

**LAPORAN HASIL PRAKTIKUM  
PEMROGRAMAN WEB I**



**NAMA : ADITYA ANANDA RAMADHANI**  
**NIM : 193030503072**  
**KELAS : A**  
**MODUL : III (PHP MySQL Database)**

**JURSAN 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.

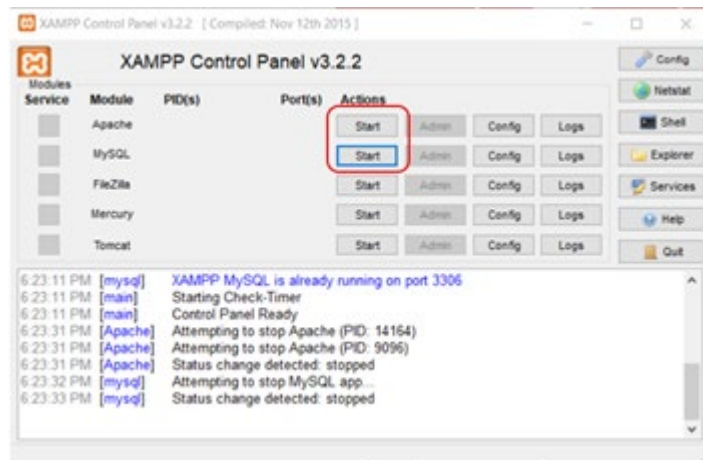
#### **1.2. LANDASAN TEORI**

##### **1.2.1. MEMBUAT KONEKSI PHP KE MYSQL**

PHPMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui World Wide Web. PHPMyAdmin mendukung berbagai operasi MySQL diantaranya mengelola basis data. PHPMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Sedangkan MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL. SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain C, Java, PHP, Smalltalk. Berikut merupakan cara mengkonesikan PHP ke MySQL :

##### **A. Pembuatan database**

Sebelumnya aktifkan dulu XAMPPnya dengan mengklik bagian Start di bagian Apache dan MySQL dan tunggu sampai tombol “Start” berwarna hijau. Berikut tampilannya:



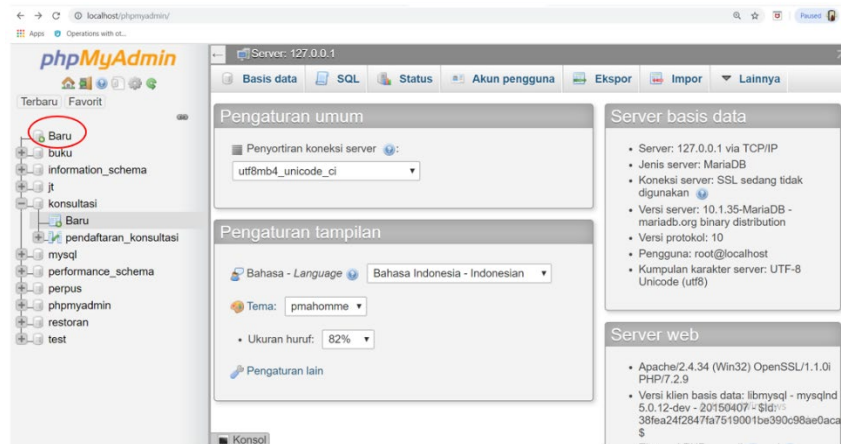
Gambar 1.1. Mengaktifkan Apache dan MySQL di XAMPP

Apabila Apache dan MySQL sudah aktif, buka PHPMyAdmin di localhost lalu pilih phpMyAdmin :



Gambar 1.2 Membuka PHPMyAdmin

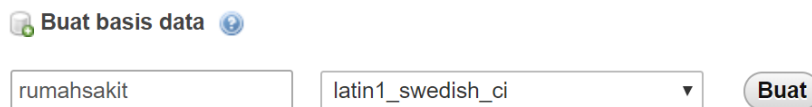
Setelah diklik phpMyAdmin, maka akan muncul tampilan seperti berikut :



Gambar 1.3. Tampilan PHPMyAdmin

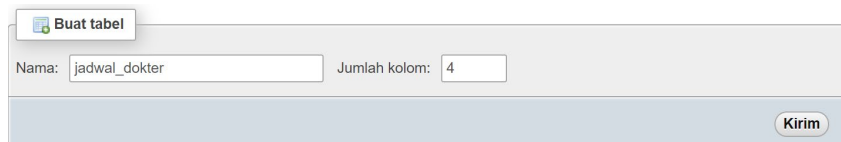
Klik tulisan “Baru” untuk membuat *database*. Buat *database* dengan nama “rumahsakit” lalu klik “Buat” seperti berikut:

## Basis data



Gambar 1.4. Membuat Database rumah sakit

Setelah itu buat tabel dengan nama “jadwal\_dokter” dengan jumlah kolom 4, lalu klik “Kirim” seperti berikut :



Gambar 1.5 Membuat Tabel

Apabila tabel berhasil dibuat, lalu tentukan nama kolom dan atributnya :

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran
id <small>Pick from Central Columns</small>	INT		Tidak ada	
nama <small>Pick from Central Columns</small>	VARCHAR	225	Tidak ada	
spesialis <small>Pick from Central Columns</small>	VARCHAR	225	Tidak ada	
hari <small>Pick from Central Columns</small>	VARCHAR	225	Tidak ada	

Gambar 1.6. Menentukan Kolom

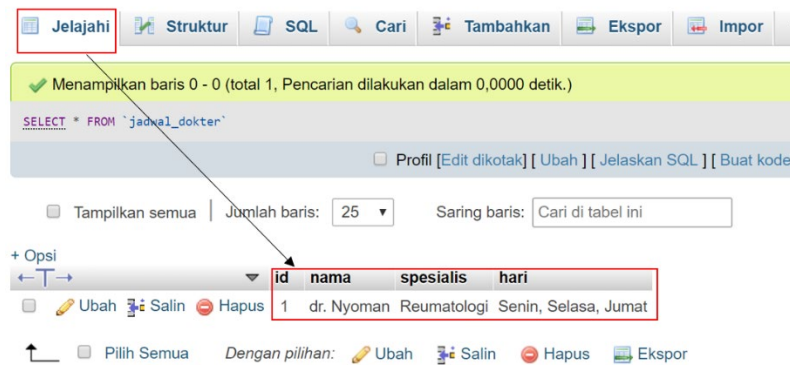
Ada 4 kolom yang dibuat yaitu “id” dengan tipe data INT (*integer*) dan diberi atribut AI (*Auto Increment*) agar idnya urut dari kecil ke besar tanpa harus menuliskan secara manual, “nama”, “spesialis”, dan “hari” dengan tipe data *varchar*. Kemudian klik “Simpan”. Kemudian untuk mengisi tabel tersebut, klik “Tambahkan” dan isi data sesuai kebutuhan lalu klik “Kirim” seperti berikut :

Kolom	Jenis	Fungsi	Tak Ternilai	Nilai
id	int(11)			1
nama	varchar(225)			dr. Nyoman
spesialis	varchar(225)			Reumatologi
hari	varchar(225)			Senin, Selasa, Jumat

**Kirim**

Gambar 1.7. Mengisi Tabel

Untuk memastikan data yang *diinput* sudah masuk ke database atau belum, klik “Jelajahi”. Jika data sudah masuk maka akan muncul data seperti berikut :



Gambar 1.8. Data yang Berhasil Diinput

## B. Koneksi PHP\_MySQL

Sebelumnya buat dulu folder untuk menyimpan script koneksi di folder “htdocs” yang berada dalam folder “XAMPP”. Di sini folder tersebut diberi nama “RS”. Setelah itu buka Sublime Text, Notepad++ atau Notepad untuk menulis scriptnya.

