**LAPORAN PRAKTIKUM**

**PEMROGRAMAN WEB & MOBILE I**

****

**NAMA : RANI MELIYANA PUTRI**

**NIM : 11191062**

**KELAS : C**

**MODUL IV**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

# BAB I

# TUJUAN DAN LANDASAN TEORI

1. **Tujuan Praktikum**

**1.1.** Mahasiswa mampu membuat program dengan menggabungkan fitur PHP dan Javascript.

**1.2.** Mahasiswa mampu membuat program web yang dinamis.

1. **Landasan Teori**

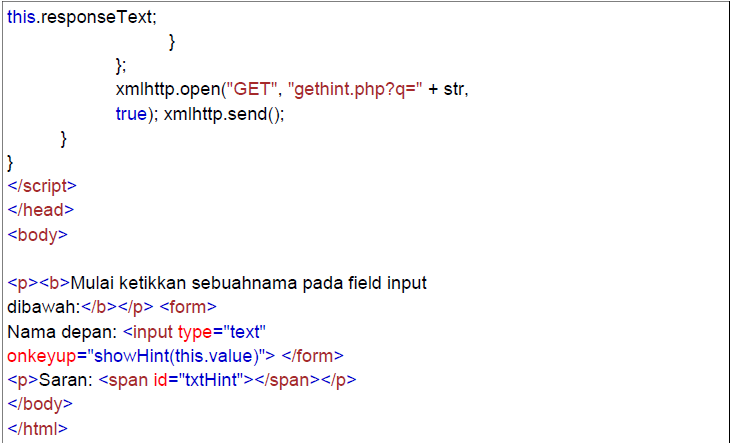
AJAX digunakan untuk membuat halaman web yang cepat dan dinamis. AJAX mengizinkan halaman web diperbaharui secara asinkron oleh pertukaran data dalam jumlah kecil dengan aktifitas server dibelakang layar. Jadi, pembaharuan halaman web dilakukan tanpa merubah seluruh tampilan halaman. Halaman web yang klasik (tidak menggunakan AJAX) harus memuat seluruh halaman jika kontennya harus berubah. Contoh aplikasi penggunaan AJAX adalah seperti Google Maps, Gmail, Youtube, dan Tab Facebook. AJAX didasari oleh Standar internet dan menggunakan kombinasi dari:

* Objek XMLHttpRequest (untuk merubah data secara asinkron dengan sebuah server)
* JavaScript/DOM (untuk menampilkan atau berinteraksi dengan informasi)
* CSS (untuk memodelkan data)
* XML (sering digunakan sebagai format pengiriman data)

DOM adalah singkatan dari Document Object Model.

Kode program berikut adalah contoh penggunaan AJAX untuk membuat saran otomatis pada text input:





Penjelasan:

Pertama, program memeriksa apakah field input kosong (str.length == 0). Jika kosong, bersihkan kontent dari placeholder txtHint dan keluar dari fungsi. Namun, jika field input tidak kosong, maka akan melakukan hal-hal sebagai berikut:

• Membuat sebuah objek XMLHttpRequest.

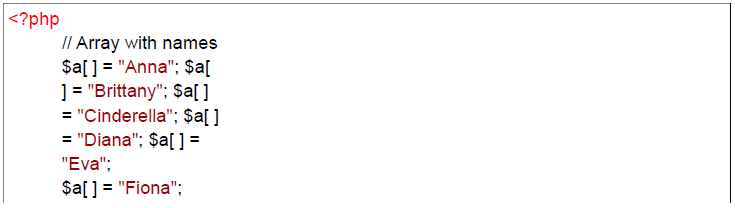
• Membuat fungsi untuk dieksekusi ketika jawaban server adalah siap (ready).

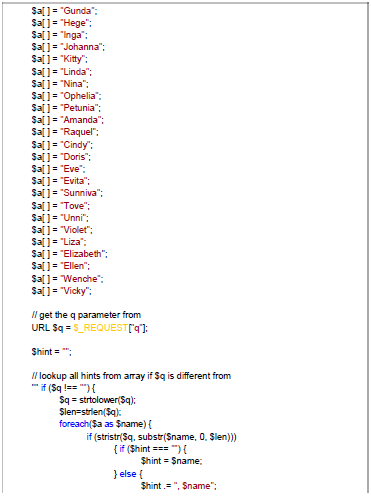
• Mengirim permintaan off ke file PHP (gethint.php) pada server.

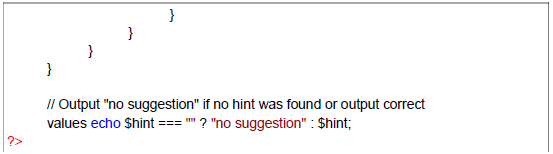
• Memberitahukan bahwa parameter q ditambah ke URL (gethint.php?q=”+str)

• Variabel str menyimpan konten field input.

Kode program diatas menggunakan file PHP dengan nama “gethint.php”, kode program dari file tersebut adalah sebagai berikut:



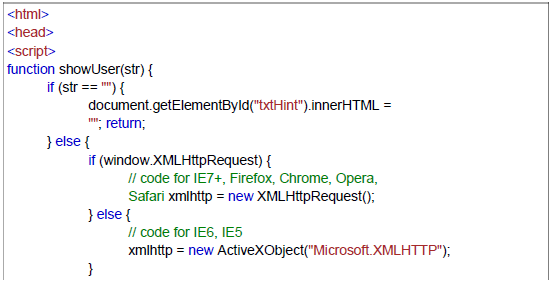




**AJAX dan MySQL**

AJAX bisa digunakan untuk komunikasi interaktif dengan sebuah database. Dibawah ini akan diberikan contoh kode program untuk mengakses database menggunakan AJAX, namun sebelum itu, perlu dibuat tabel “user” pada database “my\_db” dan diisi dengan data sebagai berikut:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **FirstName** | **LastName** | **Age** | **Hometown** | **Job** |
| 1 | Peter | Griffin | 41 | Quahog | Brewery |
| 2 | Lois | Griffin | 40 | Newport | Piano Teacher |
| 3 | Joseph | Swanson | 39 | Quahog | Police Officer |
| 4 | Glenn | Quagmire | 41 | Quahog | Pilot |





Penjelasan kode:

Pertama, memeriksa apakah ada data orang yang terpilih. Jika tidak ada maka (str

== “”), bersihkan kontent dari txtHint dan keluar dari fungsi. Jika ada data orang terpilih, maka akan melakukan hal berikut:

• Membuat objek XMLHttpRequest.

