

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



**NAMA : ETRILIAN TONGALU**  
**NIM : E1E118003**  
**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** Membuka 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** Membuka 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)) {
```

```

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

    mysqli_close($conn);
?>

```

**Gambar 1.3** 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 1.4** 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 1.5** Memasukkan Data ke Database

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);
?>

```

**Gambar 1.6** Memasukkan Data ke Database

### 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);
?>

```

**Gambar 1.7** Mengambil Data dari Database

### Menghapus Data dari Database

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

id	Nama depan	Nama belakang	Email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23

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

```

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

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

```



```

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

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

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

mysqli_close($conn);
?>

```

**Gambar 1.8** Menghapus Data dari Database

### 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);
?>
```

**Gambar 1.9** Update Data dalam Database

## BAB II

### PEMBAHASAN

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

Dari kriteria tugas di atas, kita akan membuat file untuk membuat database, input data, serta update data. File yang dibuat ada 3, yaitu index.php, input.php, dan update.php. Index.php merupakan halaman awal yang berisi source code untuk membuat database beserta tabel-tabelnya dan halaman ini menampilkan tabel yang berisi data pegawai lengkap dengan relasi tabelnya. Adapun kode programnya yaitu sebagai berikut.

```
<?php
ini_set("display_errors", 0);

// Buat Database
if ($_POST['btn'] == 'Buat Database') {
    $conn = mysqli_connect("localhost", "root", "");
    $sql = "CREATE DATABASE db_pegawai";
    if (mysqli_query($conn, $sql)) {
        echo "<script>alert('Database (db_pegawai) berhasil dibuat')</script>";
    } else {
        echo "<script>alert('Database gagal dibuat/sudah ada')</script>";
    }
    mysqli_close($conn);
}

// Buat Tabel
if ($_POST['btn'] == 'Buat Tabel') {
    $conn = mysqli_connect("localhost", "root", "", "db_pegawai");
    $sql_p = "CREATE TABLE tb_pegawai (
        nip varchar(20) primary key,
        nama varchar(25),
```

```

        alamat text,
        kd_jabatan varchar(255)
    );
$sql_j = "CREATE TABLE tb_jabatan (
    kd_jabatan varchar(20) primary key,
    nama_jabatan varchar(255)
)";

if (mysqli_query($conn, $sql_p) && mysqli_query($conn, $sql_j)) {
    $sql_j_i = "INSERT INTO tb_jabatan (kd_jabatan, nama_jabatan)
        VALUES ('001', 'Kepala'),('002','Sekretaris'),('003'
        , 'Fungsional Perencanaan'),('004','Kepala Sub Bagian
        '),('005','Kepala Bidang'),('006','Kepala Sub Bidang
        '),('007','Sub Bidang'),('008','Staaf')";
    mysqli_query($conn, $sql_j_i);

    $sql_p_i = "INSERT INTO tb_pegawai (nip, nama, alamat, kd_jabata
    n) VALUES ('197008121995121001','Zainal','Anduonohu,
    Kec. Laloara, Jl. Lumba-Lumba','001'),
    ('196706171994031010','Yono','Anduonohu, Jl. Jend. A
    H. Nasution','008'),'196512312006041114','Amrin',
    'Kambu, Jl. Lasitarda, Lrg. Tekukur','008')";
    mysqli_query($conn, $sql_p_i);

    echo "<script>alert('Tabel pegawai dan jabatan berhasil dibuat')
    </script>";
} else {
    echo "<script>alert('Tabel gagal dibuat/sudah ada')</script>";
}
mysqli_close($conn);
}

?>

<!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>Modul 3 - Etrilian Tongalu</title>
</head>