### 1) Script Koneksi

```
<?php
$kon = mysqli_connect('localhost','root','');
$link = mysqli_select_db($kon,"rumahsakit") or
die(mysqli_error());
?>
```

### 2) Script ini merupakan script untuk tampilan tabel di website.

Buka halaman baru di notepad atau sublime text, lalu copykan script di bawah ini dan simpan dengan nama “index.php”.

```
<?php
include "db_connect.php";
$query=mysqli_query ($kon, "SELECT * FROM
jadwal_dokter")or die (mysqli_error());
$jumlah = mysqli_num_rows($query);
echo "<html>";
echo "<head>";
echo "<link href='style.css' type='text/css' rel='stylesheet'>";
```

```

echo "</head>";
echo "<title>Jadwal Dokter</title>";
echo "<body>";
echo      "<font      color='darkmagenta'      face='Arial'
size=3><b><br>Jadwal      Dokter      RS      Sehat
Selalu</b></font><br><br>";
echo      "<a      href='add.php'      style=\"text-decoration:
none\"><font      face='tahoma'      size='1'>Masukkan      Jadwal
Baru</font></a><br>";
echo      "<br><table      border=\"0\"      cellpadding=\"1\"
cellspacing=\"1\" bordercolor=\"blue\" bgcolor=\"white\">
<tr bgcolor='blue' height=\"30\"><font color='white'>
      <th align='center'><font      color='white'      face='Arial'
size=2>No</font></th>
      <th align='center'><font      color='white'      face='Arial'
size=2>Nama</font></th>
      <th align='center'><font      color='white'      face='Arial'
size=2>Spesialis</font></th>
      <th align='center'><font      color='white'      face='Arial'
size=2>Hari</font></th>
      <th align='center'><font      color='yellow'      face='Arial'
size=2>Ubah?</font></th>
</tr>";
$j=0;
while ($row=mysqli_fetch_array($query)) {
    echo "<tr><td align='left' bgcolor='#657FFF'>";
    echo "<font face='Arial' size=1>";
    echo $j+1;
    echo"</font>";
    echo"</td>";
    echo "<td align='left' bgcolor='#E8D3DF'>";
    echo "<font face='Arial' size=1>";
    echo $row["nama"];
    echo"</font>";
    echo"</td>";
    echo "<td align='left' bgcolor='#E8D3DF'>";
    echo "<font face='Arial' size=1>";
    echo $row["spesialis"];
    echo"</font>";
    echo"</td>";
    echo "<td align='left' bgcolor='#E8D3DF'>";
    echo "<font face='Arial' size=1>";
    echo $row["hari"];
    echo"</font>";
    echo"</td>";
    echo"<td align='left' bgcolor='#E8D3DF'>";

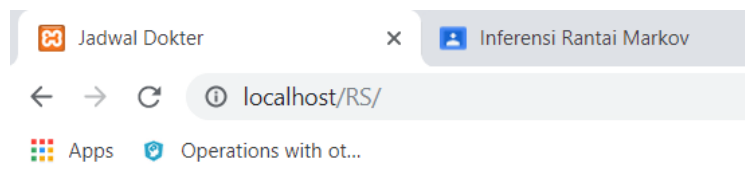
```

```

echo "<a href='delete.php?id=".$row['id']."' style=\"text-
decoration: none\" title=\"Hapus\"><font face='tahoma'
size='1'>Hapus</font></a>
<a href='edit.php?id=".$row['id']."' style=\"text-
decoration: none\" title=\"Edit\"><font face='tahoma'
size='1'>Edit</font></a>"; $j++; }
echo "</table>";
echo "</body>";
echo "</html>";
?>

```

Untuk mengecek tampilan website dengan tabel yang dibuat, ketikkan “localhost/RS” di browser. Tampilannya seperti ini:



## Jadwal Dokter RS Sehat Selalu

Masukkan Jadwal Baru

No	Nama	Spesialis	Hari	Ubah?
1	dr. Nyoman	Reumatologi	Senin, Selasa, Jumat	Hapus Edit

Gambar 1.9 tampilan awal table jadwal dokter

### 3) Script Add

Simpan script berikut dengan nama “add.php”.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Tambah Jadwal Baru</title>
<link href="style.css" type="text/css" rel="stylesheet">
</head>
<body>
<form action="insert_jadwal.php" method="POST">

```



```

<font face="Tahoma" color="green"
size="1"><b><br>Masukkan Jadwal Baru</b></font>
<table align="left">
<tr>
<td><font face="Tahoma" color="black"
size="2">nama</font></td>
<td>:</td>
<td><input type="text" name="nama"
size="30"></td>
</tr>

<tr>
<td><font face="Tahoma" color="black"
size="2">spesialis</font></td>
<td>:</td>
<td><input type="text" name="spesialis"
size="30"></td>
</tr>

<tr>
<td><font face="Tahoma" color="black"
size="2">hari</font></td>
<td>:</td>
<td><input type="text" name="hari"
size="30"></td>
</tr>

<tr>
<td></td><td></td>
<td><input type="submit" value="Add">
<font face="Tahoma" color="green" size="2">
<a href="index.php" style="text-
decoration:none">back</font></a>
</td>
</tr>
</table>
</form>
</body>
</html>

```

#### 4) Script Insert

Simpan script berikut dengan nama “insert\_jadwal.php”.

```

<?php
include "db_connect.php";
$nama = $ POST['nama'];

```

```

$spesialis = $_POST['spesialis'];
$hari      = $_POST['hari']; $query=mysqli_query($kon,
"INSERT INTO jadwal_dokter(nama, spesialis, hari)
VALUES ('$nama', '$spesialis', '$hari')")or die
(mysqli_error());

if($query) {
?>
<script language="JavaScript">
    document.location='index.php'</script>
<?php
}
?>

```

#### 5) Script Delete

Simpan script berikut dengan nama “delete.php”.

```

<?php
include "db_connect.php";
$id = $_GET['id'];

$query = "DELETE FROM jadwal_dokter WHERE id =
$id";
mysqli_query($kon, $query);

?>
<script language="JavaScript">
document.location='index.php'</script>
<?php
?>

```

#### 6) Script update

Simpan script berikut dengan nama “update.php”.

```

<?php
include "db_connect.php";
$id = $_GET['id'];

$nama = $_POST['nama'];
$spesialis = $_POST['spesialis'];
$hari = $_POST['hari'];

$query=mysqli_query ($kon, "UPDATE jadwal_dokter SET
nama='$nama',

```

```
spesialis='$spesialis', hari='$hari' WHERE id='$id')or die
(mysql_error());

if($query) {
?>
<script language="JavaScript">
document.location='index.php'</script>
<?php
}
?>
```

### 1.2.2. Fungsi Query PHP-MySQL

Untuk menjalankan query MySQL, PHP menyediakan fungsi `mysql_query()`. Fungsi ini dijalankan setelah koneksi berhasil dilakukan dengan fungsi `mysql_connect()`. Fungsi `mysql_query()` merupakan fungsi yang akan sering digunakan, karena dengan fungsi inilah PHP akan melakukan 'percakapan' dengan MySQL.

Fungsi `mysql_query()` membutuhkan 2 argumen, dimana argumen pertama diisi dengan query MySQL, dan argumen kedua diisi dengan link koneksi hasil fungsi `mysql_connect()`. Argumen kedua ini bersifat opsional. Berikut adalah format dasar penulisan fungsi `mysql_query()` dalam PHP :

```
$result = mysql_query("query_mysql",[$link_koneksi_mysql]);
```

Penjelasan :

- a) `$result` adalah variabel yang akan menampung hasil dari fungsi `mysql_query()`. Jika fungsi `mysql_query()` berhasil dijalankan, variabel `$result` akan berisi hasil query, namun jika query gagal, variabel ini akan berisi nilai boolean `FALSE`. Variabel `$result` bertipe `resources` (sama dengan hasil dari fungsi `mysql_connect()`). Nama dari variabel ini tidak harus ditulis sebagai `$result`, anda bebas jika ingin menggantinya dengan nama lain.
- b) `'query_mysql'` adalah argumen pertama fungsi `mysql_query()`. Pada bagian inilah kita menulis query MySQL. Query tersebut selanjutnya akan dikirim kepada MySQL Server untuk diproses.

Argumen ini harus pertipe string dan mengikuti aturan penulisan string didalam PHP. Contoh query ini, misalnya “SELECT \* FROM mahasiswa”, atau “CREATE DATABASE universitas”.

- c) \$link\_koneksi\_mysql adalah argumen kedua dari fungsi mysql\_query(). Argumen ini diisi dengan variabel hasil pemanggilan fungsi mysql\_connect(). Argumen ini bersifat opsional, dan jika diabaikan PHP akan menggunakan variabel mysql\_connect() yang sedang terkoneksi saat ini.

### 1.2.3. ELEMEN FORM

Element form berfungsi untuk mendefenisikan form intraktif. Elemen ini mempunyai attribute yaitu action dan method. Attribute action berisikan aksi terhadap form yang dikirim (url) dan attribute methode berisikan metode form melakukan proses pengiriman data (GET/POST). (Jayanti and Raja 2020)

Contoh :