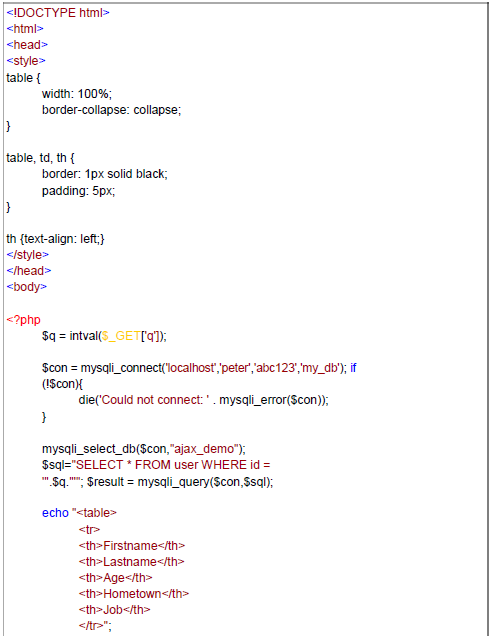
• Membuat fungsi untuk dieksekusi ketika jawaban server adalah siap (ready).

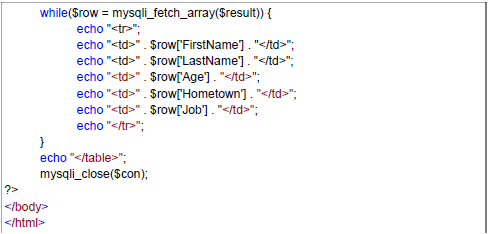
• Kirim permintaan off untuk sebuah file pada server.

• Memberitahukan bahwa sebuah parameter (q) ditambahkan ke URL (dengan

kontent dropdown list).

Contoh program diatas menggunakan file “getuser.php”. File tersebut berisi kode program untuk melakukan query pada database MySQL, Kode program dari file tersebut adalah sebagai berikut:





Penjelasan dari kode program:

Ketika query dikirimkan dari JavaScript ke file PHP, hal-hal berikut terjadi:

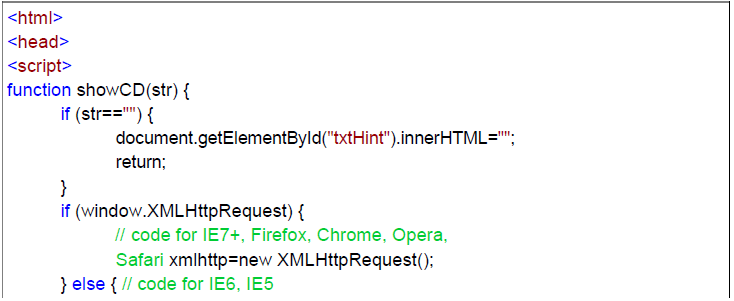
• PHP membuka koneksi ke server MySQL.

• Menemukan data orang.

• Tabel HTML dibuat, menambahkan data ke tabel tersebut, dan mengirimkannya kembali ke placeholder “txtHint”.

**AJAX dan XML**

AJAX bisa digunakan untuk komunikasi interaktif dengan file XML. Berikut adalah contoh program AJAX yang digunakan untuk menampilkan data XML:





Penjelasan kode program:

Fungsi showCD() melakukan hal-hal berikut:

• Memeriksa apakah ada CD yang terpilih.

• Membuat sebuah objek XMLHttpRequest.

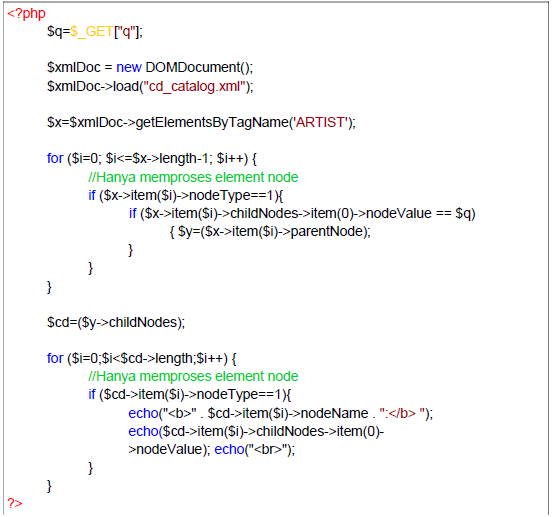
• Membuat fungsi yang akan dieksekusi ketika jawaban server adalah siap (ready).

• Mengirimkan permintaan off untuk sebuah file pada server.

• Memberitahukan bahwa sebuah parameter (q) ditambahkan ke URL (dengan kontent dari list dropdown).

Pada kode program tersebut, terdapat file “getcd.php”. Kode program dari file tersebut memuat dokumen XML “cd\_catalog.xml”, menjalankan query terhadap file XML dan mengembalikan hasilnya. File “cd\_catalog.xml” bisa didownload melalui link

https://www.w3schools.com/php/cd\_catalog.xml. Berikut adalah kode program dari file “getcd.php”:



Ketika query CD dikirimkan dari JavaScript ke halaman PHP, hal-hal berikut terjadi:

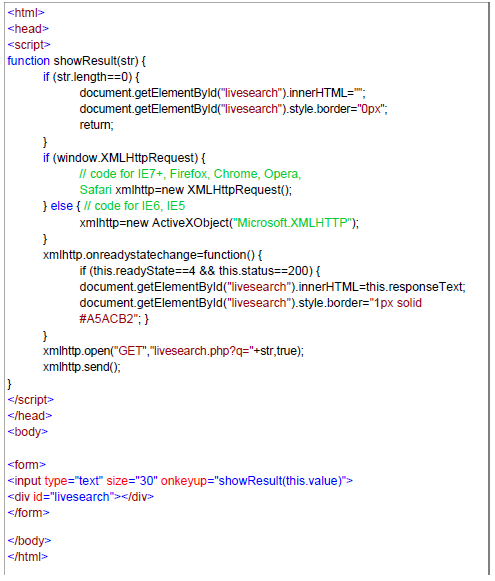
• PHP membuat sebuah objek XML DOM.

• Menemukan semua element yang sesuai dengan nama yang dikirimkan dari JavaScript.

• Mengeluarkan output berupa informasi album (mengirimkan ke placeholder “txtHint”)

**AJAX – Pencarian Langsung (Live Search)**

AJAX bisa digunakan untuk membuat pencarian keyword yang lebih interaktif dan user-friendly. Berikut adalah contoh program untuk melakukan pencarian melalui query pada sebuah file XML “links.xml”. File ini bisa didownload pada link: https://www.w3schools.com/php/links.xml . Kode program dari contoh program pencarian adalah sebagai berikut:

