

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I



Nama : Sugeng Wahyu Nugroho
NIM : 193010503005
Kelas : C
Modul : III (PHP MYSQL DATABASE)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKARAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

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

1.2. LANDASAN TEORI

1.2.1. Pengertian MySQL

MySQL adalah sistem manajemen database relasional open source (RDBMS) dengan client-server model. Sedangkan RDBMS merupakan software untuk membuat dan mengelola database berdasarkan pada model relasional.

Beberapa orang bahkan membaca MySQL seperti sedang menyebutkan “my sequel”. MySQL AB, sebuah perusahaan asal Swedia, menjadi yang pertama dalam mengembangkan MySQL di tahun 1994. Hak kepemilikan MySQL kemudian diambil secara menyeluruh oleh perusahaan teknologi Amerika Serikat, Sun Microsystems, ketika mereka membeli MySQL AB pada tahun 2008. Di tahun 2010, Oracle yang adalah salah satu perusahaan teknologi terbesar di Amerika Serikat mengakuisisi Sun Microsystems. Semenjak itulah, MySQL sepenuhnya dimiliki oleh Oracle.

1.2.2. Database

Dalam bahasa yang sederhana, database adalah sekumpulan data yang terstruktur. Anggap saja Anda sedang berpose selfie: yang dilakukan pastilah menekan tombol kamera depan, lalu mengabadikan potret diri. Foto selfie yang diambil adalah data, sedangkan galeri foto merupakan database. Database adalah tempat untuk menyimpan dan mengelola data. Kata “relasional” berarti data yang disimpan di dataset

dikelola sebagai tabel. Setiap tabel saling terkait. Jika software tidak mendukung model data relasional, maka yang dipanggil hanyalah DBMS.

1.2.3. Open Source

Jika suatu software atau tool dikatakan open source, maka itu berarti Anda bebas menginstall, menggunakan, bahkan memodifikasinya. Anda pun dapat mempelajari dan mengkustomisasikan source code-nya agar software bisa diatur dan diubah sesuai keinginan dan kebutuhan Anda. Hanya saja, aktivitas yang Anda lakukan tak hanya tergantung pada situasi dan kondisi, tapi juga ditentukan oleh GPL (GNU Public License). Lisensi berbayar tersedia bagi Anda yang menginginkan fleksibilitas terhadap kepemilikan dan bantuan tingkat lanjut (premium).

1.2.4. Client-server Model

Komputer yang memasang dan menjalankan software RDBMS disebut sebagai client. Agar bisa mengakses data, komputer harus terhubung dengan server RDBMS terlebih dulu. Keadaan seperti inilah yang disebut client-server.

MySQL adalah salah satu pilihan software RDBMS. Terkadang RDBMS dan MySQL dianggap sama karena popularitas MySQL. Aplikasi web terkenal, seperti Facebook, Twitter, YouTube, Google, dan Yahoo! menggunakan MySQL untuk menyimpan data. Pada awalnya MySQL dibuat untuk penggunaan terbatas saja, tapi sekarang software ini sudah kompatibel dengan berbagai platform computing, seperti Linux, macOS, Microsoft Windows, dan Ubuntu.

1.2.5. SQL

MySQL dan SQL adalah dua software yang berbeda. MySQL merupakan salah satu nama brand terpopuler dari software RDBMS yang

menerapkan client-server model. Lalu, bagaimana client dan server berkomunikasi di dalam ruang lingkup RDBMS? Jadi, baik client maupun server, keduanya menggunakan bahasa spesifik domain – Structured Query Language (SQL). Jika Anda pernah melihat atau membaca beberapa nama yang dikombinasikan dengan SQL, misalnya PostgreSQL dan Microsoft SQL, maka server tersebut biasanya menggunakan syntax SQL. Walaupun terkadang ditulis dalam bahasa pemrograman yang lain, software RDBMS selalu menggunakan SQL sebagai bahasa utama untuk berinteraksi dengan database. MySQL sendiri ditulis dalam C dan C++. Agar lebih mudah dipahami, kita mengambil negara-negara di Amerika Selatan sebagai contohnya. Secara geografis, negara-negara tersebut tidaklah sama, bahkan sejarahnya pun berbeda. Namun, masyarakat di semua negara di Amerika Selatan menggunakan bahasa Spanyol untuk berkomunikasi.

