

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



**NAMA : KEVIN IMMANUEL RUI COSTA**

**NIM : 193020503030**

**KELAS : A**

**MODUL : III (PHP MySQL Database)**

**JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKARAYA**

**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

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).

#### 1.2.1 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";
?>
```

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, $password);
// Periksa Koneksi
if (!$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”. 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 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);
?>
```

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.

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**

Pertama, pada nomor 1 diminta untuk membuat sebuah database pada MySQL untuk menyimpan data-data pegawai tabelnya. Berikut adalah query untuk membuat databasenya :

```
CREATE DATABASE MODUL3;
```

Lalu setelah itu membuat tabel beserta kolom-kolomnya pada database tersebut, tabel disini terdapat dua tabel yakni tabel pegawai dan tabel departemen. Berikut adalah query untuk membuat tabel dan kolom-kolomnya,

#### **Tabel Pegawai**

```
CREATE TABLE PEGAWAI (ID_Pegawai VARCHAR (10), Nama_Pegawai  
VARCHAR (16), ID_Departemen VARCHAR(10), Alamat VARCHAR (16));
```

Dapat dilihat pada query di atas, tabel memiliki nama pegawai dengan kolom-kolomnya ID\_Pegawai, Nama\_Pegawai, ID\_Departemen dan Alamat.

#### **Tabel Departemen**

```
CREATE TABLE DEPARTEMEN (ID_Departemen  
VARCHAR(10), Nama_Departemen VARCHAR (16))
```

Dapat dilihat pada query di atas, tabel memiliki nama departemen dengan kolom-kolomnya ID\_Departemen dan Nama\_Departemen

Selanjutnya membuat primary key serta foreign key, berikut adalah querynya,

Tabel Departemen

```
ALTER TABLE DEPARTEMEN ADD PRIMARY KEY  
(ID_Departemen);
```

Tabel Pegawai

```
ALTER TABLE PEGAWAI ADD PRIMARY KEY (ID_Pegawai);  
  
ALTER TABLE PEGAWAI ADD FOREIGN KEY (ID_Departemen)  
REFERENCES DEPARTEMEN (ID_Departemen);
```

Selanjutnya memasukkan data-data dari pegawai dan dari departemen ke dalam tabel dari database tersebut. Berikut adalah querynya

Tabel Departemen

```
INSERT INTO departemen VALUES  
("DPRT001","IT"),  
("DPRT002","Marketing"),  
("DPRT003","Production") ;
```

Tabel Pegawai

```
INSERT INTO pegawai (ID_Pegawai,Nama_Pegawai,ID_Departemen,Alamat)  
VALUES ("PGW001","Kevin","DPRT001","Jalan Kecil"),  
("PGW002","Immanuel","DPRT002","Jalan Besar"),  
("PGW003","RUI COSTA","DPRT003","Jalan Sedang");
```

Pada nomor 2 diminta untuk membuat sebuah program untuk menyimpan menghapus dan juga mengubah data-data pegawai. Disini terdapat 7 file php, yang masing-masing memiliki kegunaan serta fungsinya masing-masing, berikut adalah php tersebut.

### Connect.php

```
<?php
$koneksi = mysqli_connect('localhost', 'root', '', 'Modul3');
if(!$koneksi){
    exit('File tidak ada');
}
?>
```

Fungsi Utama File ini adalah mengkoneksikan php dengan database .Dapat dilihat pada sintaks di atas, terdapat variable dengan nama koneksi, variable tersebut berisi nilai dimana variable tersebut terhubung dengan database dengan nama database Modul3.

Php ini menjadi php utama, karena kebanyakan php dari 7 php tersebut terhubung dengan PHP connect ini. Hal itu disebabkan karena file connect.php merupakan file yang menghubungkan php dengan database.