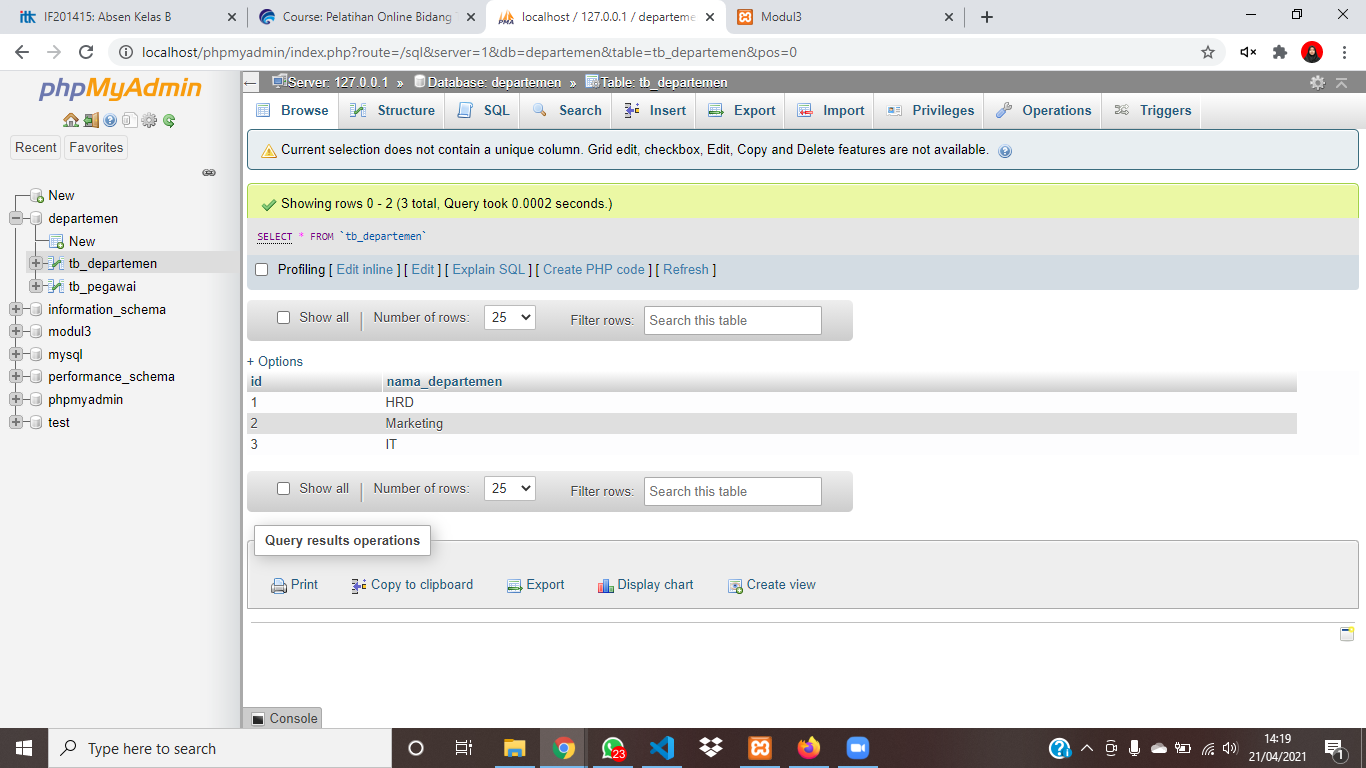


# BAB II

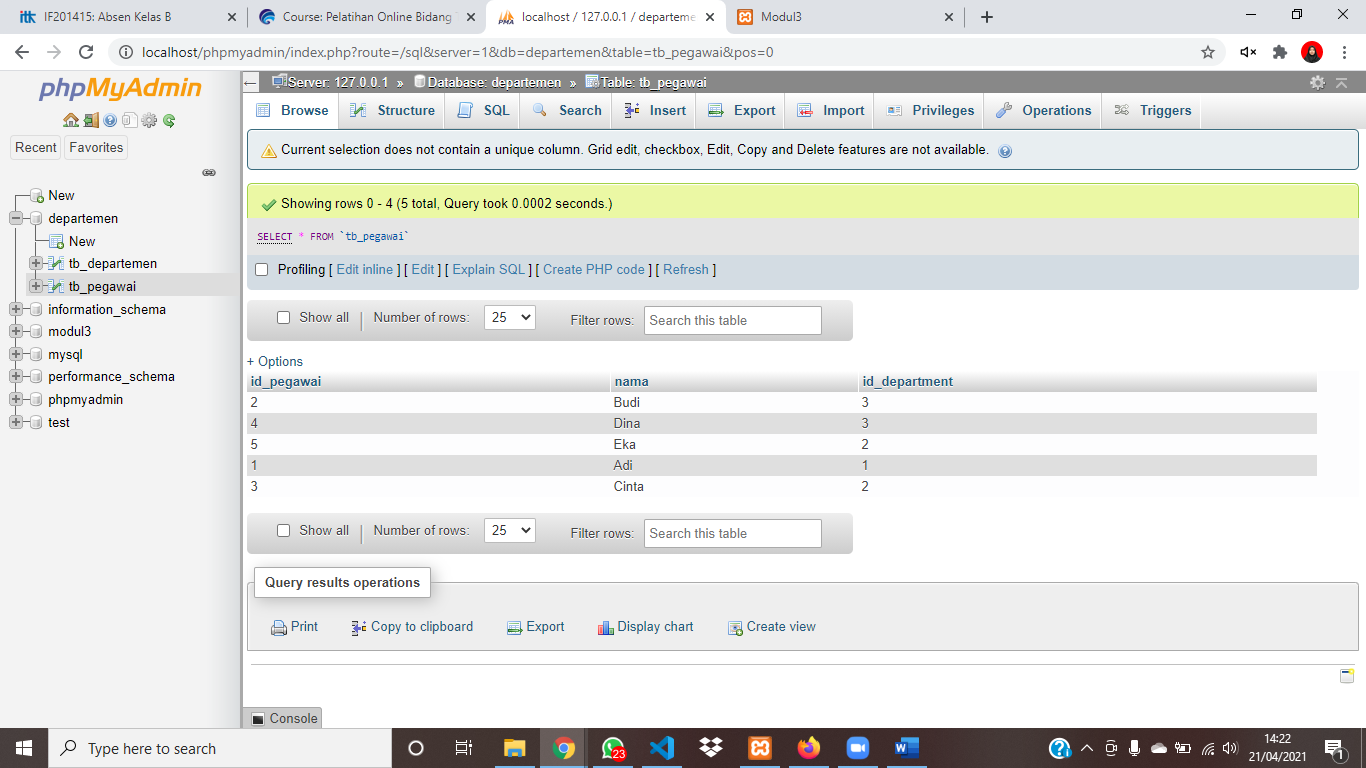
# PEMBAHASAN

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

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



**Gambar 2.1 Tabel Departemen**



**Gambar 2.2 Tabel Pegawai**

**Source Code function.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | <?php  $servername="localhost";  $database="departemen";  $username="root";  $password="";  $conn=mysqli\_connect($servername,$username,$password,$database);  function show($query){  global $conn;  $result=mysqli\_query($conn,$query);  $rows=[];  while($row=mysqli\_fetch\_assoc($result)){  $rows[]=$row;  }  return$rows;  }  function insert($data){  global $conn;  $id\_pegawai=$\_POST["id\_pegawai"];  $nama=$\_POST["nama"];  $id\_department=$\_POST["id\_department$id\_department"];    mysqli\_query($conn,"INSERT INTO tb\_pegawai VALUES('$id\_pegawai','$nama','$id\_department')");  return mysqli\_affected\_rows($conn);  }  function delete($data){  global $conn;  $id\_pegawai=$\_GET["delete"];  mysqli\_query($conn,"DELETE FROM tb\_pegawai WHERE id\_pegawai=$id\_pegawai");  return mysqli\_affected\_rows($conn);  }  function update($data){  global $conn;  $id\_pegawai=$\_POST["id\_pegawai"];  $nama=$\_POST["nama"];  $id\_department=$\_POST["id\_department $id\_department"];  mysqli\_query($conn,"UPDATE tb\_pegawai SET nama='$nama',  id\_department='$id\_department'  WHERE id\_pegawai= $id\_pegawai");  return mysqli\_affected\_rows($conn);  }  ?> |