```
<form action="..." method="...">
</form>
```

### 1.2.4. ELEMEN INPUT

Element INPUT berfungsi untuk mendefinisikan input yang akan dimasukkan pengguna. Element ini mempunyai attribute yaitu name, size, type, value, checked. Attribute name mendefinisikan nama dari input kontrol form, attribute size mendefinisikan ukuran teks pada input kontrol, type mendefinisikan bentuk-bentuk input kontrol, value mendefinisikan nilai awal/reset/submit, checked mendefinisi-kan pilihan terpilih pada type radio/checkbox. Element ini tidak mempunyai tag penutup dan harus berada di dalam element FORM. (Jayanti and Raja 2020)

Contoh :

<code>&lt;input name = "name" size = "number" type = "text"   "checkbox"   "radio"   "submit"   "reset" value = "value" checked &gt;</code>
---

#### 1.2.5. \$\_REQUEST (REQUEST)

\$\_REQUEST PHP digunakan untuk mengumpulkan data setelah submit sebuah form HTML. Contoh berikut menunjukkan sebuah form dengan field input dan tombol submit. Ketika user melakukan submit data dengan menekan “Submit”, data form dikirimkan untuk file yang dispesifikasikan dalam atribut aksi dari tag <form>. Dalam contoh ini, kita memperhatikan file ini memproses data form. Jika pengguna ingin menggunakan file PHP lain untuk memproses data form, menggantinya dengan nama file yang dipilih, maka gunakan variabel super global \$\_REQUEST untuk mengumpulkan field nilai input.

#### 1.2.6. \$\_POST (POST)

\$\_POST PHP sangat banyak digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method=“post”. \$\_POST juga banyak digunakan untuk melewati variabel. Contoh berikut menunjukkan sebuah form dengan sebuah input field dan sebuah tombol submit. Ketika user melakukan submit data dengan menekan tombol submit, data form dikirim ke file yang dispesifikasikan dalam atribut aksi dari tag <form>. Pada contoh ini, perhatikan file untuk pemrosesan data. Jika ingin menggunakan file PHP lain untuk memproses data form, ganti dengan nama file yang diinginkan. Kemudian, gunakan variabel super global \$\_POST untuk mengumpulkan nilai field input.

#### 1.2.7. \$\_GET (GET)

\$\_GET PHP bisa digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method=“get”. Juga bisa digunakan untuk mengumpulkan data yang dikirimkan dalalam URL.

### **1.2.8. PERBEDAAN GET DAN POST**

**1.2.9.** GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, ...)). Array ini menyimpan pasangan kunci/nilai, dimana kunci-kunci adalah nama-nama dari form control dan nilai-nilai adalah data input dari user. Method GET diakses menggunakan `$_GET` dan method POST diakses menggunakan `$_POST`. Kedua variabel ini adalah variabel superglobal, yang selalu bisa diakses, tanpa memperhatikan lingkup dan bisa diakses dari fungsi, class atau file yang berbeda tanpa harus melakukan teknik khusus. `$_GET` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. `$_POST` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST.

## BAB II

### PEMBAHASAN

pembahasan berdasarkan materi soal yang ada pada modul Pemograman Web adalah sebagai berikut :

1. Membuat database pada MySql untuk menyimpan data-data pegawai beserta relasi tabelnya.

pertama yang harus dilakukan adalah membuat database “pegawai”, yaitu sebagai berikut :

```
Create database pegawai;
```

Pada bagian diatas merupakan perintah SQL untuk membuat database bernama “pegawai”.

Selanjutnya adalah membuat table “pegawai” dan “departemen”, yaitu sebagai berikut :

```
CREATE TABLE `pegawai` (  
    `idpegawai` int(5) not null auto_increment,  
    `namapegawai` varchar(20),  
    `alamat` varchar(20),  
    `iddept` int(5)  
);
```

Pada bagian diatas merupakan perintah SQL untuk membuat table pegawai dengan attribute :

- 1) idpegawai dengan tipe data integer dan nilai tidak boleh null dan auto increment (otomatis naik +1 dari nilai sebelumnya)
- 2) namapegawai degan tipe data varchar dengan ukuran 20.
- 3) alamat degan tipe data varchar dengan ukuran 20.
- 4) iddept degan tipe data integer dengan ukuran 5.

```
CREATE TABLE `departemen` (  
    `iddept` int(5) NOT NULL AUTO_INCREMENT,  
    `namadep` varchar(20),  
    `lokasi` varchar(20)  
);
```

Pada bagian diatas merupakan perintah SQL untuk membuat table departemen dengan attribute :

- 1) idpegawai dengan tipe data integer dan nilai tidak boleh null dan auto increment (otomatis naik +1 dari nilai sebelumnya)
- 2) namadep dengan tipe data varchar dengan ukuran 20.
- 3) lokasi dengan tipe data varchar dengan ukuran 20.

Berikutnya membuat relasi antara table dengan menentukan primary key dan foreign key adalah sebagai berikut.

Primary key dari table pegawai, yaitu :

```
ALTER TABLE `pegawai`  
ADD PRIMARY KEY (`idpegawai`);
```

Primary key dari table pegawai adalah attribute idpegawai.

Primary key dari table departemen, yaitu :

```
ALTER TABLE `departemen`  
ADD PRIMARY KEY (`iddept`);
```

Primary key dari table departemen adalah attribute iddept.

Foreign key yang menghubungkan antara table pegawai dan table departemen, yaitu :

```
ALTER TABLE `pegawai`  
ADD FOREIGN KEY (`iddept`) REFERENCES `departemen`  
(`iddept`);
```

Foreign key yang menghubungkan antara table pegawai dan table departemen adalah attribute iddept yang berada pada table pegawai yang berefrensi terhadap primary key departemen yaitu attribute iddept.

2. Membuat program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut.
  - a. Pengkoneksian database



Sebelum mengakses data dalam database SQL, kita harus terhubung ke server database SQL. Berikut adalah kode program php yang digunakan agar terhubung dengan server SQL :

```
<?php
$servername = "localhost";
$dbname = "pegawai";
$username = "root";
$password = "";

$conn = mysqli_connect($servername, $username, $password,
$dbname);
?>
```

Pada bagian awal terdapat beberapa variable adalah sebagai berikut :

- 1) Variable servername (\$servername) yang menyimpan nilai “localhost” atau nama dari server yang digunakan.
- 2) Variable database (\$dbname) yang menyimpan nilai “pegawai” atau nama dari database yang digunakan.
- 3) Variable username (\$username) yang menyimpan nilai “root” atau username dari server yang digunakan.
- 4) Variable password (\$password) yang menyimpan nilai “” atau password dari server yang digunakan.

Setelah itu terdapat variable conn (\$conn) yang menyimpan fungsi mysqli\_connect yang berfungsi agar php menjalankan argumen agar terkoneksi ke SQL server atau database. Pada argumen pertama dari fungsi mysqli\_connect(), dimana nilai dari variable servername (\$servername) adalah alamat lokasi SQL server yang dijalankan. Pada argumen kedua dari fungsi mysqli\_connect(), dimana nilai dari variable username (\$username) adalah username dari SQL server yang dijalankan. Pada argumen ketiga dari fungsi mysqli\_connect(), dimana nilai dari variable password (\$password) adalah password dari SQL server yang dijalankan. Pada argumen keempat dari fungsi mysqli\_connect(), dimana nilai dari variable database (\$dbname) adalah database dari SQL server yang dijalankan.

- b. Menampilkan data-data yang tersimpan di database.

Pada kode program php untuk pengkoneksian database ditambahkan fungsi untuk menampilkan sebagai berikut :

```
<?php
$servername = "localhost";
$dbname = "pegawai";
$username = "root";
$password = "";

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

function show($query){
    global $conn;
    $result = mysqli_query($conn, $query);
    $rows = [];
    while( $row = mysqli_fetch_assoc($result) ){
        $rows[] = $row;
    }
    return $rows;
}
?>
```

Pada bagian diatas ditambahkan function show atau fungsi menampilkan dengan parameter variabel query (\$query) atau query yang ingin ditampilkan. Didalam terdapat variable global conn (global \$conn). Selanjutnya terdapat variable result (\$result) yang menyimpan fungsi mysqli\_query yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah parameter yang berada di function yaitu variabel query (\$query) yang berfungsi untuk menjalankan perintah query berdasarkan parameter masukan function. Kemudian dideklarasikan variable rows yang menyimpan array (\$rows=[]). Setelah itu dijalankan perulangan while dengan variable masukan yaitu variable row (\$row) yang menyimpan fungsi mysqli\_fetch\_assoc yang digunakan untuk menampilkan data dengan parameter masukan yaitu variable result (\$result). Didalam perulangan while terdapat variable array rows (\$rows[]) yang

menyimpan nilai dari variable row (\$row). Terakhir dikembalikan nilai dari variable rows (\$rows).

Berikut adalah kode program php dan html yang digunakan agar menampilkan data-data tersimpan di dalam database :

```
<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
$data2 = show("SELECT * FROM departemen");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Modul3</title>
</head>
<body>
    <table border="1" cellpadding="10">
        <tr>
            <td>id pegawai</td>
            <td>nama pegawai</td>
            <td>alamat</td>
            <td>iddept</td>
            <td>aksi</td>
        </tr>
        <?php
        foreach ($data as $d) {
            ?>
            <tr>
                <td><?php echo $d["idpegawai"]; ?></td>
                <td><?php echo $d["namapegawai"]; ?></td>
                <td><?php echo $d["alamat"]; ?></td>
                <td><?php echo $d["iddept"]; ?></td>
                <td>
                    </td>
            </tr>
        <?php
        }
        ?>
    </table>
    <br></br>
    <table border="1" cellpadding="10">
```