```

```

<body>
  <form action="index.php" method="POST">
    <ul style="list-style: none;">
      <li>
        Jika belum memiliki database, lakukan perintah berikut :
        <ul style="margin-top: 5px;">
          <li>
            Klik untuk membuat database :
            <input type="submit" name="btn" value="Buat Data
              base">
          </li>
          <li style="margin-top: 10px;">
            Klik untuk menambahkan tabel :
            <input type="submit" name="btn" value="Buat Tabe
              l">
          </li>
        </ul>
      </li>
      <li style="margin-top: 10px;">
        <form method="GET">
          <h3>Daftar Pegawai</h3>
          <table border="1">
            <tr>
              <th>No</th>
              <th>NIP</th>
              <th>Nama</th>
              <th>Alamat</th>
              <th>Jabatan</th>
              <th colspan="2">Aksi</th>
            </tr>

            <?php
            ini_set("display_errors", 0);
            $con = mysqli_connect("localhost", "root", "", "
              db_pegawai");

            // Hapus Data
            if ($_GET['btn'] == 'delete') {
              $sql = "DELETE FROM tb_pegawai WHERE nip='$_
                GET[id]'";
              mysqli_query($con, $sql);
            }

            // Menyimpan/Menambah Data
            if ($_POST['btn'] == 'Send') {

```

```

        $nip = $_POST['nip'];
        $nama = $_POST['nama'];
        $alamat = $_POST['alamat'];
        $kd_jabatan = $_POST['kd_jabatan'];
        $sql = "INSERT INTO tb_pegawai (nip,nama,alamat,kd_jabatan) VALUES ('$nip','$nama','$alamat','$kd_jabatan')";
        mysqli_query($con, $sql);
    }

    // Update Data
    if ($_POST['btn'] == 'update') {
        $sql = "UPDATE tb_pegawai SET nip='$_POST[nip]',nama='$_POST[nama]',alamat='$_POST[alamat]',kd_jabatan='$_POST[kd_jabatan]' WHERE nip='$_POST[nip]'";
        mysqli_query($con, $sql);
    }

    // Tampilkan Data
    $sql = "SELECT * FROM tb_pegawai INNER JOIN tb_jabatan ON tb_pegawai.kd_jabatan = tb_jabatan.kd_jabatan";
    $hasil = mysqli_query($con, $sql);
    $i = 1;
    foreach ($hasil as $data) :
        echo "
            <td>" . $i++ . "</td>
            <td>" . $data["nip"] . "</td>
            <td>" . $data["nama"] . "</td>
            <td>" . $data["alamat"] . "</td>
            <td>" . $data["nama_jabatan"] . "</td>
            <td><a href=\"update.php?id=$data[nip]&btn=update\">update</a></td>
            <td><a href=\"index.php?id=$data[nip]&btn=delete\">delete</a></td>
        </tr>";
    endforeach;
    ?>
</table>
<a href="input.php">Tambah Pegawai</a><br>
</form>
</li>
</ul>
</form>

```

```
</body>
</html>
```

Berdasarkan tugas yang pertama, kita mulai dengan membuat database terlebih dahulu. Database yang dibuat bisa menggunakan program PHP maupun melalui localhost/phpmyadmin. Disini kita coba dengan menggunakan program yang dimasukkan dalam index.php. Untuk membuat database dan tabel, kita membutuhkan tombol untuk menjalankan program tersebut. Kode yang digunakan yaitu:

Klik untuk membuat database :

```
<input type="submit" name="btn" value="Buat Database">
```

Klik untuk menambahkan tabel :

```
<input type="submit" name="btn" value="Buat Tabel">
```

Sedangkan kode program yang digunakan untuk membuat database yaitu:

```
if ($_POST['btn'] == 'Buat Database') {
    $conn = mysqli_connect("localhost", "root", "");
    $sql = "CREATE DATABASE db_pegawai";
    if (mysqli_query($conn, $sql)) {
        echo "<script>alert('Database (db_pegawai) berhasil dibuat')
            </script>";
    } else {
        echo "<script>alert('Database gagal dibuat/sudah ada')
            </script>";
    }
    mysqli_close($conn);
}
```

Program di atas menggunakan kondisi if else, yaitu untuk mencari tahu apakah tombol yang diklik merupakan tombol untuk membuat database atau tombol untuk membuat tabel. Jika tombol yang diklik (name="btn") yaitu "Buat Database", maka code di dalamnya akan dijalankan. Yaitu membuat database dengan SQL `CREATE DATABASE` dengan nama db\_pegawai.

