

LAPORAN HASIL PRAKTIKUM PEMROGRAMAN WEB & MOBILE 1



NAMA : SATRIA SEPTA ARIANTO

NIM : 193020503026

KELAS : A

MODUL : III (PHP MySQL Database)

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
TAHUN 2021**

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. Tujuan

- 1.1.1 Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.
- 1.1.2 Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

1.2. Landasan Teori

1.2.1 Membuka Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL.(Praktikum, 2021). Berikut adalah contoh kode program agar terhubung dengan server mySQL:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);
    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Jika kode diatas tidak berhasil, kemungkinan variabel \$connect_error sudah tidak tersedia pada versi PHP yang kita gunakan, maka gunakan kode program berikut:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    // Membuat Hubungan
    $conn = mysqli_connect($servername, $username, $password);
    // Memeriksa Hubungan
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Connected successfully";
?>

```

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut:

```
mysqli_close($conn);
```

1.2.2 Membuat Database

Database pada MySQL bisa juga dibuat menggunakan kode program PHP. Program tersebut akan berisi statement SQL “CREATE DATABASE”. Berikut adalah contoh program yang digunakan untuk membuat database “myDB” :

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error()); }

```

```
// Membuat Database $sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

1.2.3 Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Bentuk umum SQL untuk membuat suatu table secara sederhana sebagai berikut :

```
CREATE TABLE nama_tabel (
field1 tipe(panjang),
field2 tipe(panjang),
...
fieldn tipe(panjang),
PRIMARY KEY (field_key)
);
```

Bentuk umum di atas merupakan bentuk umum pembuatan tabel yang sudah disederhanakan. Penamaan tabel dan field memiliki aturan yang sama dengan penamaan database.(Achmad Solichin. S.Kom, 2016). Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

```

CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
KEY, firstname VARCHAR(30) NOT NULL, lastname
VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)

```

Berikut adalah contoh program untuk membuat tabel diatas:

```

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }
    // SQL untuk membuat tabel
    $sql = "CREATE TABLE MyGuests (id INT(6) UNSIGNED
        AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
    )";
    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    }

```

```
} else {  
    echo "Error creating table: " . mysqli_error($conn);  
}  
mysqli_close($conn);  
?>
```

1.2.4 Memasukkan Data ke Database

Untuk memasukkan data kedalam tabel di database, terdapat beberapa aturan sintaks yang harus diikuti:

- Query SQL harus diberikan kutip dalam PHP.
- Nilai string didalam query SQL harus diberikan kutip.
- Nilai numeris tidak harus diberikan kutip.
- Nilai NULL tidak harus diberikan kutip.

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:

```
INSERT INTO nama_tabel VALUES ('nilai1','nilai2',...);
```

atau dapat dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel(field1,field2,...)  
VALUES ('nilai1','nilai2',...);
```

atau dapat juga dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel  
SET field1='nilai1', field2='nilai2',...;
```

Adapun contoh kode program PHP untuk menambahkan data tersebut adalah sebagai berikut:

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

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

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

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";
    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

Jika kita melakukan perintah INSERT pada tabel yang menggunakan field dengan opsi AUTO_INCREMENT, kita bisa mendapatkan ID dari baris yang terakhir diinputkan. Caranya menggunakan fungsi `mysqli_insert_id($conn)`; nilai kembalian dari fungsi ini akan memberikan id dari record atau baris terakhir yang diinputkan.

Untuk menambahkan record dengan jumlah yang banyak, kita bisa menggunakan fungsi `mysqli_multi_query($conn, $sql)`, berikut adalah

contoh program untuk menambahkan record dengan jumlah yang banyak:

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

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

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

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Mary', 'Moe', 'mary@example.com')";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Julie', 'Dooley', 'julie@example.com')";
    if (mysqli_multi_query($conn, $sql)) {
        echo "New records created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

1.2.5 Mengambil Data dari Database

Untuk mengambil data, statement SQL yang digunakan adalah
SELECT nama_kolom() FROM nama_tabel, atau kita bisa

menggunakan karakter * untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan statement WHERE setelah nama_tabel. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:

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

    // Membuat koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }
    $sql = "SELECT id, firstname, lastname FROM
    MyGuests"; $result = mysqli_query($conn, $sql);
    if (mysqli_num_rows($result) > 0) {
        // Menampilkan data pada setiap baris
        while($row = mysqli_fetch_assoc($result)) {
            echo "id: " . $row["id"]. " - Name: " .
            $row["firstname"]. " " . $row["lastname"]. "<br>";
        }
    } else {
        echo "0 results";
    }
    mysqli_close($conn);
?>
```

1.2.6 Menghapus Data dari Database

Statement DELETE digunakan untuk menghapus baris data atau records dari tabel. Clause WHERE digunakan untuk menspesifikasikan baris yang akan dihapus. Jika statement DELETE digunakan tanpa menggunakan clausa WHERE, maka semua record yang ada pada tabel akan dihapus.

id	namadepan	namabelakang	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Mary	mary@example.com	2014-10-23 10:22:30
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23

Jika kita memiliki data seperti tabel “MyGuest” seperti terlihat diatas, kemudian kita ingin menghapus data dengan nama depan Julie, maka kode program untuk menghapus data tersebut adalah sebagai berikut :

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password,
    $dbname);
    // Memeriksa koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    // sql untuk menghapus record
    $sql = "DELETE FROM MyGuests WHERE id=3";
```