```

<tr>
  <td>iddept</td>
  <td>namadept</td>
  <td>lokasi</td>
  <td>aksi</td>
</tr>
<?php
foreach ($data2 as $d) {
?>
  <tr>
    <td><?php echo $d["iddept"]; ?></td>
    <td><?php echo $d["namadep"]; ?></td>
    <td><?php echo $d["lokasi"]; ?></td>
    <td>
    </td>
  </tr>
<?php
}
?>
</table>
</body>
</html>

```

Pada bagian diatas terdapat fungsi require(“function.php”) yang digunakan untuk menyeratakan file php lain yaitu function.php ke dalam php. Kemudian dideklarasikan variable data (\$data) yang memanggil function show dengan parameter masukan “SELECT \* FROM pegawai” dan dideklarasikan juga variable data2 (\$data2) yang memanggil function show dengan parameter masukan “SELECT \* FROM departemen”. Selanjutnya terdapat kode program html, pada bagian head dari html terdapat tag title “<title>Modul3</title>” yang digunakan untuk menampilkan judul halaman yaitu Modul3. Pada bagian body terdapat beberapa bagian yaitu :

- 1) terdapat tag table yang digunakan untuk membungkus tabel (<table border="1" cellpadding="10">). Attribute border memiliki nilai 1 yang berguna untuk mengatur ukuran garisnya dan attribute cellpadding memiliki nilai 10 yang berguna untuk mengatur jarak antara teks sel dengan garis.

Kemudian terdapat tag table row (`<tr>`) untuk membuat baris 1 pertama table. Didalam tag table row (`<tr>`), terdapat beberapa tag table data (`<td>`). pada tag table data (`<td>`) pertama adalah menampilkan kalimat “id pegawai”. tag table data (`<td>`) kedua adalah menampilkan kalimat “nama pegawai”. tag table data (`<td>`) ketiga adalah menampilkan kalimat “alamat”. tag table data (`<td>`) keempat adalah menampilkan kalimat “idept”. tag table data (`<td>`) kelima adalah menampilkan kalimat “aksi”. Setelah itu baris pertama table ditutup dengan menggunakan tag penutup table row (`</tr>`).

Setelah baris pertama selesai, untuk baris kedua terdapat perulangan foreach yang digunakan untuk melakukan perulangan terhadap data array dengan parameter masukannya adalah variable data (`$data`) yang dialiaskan menjadi variable d (`$d`). baris kedua dilanjutkan dengan menggunakan tag table row (`<tr>`). Didalam tag table row (`<tr>`) baris kedua, terdapat beberapa tag table data (`<td>`). pada tag table data (`<td>`) pertama adalah menampilkan perintah php echo dari variable d (`$d`) dengan array yaitu attribute idpegawai. tag table data (`<td>`) kedua adalah menampilkan echo dari variable d (`$d`) dengan array yaitu attribute namapegawai. tag table data `<td>` ketiga adalah menampilkan echo dari variable d (`$d`) dengan array yaitu attribute alamat. tag table data `<td>` keempat adalah menampilkan echo dari variable d (`$d`) dengan array yaitu attribute idept. Setelah itu baris kedua table ditutup dengan menggunakan tag penutup table row (`</tr>`).

- 2) terdapat tag table kedua yang digunakan untuk membungkus table yang kedua (`<table border="1" cellpadding="10">`). Attribute border memiliki nilai 1 yang berguna untuk mengatur ukuran garisnya dan attribute cellpadding memiliki nilai 10 yang berguna untuk mengatur jarak antara teks sel dengan garis.

Kemudian terdapat tag table row (<tr>) untuk membuat baris 1 pertama table. Didalam tag table row (<tr>), terdapat beberapa tag table data (<td>). pada tag table data (<td>) pertama adalah menampilkan kalimat “id departemen”. tag table data (<td>) kedua adalah menampilkan kalimat “nama departemen”. tag table data (<td>) ketiga adalah menampilkan kalimat “lokasi”. tag table data (<td>) keempat adalah menampilkan kalimat “aksi”. Setelah itu baris pertama table ditutup dengan menggunakan tag penutup table row (</tr>).

Setelah baris pertama selesai, untuk baris kedua terdapat perulangan foreach yang digunakan untuk melakukan perulangan terhadap data array dengan parameter masukannya adalah variable data2 (\$data2) yang dialiaskan menjadi variable d (\$d). baris kedua dilanjutkan dengan menggunakan tag table row (<tr>). Didalam tag table row (<tr>) baris kedua, terdapat beberapa tag table data (<td>). pada tag table data (<td>) pertama adalah menampilkan perintah php echo dari variable d (\$d) dengan array yaitu attribute iddept. tag table data (<td>) kedua adalah menampilkan echo dari variable d (\$d) dengan array yaitu attribute namadep. tag table data <td> ketiga adalah menampilkan echo dari variable d (\$d) dengan array yaitu attribute lokasi.

c. Menambahkan data pada database.

Pada kode program php untuk pengkoneksian database ditambahkan fungsi untuk menambahkan data sebagai berikut :

```
<?php
$servername = "localhost";
$dbname = "pegawai";
$username = "root";
$password = "";

$conn = mysqli_connect($servername, $username, $password,
$dbname);
```

```

function show($query){
    global $conn;
    $result = mysqli_query($conn, $query);
    $rows = [];
    while( $row = mysqli_fetch_assoc($result) ){
        $rows[] = $row;
    }
    return $rows;
}

function insert($data){
    global $conn;
    $namapegawai = $_POST["namapegawai"];
    $alamat = $_POST["alamat"];
    $iddept = $_POST["iddept"];

    mysqli_query($conn, "INSERT INTO pegawai
VALUES('$namapegawai','$alamat','$iddept')");
    return mysqli_affected_rows($conn);
}

function insert1($data2){
    global $conn;
    $namadep = $_POST["namadep"];
    $lokasi = $_POST["lokasi"];

    mysqli_query($conn, "INSERT INTO departemen
VALUES('$namadep','$lokasi')");
    return mysqli_affected_rows($conn);
}

```

Pada bagian diatas ditambahkan function insert dan insert1 yang berfungsi untuk menambahkan data dengan parameter variabel data (\$data) untuk insert dan variable data2 (\$data2) untuk insert1. Berikut merupakan penjelasan dari kedua function tersebut.

- 1) Pada function insert didalam terdapat variable global conn (global \$conn). Kemudian terdapat variable namapegawai yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute namapegawai. Selanjutnya terdapat variable alamat yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute alamat. Berikutnya terdapat variable iddept yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute iddept.

Selanjutnya terdapat fungsi `mysqli_query` yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable `conn ($conn)` yang berfungsi untuk memanggil database dan parameter kedua adalah `"INSERT INTO pegawai VALUES('$namapegawai','$alamat','$iddept')"` yang berfungsi untuk menjalankan perintah query untuk memasukan data ke table pegawai. Terakhir terdapat fungsi `mysqli_affected_rows` yang digunakan mengembalikan jumlah baris yang terpengaruh di `SELECT`, `INSERT`, `UPDATE`, `REPLACE`, atau `DELETE` query sebelumnya dengan parameter masukan adalah variabel `conn ($conn)`.

- 2) Pada function `insert2` didalam terdapat variable global `conn (global $conn)`. Kemudian terdapat variable `namadep` yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute `namadep`. Selanjutnya terdapat variable lokasi yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute lokasi. Selanjutnya terdapat fungsi `mysqli_query` yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable `conn ($conn)` yang berfungsi untuk memanggil database dan parameter kedua adalah `mysqli_query($conn, "INSERT INTO departemen VALUES('$namadep','$lokasi')");` yang berfungsi untuk menjalankan perintah query untuk memasukan data ke table departemen. Terakhir terdapat fungsi `mysqli_affected_rows` yang digunakan mengembalikan jumlah baris yang terpengaruh di `SELECT`, `INSERT`, `UPDATE`, `REPLACE`, atau `DELETE` query sebelumnya dengan parameter masukan adalah variabel `conn ($conn)`.

Setelah membuat function untuk memproses penambahan data pada database, selanjutnya adalah membuat kode program php sebagai tempat masukan bagi table yaitu table pegawai dan table departemen.



Coding program input untuk penambahan data pada table adalah sebagai berikut :

a) Insert table pegawai

```
<?php
require("function.php");
if (isset($_POST["insert"])) {
    if (insert($_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>";
    }
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Insert</title>
</head>

<body>
    <a href="index.php">home</a>
    <form action="" method="POST">
        <ul>
            <li>
                <label for="namapegawai">namapegawai</label>
                <input type="text" name="namapegawai"
id="namapegawai">
            </li>
            <li>
                <label for="alamat">alamat</label>
                <input type="text" name="alamat" id="alamat">
            </li>
            <li>
                <label for="iddept">iddept</label>
                <input type="text" name="iddept" id="iddept">
            </li>
        </ul>
    </form>

```

```
        </li>
        <li>
            <button type="submit" name="insert">insert</button>
        </li>
    </ul>
</form>
</body>

