

LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Kevin Jeremia S.P
NIM : 193020503034
KELAS : A
MODUL : III (PHP MySQL Database)

JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

1.1 Tujuan Praktikum

- 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

Data dalam database MySQL disimpan dalam tabel-tabel. Sebuah tabel adalah koleksi dari data yang berelasi dan mengandung kolom dan baris. Database sangat bermanfaat untuk menyimpan informasi secara kategori. Contoh yang akan diberikan pada modul praktikum ini adalah tabel yang mengandung data Employees (Pekerja), Products (Produk), Customers (Pelanggan) dan Orders (Pesanan).

Membuka Koneksi

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

```

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

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>

```

Gambar 1.1 Koneksi

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

```

<?php

    $servername = "localhost";
    $username = "username";
    $password = "password";

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

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

```

Gambar 1.2 Koneksi

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut: **mysqli_close(\$conn);**

Membuat Database

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

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

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

    // Membuat Database
    $sql = "CREATE DATABASE myDB";
    if (mysqli_query($conn, $sql)) {
```

Gambar 1,3 Membuat Database

```
        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

Gambar 1.4 Membuat Database

Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

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

Berikut adalah contoh program untuk membuat tabel diatas:

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

Gambar 1.5 Membuat Tabel

```
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY  
KEY, firstname VARCHAR(30) NOT NULL, lastname  
VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP  
);  
  
if (mysqli_query($conn, $sql)) {  
    echo "Table MyGuests created successfully";  
} else {  
    echo "Error creating table: " . mysqli_error($conn);  
}  
  
mysqli_close($conn);  
?>
```

Gambar 1.6 Membuat Tabel

Memasukkan Data ke Database

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

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

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:
INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)

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

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

```
$dbname = "myDB";

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

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);

?>
```

Gambar 1.7 Memasukan Data

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

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

```

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){

```

Gambar 1.8 Memasukan Data

```

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

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Mary', 'Moe', 'mary@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Julie', 'Dooley', 'julie@example.com');";

    if (mysqli_multi_query($conn, $sql)) {
        echo "New records created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.9 Memasukan Data

Mengambil Data dari

Database Untuk mengambil data, statement SQL yang digunakan adalah `SELECT nama_kolom() FROM nama_tabel`, atau kita bisa menggunakan karakter `*` untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan statement `WHERE` setelah `nama_tabel`. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:


```

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

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

```

Gambar 1.10 Mengambil Data

```

$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```

Gambar 1.11 Mengambil Data

Menghapus Data

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

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

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

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

Gambar 1.12 Menghapus

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

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

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

mysqli_close($conn);
?>
```

Gambar 1.13 Menghapus

Update Data dalam Database

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

UPDATE nama_tabel

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

WHERE kolom_penentu=nilai_penentu ...

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

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){
```

Gambar 1.14 Update Data

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

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

    if (mysqli_query($conn, $sql)) {
        echo "Record updated successfully";
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

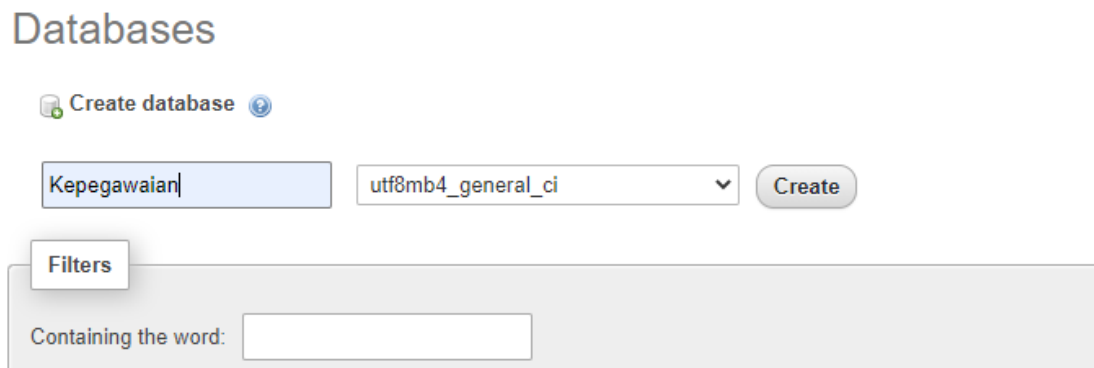
Gambar 1.15 Update Data

BAB II

PEMBAHASAN

2.1 Buatlah database pada MySql untuk menyimpan data data pegawai beserta relasi table nya.

Di soal yang pertama ini diminta untuk membuat data base. Data base yang akan di buat akan meyimpan data data dari pegawai. Di dalam database ini akan memiliki 2 tabel. Table pertama adalah table Pegawai yang memiliki fungsi menyimpan data pegawai. Lalu table kedua adalah table departemen yang memiliki fungsi untuk menyimpan data dari department. Data base akan di buat di melalui phpMyAdmin. Berikut ini adalah pembuatan nya.