**Penjelasan**

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

$servername="localhost";

$database="departemen";

$username="root";

$password="";

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

function show($query){

    global $conn;

    $result=mysqli\_query($conn,$query);

    $rows=[];

    while($row=mysqli\_fetch\_assoc($result)){

        $rows[]=$row;

    }

    return$rows;

}

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

function insert($data){

    global $conn;

    $id\_pegawai=$\_POST["id\_pegawai"];

    $nama=$\_POST["nama"];

    $id\_department=$\_POST["id\_department$id\_department"];

    mysqli\_query($conn,"INSERT INTO tb\_pegawai VALUES('$id\_pegawai','$nama','$id\_department')");

    return mysqli\_affected\_rows($conn);

}

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

function delete($data){

    global $conn;

    $id\_pegawai=$\_GET["delete"];

    mysqli\_query($conn,"DELETE FROM tb\_pegawai WHERE id\_pegawai=$id\_pegawai");

    return mysqli\_affected\_rows($conn);

}

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

function update($data){

    global $conn;

    $id\_pegawai=$\_POST["id\_pegawai"];

    $nama=$\_POST["nama"];

    $id\_department=$\_POST["id\_department $id\_department"];

    mysqli\_query($conn,"UPDATE tb\_pegawai SET nama='$nama',

    id\_department='$id\_department'

    WHERE id\_pegawai= $id\_pegawai");

    return mysqli\_affected\_rows($conn);

}

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

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

**Source Code insert.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | <?php  require("function.php");  if(isset($\_POST["insert"])){  if(insert($\_POST>0)){  echo "  <script>alert('data berhasil ditambah');  document.location.href='index.php';  </script>";  }  else{  echo"<script>  alert('data gagal ditambah');  document.location.href='index.php';  </script>";  }  }  ?>  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"content="width=device-width,initial-scale=1.0">  <title>Insert</title>  </head>  <body>  <a href="index.php">home</a>  <form action="" method="POST">  <ul>  <li>  <label for="id\_pegawai">id\_pegawai</label>  <input type="text" name="id\_pegawai" id="id\_pegawai">  </li>  <li>  <label for="nama">nama</label>  <input type="text"name="nama"id="nama">  </li>  <li>  <label for="id\_department">id\_department</label>  <input type="text"name="id\_department"id="id\_department">  </li>  <li>  <button type="submit"name="insert">insert</button>  </li>  </ul>  </form>  </body>  </html> |

**Penjelasan**

require("function.php");

if(isset($\_POST["insert"])){

    if(insert($\_POST>0)){

        echo "

        <script>alert('data berhasil ditambah');

        document.location.href='index.php';

        </script>";

    }

    else{

        echo"<script>

        alert('data gagal ditambah');

        document.location.href='index.php';

        </script>";

        }

    }

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

**Source Code update.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | <?php  require("function.php");  $id\_pegawai=$\_GET["update"];  $data=show("SELECT \* FROM tb\_pegawai WHERE id\_pegawai =$id\_pegawai");  if(isset($\_POST["ubah"])){  if(update($\_GET>0)){  echo "<script>  alert('data berhasil diupdate');  document.location.href='index.php';  </script>";  }  else{  echo"<script>  alert('data gagal diupdate');  document.location.href='index.php';</script>";  }  }?>  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"content="width=device-width,initial-scale=1.0">  <title>Update</title>  </head>  <body>  <a href="index.php">home</a>  <form action=""method="POST">  <?php foreach($data as $d){  ?>  <ul>  <li>  <label for="id\_pegawai">id\_pegawai</label>  <input type="text"name="id\_pegawai"id="id\_pegawai"value=<?php echo $d["id\_pegawai"];?>readonly>  </li>  <li>  <label for="nama">nama</label>  <input type="text"name="nama"id="nama"value=<?php echo $d["nama"];?>>  </li>  <li>  <label for="id\_department">id\_department</label>  <input type="text"name="id\_department"id="id\_department"value=<?php echo $d["id\_department"];?>>  </li>  <li>  <button type="submit"name="ubah">update</button>  </li>  </ul>  <?php  }?>  </form>  </body>  </html> |

**Penjelasan**

require("function.php");

$id\_pegawai=$\_GET["update"];

$data=show("SELECT \* FROM tb\_pegawai WHERE id\_pegawai =$id\_pegawai");

if(isset($\_POST["ubah"])){

    if(update($\_GET>0)){

        echo "<script>

        alert('data berhasil diupdate');

        document.location.href='index.php';

        </script>";

    }

    else{

        echo"<script>

        alert('data gagal diupdate');

        document.location.href='index.php';</script>";

    }

}

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

**Source Code index.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | <?php  require("function.php");  $data=show("SELECT \* FROM tb\_pegawai");  if(isset($\_GET["delete"])){  if(delete($\_GET>0)){  echo"<script>  alert('data berhasil dihapus');  document.location.href='index.php';  </script>";  }  else{  echo"<script>  alert('data gagal dihapus');  document.location.href='index.php';  </script>";}  }  ?>  <!DOCTYPEhtml>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"content="width=device-width,initial-scale=1.0">  <title>Modul3</title>  </head>  <body>  <table border="1">  <tr>  <td>id\_pegawai</td>  <td>nama</td>  <td>id\_department</td>  <td>aksi</td>  </tr>  <?php foreach($data as $d){  ?>  <tr>  <td><?php echo $d["id\_pegawai"];?></td>  <td><?php echo $d["nama"];?></td>  <td><?php echo $d["id\_department"];?></td>  <td>  <form action=""method="GET">  <button type="submit"name="delete"value=<?php echo $d["id\_pegawai"];?>>Hapus</button>  </form>  <form action="update.php"method="GET">  <button type="submit"name="update"value=<?php echo $d["id\_pegawai"];?>>Ubah</button>  </form>  </td>  </tr>  <?php  }?>  </table>  <form action="insert.php">  <button>Tambahkan Data</button>  </form>  </body>  </html> |

**Penjelasan**

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

require("function.php");

$data=show("SELECT \* FROM tb\_pegawai");

if(isset($\_GET["delete"])){

    if(delete($\_GET>0)){

        echo"<script>

        alert('data berhasil dihapus');

        document.location.href='index.php';

        </script>";