</html>
```

Pada bagian diatas terdapat fungsi require(“function.php”) yang digunakan untuk menyeratakan file php lain yaitu function.php ke dalam php. Terdapat fungsi isset pada PHP (if (isset(\$\_POST["insert"]))) digunakan untuk mengecek form dari HTML, apakah form sudah di klik submit atau belum. Penggunaan isset berdampingan dengan fungsi if. Fungsi isset pada form digunakan untuk mengecek variabel \$\_POST yang dikirim dari sebuah form. variabel \$\_POST digunakan pengiriman data dari form menggunakan method POST yaitu attribute insert. Fungsi isset kemudian akan menjalankan fungsi if yaitu jika function insert dengan parameter masukan variable POST (\$\_POST) bernilai lebih dari 1 maka akan dijalankan script untuk memberikan notifikasi “data berhasil ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php. namun, jika variable POST (\$\_POST) bernilai kurang dari 1 maka akan dijalankan script untuk memberikan notifikasi “data gagal ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php.

Selanjutnya terdapat kode program html, pada bagian head dari html terdapat tag title “<title>Insert</title>” yang digunakan untuk menampilkan judul halaman yaitu Insert. Terdapat anchor (<a>) yang menampilkan kalimat “home” dengan href atau hyperlink referensi nya ke index.php. kemudian terdapat tag form (<form>) dengan method adalah POST yang digunakan mengirim nilai form HTML ke PHP. Didalam tag form (<form>) terdapat tag unordered

list (<ul>) yang didalamnya lagi terdapat beberapa tag list (<li>) yaitu :

- 1) Tag label <label> memiliki for (id dari form control) yaitu namapegawai dengan menampilkan kalimat “namapegawai”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “namapegawai” dan id nya adalah “namapegawai”.
- 2) Tag label <label> memiliki for (id dari form control) yaitu alamat dengan menampilkan kalimat “alamat”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “alamat” dan id nya adalah “alamat”.
- 3) Tag label <label> memiliki for (id dari form control) yaitu iddept dengan menampilkan kalimat “iddept”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “iddept” dan id nya adalah “iddept”.
- 4) Tag button <button> dengan tipe button adalah submit dengan nama button adalah insert dan menampilkan kalimat “insert”. Button ini digunakan untuk mettrigger if (isset(\$\_POST["insert"]))

b) Insert table departemen

```
<?php
require("function.php");
if (isset($_POST["insert1"])) {
    if (insert1($_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>";
    }
}
?>
```

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Insert</title>
</head>

<body>
  <a href="index.php">home</a>
  <form action="" method="POST">
    <ul>

      <li>
        <label for="namadep">namadep</label>
        <input type="text" name="namadep" id="namadep">
      </li>
      <li>
        <label for="lokasi">lokasi</label>
        <input type="text" name="lokasi" id="lokasi">
      </li>
      <li>
        <button
                                type="submit"
name="insert1">insert</button>
      </li>
    </ul>
  </form>
</body>

</html>

```

Pada bagian diatas terdapat fungsi require(“function.php”) yang digunakan untuk menyeratakan file php lain yaitu function.php ke dalam php. Terdapat fungsi isset pada PHP (if (isset(\$\_POST["insert1"]))) digunakan untuk mengecek form dari HTML, apakah form sudah di klik submit atau belum. Penggunaan isset berdampingan dengan fungsi if. Fungsi isset pada form digunakan untuk mengecek variabel \$\_POST yang dikirim dari sebuah form. variabel \$\_POST digunakan pengiriman data dari form menggunakan method POST yaitu attribute insert1. Fungsi isset kemudian akan menjalankan fungsi if yaitu jika function insert1

dengan parameter masukan variable POST (`$_POST`) bernilai lebih dari 1 maka akan dijalankan script untuk memberikan notifikasi “data berhasil ditambah” dan dijalankan `document.location.href` untuk kembali ke halaman `index.php`. namun, jika variable POST (`$_POST`) bernilai kurang dari 1 maka akan dijalankan script untuk memberikan notifikasi “data gagal ditambah” dan dijalankan `document.location.href` untuk kembali ke halaman `index.php`.

Selanjutnya terdapat kode program html, pada bagian head dari html terdapat tag title “`<title>Insert</title>`” yang digunakan untuk menampilkan judul halaman yaitu Insert. Terdapat anchor (`<a>`) yang menampilkan kalimat “home” dengan href atau hyperlink referensi nya ke `index.php`. kemudian terdapat tag form (`<form>`) dengan method adalah POST yang digunakan mengirim nilai form HTML ke PHP. Didalam tag form (`<form>`) terdapat tag unordered list (`<ul>`) yang didalamnya lagi terdapat beberapa tag list (`<li>`) yaitu :

- 1) Tag label `<label>` memiliki for (id dari form control) yaitu `namadep` dengan menampilkan kalimat “namadep”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “namadep” dan id nya adalah “namadep”.
- 2) Tag label `<label>` memiliki for (id dari form control) yaitu lokasi dengan menampilkan kalimat “lokasi”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “lokasi” dan id nya adalah “lokasi”.
- 3) Tag button `<button>` dengan tipe button adalah submit dengan nama button adalah `insert1` dan menampilkan kalimat “insert1”. Button ini digunakan untuk mettrigger if (`isset($_POST["insert1"])`)

Pada kode program php dan html untuk menampilkan data ditambah beberapa coding agar dapat menjalankan perintah menambahkan data sebagai berikut :

```
<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
$data2 = show("SELECT * FROM departemen");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Modul3</title>
</head>
<body>
<form action="insert.php">
    <button>INSERT</button>
</form>
<table border="1" cellpadding="10">
    <tr>
        <td>id pegawai</td>
        <td>nama pegawai</td>
        <td>alamat</td>
        <td>iddept</td>
        <td>aksi</td>
    </tr>
    <?php
    foreach ($data as $d) {
    ?>
        <tr>
            <td><?php echo $d["idpegawai"]; ?></td>
            <td><?php echo $d["namapegawai"]; ?></td>
            <td><?php echo $d["alamat"]; ?></td>
            <td><?php echo $d["iddept"]; ?></td>
            <td>
            </td>
        </tr>
    <?php
    }
    ?>
</table>
<br></br>
<form action="insertdept.php">
```

```

<button>INSERT</button>
</form>
<table border="1" cellpadding="10">
  <tr>
    <td>id departemen</td>
    <td>nama departemen</td>
    <td>lokasi</td>
    <td>aksi</td>
  </tr>
  <?php
  foreach ($data2 as $d) {
  ?>
    <tr>
      <td><?php echo $d["iddept"]; ?></td>
      <td><?php echo $d["namadep"]; ?></td>
      <td><?php echo $d["lokasi"]; ?></td>
      <td>
      </td>
    </tr>
    <?php
    }
  ?>
</table>
</body>
</html>

```

Pada bagian diatas merupakan coding program untuk menampilkan data yang ditambahkan beberapa baris coding program agar dapat mengakses coding program menambahkan data pada table. Coding yang ditambahkan kedalam menampilkan data adalah sebagai berikut :

- 1) Pada bagian coding ini digunakan untuk menampilkan insert.php dengan menggunakan tag form (<form>). attribute action diisi dengan nilai berupa alamat halaman PHP yaitu insert.php dimana akan memproses isi form tersebut. Isi dari tag form (<form>) adalah button (<button>) yang didalamnya menampilkan kalimat "INSERT"

```

<form action="insert.php">
  <button>INSERT</button>
</form>

```

- 2) Pada bagian coding ini digunakan untuk menampilkan insertdept.php dengan menggunakan tag form (<form>). attribute action diisi dengan nilai berupa alamat halaman PHP yaitu insertdept.php dimana akan memproses isi form tersebut. Isi dari tag form (<form>) adalah button (<button>) yang didalamnya menampilkan kalimat “INSERT”

```
<form action="insertdept.php">
    <button>INSERT</button>
</form>
```

- d. Mengupdate data pada database.

Pada kode program php untuk pengkoneksian database ditambahkan fungsi untuk mengupdate data sebagai berikut :

```
<?php
$servername = "localhost";
$dbname = "pegawai";
$username = "root";
$password = "";

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

function show($query){
    global $conn;
    $result = mysqli_query($conn, $query);
    $rows = [];
    while( $row = mysqli_fetch_assoc($result) ){
        $rows[] = $row;
    }
    return $rows;
}

function insert($data){
    global $conn;
    $namapegawai = $_POST["namapegawai"];
    $alamat = $_POST["alamat"];
    $iddept = $_POST["iddept"];

    mysqli_query($conn, "INSERT INTO pegawai
VALUES('$namapegawai','$alamat','$iddept')");
    return mysqli_affected_rows($conn);
}
```



```

}
function insert1($data2){
    global $conn;
    $namadep = $_POST["namadep"];
    $lokasi = $_POST["lokasi"];

    mysqli_query($conn, "INSERT INTO departemen
VALUES('$namadep','$lokasi')");
    return mysqli_affected_rows($conn);
}

function update($data){
    global $conn;
    $idpegawaibaru = $_POST["idpegawaibaru"];
    $idpegawai = $_POST["idpegawai"];
    $namapegawai = $_POST["namapegawai"];
    $alamat = $_POST["alamat"];
    $iddept = $_POST["iddept"];
    mysqli_query($conn, "UPDATE pegawai SET namapegawai =
'$namapegawai', alamat = '$alamat', iddept='$iddept' WHERE
idpegawai = '$idpegawai'");
    mysqli_query($conn, "UPDATE pegawai set idpegawai =
'$idpegawaibaru' where idpegawai = '$idpegawai'");
    return mysqli_affected_rows($conn);
}

function update1($data2){
    global $conn;
    $iddept = $_POST["iddept"];
    $namadep = $_POST["namadep"];
    $lokasi = $_POST["lokasi"];
    mysqli_query($conn, "UPDATE departemen SET namadep =
'$namadep', lokasi = '$lokasi' WHERE iddept = '$iddept'");
    return mysqli_affected_rows($conn);
}
?>

```

Pada bagian diatas ditambahkan function update dan update1 yang berfungsi untuk mengupdate data dengan parameter variabel data (\$data) untuk update dan variable data2 (\$data2) untuk update1. Berikut merupakan penjelasan dari kedua function tersebut.