### Input.php

```
<html>
  <form action="insert.php" method="POST">
    <label>Nama</label><br/>
    <input type="text" name="nama"/>
    <br/><br/>

    <label>Alamat</label><br/>
    <input type="text" name="Alamat"/>
    <br/><br/>

    <label>ID Departemen</label><br/>
    <input type="text" name="ID_Departemen"/>
    <br/><br/>
```

```

<label>ID Pegawai</label><br/>
<input type="text" name="ID_Pegawai"/>
<br/><br/>

<button type="submit">Tambah</button>
</form>
</html>

```

Fungsi Utama File ini adalah sebagai tempat atau wadah untuk menginputkan atau memasukan data ke dalam kolom-kolom dari table pegawai. Dapat dilihat dalam sintaks tersebut terdapat tag form, yang mana form merupakan bagian pada HTML yang dapat digunakan untuk membuat elemen Form pada halaman Web, form menjadi tempat penginputan data sebelum diproses oleh system.

Dalam php tersebut terdapat form dengan tipe inputan yang berupa teks dan terdapat form dengan tipe button (tombol) yang berupa submit (menyerahkan inputan).

Lalu terdapat nama-nama dari kolom-kolom pada table pegawai, yang mana menyesuaikan dengan data-data yang akan dimasukkan ke dalam kolom-kolom pada tabel pegawai.

Dan didalam form tersebut terdapat action, yang mana action digunakan untuk mendefinisikan tindakan yang dilakukan oleh form dikirimkan. disitu tindakan yang dilakukan adalah memproses file dengan nama insert.php

### **Insert.php**

```

<?php

include('connect.php');
$nama = $_POST['nama'];
$idp = $_POST['ID_Pegawai'];
$iddd = $_POST['ID_Departemen'];

```

```

    $alamat = $_POST['Alamat'];

$query = mysqli_query($koneksi, "INSERT INTO pegawai set
Nama_Pegawai='$nama',
Alamat='$alamat',
ID_Pegawai='$idp',
ID_Departemen='$idd'");

    if ($query) {
        header('Location:tampilkan.php');
    }else {
        echo 'Input gagal';
    }
?>

```

Fungsi Utama File ini adalah sebagai tempat query INSERT INTO di jalankan, sehingga data dapat dimasukkan ke dalam tabel. Dalam file insert.php terdapat, include dimana include menghubungkan file insert.php dan file connect.php, sehingga file insert.php dapat terhubung ke database. Lalu setelah itu terdapat beberapa variable yakni \$nama,\$idp,\$idd,\$alamat, variable-variabel tersebut sama-sama mendapatkan nilai dari variable \$\_POST yang mengambil nilai dari data yang dimasukkan pada file input.php

Lalu terdapat variable \$query, variable ini berisi sintaks mysqli\_query, digunakan untuk mengirimkan perintah SQL ke server MySQL untuk melakukan aktivitas CRUD dimana meliputi CREATE (menambahkan data dengan perintah SQL INSERT), READ (membaca data dengan perintah SQL SELECT), UPDATE(mengubah data dengan perintah SQL UPDATE), DELETE (menghapus data dengan perintah SQL DELETE).

Lalu mysqli\_query tersebut menggunakan database dengan menggunakan variable \$koneksi lalu memasukkan data ke dalam tabel pegawai dengan menggunakan query INSERT INTO. Dimana nilai yang di set atau atur adalah Kolom Nama\_Pegawai nilainya sama dengan variabel \$nama yang mana nilainya didapat dari hasil inputan pada form nama. Lalu kolom Alamat yang nilainya

sama dengan variabel \$Salamat yang nilainya didapat dari hasil inputan pada form Alamat. Lalu kolom ID\_Pegawai yang nilainya sama dengan variabel \$idp dengan nilai didapat dari hasil inputan pada form ID\_Pegawai, lalu kolom ID\_Departemen yang nilainya sama dengan variabel \$idd yang nilainya didapat dari hasil inputan pada form ID\_Departemen.

Lalu terdapat percabangan if, dimana jika program berhasil dijalankan maka program akan menuju ke file tampilkan.php. Dan jika program tidak berhasil dijalankan maka program akan mengeluarkan output Input Gagal.

