

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



**NAMA : MUHAMMAD IJLAL PRAYOGA
NIM : E1E118012
KELAS : C
MODUL : III**

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

BAB I

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

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

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 (

```

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 1.4 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 1.5 Update Data

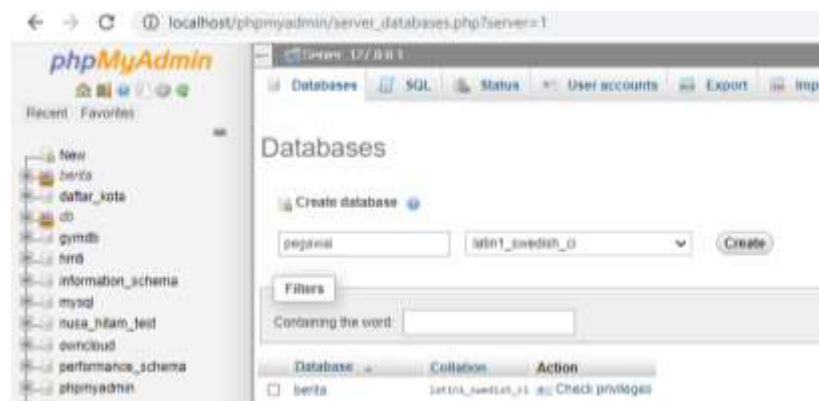
BAB II

PEMBAHASAN

Berikut ini tugas yang diminta pada modul.

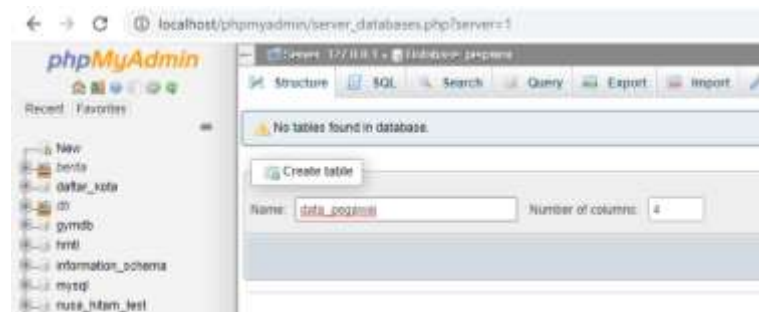
1. Buatlah database pada MySQL untuk menyimpan data-data pegawai beserta relasi tabelnya.
2. Buatlah program untuk menyimpan, menghapus dan mengubah data data pegawai tersebut.

Pada tugas yang pertama kita diminta untuk membuat database pada MySQL untuk menyimpan data-data pegawai beserta relasi tabelnya. Untuk membuat database nya dengan cara buka browser lalu ketik localhost/phpmyadmin lalu klik tombol new dan isi nama database seperti berikut :



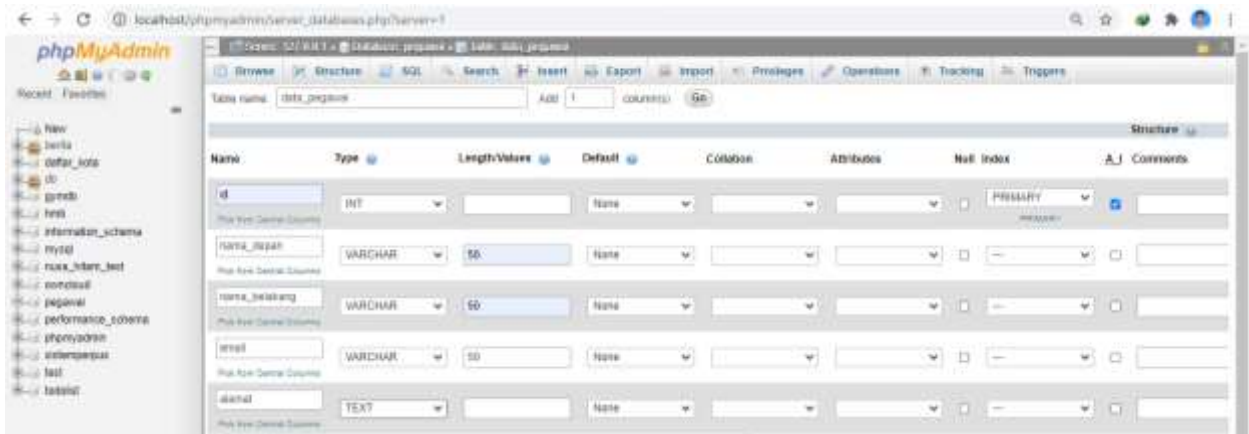
Gambar 2.1 Pembuatan Database.

Kemudian lanjutkan dengan membuat tabel. Pertama tetapkan berapa kolom dari tabel yang akan dibuat :



Gambar 2.2 Penetapan Jumlah Kolom.

Setelah itu tetapkan nama masing-masing kolom dan karakteristiknya. Untuk primary key saya membuatnya A_I atau Auto Increment sehingga ketika data ditambahkan maka id otomatis akan terisi dengan sendirinya



Gambar 2.3 Penetapan Karakteristik Kolom

Dapat dilihat tabel yang dibuat mempunyai kolom berupa id, nama_depan, nama_belakang, email dan alamat yang nantinya data tersebut akan dikelola oleh sistem.

Pada tugas yang kedua kita diminta untuk membuat program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut.

Program file yang pertama yaitu berupa index.php yang dimana merupakan tampilan dari seluruh perintah yang diminta pada tugas kedua yang berisikan data-data, insert, delete, update, serta script HTML yang berfungsi untuk menampilkan tabelnya yang dimana isi dari program tersebut adalah sebagai berikut :