    }

    else{

        echo"<script>

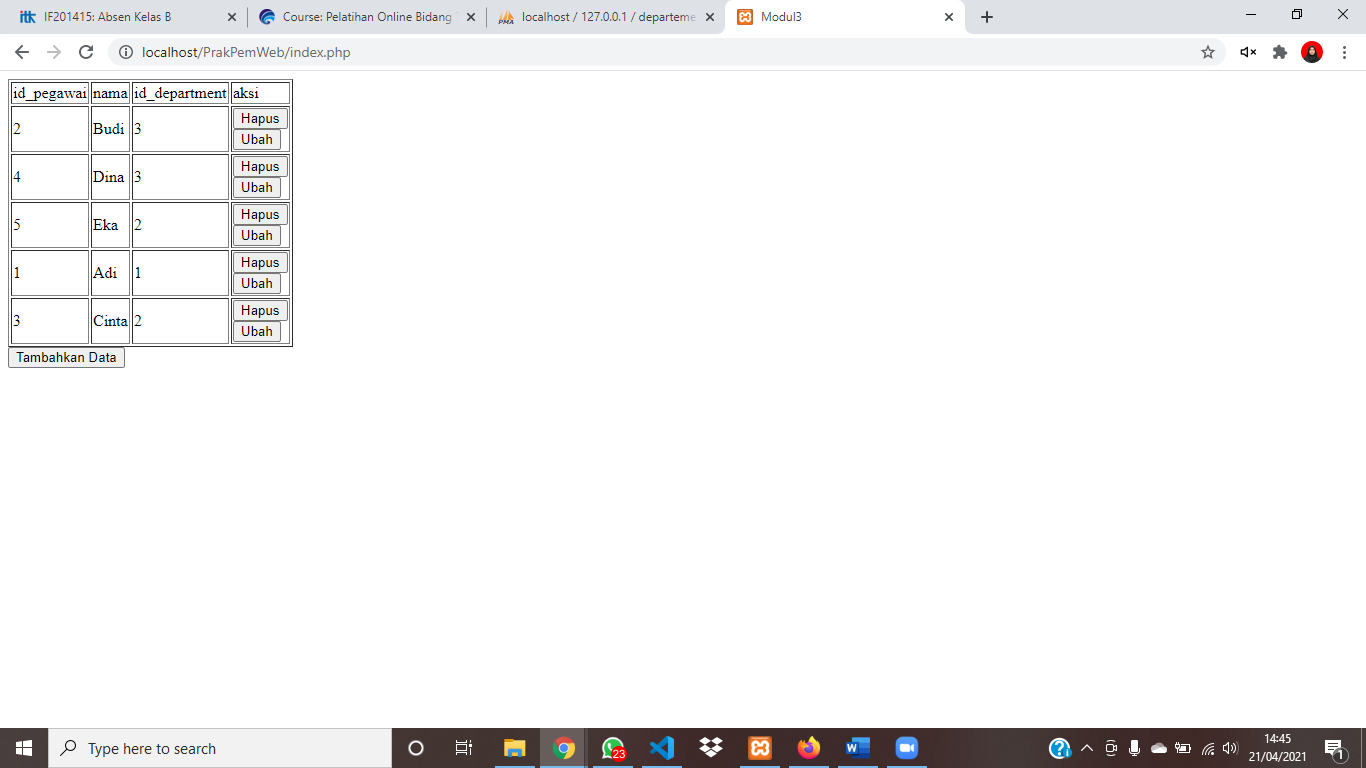
        alert('data gagal dihapus');

        document.location.href='index.php';