#### Tampilkan.php

```
<?php
include('connect.php');
$query = mysqli_query($koneksi, "SELECT *FROM pegawai");
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>

<head>
    <title>Tabel</title>
</head>

<body>

    <table border="1">
        <thead>
            <th>ID Pegawai</th>
            <th>Nama Pegawai</th>
            <th>ID Departemen</th>
            <th>Alamat</th>
        </thead>
        <tbody>
            <?php foreach ($result as $value) { ?>
                <tr>
                    <td><?= $value['ID_Pegawai'] ?></td>
                    <td><?= $value['Nama_Pegawai'] ?></td>
                    <td><?= $value['ID_Departemen'] ?></td>
                    <td><?= $value['Alamat'] ?></td>
                </tr>
            }
        </tbody>
    </table>

```



```

<a href="ubah.php?no=<?= $value['ID_Pegawai']?>">Edit</a>
    <br>
    <a href="hapus.php?no=<?= $value['ID_Pegawai']?>">Hapus</a>
    </td>
</tr>

    <?php } ?>
</tbody>
</table>
<nav>
    <a href="input.php">Input Data</a>
</nav>
</body>
</html>

```

Fungsi Utama File ini adalah sebagai file yang menampilkan data –data dari tabel. Terdapat include, seperti yang sudah dijelaskan untuk menghubungkan file Tampilkan.php dengan file Connect.php. lalu variabel query yang mana disini query sql yang digunakan adalah SELET \*FROM yang berguna untuk mengambil data dari tabel yang dikehendaki.

Lalu terdapat variabel \$result dengan nilai Mysqli\_fetch\_all yang digunakan untuk mengambil semua baris hasil dan mengembalikan kumpulan hasil sebagai larik asosiatif, larik numerik, atau keduanya.

Setelah itu terdapat tabel yang berisi kolom dengan nama ID Pegawai, Nama Pegawai, ID Departemen dan Alamat. Lalu terdapat perulangan foreach dimana variabel \$result sebagai \$value.

Variabel \$value memiliki nilai sesuai dengan nama kolom yang dipanggil, contoh \$value['ID\_Pegawai'] mengambil nilai dari kolom ID Pegawai dan nilai-nilai tersebut diletakkan sesuai dengan kolom-kolom yang telah ditentukan pada tabel sebelumnya.

Lalu terdapat tag `<a>` yang mana `<a>` element digunakan untuk mendefinisikan sebuah hyperlink. Dimana hyperlink tersebut terdapat tiga, yakni menuju ke file `ubah.php`, `hapus.php` dan terakhir `input.php`

### Ubah.php

```
<?php
include('connect.php');
$no = $_GET['no'];
$query = mysqli_query($koneksi, "SELECT * FROM pegawai WHERE pegawai
.ID_Pegawai = '$no' LIMIT 1");
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>
<html>
<form action="update.php" method="POST">

    <input type="hidden" name="no" value="<?php echo $result[0]['ID_
Pegawai']?>">
    <label>Nama</label><br />
    <input type="text" name="nama" value="<?php echo $result[0]['Nam
a_Pegawai'] ?>" />
    <br /><br />

    <label>Alamat</label><br />
    <input type="text" name="Alamat" value="<?php echo $result[0]['A
lamat'] ?>" />
    <br /><br />

    <label>ID Pegawai</label><br />
    <input type="text" name="ID_Pegawai" value="<?php echo $result[0
]['ID_Pegawai'] ?>" />
    <br /><br />

    <label>ID Departemen</label><br />
    <input type="text" name="ID_Departemen" value="<?php echo $resul
t[0]['ID_Departemen'] ?>" />
    <br /><br />

    <br/><br/>
    <button type="submit">Ubah</button>
</form>
</html>
```

Fungsi utama file tersebut untuk memasukkan data, dimana data yang dimasukkan pada file ini merupakan pembaharuan dari data yang telah ada. Sama seperti file input, didalam file ini juga terdapat form dengan tipe input teks dan tipe tombol submit.

File terhubung dengan file connect.php, lalu terdapat variabel \$no dengan nilai \$\_GET ['no'] nilai tersebut didapat dari nilai 'no'. lalu terdapat variabel \$query dengan query berisi Select \*FROM yang mana digunakan untuk mengambil data dengan kondisi pegawai.ID\_Pegawai = '\$no'

Lalu terdapat variabel \$result. Setelah itu terdapat form, dengan action ke file update.php, lalu terdapat form input yang digunakan untuk menginputkan data baru yang akan memperbaharui data lama, dan terdapat form tombol yang digunakan untuk mensubmit data yang telah di inputkan.