Selanjutnya kode yang digunakan untuk membuat tabel beserta data-datanya (isi dari tabel tersebut) yaitu sebagai berikut.

```
if ($_POST['btn'] == 'Buat Tabel') {
    $conn = mysqli_connect("localhost", "root", "", "db_pegawai");
```

```

$sql_p = "CREATE TABLE tb_pegawai (
    nip varchar(20) primary key,
    nama varchar(25),
    alamat text,
    kd_jabatan varchar(255)
)";
$sql_j = "CREATE TABLE tb_jabatan (
    kd_jabatan varchar(20) primary key,
    nama_jabatan varchar(255)
)";

if (mysqli_query($conn, $sql_p) && mysqli_query($conn, $sql_j)) {
    $sql_j_i = "INSERT INTO tb_jabatan (kd_jabatan, nama_jabatan)
        VALUES ('001', 'Kepala'),('002','Sekretaris'),('003'
        , 'Fungsional Perencanaan'),('004','Kepala Sub Bagian
        '),('005', 'Kepala Bidang'),('006','Kepala Sub Bidang
        '),('007','Sub Bidang'),('008','Staf')";
    mysqli_query($conn, $sql_j_i);

    $sql_p_i = "INSERT INTO tb_pegawai (nip, nama, alamat, kd_jabata
        n) VALUES ('197008121995121001','Zainal','Anduonohu,
        Kec. Laloara, Jl. Lumba-Lumba','001'),
        ('196706171994031010','Yono','Anduonohu, Jl. Jend. A
        H. Nasution','008'),'196512312006041114','Amrin',
        'Kambu, Jl. Lasitarda, Lrg. Tekukur','008')";
    mysqli_query($conn, $sql_p_i);

    echo "<script>alert('Tabel pegawai dan jabatan berhasil dibuat')
        </script>";
} else {
    echo "<script>alert('Tabel gagal dibuat/sudah ada')</script>";
}
mysqli_close($conn);
}

```

Berbeda dengan kode untuk membuat database, untuk membuat tabel terlebih dahulu harus ditentukan database apa yang digunakan. Maka dari itu dalam variable \$conn, di bagian kode setelah “password” harus ditambahkan lagi dengan “nama database” yaitu “db\_pegawai”. Lalu kode di atas juga menggunakan kondisi if else, dimana jika tombol yang diklik adalah “Buat Tabel” maka akan menjalankan program di dalamnya. Untuk SQL-nya yaitu **CREATE TABLE** , dimana jumlah tabel yang dibuat ada dua yaitu tb\_pegawai dan tb\_jabatan. Dalam SQL **CREATE TABLE**,



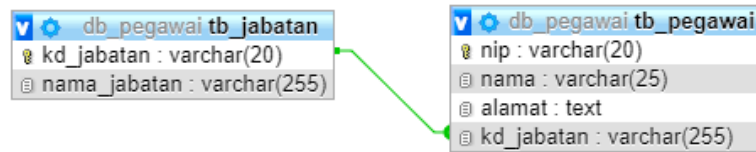
dalam programnya sekaligus dibuatkan atribut-atributnya. Salah satu contohnya yaitu `nip varchar(20) primary key`, yang berarti atribut yang dibuat bernama “nip”, tipe data yang digunakan “varchar”, jumlah datanya “20”, dan jika atribut yang dibuat merupakan primary key maka tuliskan “primary key”, jika tidak maka tulisan “primary key” dihilangkan.

Selain itu, dalam code sebelumnya juga memuat SQL untuk memasukkan data dengan menggunakan SQL `INSERT INTO ('nama-nama atribut'), VALUES ('data yang ingin dimasukkan')`. Jika value yang ingin dimasukkan lebih dari satu, maka cukup berikan lagi SQL `VALUES ('data yang ingin dimasukkan')`. yang dipisahkan dengan tanda koma “,” dari kode sebelumnya.