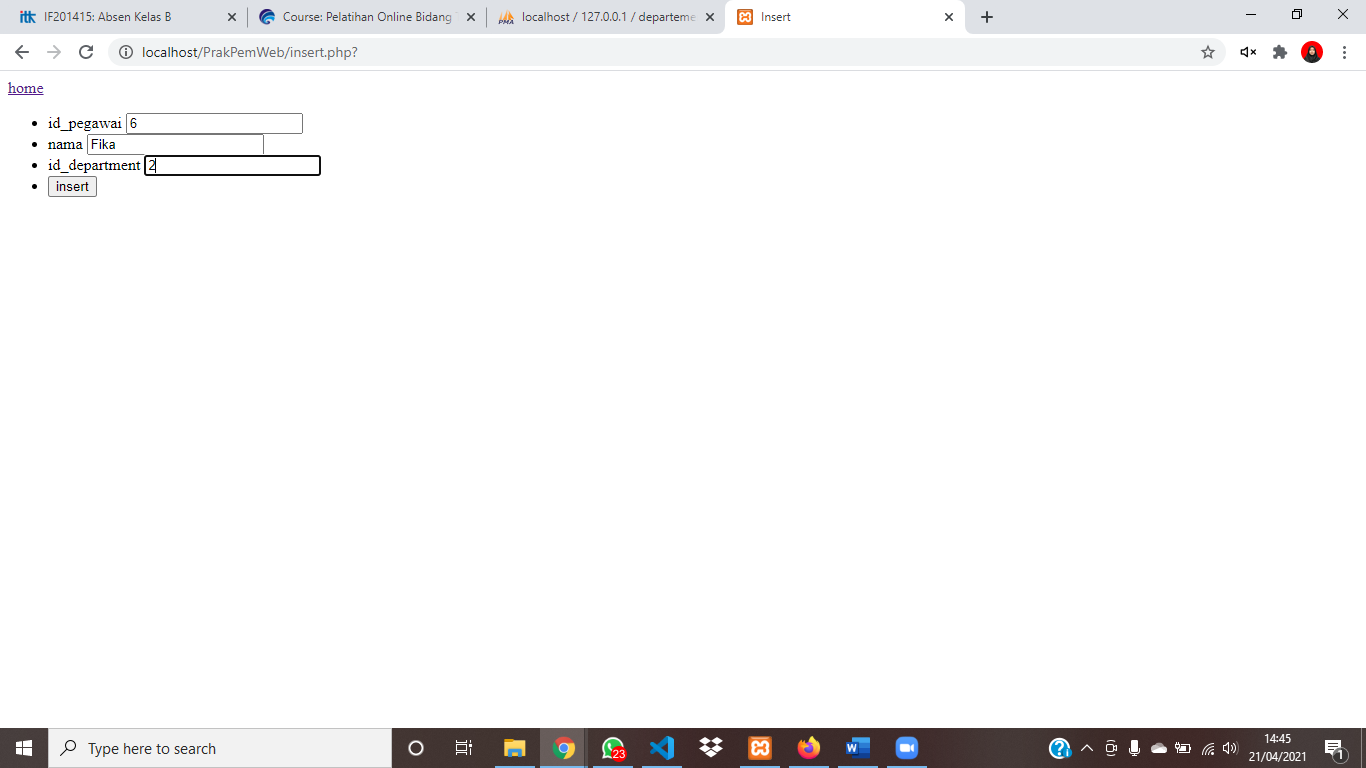
        </script>";}

    }

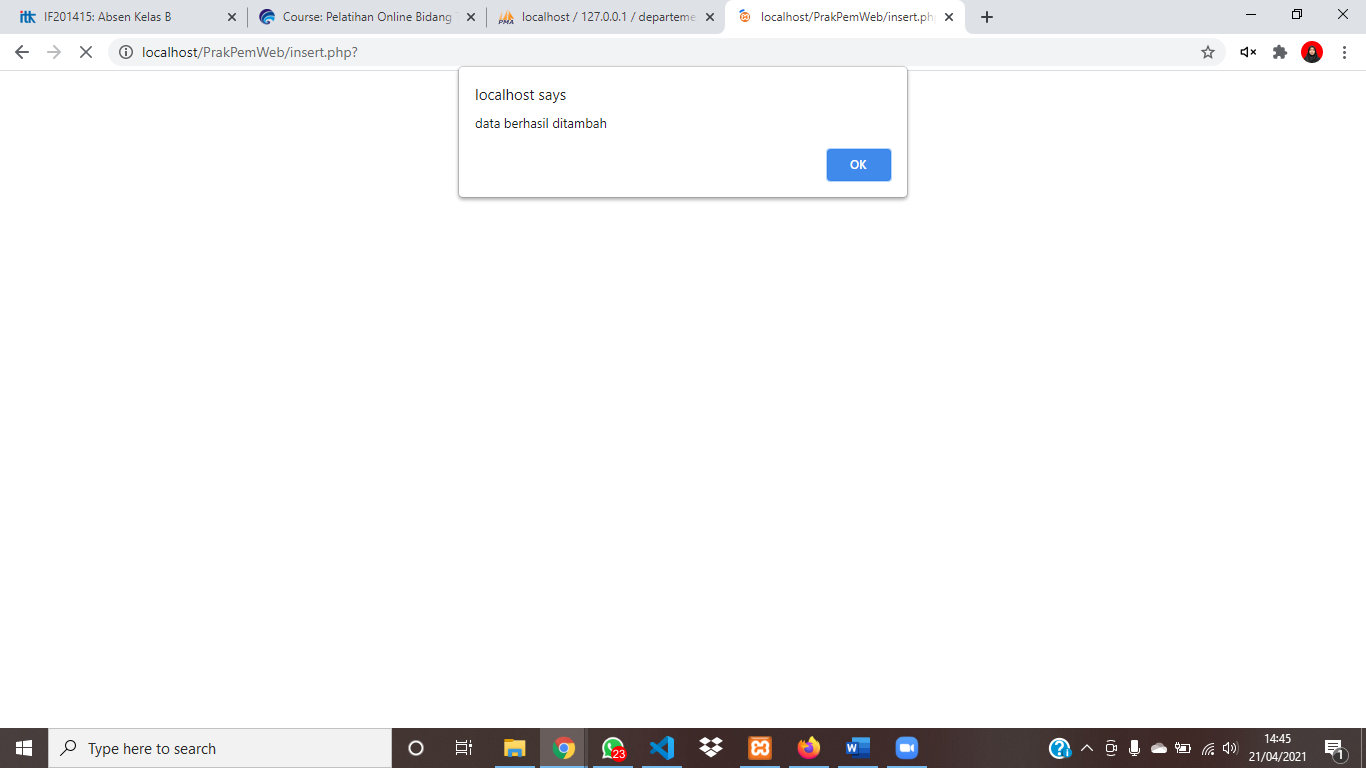
Berikut hasil program yang sudah dibuat.