### Update.php

```
<?php

include('connect.php');
$no = $_POST['no'];
$nama = $_POST['nama'];
$idp = $_POST['ID_Pegawai'];
$iddd = $_POST['ID_Departemen'];
$alamat = $_POST['Alamat'];

$query = mysqli_query($koneksi, "UPDATE pegawai set
Nama_Pegawai='$nama',
Alamat='$alamat',
ID_Pegawai = '$idp',
ID_Departemen = '$iddd' WHERE pegawai.ID_Pegawai = '$no' ");

if ($query) {
    header('Location:tampilkan.php');
}else {
    echo 'Input gagal';
}
```

Fungsi utama file tersebut adalah untuk mengupdate data pada tabel pegawai. Dimana dapat dilihat terdapat beberapa variabel, antara lain \$no, \$nama, \$idp, \$idd, \$alamat dimana variabel tersebut mengambil nilai dari data yang diinputkan pada form ubah.php. lalu terdapat variabel \$query dimana query yang digunakan adalah Update pegawai Set nama kolom yang digubah = nama variabel yang sesuai dengan form, lalu kondisinya adalah pegawai.ID\_Pegawai = variabel \$no.

Lalu terdapat percabangan if, dimana jika program berjalan dengan benar maka program akan masuk ke file tampilkan.php, jika tidak maka program akan mengeluarkan output Input Gagal.

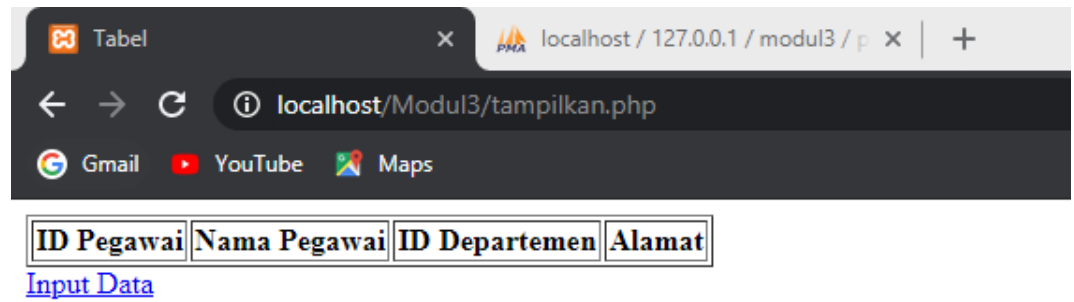
### Hapus.php

```
<?php
    include('connect.php');
    $no = $_GET['no'];
    $delete = mysqli_query($koneksi, "DELETE FROM pegawai WHERE pegawai.ID_Pegawai = '$no'");
    if ($delete) {
        header('Location:tampilkan.php');
    } else {
        echo "File tidak ada";
    }
?>
```

Fungsi utama dari file tersebut adalah untuk menghapus data pada tabel pegawai. Terdapat variabel \$delete dengan query DELETE FROM yang berfungsi untuk menghapus data dari tabel yang diinginkan dengan kondisi pegawai.ID\_Pegawai = '\$no'.

Lalu terdapat percabangan if, dimana jika program berjalan dengan benar maka program akan masuk ke file tampilkan.php, jika tidak maka program akan mengeluarkan output Input Gagal.