Pada awal tahun 1970-an, seorang ahli komputer, Ted Codd, mengembangkan SQL dengan IBM berbasis model relasional. Pada tahun 1974, SQL mulai banyak digunakan dan dengan cepat menggantikan posisi bahasa yang sudah outdated, yakni ISAM dan VISAM. Tugas SQL adalah untuk memberitahukan server tentang apa yang harus dilakukannya terhadap data. Penggambaran umumnya seperti password atau kode WordPress. Anda memasukkan password atau kode tersebut ke sistem untuk mendapatkan akses agar bisa login ke dashboard. Dalam hal ini, SQL statement menginstruksikan server untuk menjalankan operasi tertentu:

- Data query: meminta informasi yang spesifik dari database yang sudah ada.
- Manipulasi data: menambahkan, menghapus, mengubah, menyortir, melakukan operasi lainnya untuk memodifikasi data, value, atau visual.

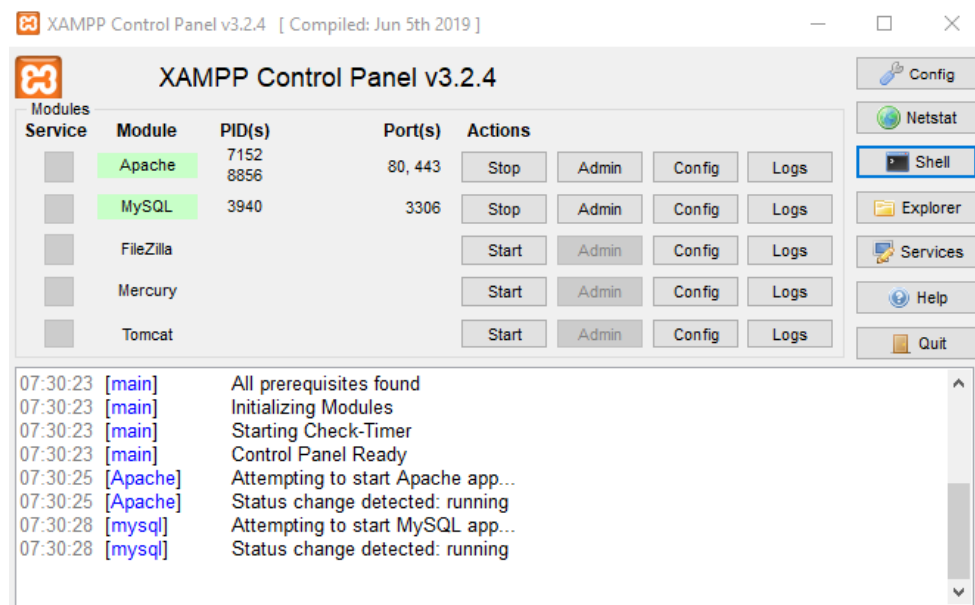
- Identitas data (data identity): menentukan tipe data, misalnya mengubah data numerik menjadi data integer. Selain itu, juga menentukan schema atau hubungan dari masing-masing tabel yang ada di database.
- Data access control: menyediakan metode keamanan untuk melindungi data, termasuk dalam menentukan siapa yang boleh melihat atau menggunakan informasi yang tersimpan di database.

BAB II

PEMBAHASAN

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

Pada soal pertama ini, praktikan arahkan untuk membuat database di MySQL sebagai tempat penyimpanan data-data pegawai. Untuk itu hal yang pertama dilakukan adalah menghidupkan aplikasi *XAMPP* terlebih dahulu dan mengklik start pada tombol *Apache* dan *MySQL*. Untuk lebih jelas bisa dilihat pada gambar berikut:

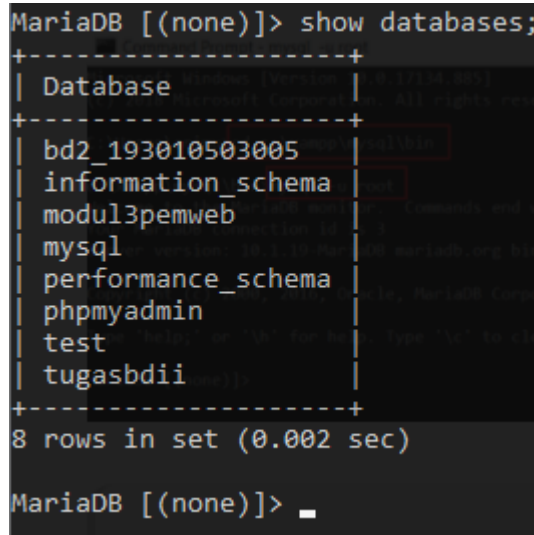


Gambar 2.1 Mengaktifkan aplikasi *XAMPP*

Apabila kedua teks berubah menjadi berwarna hijau, maka *apache* dan *mysql* sudah aktif. Langkah selanjutnya adalah membuka *commandprompt* atau bisa melalui *phpmyadmin*. Untuk penjelasan ini penulis menggunakan *commandprompt* sebagai tool. Untuk directory agar *command prompt* terbuka disesuaikan dengan dimana folder *xampp* terinstall. Karena disini *Xampp* terinstall secara default di folder *c:* Perintah yang diketikkan pada *commandprompt* adalah:

```
cd..  
  
cd..  
  
cd xampp\mysql\bin  
  
c:\xampp\mysql\bin(sesudah diinput)  
  
mysql -u root -p  
  
MariaDB [(none)]> (sesudah diinput)
```

Cd berarti mengeluarkan direktory agar bisa menuju ke direktory yang diinginkan. *Mysql -u root -p* memiliki makna bahwa ini merupakan perintah untuk masuk ke mode admin. Selanjutnya input kan perintah *show databases;* agar sistem menampilkan database apa saja yang telah dibuat sebelumnya. Disini penulis sudah membuat sebelumnya database dengan nama modul3PemWeb.



```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| bd2_193010503005 |  
| information_schema |  
| modul3pemweb |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
| tugasbdii |  
+-----+  
8 rows in set (0.002 sec)  
  
MariaDB [(none)]> _
```

Gambar 2.2 Menampilkan daftar database

Sehingga penulis langsung menggunakan saja databases yang telah dibuat, untuk melihat ada tabel apa saja yang berada pada database modul3pemweb. Ketikkan perintah *use modul3pemweb* dan jangan lupa akhir

dengan memberi tanda (;) dan apabila sudah terpilih ketikkan show tables maka table akan tampil. Untuk lebih jelas bisa dilihat pada gambar.

```
MariaDB [(none)]> use modul3pemweb;
Database changed
MariaDB [modul3pemweb]> show tables;
+-----+
| Tables_in_modul3pemweb |
+-----+
| karyawanot             |
+-----+
1 row in set (0.001 sec)

MariaDB [modul3pemweb]>
```

Gambar 2.3 Menampilkan table pada database

Dan didalam table karyawan, penulis sudah membuat field untuk masing-masing nilai agar bisa ditampilkan menggunakan bahasa php. Untuk field yang ada adalah *id*, *nama*, *nrp*, *jabatan*, *email*, *gambar*.

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

Masuk ke soal kedua, setelah membuat database dan memasukkan nilai-nilai kedalamnya, selanjutnya menghubungkan program PHP ke database MySQL. Maka dari itu praktikan harus membuat programnya agar bisa ditampilkan. Untuk code editor penulis menggunakan aplikasi *sublime text*. Buat file dengan nama *index.php*.

```
index.php x functions.php x tambah.php x hapus.php x
1 <?php
2 //menghubungkan file functions
3 require 'functions.php';
4
5 $karyawan = query("SELECT * FROM Karyawan");
6
7 ?>
8
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title>Modul 3</title>
14 </head>
15 <body>
```



```
index.php x functions.php x tambah.php x hapus.php x ubah.php

<h1>Daftar Karyawan</h1>
<a href="tambah.php">Tambah data Karyawan</a>
<br><br>
<table border="1" cellpadding="10" cellspacing="0">
  <tr>
    <th>No.</th>
    <th>Aksi</th>
    <th>Nama</th>
    <th>NRP</th>
    <th>Jabatan</th>
    <th>Email</th>
    <th>gambar</th>
  </tr>

  <?php $i = 1; ?>
  <?php foreach ($karyawan as $row): ?>

    <tr>
      <td><?= $i; ?></td>
      <td>
        <a href="ubah.php?id=<?= $row["id"]; ?>">ubah</a>|
        <a href="hapus.php?id=<?= $row["id"]; ?>" onclick=
          "return confirm('Apakah anda yakin?');">hapus</a>
      </td>
      <td><?= $row["nama"]; ?></td>
      <td><?= $row["nrp"]; ?></td>
      <td><?= $row["Jabatan"] ?></td>
      <td><?= $row["email"] ?></td>
      <td>" width=""></td>
    </tr>
  <?php $i++; ?>
  <?php endforeach; ?>
</table>
</body>
</html>
```