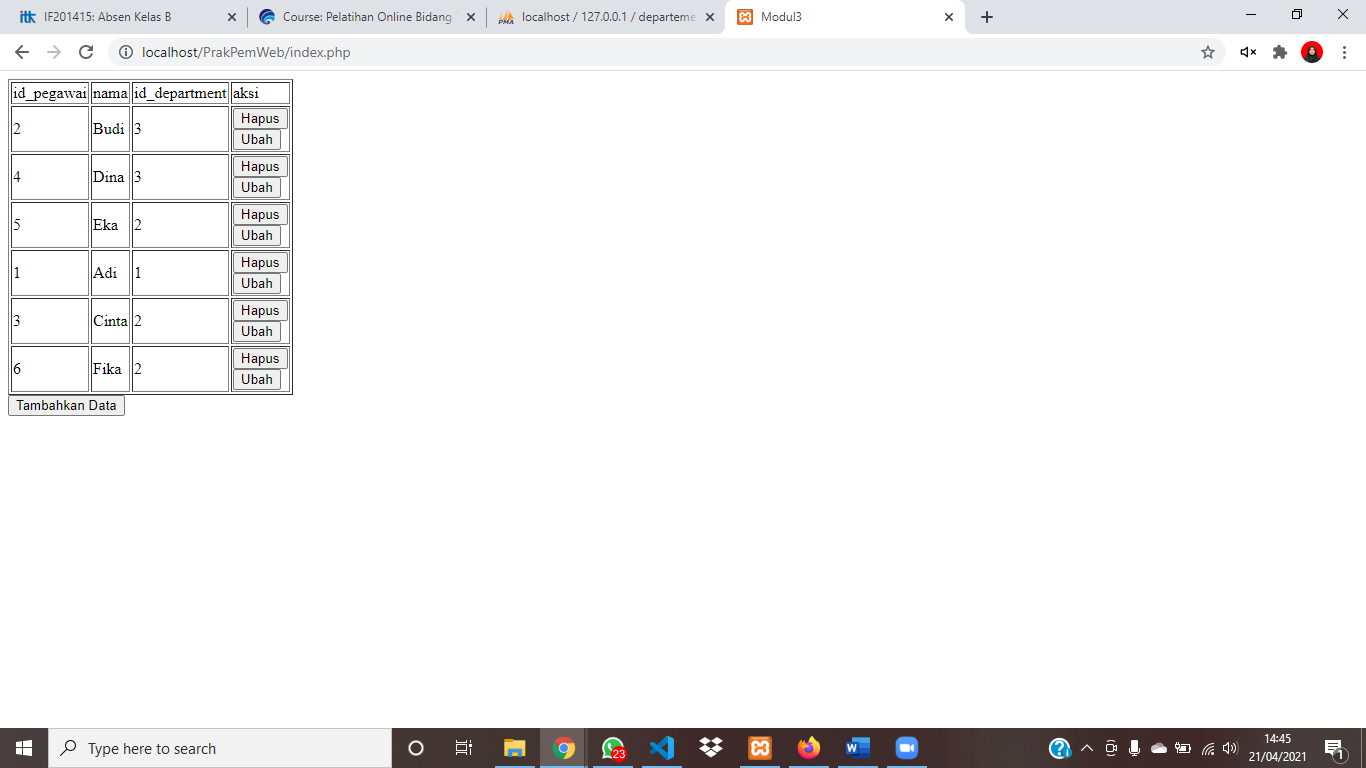
**Gambar 2.3 Tampilan awal**



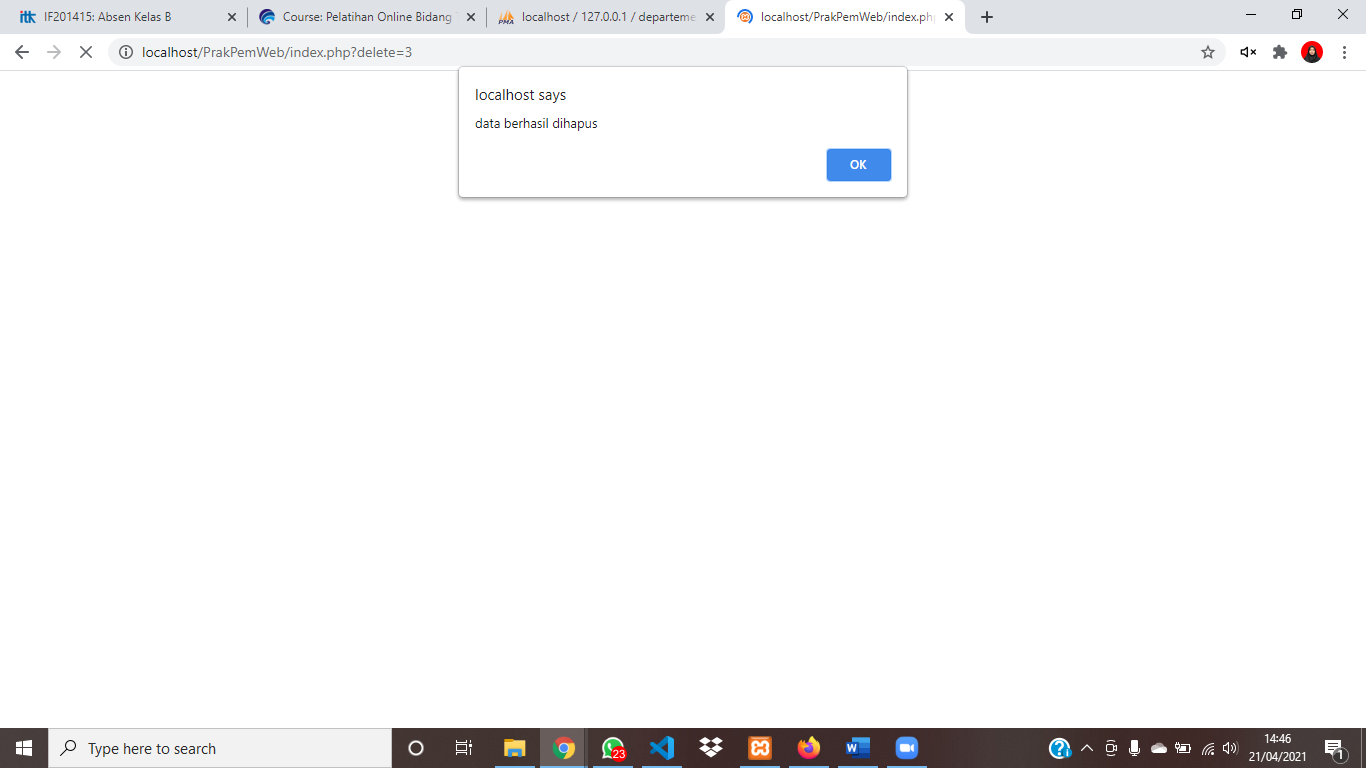
**Gambar 2.4 Tampilan Untuk Menambahkan/Insert Data**



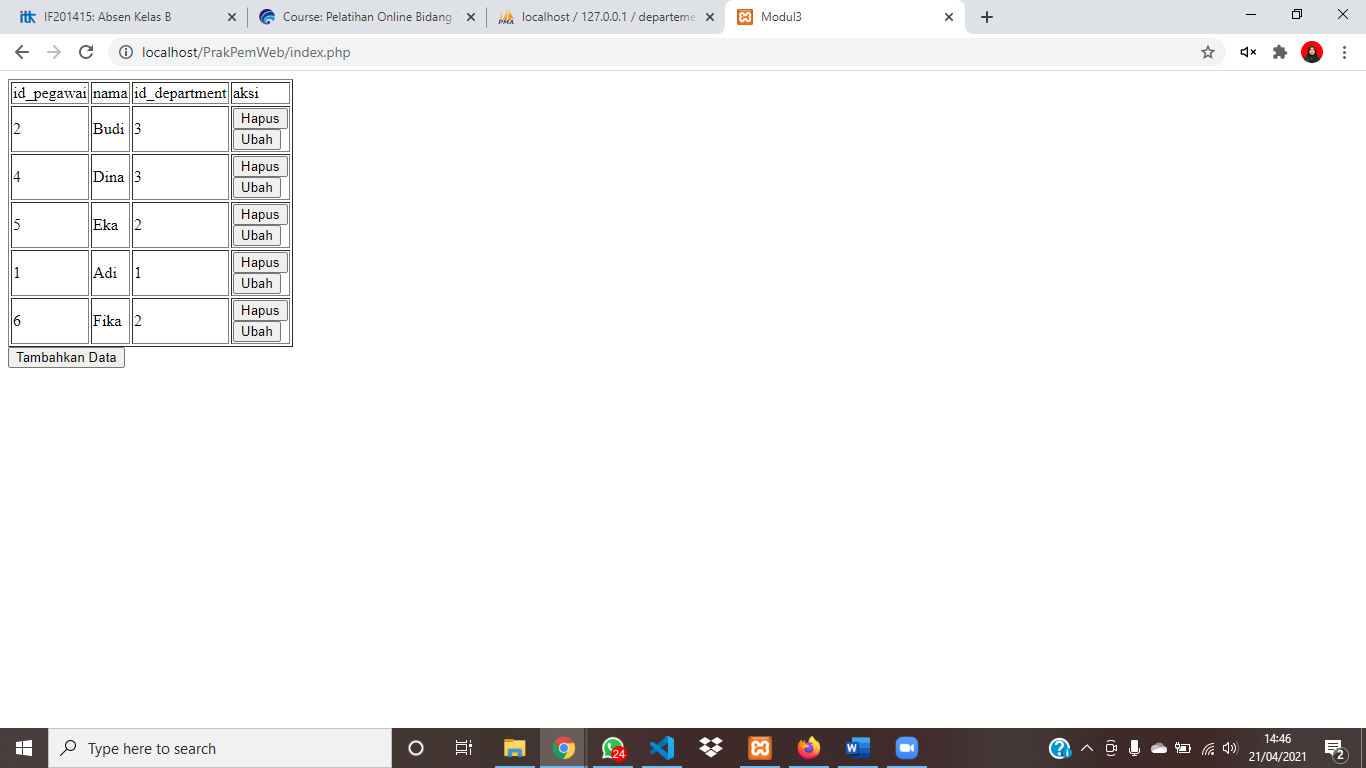
**Gambar 2.5 Tampilan jika data berhasil ditambah.**