```

        if (mysqli_query($conn, $sql)) {
            echo "Record deleted successfully";
        } else {
            echo "Error deleting record: " . mysqli_error($conn);
        }
        mysqli_close($conn);
    ?>

```

1.2.7 Update Data dalam Database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:

UPDATE nama_tabel

SET kolom1=nilai1, kolom2=nilai2, ...

WHERE kolom_penentu=nilai_penentu ...

Jika statement ini digunakan untuk update data ke dua pada tabel “MyGuest” diatas untuk mengganti nama belakang dari “Moe” menjadi “Doe”, maka kode program untuk update tersebut adalah sebagai berikut:

```

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

    // Membuat Koneksi

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

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

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
    if (mysqli_query($conn, $sql)) {

```

```
        echo "Record updated successfully";  
    } else {  
        echo "Error updating record: " . mysqli_error($conn);  
    }  
    mysqli_close($conn);  
?>
```

BAB II PEMBAHASAN

2.1 Tugas

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.

2.2 Program Tampil Database Pegawai

Program web yang diinginkan adalah untuk menampilkan data pegawai serta relasinya yang bisa menyimpan, menghapus dan mengubah data. Namun, melainkan menyimpan, saya membuat program untuk tambah data. Program web ini terdiri dari 9 file php berbeda. 2 untuk tabel database, 2 untuk fungsi 2 tabel yang ada. 3 untuk program tambah, ubah dan hapus data pegawai. Lalu, 2 program untuk tambah dan hapus relasi pegawai yaitu departemen. Berikut adalah kode program dan deskripsinya.

2.2.1 employee.php

```
<?php
require 'functionemployee.php';
$employs = query("SELECT * FROM employee INNER JOIN department ON employee.id_department=department.id");
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFaw/dA1S6JXm" crossorigin="anonymous">
  <title>Modul 3</title>
</head>
```

Gambar 2.1 Employee bagian 1

Kode program php pada awal adalah untuk menyatakan dibutuhkannya program fungsi lain untuk file ini. Lalu ada variabel \$employs yang akan menampung nilai dari query di dalam kurung tersebut. Query tersebut untuk memanggil semua atribut yang ada pada tabel employee dan department. Lalu

untuk awal html seperti biasa. Di head ada meta dan ada penggunaan bootstrap. Juga ada pembuatan judul pada title.

```
<body>
  <div class="container">
    <div class="row">
      <div class="col mt-5">
        <a href="employee_tambah.php">
          <div class="div btn btn-primary mb-4">
            Tambah Data
          </div>
        </a>

        <a href="department.php">
          <div class="div btn btn-primary mb-4">
            Ke Halaman Department
          </div>
        </a>

        <table class="table table-striped">
          <thead>
            <tr>
              <th scope="col">No.</th>
              <th scope="col">Nama</th>
              <th scope="col">ID</th>
              <th scope="col">Department</th>
              <th scope="col">Action</th>
            </tr>
          </thead>
        </table>
      </div>
    </div>
  </div>
```

Gambar 2.2 Employee bagian 2

Kode program pada badan html ini adalah untuk tampilan awal jika file ini yang terbuka. Terlihat banyak penggunaan div untuk membagi tiap bagiannya. Lalu ada 2 class button yang dapat menghubungkan ke file lain seperti tambah pegawai ataupun ke department. Lalu dibawahnya ada penggunaan table untuk menampung data yang akan ditampilkan. Sejah ini baru terlihat table headnya yaitu tiap atribut yang akan ditunjukkan.

```
|  |
| --- |
| <?php $i=1; ?> <?php foreach($employs as $employee) : ?> <td><?php echo $i; ?></td> <td><?php echo $employee["nama"]; ?></td> <td><?php echo $employee["id_employee"]; ?></td> <td><?php echo $employee["nama_department"]; ?></td> <td> <a href="employee_ubah.php?id=<?php echo $employee["id_employee"]; ?>">ubah</a> | <a href="employee_delete.php?id=<?php echo $employee["id_employee"]; ?>" onclick="return confirm('Yakin?');">hapus</a> </td> </tr> </tbody> <?php $i++; ?> <?php endforeach; ?> </table> </div> </div> </body> </html> |

```

Gambar 2.3 Employee bagian 3

Lalu pada badan tabel ada penggunaan sintaks php untuk memanggil data yang sudah disimpan sebelumnya pada variabel \$employs untuk diserahkan pada variabel \$employee dan dilakukan perulangan untuk menampilkan data sebanyak data yang ada hingga tidak ada data lagi yang akan ditampilkan maka perulangan selesai. Lalu pada setiap akhir data, ada aksi yang bisa dilakukan yaitu ubah dan hapus data. Kedua aksi ini akan berhubungan dengan file yang berkaitan. Lalu, sisanya akan menutup file html.

2.2.2 department.php

```

<?php
require 'functiondepartment.php';
$departs = query("SELECT * FROM department");
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
<title>Modul 3</title>
</head>

```

Gambar 2.4 Department bagian 1

Kode program php pada awal adalah untuk menyatakan dibutuhkannya program fungsi lain untuk file ini. Lalu ada variabel \$departs yang akan menampung nilai dari query di dalam kurung tersebut. Query tersebut untuk memanggil semua atribut yang ada pada tabel department. Lalu untuk awal

html seperti biasa. Di head ada meta dan ada penggunaan bootstrap. Juga ada pembuatan judul pada title.