Gambar 2.4 program untuk menampilkan table

Karena disini penulis membagi fungsi program menjadi beberapa file, maka penulis menggunakan fungsi *require 'functions.php'* yang berfungsi

sebagai penghubung dengan file lain dari program ini. Query yang akan ditampilkan dijadikan menjadi variable agar apabila terjadi perubahan pada salah satu field, tidak usah mengubah semuanya tetapi yang dirubah hanya pada variable saja sehingga cukup memudahkan dalam masalah pengeditan program. Bisa dilihat pada perintah *\$karyawan query("SELECT * FROM Karyawan")*;

Selanjutnya menuliskan kode program yang akan menampilkan data dengan menggunakan html. Diberi nama judul *Modul3* dan dengan header *Daftar Karyawan*. Penulis membuat agar pada program nantinya langsung ada keterangan tamba data karyawan. Oleh karena itu penulis menambahkan halaman tujuan yang dibagi ke file yang bernama *tambah.php*. agar progra terlihat rapi, digunakanlah fungsi table pada program dengan setingan seperti pada gambar diatas.

Didalam nya juga akan menampilkan pilihan *ubah* dan *hapus* tetapi akan dijelaskan pada gambar berikutnya. Namun sekarang pada file ini hanya memasukkan perintah halaman tujuan apabila user mengklik link ubah dan hapus. Dan apabila menghapus akan ada pemberitahuan apakah user yakin ingin menghapus salah satu fild pada program. Dengan menggunakan sintaks *onclick* yang berarti apabila user mengklik akan di *return* atau dikembalikan nilai dan menampilkan pemberitahuan "*Apakah anda yakin?*" apabila dihapus maka nilai akan dihapus juga pada database.

Semua field dijadikan didefinisikan sebagai variable sehingga apabila terjadi perubahan cukup mengubah dari variabel tidak usah satuper satu. Tetapi untuk menampilkan perulangan maka didefinisikanlah dengan menggunakan fungsi *foreach* yang mana fieldpada tabel karyawan dijadikan sebagai row. Apabila user menambahkan field maka akan otomatis menambah lagi semua field tabel karyawannya. Khusus untuk gambar menggunakan fungsi

img/(nama file)). Untuk $\$i++$ bermakna nilai akan otomatis ditambahkan satu-per satu.

```
1 <?php
2 //koneksi ke database
3 $conn = mysqli_connect("localhost", "root", "", "modul3PemWeb")
4 ;
5
6 function query($query){
7     global $conn;
8     $result = mysqli_query($conn,$query);
9     $rows = [];
10    while ($row = mysqli_fetch_assoc($result)){
11        $rows[] = $row;
12    }
13    return $rows;
14 }
15
```

Gambar 2.5 program function untuk query

Program diatas ditulis pada file yang berbeda yakni file *function.php* yang merupakan file khusus fungsi-fungsi pada program sehingga apabila terjadi pengeditan cukup memudahkan dan tampilan program menjad lebih modular. `$conn = mysqli_connect("localhost", "root", "", "modul3Pemweb")`; merupakan variabel koneksi yang berfungsi menghubungkan program dengan databases yang berada di *phpmyadmin*. Mendefinisikan function query dengan variabel `$query`. Global `$conn` berguna menjadikan variabel koneksi bisa diakses dibagian mana saja pada badan program. `$result` adalah varibale yang menampung hasil dari fungsi `mysqli_query`.