- 1) Pada function update didalam terdapat variable global conn (global \$conn). Kemudian terdapat variable idpegawaibaru yang mengirimkan data atau nilai langsung ke action untuk ditampung

dengan array adalah attribute idpegawaibaru. Selanjutnya terdapat variable idpegawai yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute idpegawai. Berikutnya terdapat variable alamat yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute alamat. Kemudian terdapat variable iddept yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute iddept. Selanjutnya terdapat fungsi mysqli\_query yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah " mysqli\_query(\$conn, "UPDATE pegawai SET namapegawai = '\$namapegawai', alamat = '\$alamat', iddept='\$iddept' WHERE idpegawai = '\$idpegawai'");" yang berfungsi untuk menjalankan perintah query untuk mengupdate data ke table pegawai. Berikutnya terdapat lagi mysqli query yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah " mysqli\_query(\$conn, " mysqli\_query(\$conn, "UPDATE pegawai set idpegawai = '\$idpegawaibaru' where idpegawai = '\$idpegawai'");" yang berfungsi untuk menjalankan perintah query untuk mengupdate data idpegwai ke table pegawai. Terakhir terdapat fungsi mysqli\_affected\_rows yang digunakan mengembalikan jumlah baris yang terpengaruh di SELECT, INSERT, UPDATE, REPLACE, atau DELETE query sebelumnya dengan parameter masukan adalah varibel conn (\$conn).

- 2) Pada function update didalam terdapat variable global conn (global \$conn). Kemudian terdapat variable iddept yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute iddept. Selanjutnya terdapat variable namadep yang

mengirimkan data atau nilai langsung ke action untuk ditampilkan dengan array adalah attribute namadep. Berikutnya terdapat variable lokasi yang mengirimkan data atau nilai langsung ke action untuk ditampilkan dengan array adalah attribute lokasi. Selanjutnya terdapat fungsi mysqli\_query yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah "mysqli\_query(\$conn, "UPDATE departemen SET namadep = '\$namadep', lokasi = '\$lokasi' WHERE iddept = '\$iddept'");" yang berfungsi untuk menjalankan perintah query untuk mengupdate data ke table departemen. Terakhir terdapat fungsi mysqli\_affected\_rows yang digunakan mengembalikan jumlah baris yang terpengaruh di SELECT, INSERT, UPDATE, REPLACE, atau DELETE query sebelumnya dengan parameter masukan adalah variabel conn (\$conn).

Setelah membuat function untuk memproses update data pada database, selanjutnya adalah membuat kode program php sebagai tempat masukan bagi table yaitu table pegawai dan table departemen. Coding program input untuk update data pada table adalah sebagai berikut :

a) Update data table pegawai

```
<?php
require("function.php");
$idpegawai = $_GET["update"];
$data = show("SELECT * FROM pegawai WHERE idpegawai = $idpegawai");
if (isset($_POST["ubah"])) {
    if (update($_GET > 0)) {
        echo "<script>
            alert('data berhasil diupdate');
            document.location.href = 'index.php';
        </script>";
    } else {
        echo "<script>
            alert('data gagal diupdate');
            document.location.href = 'index.php';
```

```

        </script>";
    }
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Update</title>
</head>

<body>
    <a href="index.php">home</a>
    <form action="" method="POST">
        <?php
        foreach ($data as $d) {
            ?>
            <ul>
                <li>
                    <label for="idpegawaibaru">idpegawaibaru</label>
                    <input type="text" name="idpegawaibaru"
id="idpegawaibaru" value=<?php echo $d["idpegawai"]; ?>>
                    </li>
                <li>
                    <label for="idpegawai">idpegawailama</label>
                    <input type="text" name="idpegawai"
id="idpegawai" value=<?php echo $d["idpegawai"]; ?>>
                    </li>
                <li>
                    <label for="namapegawai">namapegawai</label>
                    <input type="text" name="namapegawai"
id="namapegawai" value=<?php echo $d["namapegawai"]; ?>>
                    </li>
                <li>
                    <label for="alamat">alamat</label>
                    <input type="text" name="alamat" id="alamat"
value=<?php echo $d["alamat"]; ?>>
                    </li>
                <li>
                    <label for="iddept">iddept</label>

```

```

        <input type="text" name="iddept" id="iddept"
value=<?php echo $d["iddept"]; ?>>
        </li>
        <li>
            <button                                type="submit"
name="ubah">update</button>
        </li>
    </ul>
    <?php
    }
    ?>
</form>
</body>

</html>

```

Pada bagian diatas terdapat fungsi require(“function.php”) yang digunakan untuk menyeratakan file php lain yaitu function.php ke dalam php. Terdapat variable idpegawai (\$idpegawai) bernilai GET dari attribute update yang berada di idex.php. selain itu terdapat variable data (\$data) dengan nilai berasal dari function show dengan parameter masukan adalah "SELECT \* FROM pegawai WHERE idpegawai = \$idpegawai". Terdapat fungsi isset pada PHP (if (isset(\$\_POST["ubah"]))) digunakan untuk mengecek form dari HTML, apakah form sudah di klik submit atau belum. Penggunaan isset berdampingan dengan fungsi if. Fungsi isset pada form digunakan untuk mengecek variabel \$\_POST yang dikirim dari sebuah form. variabel \$\_POST digunakan pengiriman data dari form menggunakan method POST yaitu attribute ubah. Fungsi isset kemudian akan menjalankan fungsi if yaitu jika function update dengan parameter masukan variable GET (\$\_GET) bernilai lebih dari 1 maka akan dijalankan script untuk memberikan notifikasi “data berhasil ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php. namun, jika variable GET (\$\_GET) bernilai kurang dari 1 maka akan dijalankan script untuk memberikan notifikasi “data gagal ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php.

Selanjutnya terdapat kode program html, pada bagian head dari html terdapat tag title “<title>update</title>” yang digunakan untuk menampilkan judul halaman yaitu update. Terdapat anchor (<a>) yang menampilkan kalimat “home” dengan href atau hyperlink referensi nya ke index.php. kemudian terdapat tag form (<form>) dengan method adalah POST yang digunakan mengirim nilai form HTML ke PHP. Didalam tag form (<form>) dijalankan perulangan php yaitu foreach yang berfungsi melakukan perulangan pada array. Didalam tag form (<form>) terdapat tag unordered list (<ul>) yang didalamnya lagi terdapat beberapa tag list (<li>) yaitu :

- 1) Tag label <label> memiliki for (id dari form control) yaitu idpegawaibaru dengan menampilkan kalimat “idpegawaibaru”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “idpegawaibaru” dan id nya adalah “idpegawaibaru”.
- 2) Tag label <label> memiliki for (id dari form control) yaitu idpegawai dengan menampilkan kalimat “idpegawai”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “idpegawai” dan id nya adalah “idpegawai”.
- 3) Tag label <label> memiliki for (id dari form control) yaitu namapegawai dengan menampilkan kalimat “namapegawai”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “namapegawai” dan id nya adalah “namapegawai”.
- 4) Tag label <label> memiliki for (id dari form control) yaitu alamat dengan menampilkan kalimat “alamat”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “alamat” dan id nya adalah “alamat”.
- 5) Tag label <label> memiliki for (id dari form control) yaitu iddept dengan menampilkan kalimat “iddept”. Tag input memiliki type inputan yaitu text dengan nama inputan adalah “iddept” dan id nya adalah “iddept”.

- 6) Tag button <button> dengan tipe button adalah submit dengan nama button adalah ubah dan menampilkan kalimat “update”. Button ini digunakan untuk mettrigger if (isset(\$\_POST["ubah"])).