```
<?php
$connect = mysqli_connect("localhost", "root", "", "pegawai");

if (!$connect) {
    die("Connection failed" . mysqli_connect_error());
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tugas Modul 3</title>
</head>

<body>
    <?php
    ini_set("display_errors", 0);
    if ($_GET['btn'] == 'delete') {
        $sql = "DELETE FROM data_pegawai WHERE id='".$_GET[id]."'";
        mysqli_query($connect, $sql);
        echo "<script>
            alert('data berhasil dihapus');
            document.location.href = 'index.php';
        </script>";
    }
}
```

```

        if ($_POST['btn'] == 'update') {
            $sql = "UPDATE data_pegawai SET nama_depan='".$_POST[nama_depan]','nama_belakang='".$_POST[nama_belakang]','email='".$_POST[email]','alamat ='".$_POST[alamat]' WHERE id='".$_POST[id]'"";
            mysqli_query($connect, $sql);
            echo "<script>
                alert('data berhasil diubah');
                document.location.href = 'index.php';
            </script>";
        }
        if ($_POST['btn'] == 'input') {
            $id = $_POST['id'];
            $nama_depan = $_POST['nama_depan'];
            $nama_belakang = $_POST['nama_belakang'];
            $email = $_POST['email'];
            $alamat = $_POST['alamat'];
            $sql = "INSERT INTO data_pegawai (id,nama_depan,nama_belakang,email,alamat)
VALUES ('$id','$nama_depan','$nama_belakang','$email','$alamat')";
            mysqli_query($connect, $sql);
            echo "<script>
                alert('data berhasil ditambah');
                document.location.href = 'index.php';
            </script>";
        }
        $sql = "SELECT * from data_pegawai";
        $result = mysqli_query($connect, $sql);
        ?>
        <h1 style="text-align: center;">Data Pegawai</h1>
        <table border="1" style="margin: 0 auto;">
            <tr>
                <td>id</td>
                <td>nama depan</td>
                <td>nama belakang</td>
                <td>email</td>
                <td>alamat</td>
                <td colspan="2" align="center">aksi</td>
            </tr>
            <?php
            if (mysqli_num_rows($result)) {
                while ($row = mysqli_fetch_assoc($result)) {
                    echo "<tr>
                        <td>" . $row["id"] . "</td>
                        <td>" . $row["nama_depan"] . "</td>
                        <td>" . $row["nama_belakang"] . "</td>
                        <td>" . $row["email"] . "</td>
                        <td>" . $row["alamat"] . "</td>
                        <td><a href='\"update_data.php?id=" . $row["id"] . "&btn=update\"'
>
                            update</a></td>
                        <td><a href='\"index.php?id=" . $row['id'] . "&btn=delete\"'
delete</a> </td>
                    </tr>";
                }
            } else {
                echo "0 result in database";
            }
            ?>