Apabila fungsi dijalankan dan berhasil, maka hasil query akan tampil, akan tetapi apabila tidak berhasil maka akan muncul nilai boolean *false*. `$rows []`; berfungsi sebagai tempat sementara sebelum ditampilkannya hasil query. `Mysqli_fetch_assoc` berfungsi menampilkan nilai dalam bentuk string, sehingga memudahkan dalam memasukkan apabila hanya salah satu field yang ingin ditampilkan tinggal mengetikkan nama filed saja. Selanjutya hasil query akan dikembalikan atau ditampilkan.

```

function tambah($data){
    global $conn;

    //ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $nrp = htmlspecialchars($data["nrp"]);
    $jabatan = htmlspecialchars($data["jabatan"]);
    $email = htmlspecialchars($data["email"]);
    $gambar = htmlspecialchars($data["gambar"]);

    //query insert data
    $query = "INSERT INTO karyawan
              VALUES
              ('', '$nama', '$nrp', '$jabatan', '$email', '$
              gambar')";
    mysqli_query($conn, $query);
    return mysqli_affected_rows($conn);
}

```

Gambar 2.6 program function untuk tambah data

Membuat function untuk tambah data dengan menggunakan variabel *\$data*, sama dengan sebelumnya masih menggunakan global koneksi. Masing-masing field didefinisikan secara variabel. Penggunaan *htmlspecialchars* berfungsi sebagai penyimpan sementara dalam artian tidak langsung ditampilkan ke halaman namun disesuaikan dengan apa yang ingin ditampilkan. Hal ini sebagai pengaman pada program kita dari gangguan “Hacker”.

Pendefinisian variabel masing-masing field disesuaikan dengan urutan yang berada pada table database. Untuk perintah menambahkan data, sama dengan perintah pada saat melakukan penambahan data pada phpmyadmin. Sintaks yang digunakan adalah “*INSERT INTO*” nama table dan *VALUES*. Untuk id kenapa tidak diisi karena pada saat membuat database sebelumnya diset sebagai *primary key* yang mana akan menambahkan nilai secara otomatis. Setelah disimpan hasilnya dari koneksi dan hasil query maka akan

dikembalikan lagi ke koneksi sehingga di database akan otomatis ditambahkan juga.

```
function hapus($id){  
    global $conn;  
    mysqli_query($conn,"DELETE FROM karyawan WHERE id = $id");  
    return mysqli_affected_rows($conn);  
}
```

Gambar 2.7 program function untuk hapus data

Untuk function hapus tidak terlalu banyak, bisa dilihat pada gambar diatas. Untuk menghapus database konsep yang cukup dasar adalah cukup mencari *primary key* saja karena primary merupakan data unik yang tidak boleh terduplikasi sehingga cukup memasukkan \$id sebagai acuan dalam menghapus sebuah data.

```
function ubah($data){  
    global $conn;  
  
    //ambil data dari tiap elemen dalam form  
    $id = $data["id"];  
    $nama = htmlspecialchars($data["nama"]);  
    $nrp = htmlspecialchars($data["nrp"]);  
    $jabatan = htmlspecialchars($data["jabatan"]);  
    $email = htmlspecialchars($data["email"]);  
    $gambar = htmlspecialchars($data["gambar"]);  
  
    //query insert data  
    $query = "UPDATE karyawan SET  
            nama = '$nama',  
            nrp = '$nrp',  
            jabatan = '$jabatan',  
            email = '$email',  
            gambar = '$gambar'  
  
            WHERE id = $id  
  
            ";  
  
    mysqli_query($conn,$query);  
    return mysqli_affected_rows($conn);  
}
```

Gambar 2.8 program function untuk ubah data

Untuk program ubah ini program hampir sama dengan program tambah. Karena akan ditampilkan pada halaman yang sama. Yang berbeda adalah sintaks update pada data saja yang mana untuk fungsi update juga mengacu pada *primary key* karena kalau tidak program akan mengganti semua tabel dengan nilai yang sama.

```
<?php
require 'functions.php';
//cek tombol submit
if(isset($_POST["submit"])){

    //cek keberhasilan insert
    if(tambah($_POST) > 0){
        echo "
        <script>
            alert('data berhasil ditambah');
            document.location.href = 'index.php';
        </script>";
    }else{
        echo "
        <script>
            alert('data Gagal ditambah');
            document.location.href = 'index.php';
        </script>";
    }
}
?>
```

Gambar 2.9 program pada file tambah

Require berarti menghubungkan pada file *functions.php*, jika kita menekan submit, apabila berhasil maka akan muncul pemberitahuan *data berhasil ditambah*. Fungsi ini menggunakan javascript type alert sehingga munculnya dari atas bar halaman. Apabila yang dimasukkan tidak sesuai maka akan muncul pemberitahuan *data gagal ditambah*.