Pada kode program php dan html untuk menampilkan data ditambah beberapa coding agar dapat menjalankan perintah mengupdate data sebagai berikut :

```
<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
$data2 = show("SELECT * FROM departemen");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Modul3</title>
</head>
<body>
<form action="insert.php">
    <button>INSERT</button>
</form>
<table border="1" cellpadding="10">
    <tr>
        <td>id pegawai</td>
        <td>nama pegawai</td>
        <td>alamat</td>
        <td>iddept</td>
        <td>aksi</td>
    </tr>
    <?php
    foreach ($data as $d) {
    ?>
        <tr>
            <td><?php echo $d["idpegawai"]; ?></td>
            <td><?php echo $d["namapegawai"]; ?></td>
            <td><?php echo $d["alamat"]; ?></td>
            <td><?php echo $d["iddept"]; ?></td>
            <td>
```

```

        <form action="update.php" method="GET">
            <button type="submit" name="update"
value=<?php echo $d["idpegawai"]; ?>>UPDATE</button>
        </form>
    </td>
</tr>
<?php
}
?>
</table>
<br></br>
<form action="insertdept.php">
    <button>INSERT</button>
</form>
<table border="1" cellpadding="10">
    <tr>
        <td>id departemen</td>
        <td>nama departemen</td>
        <td>lokasi</td>
        <td>aksi</td>
    </tr>
    <?php
    foreach ($data2 as $d) {
    ?>
        <tr>
            <td><?php echo $d["iddept"]; ?></td>
            <td><?php echo $d["namadep"]; ?></td>
            <td><?php echo $d["lokasi"]; ?></td>
            <td>
                <form action="updatedept.php" method="GET">
                    <button type="submit" name="update1"
value=<?php echo $d["iddept"]; ?>>UPDATE</button>
                </form>
            </td>
        </tr>
    <?php
    }
    ?>
</table>
</body>
</html>

```

Pada bagian diatas merupakan coding program untuk menampilkan data yang ditambahkan beberapa baris coding program agar dapat mengakses coding program mengupdate data pada table. Coding



yang ditambahkan kedalam menampilkan data adalah sebagai berikut :

- 1) Pada bagian coding ini digunakan untuk menampilkan update.php dengan menggunakan tag form (<form>). attribute action diisi dengan nilai berupa alamat halaman PHP yaitu update.php dimana akan memproses isi form tersebut. Method yang digunakan pada tag form ini adalah GET. Isi dari tag form (<form>) adalah button (<button>) dengan tipe submit yang memiliki nama update dan memiliki nilai hasil dari php echo \$d["idpegawai"]. Didalam button menampilkan kalimat "UPDATE"

```
<form action="update.php" method="GET">
    <button type="submit" name="update"
value=<?php echo $d["idpegawai"]; ?>>UPDATE</button>
</form>
```

- 2) Pada bagian coding ini digunakan untuk menampilkan updatedept.php dengan menggunakan tag form (<form>). attribute action diisi dengan nilai berupa alamat halaman PHP yaitu updatedept.php dimana akan memproses isi form tersebut. Method yang digunakan pada tag form ini adalah GET. Isi dari tag form (<form>) adalah button (<button>) dengan tipe submit yang memiliki nama update1 dan memiliki nilai hasil dari php echo \$d["iddept"]. Didalam button menampilkan kalimat "UPDATE"

```
<form action="updatedept.php" method="GET">
    <button type="submit" name="update1"
value=<?php echo $d["iddept"]; ?>>UPDATE</button>
</form>
```

- e. Menghapus data pada database.

Pada kode program php untuk pengkoneksian database ditambahkan fungsi untuk menghapus data sebagai berikut :

```

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

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

function show($query){
    global $conn;
    $result = mysqli_query($conn, $query);
    $rows = [];
    while( $row = mysqli_fetch_assoc($result) ){
        $rows[] = $row;
    }
    return $rows;
}

function insert($data){
    global $conn;
    $namapegawai = $_POST["namapegawai"];
    $alamat = $_POST["alamat"];
    $iddept = $_POST["iddept"];

    mysqli_query($conn, "INSERT INTO pegawai
VALUES('$namapegawai','$alamat','$iddept')");
    return mysqli_affected_rows($conn);
}

function insert1($data2){
    global $conn;
    $namadep = $_POST["namadep"];
    $lokasi = $_POST["lokasi"];

    mysqli_query($conn, "INSERT INTO departemen
VALUES('$namadep','$lokasi')");
    return mysqli_affected_rows($conn);
}

function delete($data){
    global $conn;
    $idpegawai = $_GET["delete"];
    mysqli_query($conn, "DELETE FROM pegawai WHERE
idpegawai = $idpegawai");
    return mysqli_affected_rows($conn);
}

```

```

function delete1($data2){
    global $conn;
    $iddept = $_GET["delete1"];
    mysqli_query($conn, "DELETE FROM departemen WHERE
iddept = $iddept");
    return mysqli_affected_rows($conn);
}

function update($data){
    global $conn;
    $idpegawaibaru = $_POST["idpegawaibaru"];
    $idpegawai = $_POST["idpegawai"];
    $namapegawai = $_POST["namapegawai"];
    $alamat = $_POST["alamat"];
    $iddept = $_POST["iddept"];
    mysqli_query($conn, "UPDATE pegawai SET namapegawai =
'$namapegawai', alamat = '$alamat', iddept='$iddept' WHERE
idpegawai = '$idpegawai'");
    mysqli_query($conn, "UPDATE pegawai set idpegawai =
'$idpegawaibaru' where idpegawai = '$idpegawai'");
    return mysqli_affected_rows($conn);
}

function update1($data2){
    global $conn;
    $iddept = $_POST["iddept"];
    $namadep = $_POST["namadep"];
    $lokasi = $_POST["lokasi"];
    mysqli_query($conn, "UPDATE departemen SET namadep =
'$namadep', lokasi = '$lokasi' WHERE iddept = '$iddept'");
    return mysqli_affected_rows($conn);
}
?>

```

Pada bagian diatas ditambahkan function delete dan delete1 yang berfungsi untuk menghapus data dengan parameter variabel data (\$data) untuk delete dan variable data2 (\$data2) untuk delete1. Berikut merupakan penjelasan dari kedua function tersebut.

- 1) Pada function delete didalam terdapat variable global conn (global \$conn). Kemudian terdapat variable idpegawai yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute delete. Selanjutnya terdapat fungsi mysqli\_query

yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah "DELETE FROM pegawai WHERE idpegawai = \$idpegawai" yang berfungsi untuk menjalankan perintah query untuk menghapus data ke table pegawai. Terakhir terdapat fungsi mysqli\_affected\_rows yang digunakan mengembalikan jumlah baris yang terpengaruh di SELECT, INSERT, UPDATE, REPLACE, atau DELETE query sebelumnya dengan parameter masukan adalah variabel conn (\$conn).

- 2) Pada function delete1 didalam terdapat variable global conn (global \$conn). Kemudian terdapat variable iddept yang mengirimkan data atau nilai langsung ke action untuk ditampung dengan array adalah attribute delete1. Selanjutnya terdapat fungsi mysqli\_query yang digunakan untuk menjalankan query yang berada di sql dengan parameter pertama adalah variable conn (\$conn) yang berfungsi untuk memanggil database dan parameter kedua adalah "DELETE FROM departemen WHERE iddept = \$idept" yang berfungsi untuk menjalankan perintah query untuk menghapus data ke table pegawai. Terakhir terdapat fungsi mysqli\_affected\_rows yang digunakan mengembalikan jumlah baris yang terpengaruh di SELECT, INSERT, UPDATE, REPLACE, atau DELETE query sebelumnya dengan parameter masukan adalah variabel conn (\$conn).

Pada kode program php dan html untuk menampilkan data ditambah beberapa coding agar dapat menjalankan proses dan perintah menghapus data sebagai berikut :

```
<?php
require("function.php");
$data = show("SELECT * FROM pegawai");
$data2 = show("SELECT * FROM departemen");
if (isset($_GET["delete"])) {
    if (delete($_GET > 0)) {
```

```

        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>";
    }
}
}
if (isset($_GET["delete1"])) {
    if (delete1($_GET > 0)) {
        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>";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Modul3</title>
</head>
<body>
    <form action="insert.php">
        <button>INSERT</button>
    </form>
    <table border="1" cellpadding="10">
        <tr>
            <td>id pegawai</td>
            <td>nama pegawai</td>
            <td>alamat</td>
            <td>iddept</td>
            <td>aksi</td>
        </tr>
    </table>
</body>
</html>
<?php