Gambar 2.1 Membuat database

Database yang akan di buat di beri nama sebagai database Kepegawaian. Dimana database ini akan meyimpan data pegawai, seperti nama, department, id pegawai dll. Di dalam database ini terdapat dua buah table yaitu table employe dan table deparment. Dimana kedua table ini akan berelasi. Berikut ini cara pembuatan kolom di dalam phpMyAdmin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_employe	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	id_department	int(11)			Yes	NULL			Change Drop More

Gambar 2.2 Tabel Employee

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_depart	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 2.3 Tabel Department

Tabel sudah sudah dibuat dimana table Pegawai ini berelasi dengan table jabatan. Tabel pegawai memiliki primary key id_pegawai. Lalu karena table pegawai ini berelasi dengan table jabatan maka id_jabatan yang terdapat table pegawai adalah sebuah foreign key. Untuk nama dan jabatan menggunakan tipe data varchar dan untuk tipe data primary key menggunakan int. Setelah itu memasukkan data kedalam database dan berikut adalah data yang sudah di masukan ke dalam database nya.

✓ Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

```
SELECT * FROM `employee`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key

+ Options

	id_employee	nama	id_department
<input type="checkbox"/> Edit Copy Delete	1	KEVIN	1
<input type="checkbox"/> Edit Copy Delete	2	Rio	2
<input type="checkbox"/> Edit Copy Delete	3	Abi	1

↑ ☐ Check all | With selected: Edit Copy Delete Export

Gambar 2.4 Isi Tabel Employee

✓ Showing rows 0 - 3 (4 total, Query took 0.0008 seconds.)

```
SELECT * FROM `department`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: N

+ Options

	id	nama_depart
<input type="checkbox"/> Edit Copy Delete	1	Pengembang
<input type="checkbox"/> Edit Copy Delete	2	Alat
<input type="checkbox"/> Edit Copy Delete	3	Survey
<input type="checkbox"/> Edit Copy Delete	4	Kesehatan

↑ ☐ Check all | With selected: Edit Copy Delete Export

Gambar 2.5 Isi Tabel Department

2.2 Buatlah Program untuk menyimpan, menghapus, dan mengubah data – data pegawai tersebut.

Selanjutnya di nomor kedua ini diminta untuk membuat program yang akan berfungsi untuk menyimpan, menghapus dan mengubah data. Program yang akan di gunakan akan berbasis web. Oleh sebab itulah akan di gunakan Bahasa PHP. Cara kerja program ini nantinya program dapat menerima inputan, lalu merekam dan menyimpan nya di dalam database kepegawaian. Selain menginput data program ini menghapus serta mengubah data yang telah di rekam.

Program ini akan menggunakan 4 file PHP. Setiap file mempunyai peran nya masing masing dalam menjalankan tugasnya. Pembuatan 4 file secara terpisah ini dipilih agar Ketika baris program di jalankan tidak ada baris program yang tabrakan. Dan berikut ini adalah penjelasn dari setiap file nya . Yang pertama adalah function.php dibawah ini :



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

Gambar 2.6 Function.php

Dapat dilihat di gambar 2.6 adalah gambar dari function.php. Function ini diawali oleh tag php. Setelah ini pendeklarasian variable. \$servername merupakan variabel dari server yang di gunakan database yaitu "localhost". \$database adalah variable dari database yang digunakan yaitu kepegawaian. Dan \$username dan

\$password merupakan variabel dari username dan password perangkat. Setelah semua variabel di deklarsi selanjutnya di tahap koneksi, Variable yang di gunakan sebagai koneksi adalah \$conn yang berisi perintah mysqli_connect denga seluruh variable diatas untuk mengkonekan nya ke database. Selanjutnya pembahasan lanjutan bari program function.php

```
7
8  function show($query){
9      global $conn;
10     $result = mysqli_query($conn, $query);
11     $rows = [];
12     while( $row = mysqli_fetch_assoc($result) ){
13         $rows[] = $row;
14     }return $rows;
15 }
16
17 function insert($data){
18     global $conn;
19     $id_employe = $_POST["id_employe"];
20     $nama = $_POST["nama"];
21     $id_department = $_POST["id_department$id_department"];
22     mysqli_query($conn, "INSERT INTO employee VALUES('$id_employe','$nama','$id_department')");
23     return mysqli_affected_rows($conn);
24 }
25
26 function delete($data){
27     global $conn;
28     $id_employe = $_GET["delete"];
29     mysqli_query($conn, "DELETE FROM employee WHERE id_employe = $id_employe");
30     return mysqli_affected_rows($conn);
31 }
32
33 function update($data){
34     global $conn;
35     $id_employe = $_POST["id_employe"];
36     $nama = $_POST["nama"];
37     $id_department = $_POST["id_department$id_department"];
38     mysqli_query($conn, "UPDATE employee SET nama = '$nama',
39     id_department = '$id_department' WHERE id_employe = $id_employe");
40     return mysqli_affected_rows($conn);
41 }
42 ?>
```

Gambar 2.7 Lanjutan Function.php

Fungsi yang akan dibahas pertama adalah fungsi show. Show memiliki fungsi untuk menyimpan seluruh data di dalam database. Data yang disimpan di masukan ke dalam array. Dan di lakukan pengulangan untuk menambahkan kolom

agar data baru juga dapat di simpan, fungsi ini memiliki nilai balikan \$rows yang akan mengembalikan nilai array.