Sedangkan untuk menampilkan data-data yang dimasukkan (menggunakan relasi antar tabel), kode yang digunakan yaitu:

```
$sql = "SELECT * FROM tb_pegawai INNER JOIN tb_jabatan ON tb_pegawai.kd_
jabatan = tb_jabatan.kd_jabatan";
$hasil = mysqli_query($con, $sql);
$i = 1;
foreach ($hasil as $data) :
    echo "
        <td>" . $i++ . "</td>
        <td>" . $data["nip"] . "</td>
        <td>" . $data["nama"] . "</td>
        <td>" . $data["alamat"] . "</td>
        <td>" . $data["nama_jabatan"] . "</td>
        <td><a href=\"update.php?id=$data[nip]&btn=update\">update</a></td>
        <td><a href=\"index.php?id=$data[nip]&btn=delete\">delete</a></td>
    </tr>";
endforeach;
```

Untuk merelasikan kedua tabel, disini kita menggunakan `INNER JOIN` untuk merelasikan `tb_pegawai` dan `tb_jabatan`. Adapun atribut yang ditampilkan yaitu `nip`, `nama`, dan `alamat` dari `tb_pegawai`, sedangkan `nama_jabatan` diambil dari `tb_jabatan`. Dimana `kd_pegawai` pada tabel `tb_pegawai` bertindak sebagai foreign key dari `kd_pegawai` pada tabel `tb_jabatan`. Berikut adalah relasi dari kedua tabel tersebut.



**Gambar 2.1** Relasi antar tabel

Kemudian untuk tugas yang kedua, kita akan menambahkan 2 file lain yaitu “input.php” untuk menambah data pegawai dan “update.php” untuk mengubah/update data pegawai. Sebelum membuat program untuk menambah data pegawai, kita berikan terlebih dahulu link (pada index.php) untuk mengarahkan ke input.php, kode programnya yaitu:

```
<a href="input.php">Tambah Pegawai</a>
```

Setelah link di atas diklik maka akan mengarah ke input.php yang berisi kode berikut:

```
<h3>Tambah Pegawai</h3>
<form method="post" action="index.php">
  NIP : <input type="text" name="nip"><br>
  Nama : <input type="text" name="nama"><br>
  Alamat : <input type="text" name="alamat"><br>
  Jabatan : <select name="kd_jabatan">
    <option value="001">Kepala</option>
    <option value="002">Sekretaris</option>
    <option value="003">Fungsional Perencanaan</option>
    <option value="004">Kepala Sub Bagian</option>
    <option value="005">Kepala Bidang</option>
    <option value="006">Kepala Sub Bidang</option>
    <option value="007">Sub Bidang</option>
    <option value="008">Staf</option>
  </select>
  <input type="submit" name="btn" value="Send">
</form>
```

Dapat dilihat bahwa action diisi dengan index.php, yang berarti jika data-datanya telah diisi dan tombol submit diklik, maka kita akan kembali diarahkan ke halaman index.php, yang didalamnya terdapat kode untuk menambahkan data pegawai. Kodenya yaitu:

```
if ($_POST['btn'] == 'Send') {
  $nip = $_POST['nip'];
```

```

$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$kd_jabatan = $_POST['kd_jabatan'];
$sql = "INSERT INTO tb_pegawai (nip,nama,alamat,kd_jabatan)
        VALUES ('$nip','$nama','$alamat','$kd_jabatan')";
mysqli_query($con, $sql);
}

```

Sedangkan untuk mengubah/update data dan delete data, kita perlu menggunakan link juga seperti sebelumnya, kodenya yaitu:

```

<td><a href="\update.php?id=$data[nip]&btn=update\">update</a></td>
<td><a href="\index.php?id=$data[nip]&btn=delete\">delete</a></td>