```
<body>
  <div class="container">
    <div class="row">
      <div class="col mt-5">
        <a href="department_tambah.php">
          <div class="div btn btn-primary mb-4">
            | Tambah Data
          </div>
        </a>

        <a href="employee.php">
          <div class="div btn btn-primary mb-4">
            | Kembali Ke Halaman Employee
          </div>
        </a>

        <table class="table table-striped">
          <thead>
            <tr>
              <th scope="col">No.</th>
              <th scope="col">Nama</th>
              <th scope="col">Action</th>
            </tr>
          </thead>
        </table>
      </div>
    </div>
  </div>
```

Gambar 2.5 Department bagian 2

Kode program pada badan html ini adalah untuk tampilan awal jika file ini yang terbuka. Terlihat banyak penggunaan div untuk membagi tiap bagiannya. Lalu ada 2 class button yang dapat menghubungkan ke file lain seperti tambah data untuk department ataupun ke employee. Lalu dibawahnya ada penggunaan table untuk menampung data yang akan ditampilkan. Sejauh ini baru terlihat table headnya yaitu tiap atribut yang akan ditunjukkan.


```

<tbody>
<tr>
<?php $i=1; ?>
<?php foreach($departs as $depart) : ?>
<td><?php echo $i; ?></td>
<td><?php echo $depart["nama_department"]; ?></td>
<td>
<a href="department_delete.php?id=<?php echo $depart["id"]; ?>" onclick="return confirm('Yakin?');">hapus</a>
</td>
</tr>
</tbody>
<?php $i++; ?>
<?php endforeach; ?>
</table>
</div>
</div>
</body>
</html>

```

Gambar 2.6 Department bagian 3

Lalu pada badan tabel ada penggunaan sintaks php untuk memanggil data yang sudah disimpan sebelumnya pada variabel \$departs untuk diserahkan pada variabel \$depart dan dilakukan perulangan untuk menampilkan data sebanyak data yang ada hingga tidak ada data lagi yang akan ditampilkan maka perulangan selesai. Lalu pada setiap akhir data, ada aksi yang bisa dilakukan yaitu hapus data. Aksi ini akan berhubungan dengan file yang berkaitan. Lalu, sisanya akan menutup file html.

2.2.3 functionemployee.php

```

<?php
$db = mysqli_connect("localhost", "root", "", "modul3");

function query($query){
    global $db;
    $result = mysqli_query($db, $query);
    $rows = [];
    while($row = mysqli_fetch_assoc($result)){
        $rows[]=$row;
    }
    return $rows;
}

function tambah($data){
    global $db;
    // ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $id_department = htmlspecialchars($data["id_department"]);

    $query = "INSERT INTO employee
    VALUES ('', '$nama', '$id_department')
    ";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

```

Gambar 2.7 Fungsi Employee bagian 1

Pada file functionemployee diatas terlihat bahwa file ini berisi fungsi yang ada pada setiap tabel atau kelompok data pada database. Contohnya adalah ini, employee atau pegawai yang didalamnya kita bisa menambah, mengubah ataupun menghapus data. Terlihat pada awal file php ada variabel \$db untuk menampung file database yang ada dengan cara menghubungkannya ke database yang kita inginkan. Lalu, ada function query dengan parameter \$query untuk memanggil query di database ke tampilan web. Lalu yang kedua ada function tambah dengan parameter \$data untuk menambah data pegawai dengan atribut nama dan id departemennya.

```
function ubah($data){
    global $db;
    $id_employee = $data["id_employee"];
    // ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $id_department = htmlspecialchars($data["id_department"]);
    $query = "UPDATE employee
    SET nama = '$nama', id_department = '$id_department'
    WHERE id_employee = $id_employee";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

function hapus($id){
    global $db;
    mysqli_query($db, "DELETE FROM employee WHERE id_employee = $id");
    return mysqli_affected_rows($db);
}
?>
```

Gambar 2.8 Fungsi Employee bagian 2

Selanjutnya, ada function ubah dengan parameter \$data untuk mengubah data nama dan id departemen pegawai berdasarkan idnya. Terakhir, ada function hapus dengan parameter \$id berdasarkan id employeenya.

2.2.4 functiondepartment.php

```
<?php
$db = mysqli_connect("localhost", "root", "", "modul3");

function query($query){
    global $db;
    $result = mysqli_query($db, $query);
    $rows = [];
    while($row = mysqli_fetch_assoc($result)){
        $rows[]=$row;
    }
    return $rows;
}

function tambah($data){
    global $db;
    // ambil data dari tiap elemen dalam form
    $nama_depart = htmlspecialchars($data["nama_depart"]);

    $query = "INSERT INTO department
    VALUES ('', '$nama_depart')
    ";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

function hapus($id){
    global $db;
    mysqli_query($db, "DELETE FROM department WHERE id = $id");
    return mysqli_affected_rows($db);
}
```

Gambar 2.9 Fungsi Department

Pada file functiondepartment diatas terlihat bahwa file ini berisi fungsi yang ada pada setiap tabel atau kelompok data pada database. Contohnya adalah ini, departemen yang didalamnya kita bisa mengubah ataupun menghapus data. Terlihat pada awal file php ada variabel \$db untuk menampung file database yang ada dengan cara menghubungkannya ke database yang kita inginkan. Lalu, ada function query dengan parameter \$query untuk memanggil query di database ke tampilan web. Lalu yang kedua ada

function tambah dengan parameter \$data untuk menambah data departemen dengan atribut nama departemennya. Terakhir, ada function hapus dengan parameter \$id berdasarkan id departemennya.

2.2.5 employee_ubah.php

```
<?php
require 'functionemployee.php';

// ambil data di URL
$id = $_GET["id"];