Selanjutnya adalah fungsi insert. Insert di gunakan untuk menginputkan data para pegawai kedalam database kepegawaan. Terdapat global \$conn yang merupakan variabel global dari koneksi.terdapat variable \$id_employe, \$nama, \$id_department yang berfungsi menyimpan data Ketika di insert. Lalu data yang di simpan akan di lakukan query di mysqli_query dengan perintah insert into dan variable \$id_employe, \$nama, \$id_departmen. Lalu akan membalikan nilai mysqli_query yang disimpan di dalam data base.

Fungsi selanjutnya adalah delete. Fungsi ini di buat untuk menghapus data yang tedapat di database. Dimana \$id_employe yang berisi delete. Lalu mysqli_query untuk melakukan query. Query yang akan di lakukan adalah delete from employee where id_employe = \$id_employe. Dimana Ketika perintah delete di lakukan maka adata yang akan diplih akan otomatis terhapus dari dalam database.

Dan fungsi terakhir yang terdapat di function.php ini adalah fungsi update. Update berfungsi untk mengupdate data yang terdapat di database. \$id_employe, \$nama, \$id_department berfungsi untuk meyimpan data yang ingi di update. Lalu data yang di simpan akan di lakukan query di mysqli_query dengan perintah update employe set nama, id department dengan kondisi id_employee. Ketika data yang ingin di update di masukan otomatis akan di update di dalam database .

Selanjutnya akan menjelaskan file selanjutnya. File yang akan dijlaskan adalah insert.php. File ini memiliki fungsi untuk membuat tampilan yang dapat untuk menginputkan nya kedalam database. Berikut ini adalah penjelasan dari baris program nya.

```

function.php x insert.php
insert.php
1 <?php
2 require("function.php");
3 if (isset($_POST["insert"])) {
4     if (isset($_POST["id_employee"]) &&
5         isset($_POST["nama"]) &&
6         isset($_POST["id_department"])) {
7         echo "<script>";
8         alert('data berhasil ditambah');
9         document.location.href = 'index.php';
10        </script>";
11    } else {
12        echo "<script>";
13        alert('data gagal ditambah');
14        document.location.href = 'index.php';
15        </script>";
16    }
17 }
18 <?>
19 <!DOCTYPE html>
20 <html lang="en">

```

Gambar 2.8 Insert.php

Bagian pertama dari baris kode ini adalah file ini require dari file function.php. Terdapat percabangan dimana Ketika ada data yang masuk maka akan menampilkan pop out ke layar data berhasil. Dan Ketika kondisi sebaliknya maka akan menampilkan data gagal di tambah.

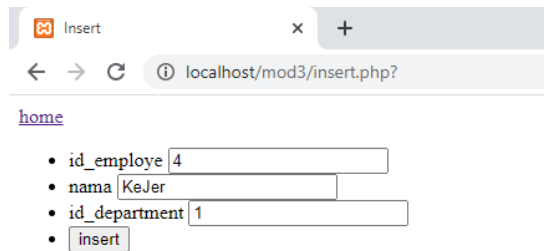
```

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

```

Gambar 2.9 Lanjutan Insert.php

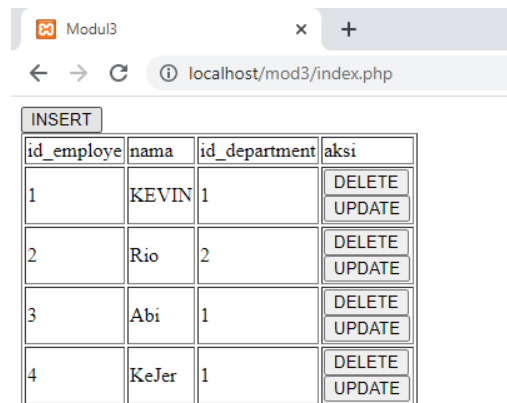
Selanjutnya adalah baris program untuk menampilkannya di web. Menggunakan ui dan li untuk membuat list. Lalu membuat label, label di gunakan untuk measukan data. Dimana setiap id nya haruslah sesuai dengan data yang ingin kita masukan , seperti id_employe, nama dan id _department. Dan berikut ini adalah hasil tampilannya di web dan perubahan data base nya .



home

- id_employe 4
- nama KeJer
- id_department 1
- insert

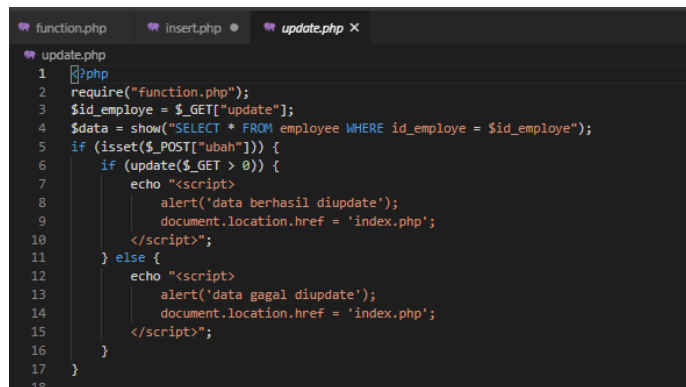
Gambar 2.10 Insert di web



id_employe	nama	id_department	aksi
1	KEVIN	1	DELETE UPDATE
2	Rio	2	DELETE UPDATE
3	Abi	1	DELETE UPDATE
4	KeJer	1	DELETE UPDATE