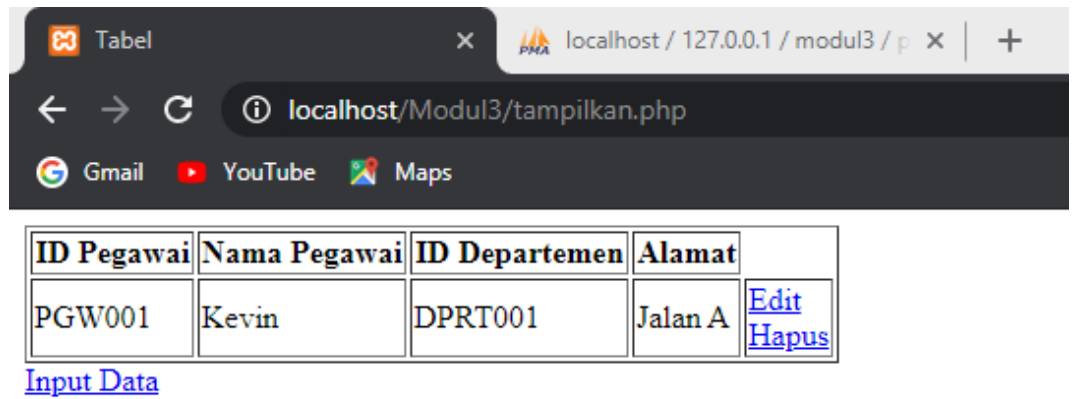
Berikut gambar dari hasil file-file tersebut.



**Gambar 2.1 Data Masih Kosong**

The screenshot shows a web browser window with the address bar displaying 'localhost/Modul3/input.php'. Below the browser window, there is a form with four input fields: 'Nama', 'Alamat', 'ID Departemen', and 'ID Pegawai'. Below the fields, there is a button labeled 'Tambah'.

**Gambar 2.2 Form Input Data**



**Gambar 2.3 Data Diinputkan**

The screenshot shows a web browser window with the address bar displaying 'localhost/Modul3/ubah.php?no=PGW001'. Below the browser, there is a form with the following fields:

Nama

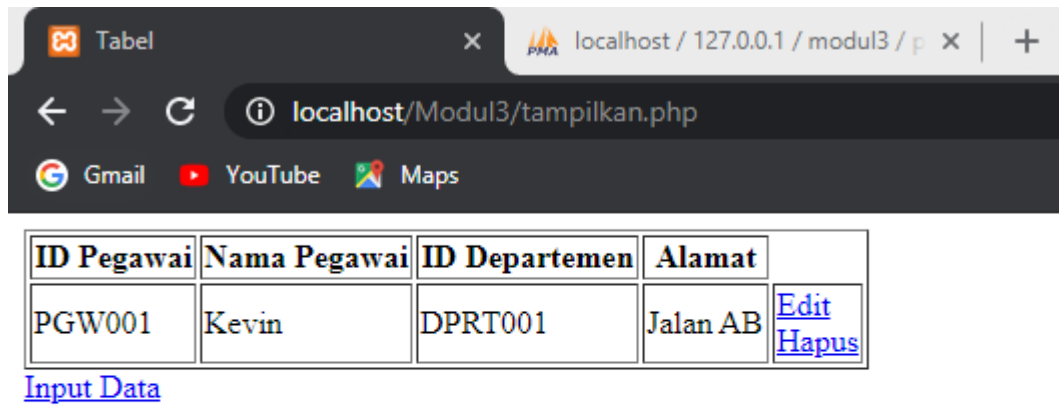
Alamat

ID Pegawai

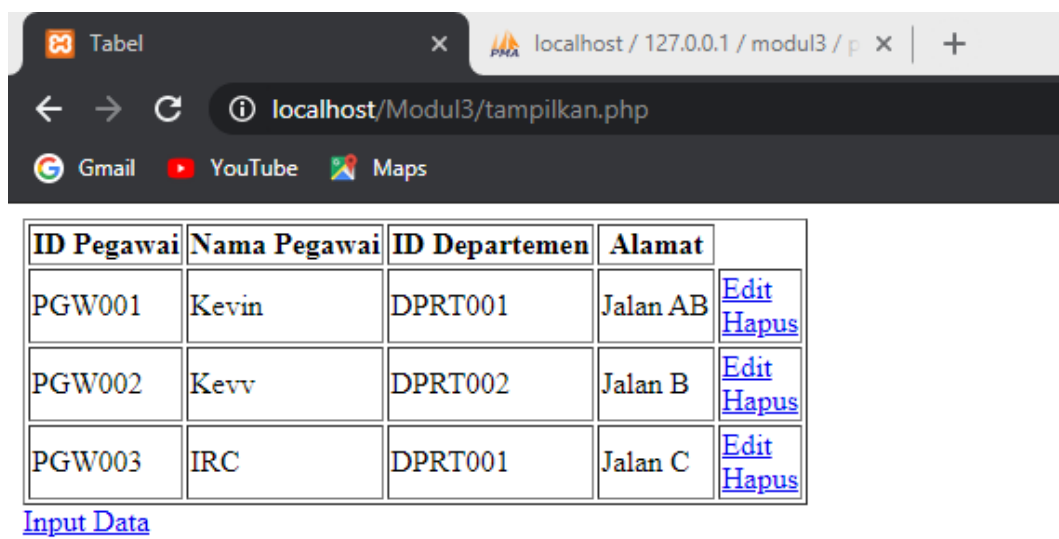
ID Departemen

Ubah

**Gambar 2.4 Form Update Data**



**Gambar 2.5 Data Diupdate**



**Gambar 2.6 Data Sebelum Dihapus (3 Baris Data)**

localhost / 127.0.0.1 / modul3 / p x +

localhost/Modul3/tampilkan.php

Gmail YouTube Maps

ID Pegawai	Nama Pegawai	ID Departemen	Alamat	
PGW001	Kevin	DPRT001	Jalan AB	<a href="#">Edit</a> <a href="#">Hapus</a>
PGW003	IRC	DPRT001	Jalan C	<a href="#">Edit</a> <a href="#">Hapus</a>

[Input Data](#)

**Gambar 2.7 Data Setelah Dihapus (2 Baris Data)**



### **BAB III**

### **KESIMPULAN**

Kesimpulannya dengan adanya mysql serta php membuat pembuatan website yang menggunakan database dapat dilakukan, penggunaannya yang dapat dilakukan secara sederhana dapat membantu dalam pembelajaran pembuatan web bagi pemula, sehingga mudah dipahami dan dipraktekkan.

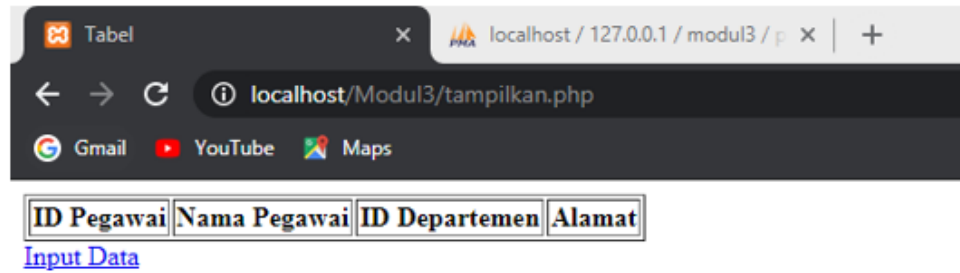
## **DAFTAR PUSTAKA**

MODUL PRAKTIKUM PEMROGRAMAN WEB DAN MOBILE I Jurusan  
Teknik Informatika Fakultas Teknik Universitas Palangkaraya, 2021

Muhardian, Ahmad.10 Juni 2017. Tutorial PHP dan MySQL: Membuat Aplikasi  
CRUD

<https://www.petanikode.com/tutorial-php-mysql/>

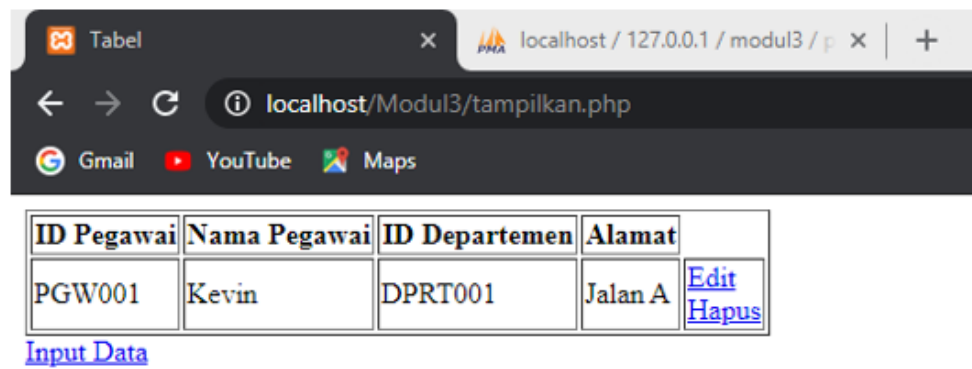
## LAMPIRAN



**Gambar 2.1 Data Masih Kosong**

The screenshot shows a web browser window with the address bar displaying 'localhost/Modul3/input.php'. Below the browser window, there is a form with four input fields: 'Nama', 'Alamat', 'ID Departemen', and 'ID Pegawai'. Below the fields, there is a button labeled 'Tambah'.