        </table>
        <a href="input_data.php">
            <p align="center">Input Data</p>
        </a>
    </body>
</html>

```

Pada baris pertama dari program diatas merupakan fungsi untuk mengecek koneksi dari database dan apabila koneksi gagal akan keluar pemberitahuan “Connection failed”.

Kemudian pada script PHP selanjutnya merupakan fungsi utama dari program ini. Dalam script tersebut terdapat fungsi menampilkan data-data yang terdapat di database, fungsi menghapus data, mengedit data, dan menambahkan data.

Pada fungsi delete merupakan fungsi untuk menghapus data pegawai pada database yang dimana untuk menghapusnya menggunakan id, dimana ketika button pada tampilan awal ditekan akan langsung menghapus data pegawai tersebut. Fungsi ini menggunakan perintah \$_GET karena hanya mengambil id dari data dan langsung menghapusnya

Pada fungsi update merupakan fungsi untuk mengubah data pegawai pada database dan tabel data_pegawai. Yang dimana melakukan update atau perubahan data berdasarkan pada id. Fungsi ini menggunakan perintah \$_POST karena pengisian data yang akan diupdate berada di laman lain yang data barunya nanti akan dikirim ke laman utama

Pada fungsi input merupakan fungsi untuk menambahkan data pegawai yang baru pada database dan tabel data_pegawai. Yang dimana ketika tombol ditekan, otomatis user akan dibawa ke laman untuk mengisi data-data dari pegawai. Fungsi ini menggunakan perintah \$_POST karena pengisian data yang akan baru berada di laman lain yang data barunya nanti akan dikirim ke laman utama

Program file kedua yaitu membuat program untuk form update data pegawai yang nantinya datanya akan dikirim ke laman utama, berikut adalah isi dari program untuk menyimpan data pegawai :

```
<?php
$conn = mysqli_connect("localhost", "root", "", "pegawai");

if (!$conn) {
    die("Connection failed" . mysqli_connect_error());
}
$sql = "SELECT * FROM data_pegawai WHERE id= '$_GET[id]'";
$result = mysqli_query($conn, $sql);
$row = mysqli_fetch_assoc($result);
?>

<form method="post" action="index.php">
    Id : <input type="text" name="id" value="<?php echo $row["id"]; ?>"><br>
    Nama Depan : <input type="text" name="nama_depan" value="<?php echo $row["nama_depan"]; ?>"><br>
    Nama Belakang : <input type="text" name="nama_belakang" value="<?php echo $row["nama_belakang"]; ?>"><br>
```

```

Email : <input type="text" name="email" value="<?php echo $row["email"]; ?>"><br>
>
Alamat : <input type="text" name="alamat" value="<?php echo $row["alamat"]; ?>">
<br>
<input type="submit" name="btn" value="update"><br><br>

<p>Catatan:<br>
    Primary Key: ID<br>
    Primary Key Tidak Dapat DI Ubah</p>
</form>

```

Program diatas menggunakan query SQL yang berfungsi mengambil data yang akan diupdate nantinya. Sedangkan dalam form terdapat fungsi echo yang menampilkan row data yang akan diupdate, serta dalam form menggunakan method post yang berfungsi mengirim datanya ke laman utama.

Pada file berikutnya adalah membuat program untuk form untuk menginput data pegawai baru yang nantinya datanya akan dikirim ke laman utama, berikut adalah isi dari program untuk menyimpan data pegawai baru :