Gambar 2.11 Database yang di tambahkan

Selanjutnya akan menjelaskan file selanjutnya. File yang akan dijlaskan adalah update.php. File ini memiliki fungsi untuk membuat tampilan update data di database. Berikut ini adalah penjelasan dari baris program nya



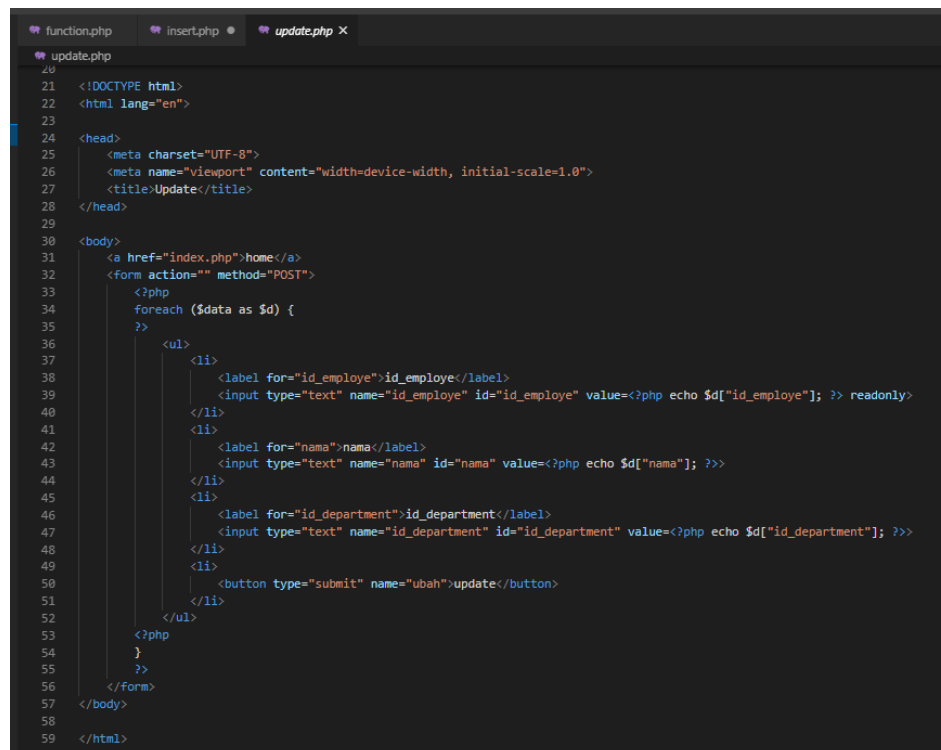
```

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

```

Gambar 2.12 Update.php

Bagian pertama dari baris kode ini adalah file ini require dari file function.php. Terdapat percabangan dimana Ketika ada data yang masuk maka akan menampilkan pop out ke layar data berhasil. Dimana terdapat select*from employe untuk menentukan data nya. Dan ketika salah maka data gagal di update



```

20
21 <!DOCTYPE html>
22 <html lang="en">
23
24 <head>
25     <meta charset="UTF-8">
26     <meta name="viewport" content="width=device-width, initial-scale=1.0">
27     <title>Update</title>
28 </head>
29
30 <body>
31     <a href="index.php">home</a>
32     <form action="" method="POST">
33         <?php
34         foreach ($data as $d) {
35             <?>
36             <ul>
37                 <li>
38                     <label for="id_employe">id_employe</label>
39                     <input type="text" name="id_employe" id="id_employe" value="<?php echo $d["id_employe"]; >> readonly>
40                 </li>
41                 <li>
42                     <label for="nama">nama</label>
43                     <input type="text" name="nama" id="nama" value="<?php echo $d["nama"]; >>
44                 </li>
45                 <li>
46                     <label for="id_department">id_department</label>
47                     <input type="text" name="id_department" id="id_department" value="<?php echo $d["id_department"]; >>
48                 </li>
49                 <li>
50                     <button type="submit" name="ubah">update</button>
51                 </li>
52             </ul>
53             <?php
54             }
55         <?>
56     </form>
57 </body>
58
59 </html>

```

Gambar 2.13 Lanjutan Update.php

Gambar 2.13 adalah lanjutan baris program. untuk menampilkannya di web. Menggunakan ui dan li untuk membuat list. Lalu membuat label, label di gunakan untuk memasukan data. Dimana setiap id nya haruslah sesuai dengan data yang ingin kita masukan , seperti id_employe, nama dan id _department.Data yang dimasukan di gunakan untuk mengupdate data yang terdapat di basis data. Dan berikut ini adalah hasil tampilannya di web dan perubahan data base nya .

