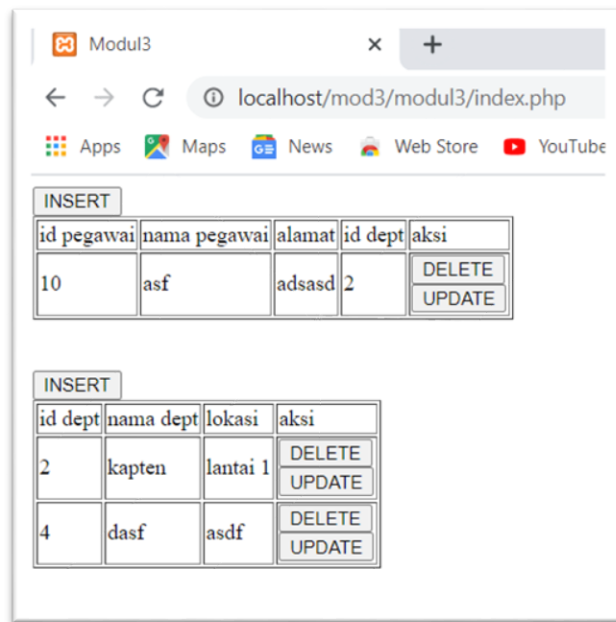
Gambar 2.22. updatedept.php(1)

```

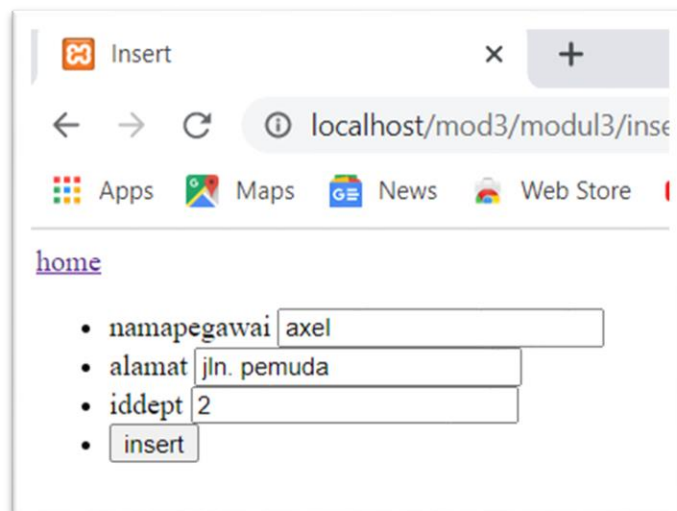
21  <!DOCTYPE html>
22  <html lang="en">
23  <head>
24      <meta charset="UTF-8">
25      <meta name="viewport" content="width=device-width, initial-scale=1.0">
26      <title>Update</title>
27  </head>
28  <body>
29      <a href="index.php">home</a>
30      <form action="" method="POST">
31          <?php
32              foreach ($data2 as $d) {
33                  ?>
34                  <ul>
35                      <li>
36                          <label for="iddept">iddept</label>
37                          <input type="text" name="iddept" id="iddept" value=?php echo $d["iddept"]; ?>
38                      </li>
39                      <li>
40                          <label for="namadep">namadep</label>
41                          <input type="text" name="namadep" id="namadep" value=?php echo $d["namadep"]; ?>
42                      </li>
43                      <li>
44                          <label for="lokasi">lokasi</label>
45                          <input type="text" name="lokasi" id="lokasi" value=?php echo $d["lokasi"]; ?>
46                      </li>
47                      <li>
48                          <button type="submit" name="ubah1">update</button>
49                      </li>
50                  </ul>
51                  <?php}
52                  ?>
53              </form>
54  </body>
55  </html>

```

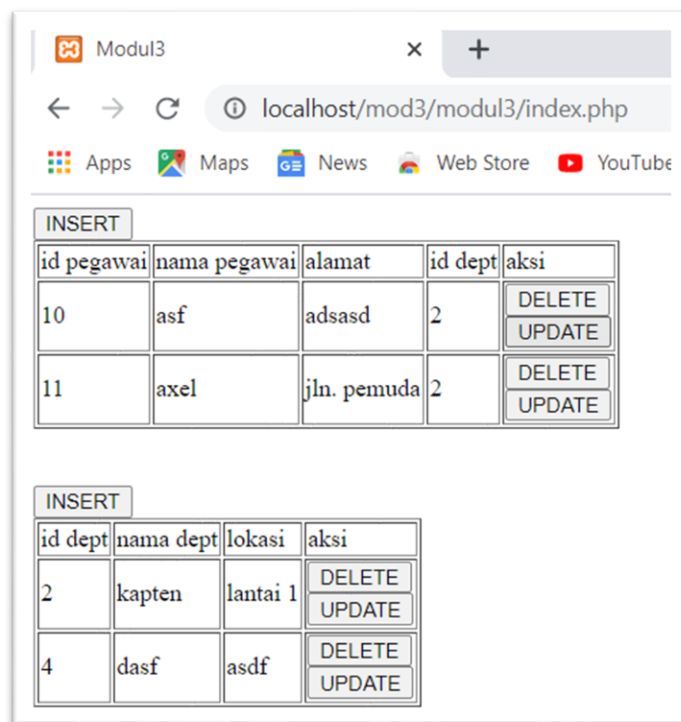
Gambar 2.23. updatedept.php (2)