// query data mahasiswa berdasarkan ID
$employee = query("SELECT * FROM employee
INNER JOIN department
ON employee.id_department=department.id
WHERE employee.id_employee = $id")[0];
$departs = query("SELECT * FROM department");

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di ubah atau tidak
    if (ubah($_POST)>0){
        echo "
        <script>
        | alert('data berhasil diubah!');
        | document.location.href = 'employee.php';
        </script>
        ";
    } else {
        echo "
        <script>
        | alert('data gagal diubah!');
        | document.location.href = 'employee.php';
        </script>
        ";
    }
}
```

Gambar 2.10 Ubah Employee bagian 1

Pada file php ini merupakan file yang berfokus untuk mengubah data pada tabel employee. Pada awal file php ini membutuhkan file fungsi employee sebelumnya. Lalu menggunakan GET id sebagai data untuk acuan perubahan data employee. Yang diubah juga bisa departemen yang menaungi pegawai yang diubah datanya. Akan muncul notifikasi nantinya jika perubahan data berhasil atau gagal.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoKhXA+058RXPxPg6fy4IWVTNh0E263XmFcJlSAw1GgFAM/da156JXm" crossorigin="anonymous">

<title>Modul 3</title>
</head>
<body>

<div class="container">
<div class="row">
<div class="col">
<h1>Ubah Data Employee</h1>
<a href="employee.php">
<div class="div btn btn-primary mb-4">
Kembali
</div>
</a>

<form action="" method="post">
<ul>
<li>
<label for="nama">Nama : </label>
<input type="text" name="nama" id="nama" required value="{?php echo $employee["nama"]; ?}">
</li>
<li>
<label for="nama">ID : </label>
<input type="text" name="id_employee" id="id_employee" required value="{?php echo $employee["id_employee"]; ?}">
</li>

```

Gambar 2.11 Ubah Employee bagian 2

```

<li class="mb-2">
<div class="form-group">
<label for="id_department">id_department</label>
<select name="id_department" class="form-control" id="id_department">
<option value="{?php echo $employee["id_department"]; ?}" selected="{?php echo $employee["nama_department"]; ?}">
<?php foreach($departs as $depart) : ?>
<option value="{? $depart["id"] ?}">{? $depart["nama_department"] ?}</option>
<?php endforeach; ?>
</select>
</div>
</li>

<li class="mt-2">
<button type="submit" name="submit">Tambah Data</button>
</li>
</ul>
</form>
</div>
</div>
</body>
</html>

```

Gambar 2.12 Ubah Employee bagian 3

Untuk bagian htmlnya adalah untuk tampilan saat ingin mengubah data seperti apa yang ingin diubah apakah nama, id employee, departemennya, dll. Lalu ada tombol kembali untuk ke tampilan awal data employee. Lalu setelah selesai diubah tinggal menekan tombol tambah data.

2.2.6 employee_tambah.php

```
<?php
require 'functionemployee.php';

$departs = query("SELECT * FROM department");

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
        <script>
            alert('data berhasil ditambahkan!');
            document.location.href = 'employee_tambah.php';
        </script>
        ";
    } else {
        echo "
        <script>
            alert('data gagal ditambahkan!');
            document.location.href = 'employee_tambah.php';
        </script>
        ";
    }
}
>>
```

Gambar 2.13 Tambah Employee bagian 1

Pada file php ini merupakan file yang berfokus untuk menambah data pada tabel employee. Pada awal file php ini membutuhkan file fungsi employee sebelumnya. Juga mengisi pada variabel \$departs yaitu nilai dari tabel department karena tambah data pada employee juga membutuhkan data department. Akan muncul notifikasi nantinya jika penambahan data berhasil atau gagal.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
    integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

  <title>Modul 3</title>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Tambah Data Employee</h1>
        <a href="employee.php">
          <div class="div btn btn-primary mb-4">
            kembali
          </div>
        </a>

        <form action="" method="post">
          <ul>
            <li class="mb-2">
              <label for="nama">Nama : </label>
              <input type="text" name="nama" id="nama" required>
            </li>

```

Gambar 2.14 Tambah Employee bagian 2

```

            <li class="mb-2">
              <div class="form-group">
                <label for="id_department">id_department</label>
                <select name="id_department" class="form-control" id="id_department">
                  <option selected>Choose One</option>
                  <?php foreach($departs as $depart) : ?>
                    <option value="<?=$depart["id"] ?>"><?=$depart["nama_department"] ?></option>
                  </select>
                </div>
              </li>

              <li class="mt-2">
                <button type="submit" name="submit">Tambah Data!</button>
              </li>
            </ul>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>

```

Gambar 2.15 Tambah Employee bagian 3

Untuk bagian htmlnya adalah untuk tampilan saat ingin menambah data employee seperti nama, id department yang akan terpilih saat kita memilih department. Lalu ada tombol kembali untuk ke tampilan awal data employee. Lalu setelah selesai menambah tinggal menekan tombol tambah data.

2.2.7 employee_hapus.php

```
#!/php
require 'functionemployee.php';

$id = $_GET["id"];
if(hapus($id) > 0 ){
    echo "
        <script>
            alert('data berhasil dihapus!');
            document.location.href = 'employee.php';
        </script>
    ";
} else {
    echo "
        <script>
            alert('data gagal dihapus!');
            document.location.href = 'employee.php';
        </script>
    ";
}
```

Gambar 2.16 Hapus Employee

Untuk penghapusan data employee, file ini berguna untuk mendapatkan id yang akan dihapus dan menampilkan notifikasi saat data berhasil dihapus atau gagal dihapus.

2.2.8 department_tambah.php