Update

localhost/mod3/update.php?update=1

[home](#)

- id_employe 1
- nama Mitchel
- id_department 1
-

Gambar 2.14 Update data di web

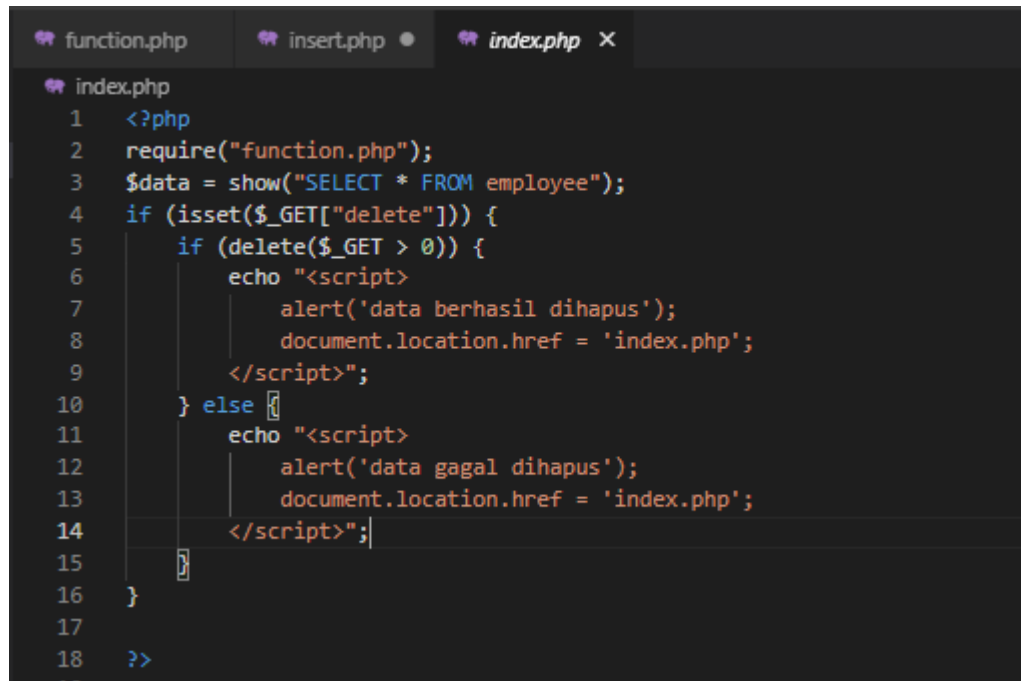
Modul3

localhost/mod3/index.php

id_employe	nama	id_department	aksi
1	Mitchel	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	Rio	2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	Abi	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
4	KeJer	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.15 data yang sudah di update

File selanjutnya adalah index.php. File ini memiliki fungsi sebagai halaman utama. Halaman ini di gunakan untuk menampilkan basis data, dari table employe. Berikut ini adalah baris program nya.



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

Gambar 2.16 index.php

Bagian pertama dari baris kode ini adalah file ini require dari file function.php. Terdapat percabangan dimana Ketika ada data yang masuk maka akan menampilkan pop out ke layar data berhasil. Dimana terdapat select*from employe untuk menentukan data nya. Dan ketika salah maka data gagal di update.

```

19
20 <!DOCTYPE html>
21 <html lang="en">
22
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Modul3</title>
27 </head>
28
29 <body>
30     <form action="insert.php">
31         <button>INSERT</button>
32     </form>
33     <table border="1">
34         <tr>
35             <td>id_employe</td>
36             <td>nama</td>
37             <td>id_department</td>
38             <td>aksi</td>
39         </tr>
40         <?php
41         foreach ($data as $d) {
42             >
43                 <tr>
44                     <td><?php echo $d["id_employe"]; ?></td>
45                     <td><?php echo $d["nama"]; ?></td>
46                     <td><?php echo $d["id_department"]; ?></td>
47                     <td>
48                         <form action="" method="GET">
49                             <button type="submit" name="delete" value=<?php echo $d["id_employe"]; ?>>DELETE</button>
50                         </form>
51                         <form action="update.php" method="GET">
52                             <button type="submit" name="update" value=<?php echo $d["id_employe"]; ?>>UPDATE</button>
53                         </form>
54                     </td>
55                 </tr>
56             <?php
57             }
58             >
59         </table>
60 </body>
61 </html>

```

Gambar 2.17 Lanjutan index.php

Untuk menampilkan database menggunakan tag tr dan td untuk membuat table. Lalu mengechi id_employe, nama, dan id_department. Setelah itu terdapat baris kode untuk membuat button, untuk delete. Dimana Ketika button delete di tekan akan menjalankan fungsi delete di fuction.php, dan akan menjalankan query hapus. Berikut ada hasil nya .

Modul3 x +

localhost/mod3/index.php

INSERT

id_employe	nama	id_department	aksi
1	Mitchel	1	DELETE UPDATE
2	Rio	2	DELETE UPDATE
4	KeJer	1	DELETE UPDATE

Gambar 2.18 tampilan table di web

BAB III

KESIMPULAN

Dari praktikum yang telah di lakukan ini dapat di simpulan bahwa database di gunakan untuk meyimpan data. Selain itu Database dapat di hubungkan dengan web melalu Bahasa PHP, sehingga untuk menampilkan data dari dalam database dapat di lakukan di program web. Dan pada Bahasa PHP bisa menggunakan query sql sehingga dapat melakukan perintah untuk memanipulasi data di database seperti menghapus, mengupdate, menambahkan data.

