

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Muhammad Fauzan

NIM : 193020503024

KELAS : A

MODUL : III (PHP MySQL Database)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

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).

A. 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);**

B. 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

C. 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

D. 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

E. Mengambil Data Diri

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

F. 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

G. 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

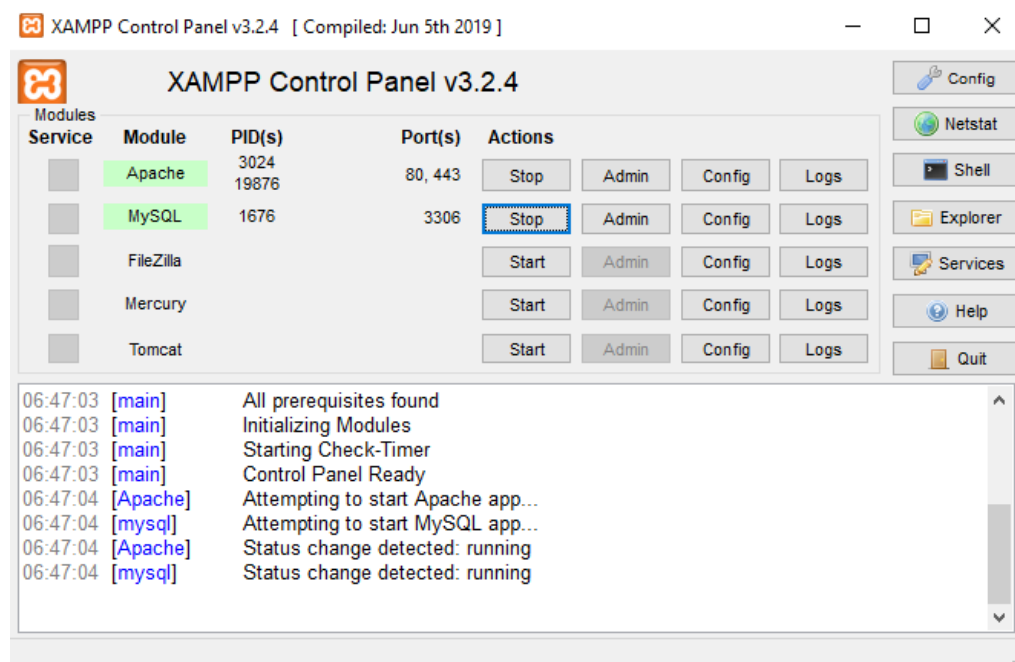
BAB II

PEMBAHASAN

2.1. Buatlah database pada MySQL untuk menyimpan data – data pegawai beserta relasi tabelnya.

Pada tugas praktikum ini praktikan diperintahkan untuk membuat sebuah database yang mempunyai sebuah relasi untuk menyimpan data data pegawai dengan relasi. Untuk membuat sebuah database dibutuhkan suatu web server, disini saya menggunakan Xampp sebagai web server.

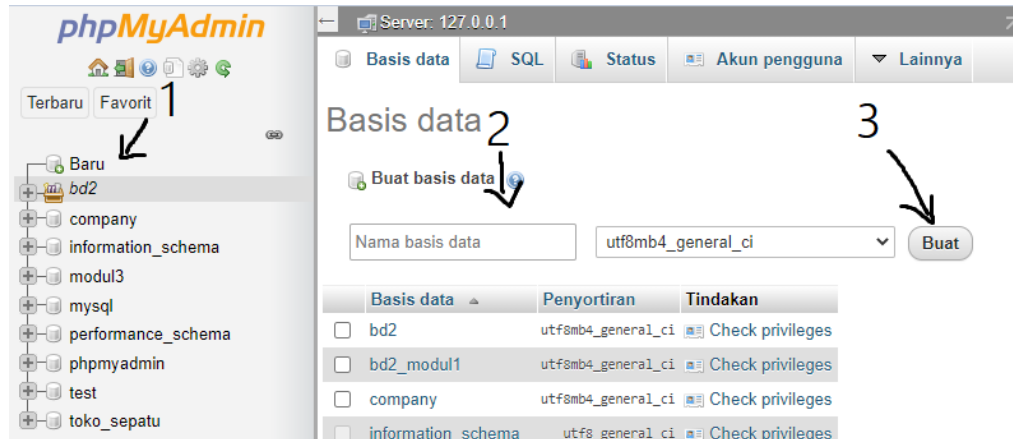
Pertama, Jalankan aplikasi web server dengan menyalakan Apache dan MySQL. Apache berfungsi untuk menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh developer. Dan MySQL berperan untuk mengolah, mengedit, dan menghapus data melalui database.