```

<form method="post" action="index.php">
    Id : <input type="text" name="id"><br>
    Nama Depan : <input type="text" name="nama_depan"><br>
    Nama Belakang : <input type="text" name="nama_belakang"><br>
    Email : <input type="text" name="email"><br>
    Alamat : <input type="text" name="alamat"><br>
    <input type="submit" name="btn" value="input">

</form>
<p>Catatan:<br>
    Primary Key: ID (Auto Increment/AI)<br>
    Primary Key Tidak Perlu Diisi karena sudah dalam kondisi AI</p>

```

Dari program diatas merupakan tempat untuk mengisi data pegawai yang baru dan nantinya datanya akan dikirim dengan method post ke laman utama.

id	nama depan	nama belakang	email	alamat	aksi
1	ijai	gracyoga	ijai@gmail.com	perumahan kendari indah	input data delete
2	dimas	eka purno	dimas@gmail.com	pacomeeto	input data delete
3	malin	kesodang	malin@gmail.com	jawa tengah	input data delete

[Input Data](#)

Gambar 2.4 Tampilan awal website.

Id:

Nama Depan:

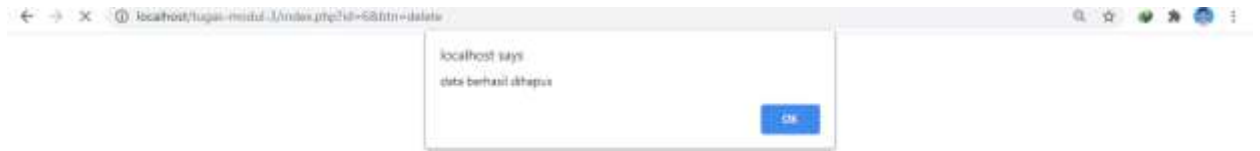
Nama Belakang:

Email:

Alamat:

Catatan:
 Primary Key: ID (Auto Increment AI)
 Primary Key: Tidak Perlu Disi karena sudah dalam kondisi AI

Gambar 2.5 Insert data pegawai.



Gambar 2.6 Delete Data pegawai.



Gambar 2.7 Update Data Pegawai.

BAB III