```
<?php
require 'functiondepartment.php';

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
        <script>
            alert('data berhasil ditambahkan!');
            document.location.href = 'department_tambah.php';
        </script>
        ";
    } else {
        echo "
        <script>
            alert('data gagal ditambahkan!');
            document.location.href = 'department_tambah.php';
        </script>
        ";
    }
}
?>
```

Gambar 2.17 Tambah Department bagian 1

Pada file php ini merupakan file yang berfokus untuk menambah data pada tabel department. Pada awal file php ini membutuhkan file fungsi department sebelumnya. Akan muncul notifikasi nantinya jika penambahan data berhasil atau gagal.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <title>Modul 3</title>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col">
          <h1>Tambah Data Department</h1>
          <a href="department.php">
            <div class="div btn btn-primary mb-4">
              kembali
            </div>
          </a>
          <form action="" method="post">
            <ul>
              <li class="mb-2">
                <label for="nama_depart">Nama : </label>
                <input type="text" name="nama_depart" id="nama_depart" required>
              </li>

```

Gambar 2.18 Tambah Department bagian 2

```

              <li class="mt-2">
                <button type="submit" name="submit">Tambah Data!</button>
              </li>
            </ul>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>

```

Gambar 2.19 Tambah Department bagian 3

Untuk bagian htmlnya adalah untuk tampilan saat ingin menambah data department seperti nama department. Lalu ada tombol kembali untuk ke tampilan awal data department. Lalu setelah selesai menambah tinggal menekan tombol tambah data.

2.2.9 department_hapus.php

```
1?php
require 'functiondepartment.php';

$id = $_GET["id"];
if(hapus($id) > 0 ){
    echo "
        <script>
            alert('data berhasil dihapus!');
            document.location.href = 'department.php';
        </script>
    ";
} else {
    echo "
        <script>
            alert('data gagal dihapus!');
            document.location.href = 'department.php';
        </script>
    ";
}
```

Gambar 2.20 Hapus Department

Untuk penghapusan data department, file ini berguna untuk mendapatkan id yang akan dihapus dan menampilkan notifikasi saat data berhasil dihapus atau gagal dihapus.

2.2.10 Hasil

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Budi	19	Marketing	ubah hapus
3	Andi	20	HRD	ubah hapus

Gambar 2.21 Tampil Employee

Tambah Data

Kembali Ke Halaman Employee

No.	Nama	Action
1	Administrasi	hapus
2	Marketing	hapus
3	HRD	hapus

Gambar 2.22 Tampil Department

Tambah Data Employee

[kembali](#)

- Nama :
- id_department
-

Gambar 2.23 Tambah Data Employee

Tambah Data

Ke Halaman Department

No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Budi	19	Marketing	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.24 Setelah Tambah Data Employee

Ubah Data Employee

[Kembali](#)

- Nama :
- ID :
- id_department
- [Tambah Data!](#)

Gambar 2.25 Ubah Data Employee

[Tambah Data](#) [Ke Halaman Department](#)

No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Sandy	19	HRD	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.26 Setelah Ubah Data Employee

[Tambah Data](#) [Ke Halaman Department](#)

localhost says
Yakin?
[OK](#) [Cancel](#)

No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Sandy	19	HRD	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.26 Hapus Data Employee

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Alex	21	Administrasi	ubah hapus
3	Sandy	19	HRD	ubah hapus

Gambar 2.27 Setelah Hapus Data Employee

Tambah Data Department

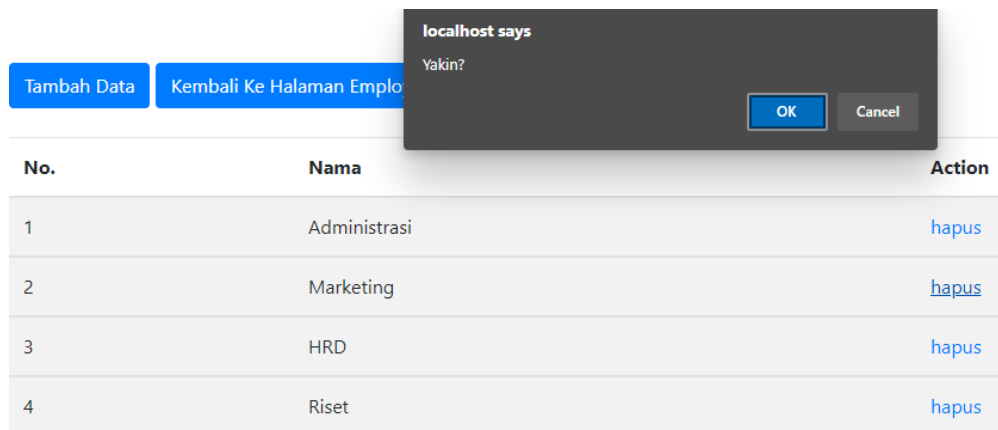
kembali

- Nama :
-

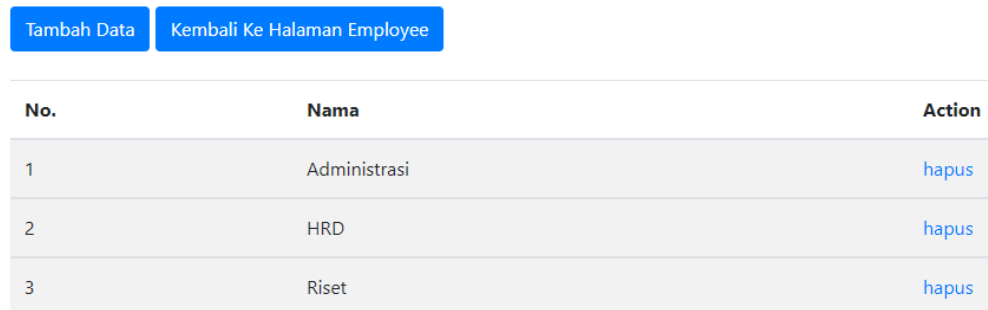
Gambar 2.28 Tambah Data Department

Tambah Data		Kembali Ke Halaman Employee	
No.	Nama	Action	
1	Administrasi	hapus	
2	Marketing	hapus	
3	HRD	hapus	
4	Riset	hapus	