BAB IV

DAFTAR PUSTAKA

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

LAMPIRAN

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

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Gambar 1.1 Koneksi

```
<?php

    $servername = "localhost";
    $username = "username";
    $password = "password";

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

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

Gambar 1.2 Koneksi

```

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

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

    // Membuat Database
    $sql = "CREATE DATABASE myDB";
    if (mysqli_query($conn, $sql)) {

```

Gambar 1,3 Membuat Database

```

        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.4 Membuat Database

```

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

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

    // SQL untuk membuat tabel
    $sql = "CREATE TABLE MyGuests (

```

Gambar 1.5 Membuat Tabel

```

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

    );

    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.6 Membuat Tabel

```

$dbname = "myDB";

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

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

Gambar 1.7 Memasukan Data

```

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){

```

Gambar 1.8 Memasukan Data

```

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

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Mary', 'Moe', 'mary@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Julie', 'Dooley', 'julie@example.com');";

    if (mysqli_multi_query($conn, $sql)) {
        echo "New records created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.9 Memasukan Data

```

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

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

```

Gambar 1.10 Mengambil Data

```

$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```

Gambar 1.11 Mengambil Data

```

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

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

```

Gambar 1.12 Menghapus

```

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

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

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

mysqli_close($conn);
?>

```

Gambar 1.13 Menghapus

```

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

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){

```

Gambar 1.14 Update Data

```

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

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

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

Gambar 1.15 Update Data

Databases

Create database ?

Kepegawaian

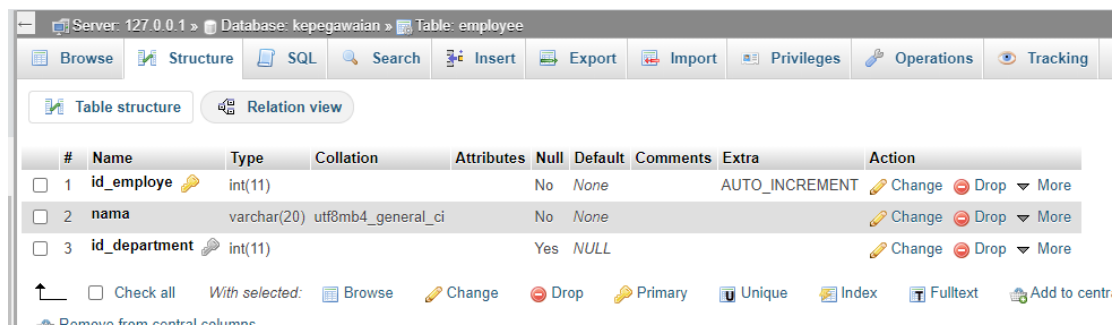
utf8mb4_general_ci

Create

Filters

Containing the word:

Gambar 2.1 Membuat database



Server: 127.0.0.1 » Database: kepegawaian » Table: employee

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking

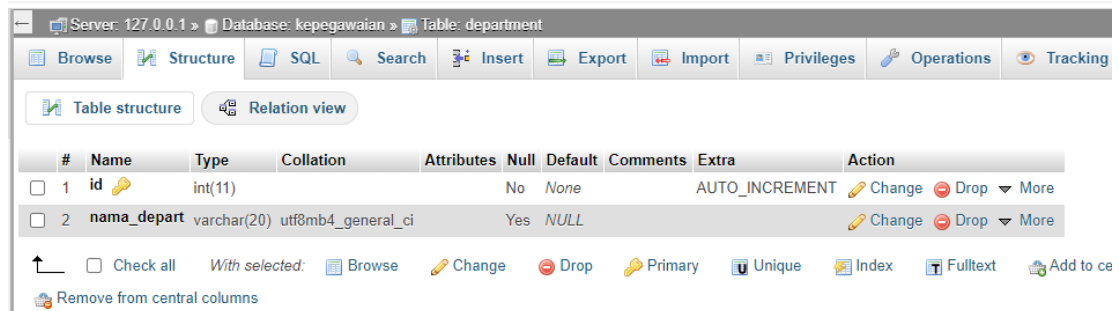
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_employe	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama	varchar(20) utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3	id_department	int(11)		Yes	NULL			Change Drop More

[Check all](#) [With selected:](#) [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to center](#)

[Remove from central columns](#)

Gambar 2.2 Tabel Employee



Server: 127.0.0.1 » Database: kepegawaian » Table: department

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama_depart	varchar(20) utf8mb4_general_ci		Yes	NULL			Change Drop More

[Check all](#) [With selected:](#) [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to center](#)

[Remove from central columns](#)

Gambar 2.3 Tabel Department

✓ Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

```
SELECT * FROM `employee`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key

+ Options

	id_employee	nama	id_department
<input type="checkbox"/> Edit Copy Delete	1	KEVIN	1
<input type="checkbox"/> Edit Copy Delete	2	Rio	2
<input type="checkbox"/> Edit Copy Delete	3	Abi	1