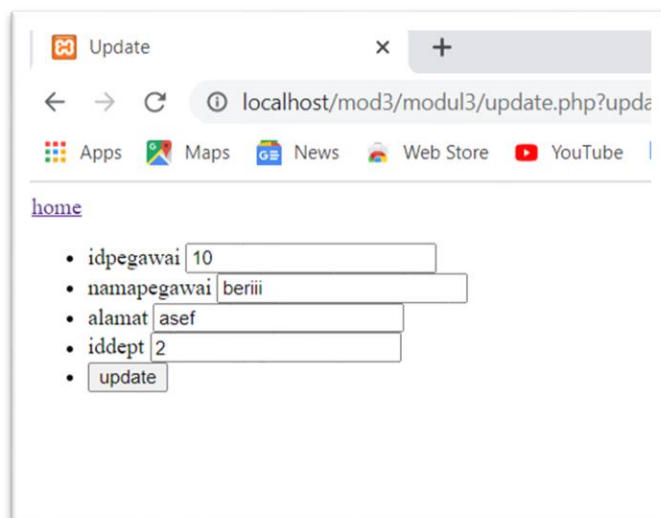
Gambar 2.24 Tampilan awal web



Gambar 2.25 Insert data pegawai



Gambar 2.26. Hasil insert data pegawai



Gambar 2.27. Update data pegawai dengan id pegawai 10

INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
10	beriii	asef	2	DELETE UPDATE
11	axel	jln. pemuda	2	DELETE UPDATE

INSERT

id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE

Gambar 2.28. Hasil update data pegawai

Modul3

localhost/mod3/modul3/index.php

Apps Maps News Web Store YouTube

INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
11	axel	jln. pemuda	2	DELETE UPDATE

INSERT

id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE

Gambar 2.29. Hasil Delete data pegawai dengan id pegawai 10

Insert

localhost/mod3/modul3/i

Apps Maps News Web Store

[home](#)

- namadep
- lokasi
-

Gambar 2.30. Insert data departemen

INSERT

id dept	nama dept	lokasi	aksi
2	kapten	lantai 1	DELETE UPDATE
4	dasf	asdf	DELETE UPDATE
5	administra	kalimantan tengah	DELETE UPDATE

Gambar 2.31. Hasil insert data departemen

Update

localhost/mod3/modul3/

Apps Maps News Web Store

[home](#)

- iddept
- namadep
- lokasi
-

Gambar 2.32. Update data departemen dengan iddept 2

Modul3

localhost/mod3/modul3/index.php

Apps Maps News Web Store YouTube

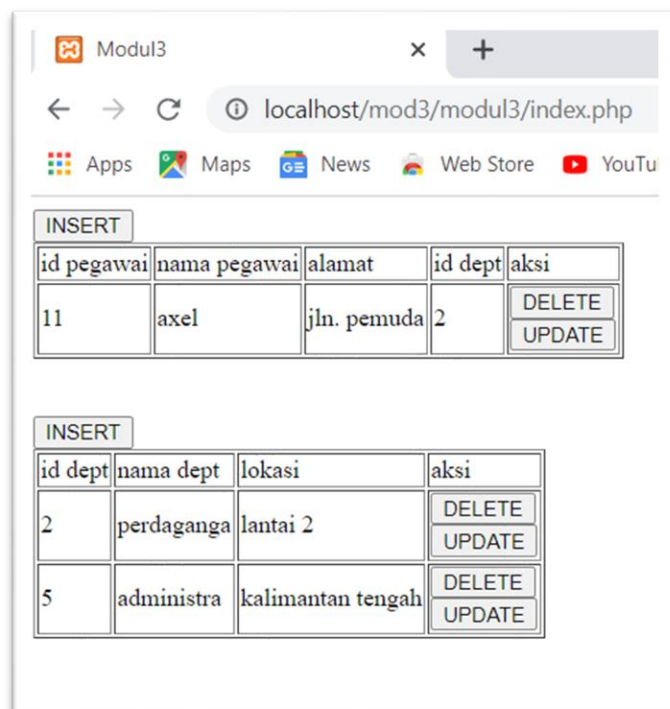
INSERT

id pegawai	nama pegawai	alamat	id dept	aksi
11	axel	jln. pemuda	2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

INSERT

id dept	nama dept	lokasi	aksi
2	perdaganga	lantai 2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
4	dasf	asdf	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
5	administra	kalimantan tengah	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.33. Hasil update data departemen



Gambar 2.34. Hasil delete data pegawai dengan iddept 4