**Gambar 2.2 Form Input Data**



**Gambar 2.3 Data Diinputkan**

The screenshot shows a web browser window with the address bar displaying 'localhost/Modul3/ubah.php?no=PGW001'. Below the browser window, there is a form with the following fields:

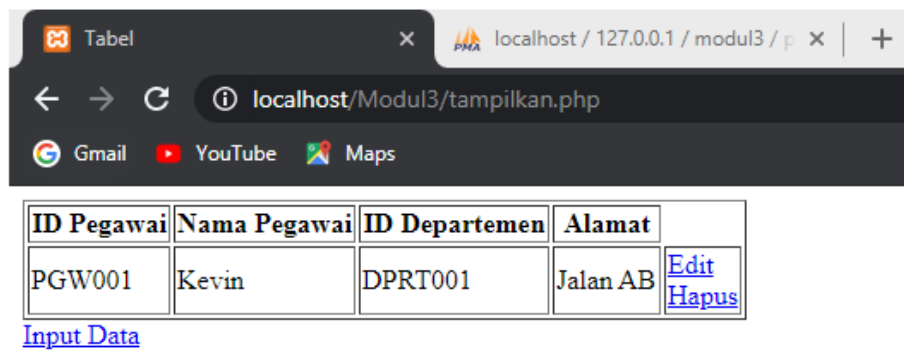
Nama

Alamat

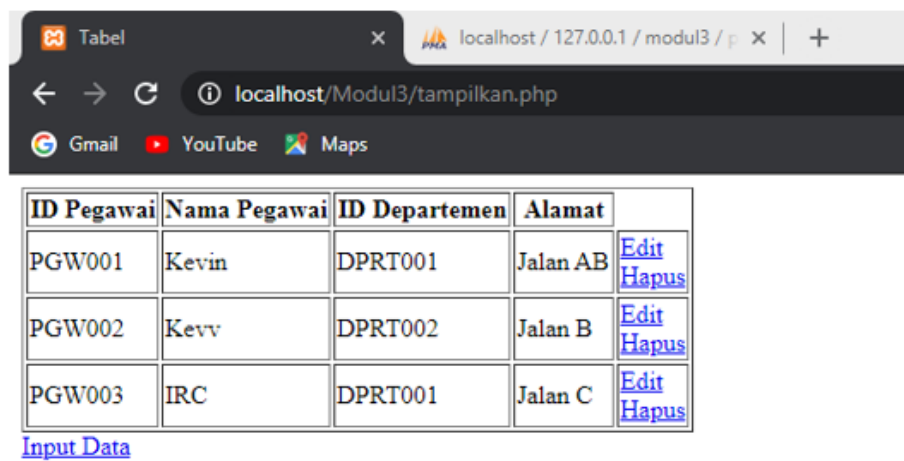
ID Pegawai

ID Departemen

**Gambar 2.4 Form Update Data**



**Gambar 2.5 Data Diupdate**



**Gambar 2.6 Data Sebelum Dihapus (3 Baris Data)**

Browser window showing a table of employee data. The browser address bar indicates the URL is localhost/Modul3/tampilkan.php. The table has columns: ID Pegawai, Nama Pegawai, ID Departemen, Alamat, and a column with links (Edit, Hapus). The data rows are:

ID Pegawai	Nama Pegawai	ID Departemen	Alamat	
PGW001	Kevin	DPRT001	Jalan AB	<a href="#">Edit</a> <a href="#">Hapus</a>
PGW003	IRC	DPRT001	Jalan C	<a href="#">Edit</a> <a href="#">Hapus</a>

[Input Data](#)

**Gambar 2.7 Data Setelah Dihapus (2 Baris Data)**