Gambar 2.29 Setelah Tambah Data Department



Gambar 2.30 Hapus Data Department



Gambar 2.31 Setelah Hapus Data Department

BAB III

KESIMPULAN

Pembuatan database dalam menggunakan file php dapat dilakukan, namun lebih baik membuat database pada aplikasi yang biasanya digunakan untuk menggunakan database. Lalu, jika ingin menggunakannya tinggal mengkoneksikan MySQL pada php yang kita buat. Query yang kita lakukan pada php pun dapat dibaca MySQL jika kita menggunakan koneksinya dengan tepat.

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>
- Praktikum. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

LAMPIRAN

```
<?php
require 'functionemployee.php';
$employs = query("SELECT * FROM employee INNER JOIN department ON employee.id_department=department.id");
?>

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <title>Modul 3</title>
  </head>
```

Gambar 2.1 Employee bagian 1

```
<body>

  <div class="container">
    <div class="row">
      <div class="col mt-5">
        <a href="employee_tambah.php">
          <div class="div btn btn-primary mb-4">
            Tambah Data
          </div>
        </a>

        <a href="department.php">
          <div class="div btn btn-primary mb-4">
            Ke Halaman Department
          </div>
        </a>

        <table class="table table-striped">
          <thead>
            <tr>
              <th scope="col">No.</th>
              <th scope="col">Nama</th>
              <th scope="col">ID</th>
              <th scope="col">Department</th>
              <th scope="col">Action</th>
            </tr>
          </thead>
```

Gambar 2.2 Employee bagian 2

```
<tbody>
<tr>
<?php $i=1; ?>
<?php foreach($employs as $employee) : ?>
<td><?php echo $i; ?></td>
<td><?php echo $employee["nama"]; ?></td>
<td><?php echo $employee["id_employee"]; ?></td>
<td><?php echo $employee["nama_department"]; ?></td>
<td>
<a href="employee_ubah.php?id=<?php echo $employee["id_employee"]; ?>">ubah</a> |
<a href="employee_delete.php?id=<?php echo $employee["id_employee"]; ?>" onclick="return confirm('Yakin?');">hapus</a>
</td>
</tr>
</tbody>
<?php $i++; ?>
<?php endforeach; ?>
</table>
</div>
</div>
</div>
</body>
</html>
```

Gambar 2.3 Employee bagian 3

```
<?php
require 'functiondepartment.php';
$departs = query("SELECT * FROM department");
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
</head>
<title>Modul 3</title>
</html>
```

Gambar 2.4 Department bagian 1

```

<body>
  <div class="container">
    <div class="row">
      <div class="col mt-5">
        <a href="department_tambah.php">
          <div class="div btn btn-primary mb-4">
            | Tambah Data
          </div>
        </a>

        <a href="employee.php">
          <div class="div btn btn-primary mb-4">
            | Kembali Ke Halaman Employee
          </div>
        </a>

        <table class="table table-striped">
          <thead>
            <tr>
              <th scope="col">No.</th>
              <th scope="col">Nama</th>
              <th scope="col">Action</th>
            </tr>
          </thead>

```

Gambar 2.5 Department bagian 2

```

<tbody>
  <tr>
    <?php $i=1; ?>
    <?php foreach($depart as $depart) : ?>
      <td><?php echo $i; ?></td>
      <td><?php echo $depart["nama_department"]; ?></td>
      <td>
        <a href="department_delete.php?id=<?php echo $depart["id"]; ?>" onclick="return confirm('Yakin?');">hapus</a>
      </td>
    </tr>
  </tbody>
</table>
<?php $i++; ?>
<?php endforeach; ?>
</div>
</div>
</div>
</body>
</html>

```

Gambar 2.6 Department bagian 3

```

<?php
$db = mysqli_connect("localhost", "root", "", "modul3");

function query($query){
    global $db;
    $result = mysqli_query($db, $query);
    $rows = [];
    while($row = mysqli_fetch_assoc($result)){
        $rows[]=$row;
    }
    return $rows;
}

function tambah($data){
    global $db;
    // ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $id_department = htmlspecialchars($data["id_department"]);

    $query = "INSERT INTO employee
    VALUES ('', '$nama', '$id_department')
    ";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

```

Gambar 2.7 Fungsi Employee bagian 1

```

function ubah($data){
    global $db;
    $id_employee = $data["id_employee"];
    // ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $id_department = htmlspecialchars($data["id_department"]);
    $query = "UPDATE employee
    SET nama = '$nama', id_department = '$id_department'
    WHERE id_employee = $id_employee
    ";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

function hapus($id){
    global $db;
    mysqli_query($db, "DELETE FROM employee WHERE id_employee = $id");
    return mysqli_affected_rows($db);
}
?>