Gambar 2.1. Xampp

Kedua, setelah menyalakan apache dan MySQL masuk kehalaman browser dan ketikkan localhost/phpMyAdmin/ pada menu pencarian untuk

membuka halaman web server. Untuk membuat database baru klik new menu kiri yang fungsinya untuk membuat database baru. Lalu masukkan nama database setelah itu klik create.



Gambar 2.2. Create New Database

Ketiga, setelah membuat database lalu buat dua buah tabel dengan nama pegawai dan department. Dengan masing masing berisi atribut seperti dibawah ini.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	NIK	bigint(20)			Tidak	Tidak ada			Ubah Hapus Lainnya
2	nama	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	alamat	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	id_department	int(11)			Ya	NULL			Ubah Hapus Lainnya

Gambar 2.3. Tabel Pegawai

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama_depart	varchar(20)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya

Gambar 2.4. Tabel Departement

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

Pada tugas praktikum kali ini praktikan diminta untuk membuat sebuah system program yang dapat menyimpan atau menambah sebuah data, menghapus dan mengubah data pegawai yang berada pada database tersebut.

Agar program dapat terkoneksi dengan database yang ingin digunakan dibutuhkan suatu koneksi yang membuatnya terhubung. Berikut adalah syntax query yang dibutuhkan untuk membuat program website terhubung dengan database.

```
mysqli_connect("localhost", "root", "", "modul3");
```

Gambar 2.5. Koneksi Database

Terdapat juga beberapa function yang nantinya digunakan didalam program seperti program perulangan yang berfungsi untuk mengulangi untuk menampilkan data dari database untuk setiap barisnya. Dan terdapat 3 buah function yang berfungsi untuk menambahkan, menghapus dan mengubah data yang terkoneksi dengan database. Karena terdapat dua buah tabel dalam database maka disini saya menggunakan dua buah function berbeda untuk menambah kedalam data yaitu function tambah data untuk data pegawai dan function tambah data untuk data department. Namun karena terkesan sama jadi saya hanya akan menjelaskan salah satu dari dua tersebut.

```
function tambah($data){  
    global $db;  
    // ambil data dari tiap elemen dalam form  
    $NIK = htmlspecialchars($data["NIK"]);  
    $nama = htmlspecialchars($data["nama"]);  
    $alamat = htmlspecialchars($data["alamat"]);  
    $id_department = htmlspecialchars($data["id_department"]);  
  
    $query = "INSERT INTO pegawai  
            VALUES  
            ('$NIK', '$nama', '$alamat', '$id_department')  
            ";  
    // query insert data  
    mysqli_query($db, $query);  
  
    return mysqli_affected_rows($db);  
}
```

Gambar 2.6. Function Tambah

Pada baris pertama dalam badan function terdapat global db yang berfungsi untuk membaca variabel global. Terdapat juga 4 buah variabel dengan `htmlspecialchars` yang artinya jika user menginputkan namun dengan kode html untuk merusak system yang dibuat maka tidak akan bisa. Lalu terdapat variabel query yang berfungsi untuk menambahkan data kedalam database dengan menjalankan query `INSERT INTO table_name VALUES('attribut1', 'attribut2', 'attribut3',....);`

Terdapat function ubah data yang berfungsi untuk mengubah data yang telah ada. Dengan menggunakan statement query `UPDATE table_name SET('attribut1', 'attribut2', 'attribut3',....);`. Berikut ini adalah full source code dari function ubah data.

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

    return mysqli_affected_rows($db);
}
```

Gambar 2.7. Function Ubah

Terdapat juga function hapus yang berfungsi untuk menghapus data yang terhubung dengan database. Dengan menggunakan statement query `DELETE FROM table_name WHERE kondisi.`

```

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

```

Gambar 2.8. Function Hapus

setelah menjelaskan bagian dari back end program atau program yang berjalan di latar belakang kali ini saya akan menjelaskan bagian front end program yang berkaitan dengan tampilan program yang terdapat pada index.php.