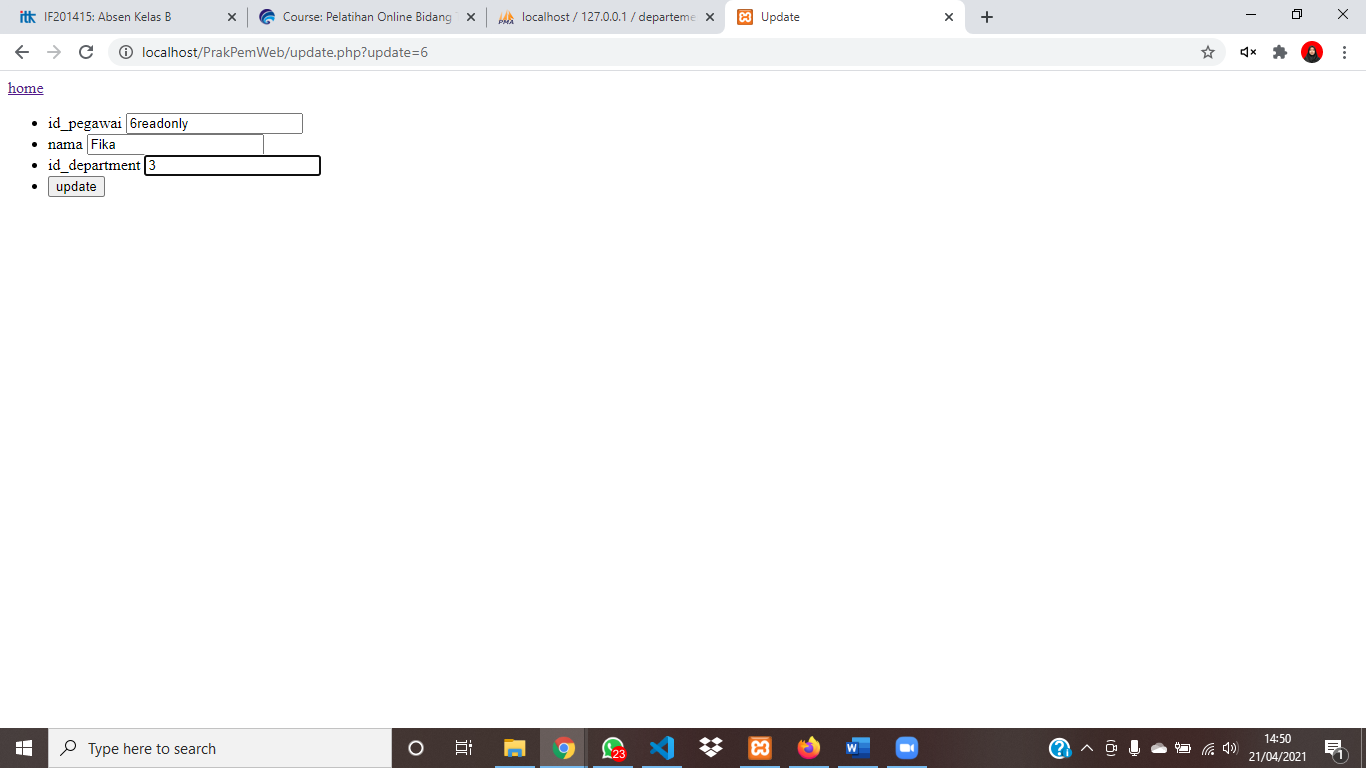
**Gambar 2.6 Tampilan jika data sudah berhasil ditambahkan.**



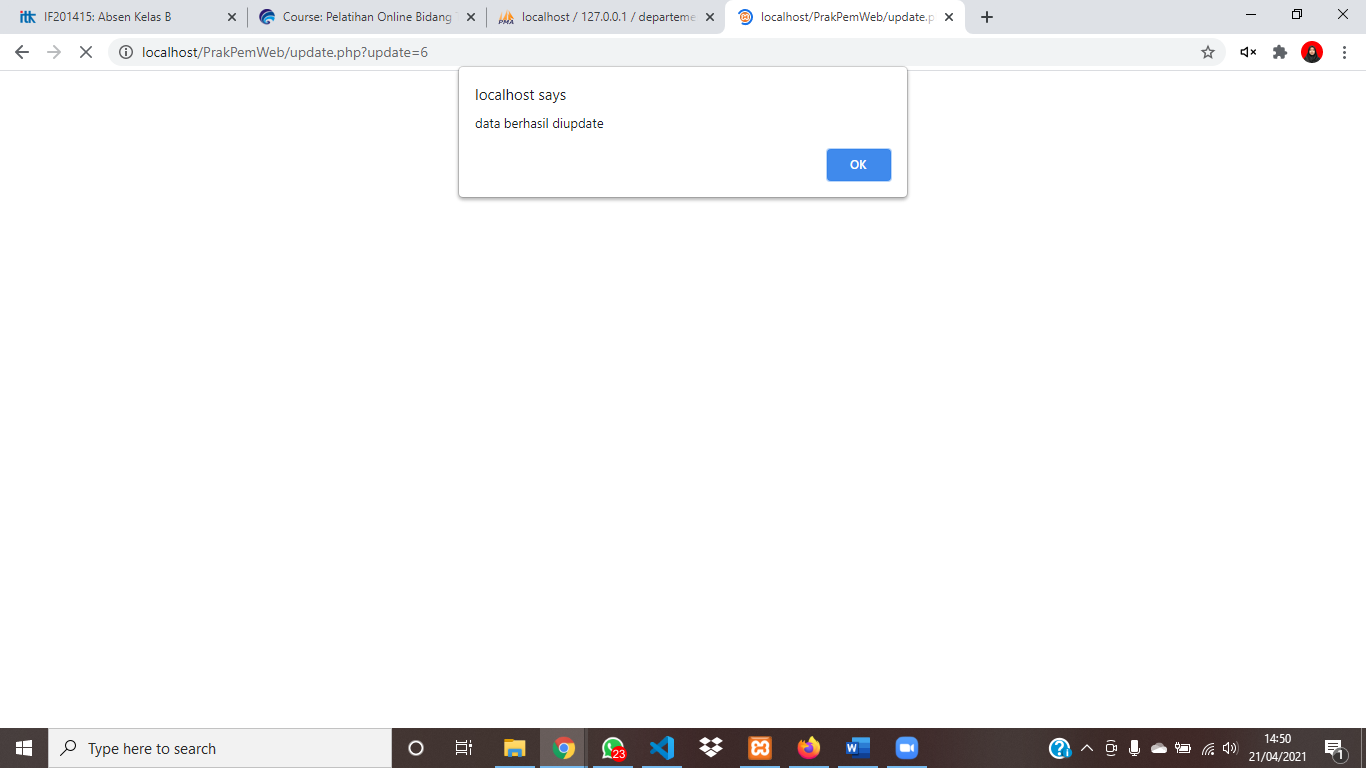
**Gambar 2.7 Tampilan jika data berhasil dihapus.**



**Gambar 2.8 Tampilan setelah data dihapus.**



**Gambar 2.9 Tampilan jika mengubah data.**



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

# BAB III

# KESIMPULAN

PHP and MySQL merupakan kolaborasi antara bahasa pemrograman dan layanan database yang populer saat ini. PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user.

# DAFTAR PUSTAKA

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

*Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

# LAMPIRAN