Pada bagian ini menggunakan permissalan *if* jika file tambah saat di post bernilai benar atau lebih besar dari 0, maka menampilkan pemberitahuan seperti tadi. Namun apabila sebaliknya maka akan menampilkan juga pemberitahuan sebaliknya.

```

<!DOCTYPE html>
<html>
<head>
  <title>Tambah data Karyawan</title>
</head>
<body>
  <h1>Tambah data karyawan</h1>
  <form action="" method="post" >
    <ul>
      <li>
        <label for ="nama">Nama : </label>
        <input type="text" name="nama" id="nama" required>
      </li>
      <li>
        <label for ="nrp">NRP : </label>
        <input type="text" name="nrp" id="nrp" required>
      </li>
      <li>
        <label for ="jabatan">Jabatan : </label>
        <input type="text" name="jabatan" id="jabatan" required>
      </li>
      <li>
        <label for ="email">Email : </label>
        <input type="text" name="email" id="email" required>
      </li>
      <li>
        <label for ="gambar">Gambar : </label>
        <input type="text" name="gambar" id="gambar">
      </li>
      <li>
        <button type="submit" name="submit">Tambah Data!</button>
      </li>
    </ul>
  </form>

```

Gambar 2.10 program pada file tambah

Pada bagian ini hanya menambahkan apa saja yang akan ditampilkan, sebagai tempat masing-masing field. Untuk penempatannya harus berurutan dengan field database yang berada pada phpmyadmin. Fungsi *required* berfungsi menandakan kolom tidak boleh kosong atau dalam artian wajib diisi oleh user.

```

<?php
require 'functions.php';

$id = $_GET["id"];

if(hapus($id) > 0 ){
  //cek keberhasilan Delete
  echo "
    <script>
      alert('data berhasil dihapus');
      document.location.href = 'index.php';
    </script>";
}else{
  echo "
    <script>
      alert('data Gagal dihapus');
      document.location.href = 'index.php';
    </script>";
}
}
>>

```

Fungsi hapus data, hampir sama dengan program sebelumnya yang berbeda adalah pemberitahuannya saja.

```
<?php
require 'functions.php';

//ambil data di URL
$id = $_GET["id"];

//query data mahasiswa berdasarkan id
$kry = query("SELECT * FROM Karyawan WHERE id = $id")[0];

//cek tombol submit
if(isset($_POST["submit"])){

    //cek keberhasilan ubah
    if(ubah($_POST) > 0){
        echo "
        <script>
            alert('data berhasil diubah');
            document.location.href = 'index.php';
        </script>";
    }else{
        echo "
        <script>
            alert('data Gagal diubah');
            document.location.href = 'index.php';
        </script>";
    }
}
?>
```

Gambar 2.11 *program pada file tambah*

Sama dengan sebelumnya disini yang menjadi acuan adalah id primary key sehingga program dapat mengetahui id yang mana yang ingin diubah.