```

<?php
require 'functionpegawai.php';
$pegawai = query("SELECT * FROM pegawai INNER JOIN department ON pegawai.id_department=department.id");
?>

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

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

            <table class="table table-striped">
                <thead>
                    <tr>
                        <th scope="col">No</th>
                        <th scope="col">NIK</th>
                        <th scope="col">Nama</th>
                        <th scope="col">Alamat</th>
                        <th scope="col">Department</th>
                        <th scope="col">Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php $i=1; ?>
                    <?php foreach($pegawai as $p) : ?>
                        <td><?php echo $i; ?></td>
                        <td><?php echo $p["NIK"]; ?></td>
                        <td><?php echo $p["nama"]; ?></td>
                        <td><?php echo $p["alamat"]; ?></td>
                        <td><?php echo $p["nama_depart"]; ?></td>
                        <td>
                            <a href="ubah_pegawai.php?id=<?php echo $p["NIK"]; ?>">ubah</a> |
                            <a href="hapus_pegawai.php?id=<?php echo $p["NIK"]; ?>" onclick="return confirm('yakin?');">hapus</a>
                        </td>
                    </tr>
                </tbody>
                <?php $i++; ?>
            <?php endforeach; ?>
            </table>
        </div>
    </div>
</div>

```

Gambar 2.9. Index

Dapat dilihat pada gambar diatas terdapat fungsi require 'functionpegawai.php'; yang berarti file index ini membutuhkan fungsi fungsi yang sudah dideklarasikan yang terdapat pada file php tersebut. Lalu terdapat variabel pegawai yang memiliki statement query *SELECT * FROM pegawai INNER JOIN department ON pegawai.id_department = department.id*);. Bagian divisi pada html ini berfungsi untuk menggabungkan menjadi satu divisi. Ada juga a href yang berfungsi untuk hyperlink ke arah file lain. Terdapat juga table yang berguna untuk membuat tabel dalam html dan pada html terdapat fungsi php yang membutuhkan function pada file yang berbeda. Yaitu pada bagian `<?php foreach($pegawai as $p) : ?>` sebagai pembuka perulangan lalu pembacaan data pada setiap baris `td <?= $p ["NIK"]?>` dan statement endforeach untuk menutup perulangan.

BAB III

KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa dalam pembuatan website yang menggunakan database dapat dilakukan menggunakan php dan mysql, dengan adanya mysql ini membuat programmer dapat menyimpan banyak data kedalam sebuah database dan dapat ditampilkan kedalam web dengan menggunakan php. Cara pengaplikasiannya yang sederhana dapat membantu para pemula belajar dalam pembuatan sebuah web yang di koneksikan dengan database sehingga data yang di inputkan kedalam web tersebut tidak hilang melainkan ikut masuk kedalam database.

DAFTAR PUSTAKA

- Alifia Seftin Oktriwina. (n.d.). *CRUD: Definisi, Fungsi, dan Cara Mengaplikasikannya* - *Glints Blog*. Diambil 12 April 2021, dari <https://glints.com/id/lowongan/crud-adalah/#.YHOkZ-gzb4Z>
- Praktikum, K. (n.d.). *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