↑ ☐ Check all | With selected: Edit Copy Delete Export

Gambar 2.4 Isi Tabel Employee

✓ Showing rows 0 - 3 (4 total, Query took 0.0008 seconds.)

```
SELECT * FROM `department`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: N

+ Options

	id	nama_depart
<input type="checkbox"/> Edit Copy Delete	1	Pengembang
<input type="checkbox"/> Edit Copy Delete	2	Alat
<input type="checkbox"/> Edit Copy Delete	3	Survey
<input type="checkbox"/> Edit Copy Delete	4	Kesehatan

↑ ☐ Check all | With selected: Edit Copy Delete Export

Gambar 2.5 Isi Tabel Department

```

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

```

Gambar 2.6 Function.php

```

8  function show($query){
9      global $conn;
10     $result = mysqli_query($conn, $query);
11     $rows = [];
12     while( $row = mysqli_fetch_assoc($result) ){
13         $rows[] = $row;
14     }return $rows;
15 }
16
17 function insert($data){
18     global $conn;
19     $id_employe = $_POST["id_employe"];
20     $nama = $_POST["nama"];
21     $id_department = $_POST["id_department$id_department"];
22     mysqli_query($conn, "INSERT INTO employee VALUES('$id_employe','$nama','$id_department')");
23     return mysqli_affected_rows($conn);
24 }
25
26 function delete($data){
27     global $conn;
28     $id_employe = $_GET["delete"];
29     mysqli_query($conn, "DELETE FROM employee WHERE id_employe = $id_employe");
30     return mysqli_affected_rows($conn);
31 }
32
33 function update($data){
34     global $conn;
35     $id_employe = $_POST["id_employe"];
36     $nama = $_POST["nama"];
37     $id_department = $_POST["id_department$id_department"];
38     mysqli_query($conn, "UPDATE employee SET nama = '$nama',
39     id_department = '$id_department' WHERE id_employe = $id_employe");
40     return mysqli_affected_rows($conn);
41 }
42 ?>

```

Gambar 2.7 Lanjutan Function.php

```

function.php x insert.php
insert.php
1 <?php
2 require("function.php");
3 if (isset($_POST["insert"])) {
4     if (insert($_POST) > 0) {
5         echo "<script>
6             alert('data berhasil ditambah');
7             document.location.href = 'index.php';
8         </script>";
9     } else {
10        echo "<script>
11            alert('data gagal ditambah');
12            document.location.href = 'index.php';
13        </script>";
14    }
15 }
16 ?>
17
18 <!DOCTYPE html>
19 <html lang="en">
20

```

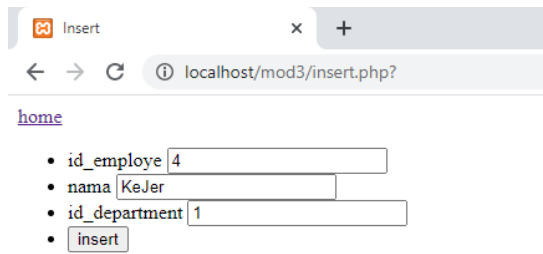
Gambar 2.8 Insert.php

```

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

```

Gambar 2.9 Lanjutan Insert.php



Gambar 2.10 Insert di web

Modul3

localhost/mod3/index.php

id_employe	nama	id_department	aksi
1	KEVIN	1	DELETE UPDATE
2	Rio	2	DELETE UPDATE
3	Abi	1	DELETE UPDATE
4	KeJer	1	DELETE UPDATE

Gambar 2.11 Database yang di tambahkan

```

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

```

Gambar 2.12 Update.php