BAB III

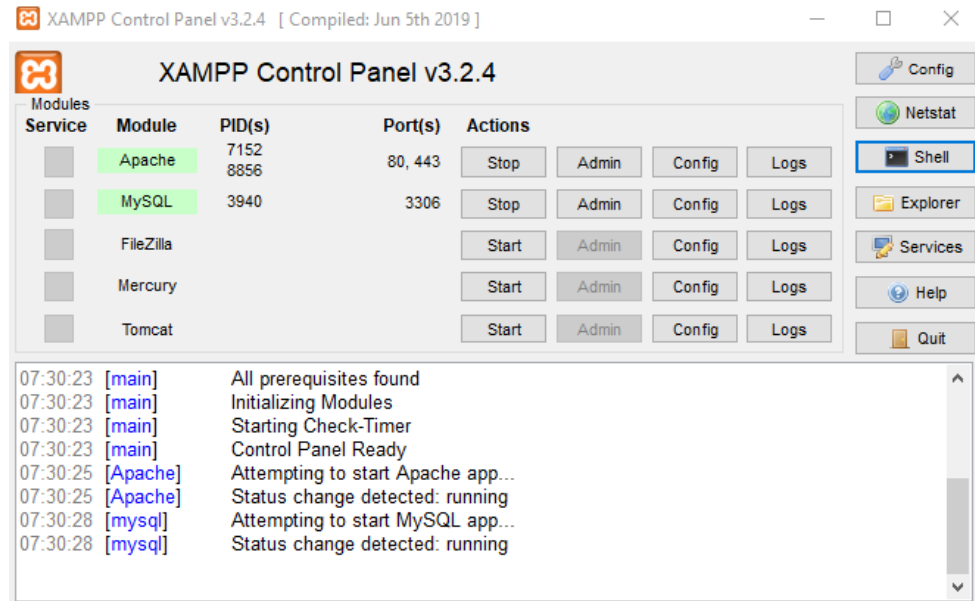
KESIMPULAN

Kesimpulan dari pembahasan diatas adalah, dalam membuat program yang ingin ditampilkan apalagi yang berhubungan dengan database, pada pembahasan ini banyak belajar tentang bagaimana cara menghubungkan program dengan database. Memisahkan atau memecah file menjadi beberapa file yang khusus menangani masing-masing fungsi. Menggunakan program yang berhubungang dengan data seperti data Karyawan tentunya sangatlah riskan jika tidak seefisien mungkin dalam mengelolanya. Oleh karena itu digunakanlah beberapa sintaks khusu dalam mengelola program.

DAFTAR ISI

- [1] Setiawan Dimas, “Cara membuat tabel di MySQL dengan CMD | Kelas Programmer,” *kelasprogrammer.com*, Aug. 06, 2019. <https://kelasprogrammer.com/cara-membuat-tabel-di-mysql-dengan-cmd/> (accessed Apr. 13, 2021).
- [2] A. C, “Apa Itu MySQL: Pembahasan Lengkap Tentang MySQL Bagi Pemula,” *www.hostinger.co.id*, 2020. <https://www.hostinger.co.id/tutorial/apa-itu-mysql> (accessed Apr. 13, 2021).
- [3] Jurusan Teknik Informatika, *MODUL PRAKTIKUM PEMROGRAMAN WEB I*. 2021.

LAMPIRAN



Gambar 2.1 Mengaktifkan aplikasi XAMPP

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| bd2_193010503005 |
| information_schema |
| modul3pemweb |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
| tugasbdii |
+-----+
8 rows in set (0.002 sec)

MariaDB [(none)]> _
```

Gambar 2.2 Menampilkan daftar database

```
MariaDB [(none)]> use modul3pemweb;
Database changed
MariaDB [modul3pemweb]> show tables;
+-----+
| Tables_in_modul3pemweb |
+-----+
| karyawan |
+-----+
1 row in set (0.001 sec)

MariaDB [modul3pemweb]>
```

Gambar 2.3 Menampilkan table pada database

```
index.php x functions.php x tambah.php x hapus.php x
1 <?php
2 //menghubungkan file functions
3 require 'functions.php';
4
5 $karyawan = query("SELECT * FROM Karyawan");
6
7 ?>
8
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title>Modul 3</title>
14 </head>
15 <body>

index.php x functions.php x tambah.php x hapus.php x ubah.php
<h1>Daftar Karyawan</h1>
<a href="tambah.php">Tambah data Karyawan</a>
<br><br>
<table border="1" cellpadding="10" cellspacing="0">
    <tr>
        <th>No.</th>
        <th>Aksi</th>
        <th>Nama</th>
        <th>NRP</th>
        <th>Jabatan</th>
        <th>Email</th>
        <th>gambar</th>
    </tr>

    <?php $i = 1; ?>
    <?php foreach ($karyawan as $row): ?>

        <tr>
            <td><?= $i; ?></td>
            <td>
                <a href="ubah.php?id=<?= $row["id"]; ?>">ubah</a>|
                <a href="hapus.php?id=<?= $row["id"]; ?>" onclick=
                    "return confirm('Apakah anda yakin?');">hapus</a>
            </td>
            <td><?= $row["nama"]; ?></td>
            <td><?= $row["nrp"]; ?></td>
            <td><?= $row["Jabatan"] ?></td>
            <td><?= $row["email"] ?></td>
            <td>" width=""></td>
        </tr>

    <?php $i++; ?>
    <?php endforeach; ?>
</table>
46
47
48 </body>
49
50 </html>
51
```

Gambar 2.4 program untuk menampilkan table