```

<?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

```

<?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());
    }

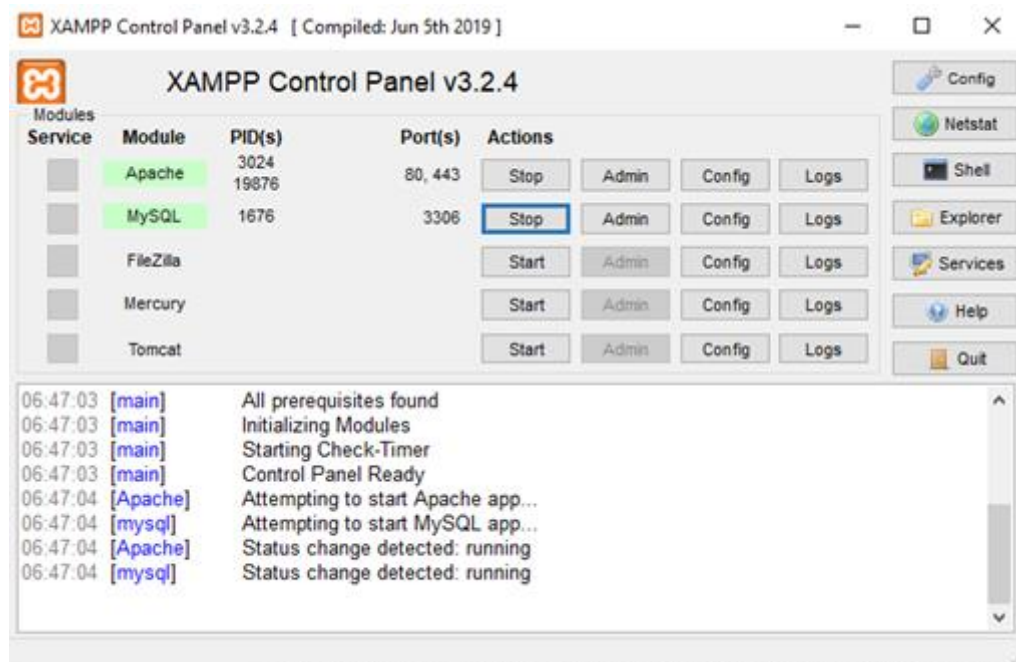
    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

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

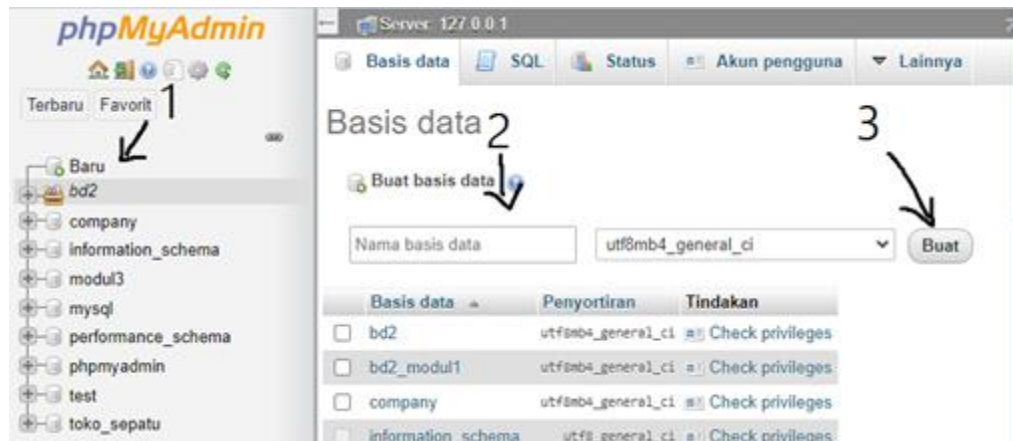
    mysqli_close($conn);
?>

```

Gambar 1.15. Update Data



Gambar 2.1. Xampp



Gambar 2.2. Create New Database

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	NiK	bigint(20)		Tidak	Tidak ada				Ubah Hapus Lainnya
2	nama	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	alamat	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	id_department	int(11)		Ya	NULL				Ubah Hapus Lainnya

Gambar 2.3. Tabel Pegawai

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)		Tidak	Tidak ada			AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama_depart	varchar(20)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya

Gambar 2.4. Tabel Departement

```
mysqli_connect("localhost", "root", "", "modul3");
```

Gambar 2.5. Koneksi Database

```

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

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

    return mysqli_affected_rows($db);
}

```

Gambar 2.6. Function Tambah

```

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

    return mysqli_affected_rows($db);
}

```

Gambar 2.7. Function Ubah

```

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

```

Gambar 2.8. Function Hapus

```

</php>
require 'functionpegawai.php';
$pegawai = query("SELECT * FROM pegawai INNER JOIN department ON pegawai.id_department=department.id");
?>

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

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

      <table class="table table-striped">
        <thead>
          <tr>
            <th scope="col">No</th>
            <th scope="col">NIK</th>
            <th scope="col">Nama</th>
            <th scope="col">Alamat</th>
            <th scope="col">Department</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
        <tbody>
          <php $i=1; ?>
          <php foreach($pegawai as $p) : ?>
            <td><php echo $i; ?></td>
            <td><? $p["NIK"]; ?></td>
            <td><? $p["nama"]; ?></td>
            <td><? $p["alamat"]; ?></td>
            <td><? $p["nama_depart"]; ?></td>
            <td>
              <a href="ubah_pegawai.php?id=<php echo $p["NIK"]; ?>">ubah</a> |
              <a href="hapus_pegawai.php?id=<php echo $p["NIK"]; ?>" onclick="return confirm('yakin?');">hapus</a>
            </td>
          </tr>
          </tbody>
          <php $i++; ?>
          <php endforeach; ?>
        </table>
      </div>
    </div>
  </div>
</div>

```

Gambar 2.9. Index