```

Dari kode di atas dapat disimpulkan bahwa method yang digunakan yaitu `$_GET`, yang berarti jika link tersebut diklik, maka nilai `$data[nip]` otomatis dimasukkan ke dalam id dan btn akan terisi update atau delete (sesuai link yang diklik). Jika yang diklik adalah link update maka akan diarahkan ke `update.php`, adapun kode programnya yaitu:

```

<h3>Update Pegawai</h3>
<?php
$con = mysqli_connect("localhost", "root", "", "db_pegawai");
$sql = "SELECT * FROM tb_pegawai WHERE nip = '$_GET[id]'";
$hasil = mysqli_query($con, $sql);
$data = mysqli_fetch_row($hasil);
?>
<form method="post" action="index.php">
    NIP : <input type="text" name="nip" value="<?php echo $data[0] ?>">
    <br>
    Nama : <input type="text" name="nama" value="<?php echo $data[1] ?>">
    <br>
    Alamat : <input type="text" name="alamat" value="<?php echo $data[2] ?>">
    <br>
    Jabatan :
    <select name="kd_jabatan">
        <option value="001" <?=$data[3] == '001' ? "selected='selected'"
" : "" ?>>Kepala</option>
        <option value="002" <?=$data[3] == '002' ? "selected='selected'"
" : "" ?>>Sekretaris</option>
        <option value="003" <?=$data[3] == '003' ? "selected='selected'"
" : "" ?>>Fungsional Perencanaan</option>

```

```

        <option value="004" <?= $data[3] == '004' ? "selected='selected'"
" : "" ?>>Kepala Sub Bagian</option>
        <option value="005" <?= $data[3] == '005' ? "selected='selected'"
" : "" ?>>Kepala Bidang</option>
        <option value="006" <?= $data[3] == '006' ? "selected='selected'"
" : "" ?>>Kepala Sub Bidang</option>
        <option value="007" <?= $data[3] == '007' ? "selected='selected'"
" : "" ?>>Sub Bidang</option>
        <option value="008" <?= $data[3] == '008' ? "selected='selected'"
" : "" ?>>Staf</option>
    </select>
    <input type="submit" name="btn" value="update">
</form>

```

Pertama-tama, nilai id diambil terlebih dahulu menggunakan method \$\_GET dan dimasukkan ke dalam atribut “nip”, yang berarti data yang diubah bergantung pada nilai dari nip yang dimasukkan tersebut. Jika tombol submit diklik maka akan mengarah Kembali ke index.php yang berisi kode untuk update data, yaitu:

```

if ($_POST['btn'] == 'update') {
    $sql = "UPDATE tb_pegawai SET nip='$_POST[nip]', nama='$_POST[nama]',
        alamat='$_POST[alamat]', kd_jabatan='$_POST[kd_jabatan]'
        WHERE nip='$_POST[nip]'";
    mysqli_query($con, $sql);
}

```

Sedangkan jika link yang diklik adalah delete/hapus data, maka tetap akan diarahkan ke halaman yang sama (index.php) dengan kode sebagai berikut.