```

1  <?php
2  //koneksi ke database
3  $conn = mysqli_connect("localhost", "root", "", "modul3PemWeb")
4      ;
5
6  function query($query){
7      global $conn;
8      $result = mysqli_query($conn,$query);
9      $rows = [];
10     while ($row = mysqli_fetch_assoc($result)){
11         $rows[] = $row;
12     }
13     return $rows;
14 }
15

```

Gambar 2.5 *program function untuk query*

```

function tambah($data){
    global $conn;

    //ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);
    $nrp = htmlspecialchars($data["nrp"]);
    $jabatan = htmlspecialchars($data["jabatan"]);
    $email = htmlspecialchars($data["email"]);
    $gambar = htmlspecialchars($data["gambar"]);

    //query insert data
    $query = "INSERT INTO karyawan
              VALUES
              ('','$nama','$nrp','$jabatan','$email','$
              gambar')";
    mysqli_query($conn,$query);
    return mysqli_affected_rows($conn);
}

```

Gambar 2.6 *program function untuk tambah data*

```
function hapus($id){
    global $conn;
    mysqli_query($conn,"DELETE FROM karyawan WHERE id = $id");
    return mysqli_affected_rows($conn);
}
```

Gambar 2.7 *program function untuk hapus data*

```
function ubah($data){
    global $conn;

    //ambil data dari tiap elemen dalam form
    $id = $data["id"];
    $nama = htmlspecialchars($data["nama"]);
    $nrp = htmlspecialchars($data["nrp"]);
    $jabatan = htmlspecialchars($data["jabatan"]);
    $email = htmlspecialchars($data["email"]);
    $gambar = htmlspecialchars($data["gambar"]);

    //query insert data
    $query = "UPDATE karyawan SET
        nama = '$nama',
        nrp = '$nrp',
        jabatan = '$jabatan',
        email = '$email',
        gambar = '$gambar'

        WHERE id = $id

    ";

    mysqli_query($conn,$query);
    return mysqli_affected_rows($conn);
}
```

Gambar 2.8 *program function untuk ubah data*

```

<?php
require 'functions.php';
//cek tombol submit
if(isset($_POST["submit"])){

    //cek keberhasilan insert
    if(tambah($_POST) > 0){
        echo "
        <script>
            alert('data berhasil ditambah');
            document.location.href = 'index.php';
        </script>";
    }else{
        echo "
        <script>
            alert('data Gagal ditambah');
            document.location.href = 'index.php';
        </script>";
    }
}
?>

```

Gambar 2.9 program pada file tambah

```

<!DOCTYPE html>
<html>
<head>
    <title>Tambah data Karyawan</title>
</head>
<body>
    <h1>Tambah data karyawan</h1>
    <form action="" method="post" >
        <ul>
            <li>
                <label for ="nama">Nama : </label>
                <input type="text" name="nama" id="nama" required>
            </li>
            <li>
                <label for ="nrp">NRP : </label>
                <input type="text" name="nrp" id="nrp" required>
            </li>
            <li>
                <label for ="jabatan">Jabatan : </label>
                <input type="text" name="jabatan" id="jabatan" required>
            </li>
            <li>
                <label for ="email">Email : </label>
                <input type="text" name="email" id="email" required>
            </li>
            <li>
                <label for ="gambar">Gambar : </label>
                <input type="text" name="gambar" id="gambar">
            </li>
            <li>
                <button type="submit" name="submit">Tambah Data!</button>
            </li>
        </ul>
    </form>

```

Gambar 2.10 program pada file tambah

```

<?php
require 'functions.php';

$id = $_GET["id"];

if(hapus($id) > 0 ){
    //cek keberhasilan Delete
    echo "
        <script>
            alert('data berhasil dihapus');
            document.location.href = 'index.php';
        </script>";
    }else{
        echo "
        <script>
            alert('data Gagal dihapus');
            document.location.href = 'index.php';
        </script>";
    }
}
?>

```

```

<?php
require 'functions.php';

//ambil data di URL
$id = $_GET["id"];

//query data mahasiswa berdasarkan id
$kry = query("SELECT * FROM Karyawan WHERE id = $id")[0];

//cek tombol submit
if(isset($_POST["submit"])){
    //cek keberhasilan ubah
    if(ubah($_POST) > 0){
        echo "
            <script>
                alert('data berhasil diubah');
                document.location.href = 'index.php';
            </script>";
    }else{
        echo "
            <script>
                alert('data Gagal diubah');
                document.location.href = 'index.php';
            </script>";
    }
}
?>

```