```

Gambar 2.8 Fungsi Employee bagian 2

```

<?php
$db = mysqli_connect("localhost", "root", "", "modul3");

function query($query){
    global $db;
    $result = mysqli_query($db, $query);
    $rows = [];
    while($row = mysqli_fetch_assoc($result)){
        $rows[]=$row;
    }
    return $rows;
}

function tambah($data){
    global $db;
    // ambil data dari tiap elemen dalam form
    $nama_depart = htmlspecialchars($data["nama_depart"]);

    $query = "INSERT INTO department
    VALUES ('', '$nama_depart')
    ";
    // query insert data
    mysqli_query($db, $query);

    return mysqli_affected_rows($db);
}

function hapus($id){
    global $db;
    mysqli_query($db, "DELETE FROM department WHERE id = $id");
    return mysqli_affected_rows($db);
}
?>

```

Gambar 2.9 Fungsi Department

```

<?php
    require 'functionemployee.php';

    // ambil data di URL
    $id = $_GET["id"];

    // query data mahasiswa berdasarkan ID
    $employee = query("SELECT * FROM employee
        INNER JOIN department
        ON employee.id_department=department.id
        WHERE employee.id_employee = $id")[0];
    $departs = query("SELECT * FROM department");

    // cek tombol submit sdh ditekan atau belum
    if(isset($_POST["submit"])) {
        // cek apakah data berhasil di ubah atau tidak
        if (ubah($_POST)>0){
            echo "
                <script>
                    alert('data berhasil diubah!');
                    document.location.href = 'employee.php';
                </script>
            ";
        } else {
            echo "
                <script>
                    alert('data gagal diubah!');
                    document.location.href = 'employee.php';
                </script>
            ";
        }
    }
}
?>

```

Gambar 2.10 Ubah Employee bagian 1

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoGXMA+0sXRXPxPg6fy4IWVTNh0E263XmFcJlSAwiGgFAW/dAi16GJXm" crossorigin="anonymous">
    <title>Modul 3</title>
</head>
<body>

    <div class="container">
        <div class="row">
            <div class="col">
                <h1>Ubah Data Employee</h1>
                <a href="employee.php">
                    <div class="div btn btn-primary mb-4">
                        Kembali
                    </div>
                </a>

                <form action="" method="post">
                    <ul>
                        <li>
                            <label for="nama">Nama : </label>
                            <input type="text" name="nama" id="nama" required value="<?php echo $employee["nama"]; ?>">
                        </li>

                        <li>
                            <label for="nama">ID : </label>
                            <input type="text" name="id_employee" id="id_employee" required value="<?php echo $employee["id_employee"]; ?>">
                        </li>
                    </ul>
                </form>
            </div>
        </div>
    </div>

```

Gambar 2.11 Ubah Employee bagian 2

```

<li class="mb-2">
  <div class="form-group">
    <label for="id_department">id_department</label>
    <select name="id_department" class="form-control" id="id_department">
      <option value="{?php echo $employee["id_department"]; ?}" selected="{?php echo $employee["nama_department"]};
      <?php foreach($departs as $depart) : ?>
        <option value="{?=$depart["id"] ?}">{?=$depart["nama_department"] ?}</option>
      <?php endforeach; ?>
    </select>
  </div>
</li>

<li class="mt-2">
  <button type="submit" name="submit">Tambah Data!</button>
</li>
</ul>
</div>
</form>
</div>
</div>
</body>
</html>

```

Gambar 2.12 Ubah Employee bagian 3

```

<?php
require 'functionemployee.php';

$departs = query("SELECT * FROM department");

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
  // cek apakah data berhasil di tambahkan atau tidak
  if (tambah($_POST)>0){
    echo "
    <script>
      alert('data berhasil ditambahkan!');
      document.location.href = 'employee_tambah.php';
    </script>
    ";
  } else {
    echo "
    <script>
      alert('data gagal ditambahkan!');
      document.location.href = 'employee_tambah.php';
    </script>
    ";
  }
}
}
?>

```

Gambar 2.13 Tambah Employee bagian 1


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
    integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

  <title>Modul 3</title>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Tambah Data Employee</h1>
        <a href="employee.php">
          <div class="div btn btn-primary mb-4">
            kembali
          </div>
        </a>

        <form action="" method="post">
          <ul>
            <li class="mb-2">
              <label for="nama">Nama : </label>
              <input type="text" name="nama" id="nama" required>
            </li>

```

Gambar 2.14 Tambah Employee bagian 2

```

            <li class="mb-2">
              <div class="form-group">
                <label for="id_department ">id_department</label>
                <select name="id_department" class="form-control" id="id_department ">
                  <option selected>Choose One</option>
                  <?php foreach($departments as $depart) : ?>
                    <option value="<? $depart["id"] ?>"><? $depart["nama_department"] ?></option>
                  </select>
                </div>
              </li>

              <li class="mt-2">
                <button type="submit" name="submit">Tambah Data!</button>
              </li>
            </ul>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>

```

Gambar 2.15 Tambah Employee bagian 3

```

<?php
require 'functionemployee.php';

$id = $_GET["id"];
if(hapus($id) > 0 ){
    echo "
        <script>
            alert('data berhasil dihapus!');
            document.location.href = 'employee.php';
        </script>
    ";
} else {
    echo "
        <script>
            alert('data gagal dihapus!');
            document.location.href = 'employee.php';
        </script>
    ";
}
}

```

Gambar 2.16 Hapus Employee

```

<?php
require 'functiondepartment.php';

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
            <script>
                alert('data berhasil ditambahkan!');
                document.location.href = 'department_tambah.php';
            </script>
        ";
    } else {
        echo "
            <script>
                alert('data gagal ditambahkan!');
                document.location.href = 'department_tambah.php';
            </script>
        ";
    }
}
}