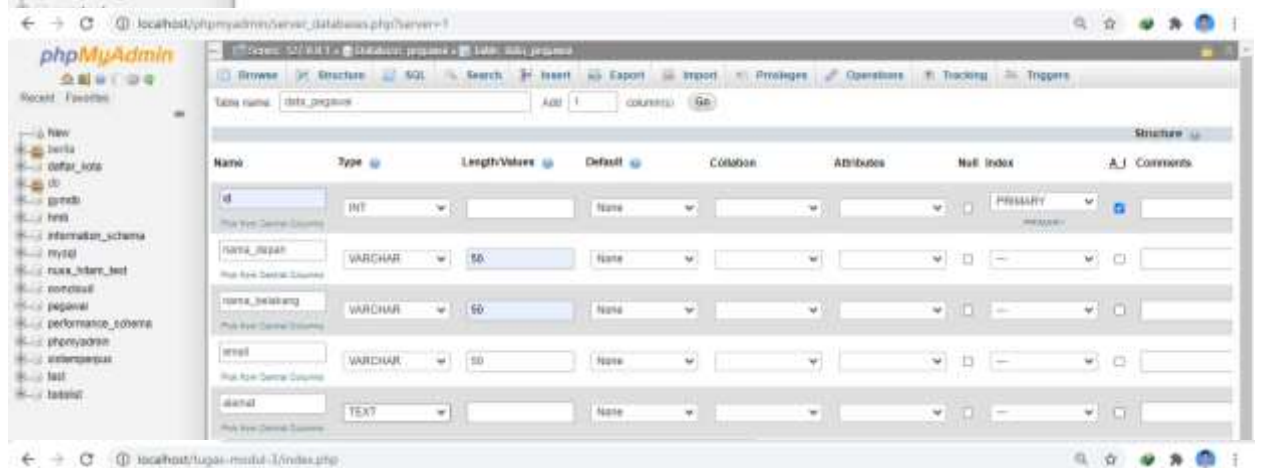
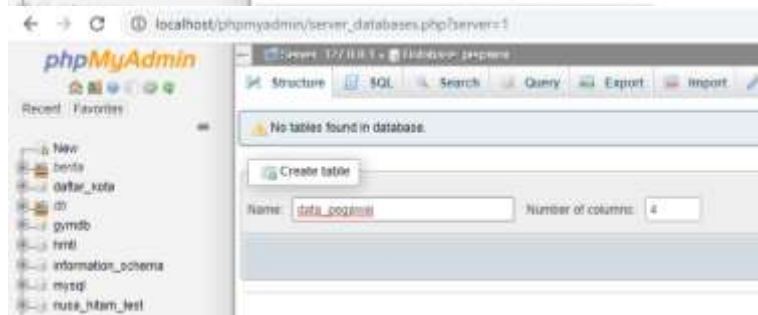
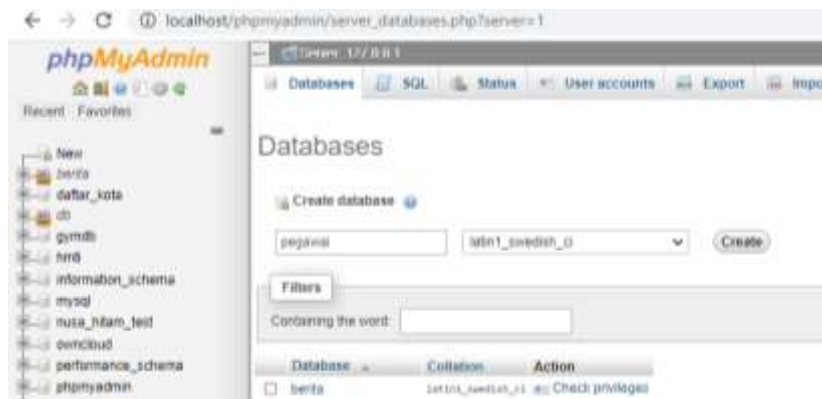
KESIMPULAN

Jadi dari praktikum kali ini dapat disimpulkan bahwa dengan adanya mysql serta php membuat pembuatan website yang menggunakan database membuat website menjadi lebih kaya akan data dan data tersebut dapat dimanfaatkan untuk menjalankan beberapa fitur yang menghasilkan informasi yang berguna bagi user.

DAFTAR PUSTAKA

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

LAMPIRAN



Data Pegawai

id	nama depan	nama belakang	email	alamat	aksi
1	laila	guz-cga	laila@gmail.com	perumahan kendari indah	update delete
2	dimas	eka putro	dimas@gmail.com	raoneero	update delete
3	muhammad	hammad	muhammad@gmail.com	zma tengah	update delete

[Input Data](#)

← → 🔄 localhost/tugas-modul-1/input_data.php

Id :

Nama Depan :

Nama Belakang :

Email :

Alamat :

Catatan:
Primary Key: ID (Auto Increment/AI)
Primary Key Tidak Perlu Disi karena sudah dalam kondisi AI

← → ✕ 🔄 localhost/tugas-modul-1/index.php?id=5&btn=delete

localhost says
data berhasil dihapus

← → 🔄 localhost/tugas-modul-1/update_data.php?id=5&btn=update

Id :

Nama Depan :

Nama Belakang :

Email :

Alamat :

Catatan:
Primary Key: ID
Primary Key Tidak Dapat Di Ubah