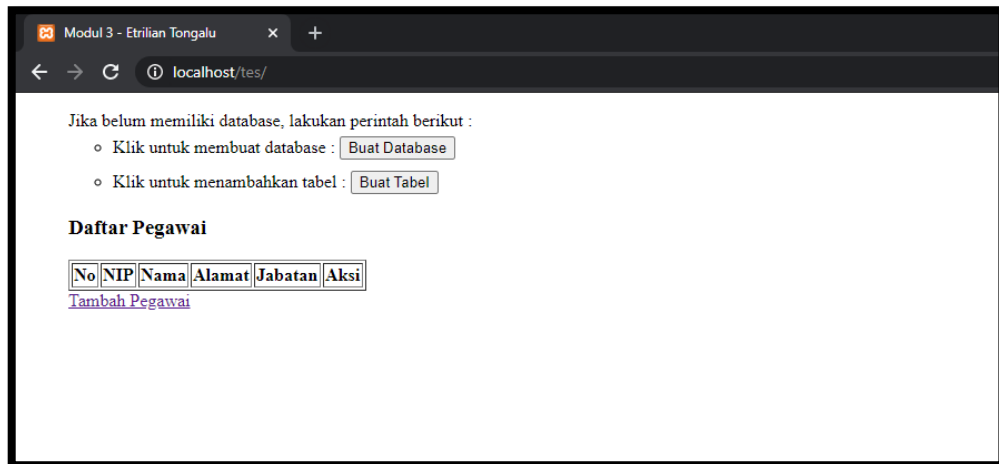
```

if ($_GET['btn'] == 'delete') {
    $sql = "DELETE FROM tb_pegawai WHERE nip='$_GET[id]'";
    mysqli_query($con, $sql);
}

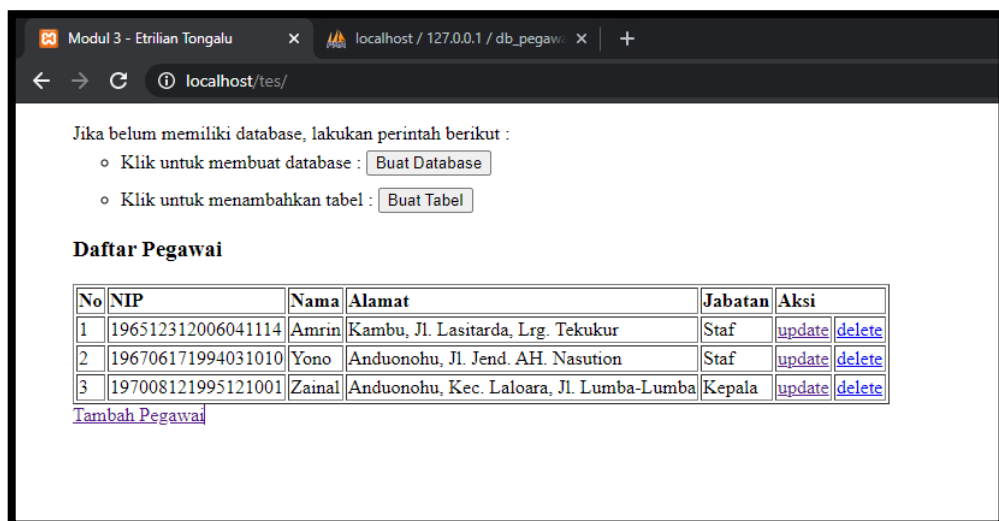
```

Data yang dihapus sesuai dengan nip yang dimasukkan dari nilai id yang diambil menggunakan method \$\_GET.

Output dari program ini jika dijalankan adalah sebagai berikut.



**Gambar 2.2** Output index.php jika database belum dibuat



**Gambar 2.3** Output index.php jika database sudah dibuat

localhost/tes/input.php

### Tambah Pegawai

NIP :

Nama :

Alamat :

Jabatan :

- Kepala
- Sekretaris
- Fungsional Perencanaan
- Kepala Sub Bagian
- Kepala Bidang
- Kepala Sub Bidang
- Sub Bidang
- Staf

**Gambar 2.4** Output input.php

localhost/tes/update.php?id=19670617199403101

### Update Pegawai

NIP :

Nama :

Alamat :

Jabatan :

**Gambar 2.5** Output update.php

## **KESIMPULAN**

MySQL adalah suatu program yang digunakan untuk mengelola database. Dengan menggunakan MySQL kita bisa mengelola database dengan menuliskan program dalam bahasa SQL, yaitu seperti membuat database, membuat tabel, memasukkan data, mengubah data, menghapus data, dan lain-lain.

Dalam membuat database, biasanya memerlukan hubungan antar tabel yang biasa disebut relasi antar tabel. Hal ini dapat dilakukan dengan Bahasa SQL, dengan menggunakan relasi INNER JOIN, LEFT JOIN, RIGHT JOIN, maupun ALIAS AS.

## **DAFTAR PUSTAKA**

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



# LAMPIRAN