```

Gambar 2.17 Tambah Department bagian 1

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
    integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
  <title>Modul 3</title>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Tambah Data Department</h1>
        <a href="department.php">
          <div class="div btn btn-primary mb-4">
            kembali
          </div>
        </a>
        <form action="" method="post">
          <ul>
            <li class="mb-2">
              <label for="nama_depart">Nama : </label>
              <input type="text" name="nama_depart" id="nama_depart" required>
            </li>

```

Gambar 2.18 Tambah Department bagian 2

```

            <li class="mt-2">
              <button type="submit" name="submit">Tambah Data!</button>
            </li>
          </ul>
        </form>
      </div>
    </div>
  </div>
</body>
</html>

```

Gambar 2.19 Tambah Department bagian 3

```

?php
require 'functiondepartment.php';

$id = $_GET["id"];
if(hapus($id) > 0 ){
    echo "
        <script>
            alert('data berhasil dihapus!');
            document.location.href = 'department.php';
        </script>
    ";
} else {
    echo "
        <script>
            alert('data gagal dihapus!');
            document.location.href = 'department.php';
        </script>
    ";
}
}
?>

```

Gambar 2.20 Hapus Department

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Budi	19	Marketing	ubah hapus
3	Andi	20	HRD	ubah hapus

Gambar 2.21 Tampil Employee

Tambah Data		Kembali Ke Halaman Employee		
No.	Nama	Action		
1	Administrasi	hapus		
2	Marketing	hapus		
3	HRD	hapus		

Gambar 2.22 Tampil Department

Tambah Data Employee

[kembali](#)

- Nama :
- id_department
-

Gambar 2.23 Tambah Data Employee

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Budi	19	Marketing	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.24 Setelah Tambah Data Employee

Ubah Data Employee

Kembali

- Nama :
- ID :
- id_department
-

Gambar 2.25 Ubah Data Employee

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Sandy	19	HRD	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.26 Setelah Ubah Data Employee

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Sandy	19	HRD	ubah hapus
3	Andi	20	HRD	ubah hapus
4	Alex	21	Administrasi	ubah hapus

Gambar 2.26 Hapus Data Employee

Tambah Data		Ke Halaman Department		
No.	Nama	ID	Department	Action
1	Victor	18	Administrasi	ubah hapus
2	Alex	21	Administrasi	ubah hapus
3	Sandy	19	HRD	ubah hapus

Gambar 2.27 Setelah Hapus Data Employee

Tambah Data Department

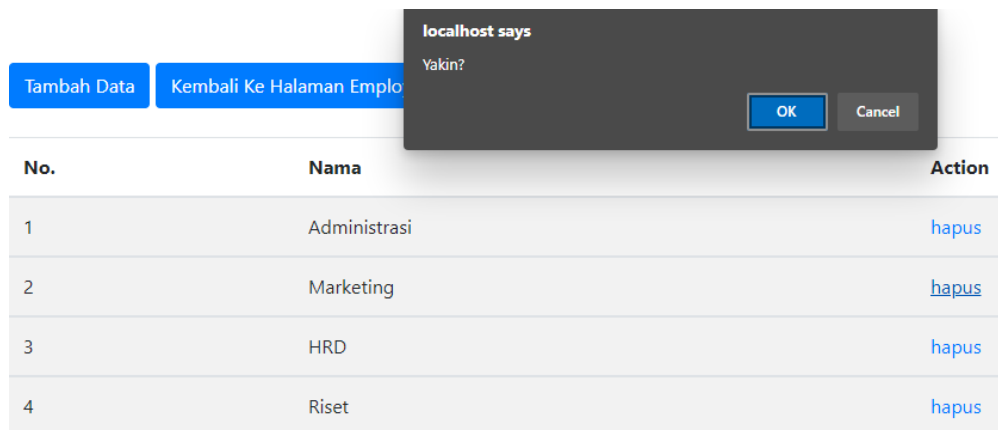
kembali

- Nama :
-

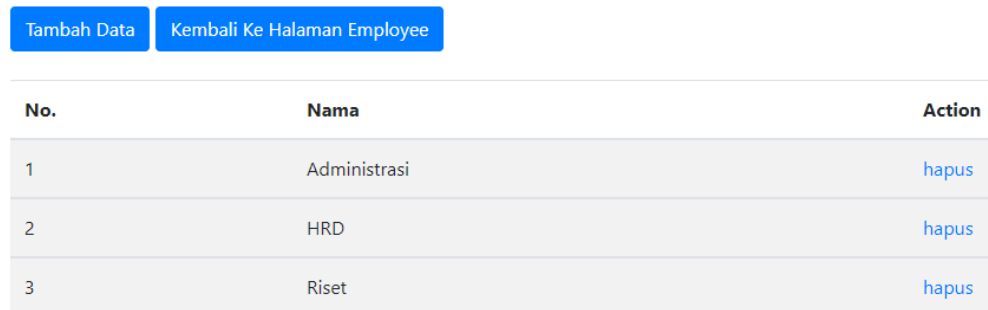
Gambar 2.28 Tambah Data Department

Tambah Data		Kembali Ke Halaman Employee	
No.	Nama	Action	
1	Administrasi	hapus	
2	Marketing	hapus	
3	HRD	hapus	
4	Riset	hapus	

Gambar 2.29 Setelah Tambah Data Department



Gambar 2.30 Hapus Data Department



Gambar 2.31 Setelah Hapus Data Department