```
function.php insert.php update.php x
update.php
19
21 <!DOCTYPE html>
22 <html lang="en">
23
24 <head>
25 <meta charset="UTF-8">
26 <meta name="viewport" content="width=device-width, initial-scale=1.0">
27 <title>Update</title>
28 </head>
29
30 <body>
31 <a href="index.php">home</a>
32 <form action="" method="POST">
33 <?php
34 foreach ($data as $d) {
35 >
36 <ul>
37 <li>
38 <label for="id_employe">id_employe</label>
39 <input type="text" name="id_employe" id="id_employe" value=?php echo $d["id_employe"]; ?> readonly>
40 </li>
41 <li>
42 <label for="nama">nama</label>
43 <input type="text" name="nama" id="nama" value=?php echo $d["nama"]; ?>
44 </li>
45 <li>
46 <label for="id_department">id_department</label>
47 <input type="text" name="id_department" id="id_department" value=?php echo $d["id_department"]; ?>
48 </li>
49 <li>
50 <button type="submit" name="ubah">update</button>
51 </li>
52 </ul>
53 <?php
54 }
55 >
56 </form>
57 </body>
58
59 </html>
```

Gambar 2.13 Lanjutan Update.php

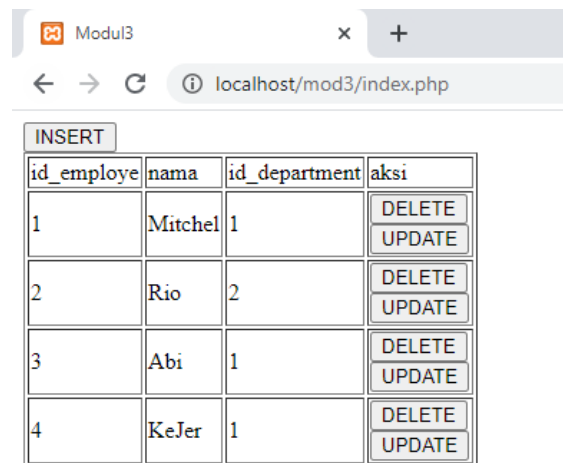
Update

localhost/mod3/update.php?update=1

[home](#)

- id_employe 1
- nama Mitchel
- id_department 1
- update

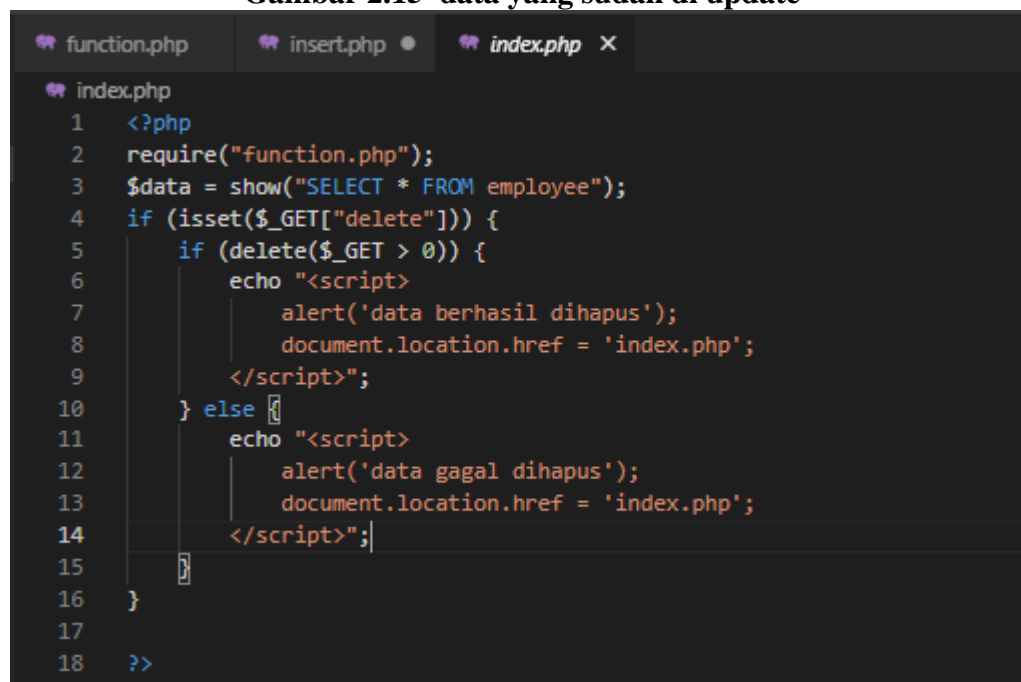
Gambar 2.14 Update data di web



The screenshot shows a web browser window with the address bar displaying 'localhost/mod3/index.php'. Below the browser, there is a table with the following structure:

INSERT			
id_employe	nama	id_department	aksi
1	Mitchel	1	DELETE UPDATE
2	Rio	2	DELETE UPDATE
3	Abi	1	DELETE UPDATE
4	KeJer	1	DELETE UPDATE

Gambar 2.15 data yang sudah di update



The screenshot shows a code editor with three tabs: 'function.php', 'insert.php', and 'index.php'. The 'index.php' tab is active, displaying the following PHP code:

```

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

```

Gambar 2.16 index.php

```
function.php X insert.php ● index.php ●
index.php
19
20 <!DOCTYPE html>
21 <html lang="en">
22
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Modul3</title>
27 </head>
28
29 <body>
30     <form action="insert.php">
31         <button>INSERT</button>
32     </form>
33     <table border="1">
34         <tr>
35             <td>id_employe</td>
36             <td>nama</td>
37             <td>id_department</td>
38             <td>aksi</td>
39         </tr>
40         <?php
41         foreach ($data as $d) {
42             ?>
43             <tr>
44                 <td><?php echo $d["id_employe"]; ?></td>
45                 <td><?php echo $d["nama"]; ?></td>
46                 <td><?php echo $d["id_department"]; ?></td>
47                 <td>
48                     <form action="" method="GET">
49                         <button type="submit" name="delete" value=<?php echo $d["id_employe"]; ?>>DELETE</button>
50                     </form>
51                     <form action="update.php" method="GET">
52                         <button type="submit" name="update" value=<?php echo $d["id_employe"]; ?>>UPDATE</button>
53                     </form>
54                 </td>
55             </tr>
56         <?php
57         }
58         ?>
59     </table>
60 </body>
61 </html>
```

Gambar 2.17 Lanjutan index.php

Modul3 x +

localhost/mod3/index.php

INSERT

id_employe	nama	id_department	aksi
1	Mitchel	1	DELETE UPDATE
2	Rio	2	DELETE UPDATE
4	KeJer	1	DELETE UPDATE

Gambar 2.18 tampilan table di web