```

```

        foreach ($data as $d) {
            ?>
            <tr>
                <td><?php echo $d["idpegawai"]; ?></td>
                <td><?php echo $d["namapegawai"]; ?></td>
                <td><?php echo $d["alamat"]; ?></td>
                <td><?php echo $d["iddept"]; ?></td>
                <td>
                    <form action="" method="GET">
                        <button type="submit" name="delete" value=<?php
echo $d["idpegawai"]; ?>>DELETE</button>
                    </form>
                    <form action="update.php" method="GET">
                        <button type="submit" name="update" value=<?php
echo $d["idpegawai"]; ?>>UPDATE</button>
                    </form>
                </td>
            </tr>
            <?php
        }
    ?>
</table>
<br><br>
<form action="insertdept.php">
    <button>INSERT</button>
</form>
<table border="1" cellpadding="10">
    <tr>
        <td>iddept</td>
        <td>namadept</td>
        <td>lokasi</td>
        <td>aksi</td>
    </tr>
    <?php
    foreach ($data2 as $d) {
        ?>
        <tr>
            <td><?php echo $d["iddept"]; ?></td>
            <td><?php echo $d["namadep"]; ?></td>
            <td><?php echo $d["lokasi"]; ?></td>
            <td>
                <form action="" method="GET">
                    <button type="submit" name="delete1" value=<?php
echo $d["iddept"]; ?>>DELETE</button>
                </form>
                <form action="updatedept.php" method="GET">

```

```

        <button type="submit" name="update1" value=<?php
echo $d["iddept"]; ?>>UPDATE</button>
    </form>
</td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

Pada bagian diatas merupakan coding program untuk menampilkan data yang ditambahkan beberapa baris coding program agar dapat memproses coding program menghapus data pada table. Coding yang ditambahkan kedalam menampilkan data adalah sebagai berikut :

- 1) Terdapat fungsi isset pada PHP (if (isset(\$\_GET["delete"]))) digunakan untuk mengecek form dari HTML, apakah form sudah di klik submit atau belum. Penggunaan isset berdampingan dengan fungsi if. Fungsi isset pada form digunakan untuk mengecek variabel \$\_GET yang dikirim dari sebuah form. Menggunakan variabel \$\_GET untuk pengiriman form pada HTML menggunakan method GET yaitu attribute delete. Fungsi isset kemudian akan menjalankan fungsi if yaitu jika function delete dengan parameter masukan variable GET (\$\_GET) bernilai lebih dari 1 maka akan dijalankan script untuk memberikan notifikasi “data berhasil ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php. namun, jika variable GET (\$\_GET) bernilai kurang dari 1 maka akan dijalankan script untuk memberikan notifikasi “data gagal ditambah” dan dijalankan document.location.href untuk kembali ke halaman index.php.

```

if (isset($_GET["delete"])) {
    if (delete($_GET > 0)) {
        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>";
    }
}

```

- 2) Terdapat fungsi `isset` pada PHP (`if (isset($_GET["delete1"]))`) digunakan untuk mengecek form dari HTML, apakah form sudah di klik submit atau belum. Penggunaan `isset` berdampingan dengan fungsi `if`. Fungsi `isset` pada form digunakan untuk mengecek variabel `$_GET` yang dikirim dari sebuah form. Menggunakan variabel `$_GET` untuk pengiriman form pada HTML menggunakan method GET yaitu attribute `delete1`. Fungsi `isset` kemudian akan menjalankan fungsi `if` yaitu jika function `delete` dengan parameter masukan variable GET (`$_GET`) bernilai lebih dari 1 maka akan dijalankan script untuk memberikan notifikasi “data berhasil ditambah” dan dijalankan `document.location.href` untuk kembali ke halaman `index.php`. namun, jika variable GET (`$_GET`) bernilai kurang dari 1 maka akan dijalankan script untuk memberikan notifikasi “data gagal ditambah” dan dijalankan `document.location.href` untuk kembali ke halaman `index.php`.

```

if (isset($_GET["delete1"])) {
    if (delete1($_GET > 0)) {
        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>";
    }
}
}

```



- 3) Pada bagian coding ini digunakan untuk menjalankan proses penghapusan dengan menggunakan tag form (<form>). attribute action diisi kosong dikarenakan berada di file yang sama. Method yang digunakan pada tag form ini adalah GET. Isi dari tag form (<form>) adalah button (<button>) dengan tipe submit yang memiliki nama delete dan memiliki nilai hasil dari php echo \$d["idpegawai"]. Didalam button menampilkan kalimat "DELETE".

```
<form action="" method="GET">
    <button type="submit" name="delete"
value=<?php echo $d["idpegawai"]; ?>>DELETE</button>
</form>
```

- 4) Pada bagian coding ini digunakan untuk menjalankan proses penghapusan dengan menggunakan tag form (<form>). attribute action diisi kosong dikarenakan berada di file yang sama. Method yang digunakan pada tag form ini adalah GET. Isi dari tag form (<form>) adalah button (<button>) dengan tipe submit yang memiliki nama delete1 dan memiliki nilai hasil dari php echo \$d["iddept"]. Didalam button menampilkan kalimat "DELETE".

```
<form action="" method="GET">
    <button type="submit" name="delete1"
value=<?php echo $d["iddept"]; ?>>DELETE</button>
</form>
```

Berikut merupakan output dari program untuk menyimpan, menghapus dan mengubah data data pegawai adalah sebagai berikut :

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.1. tampilan program

[home](#)

- namapegawai
- alamat
- iddept
- 

Gambar 2.2 tampilan input insert data table pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE
15	dani	j.marini	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.3. setelah dilakukan input insert data pegawai

[home](#)

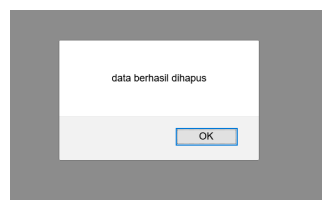
- idpegawaibaru 4
- idpegawailama 15
- namapegawai dani
- alamat jl.bandung
- iddept 2
- update

Gambar 2.4. tampilan input update data pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE
4	dani	jl.bandung	2	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.5 setelah dilakukan input update data table pegawai



Gambar 2.6 tampilan delete data table pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.7 setelah dilakukan delete data table pegawai

[home](#)

- namadep
- lokasi
- 

Gambar 2.8 tampilan input insert data table departemen

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	andi	Jl. Dara	2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	raden	jl.lima	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

INSERT			
iddept	namadep	lokasi	aksi
1	perhubungan	jl.satria	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	administrasi	jl.amanah1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	pabrik	jl.perkutu	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.9 setelah dilakukan insert data table departemen

[home](#)

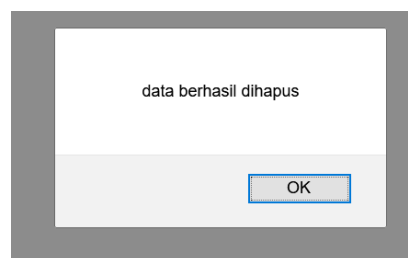
- iddept
- namadep
- lokasi
- 

Gambar 2.10. tampilan input update data table departemen

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE
3	pabrik	jl.merpati	DELETE UPDATE

Gambar 2.11 setelah dilakukan update data table departemen



Gambar 2.12. tampilan delete data table departemen

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	<div>DELETE</div> <div>UPDATE</div>
2	andi	Jl. Dara	2	<div>DELETE</div> <div>UPDATE</div>
3	raden	jl.lima	1	<div>DELETE</div> <div>UPDATE</div>

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	<div>DELETE</div> <div>UPDATE</div>
2	administrasi	jl.amanah1	<div>DELETE</div> <div>UPDATE</div>

Gambar 2.14 setelah dilakukan delete data table departemen

### **BAB III**

#### **KESIMPULAN**

Hal-hal yang dibutuhkan dalam mengkoneksikan antara PHP dan MySQL adalah SQL, PHPMyAdmin, dan PHP Script. PHPMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui World Wide Web sedangkan SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Untuk menjalankan query MySQL, PHP menyediakan fungsi `mysql_query()`. Fungsi ini dijalankan setelah koneksi berhasil dilakukan dengan fungsi `mysql_connect()`. Fungsi `mysql_query()` merupakan fungsi yang akan sering kita gunakan, karena dengan fungsi inilah PHP akan melakukan 'percakapan' dengan MySQL.



## DAFTAR PUSTAKA

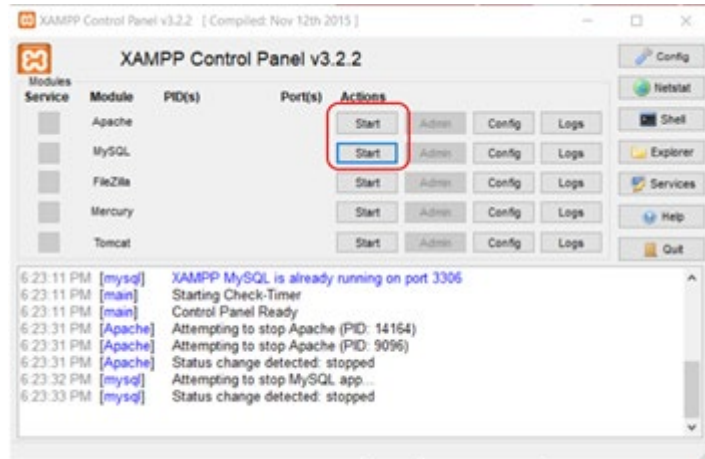
Jayanti, Sabaruddin, and Jayanti Raja. 2020. *Jago Ngoding Pemrograman Web Dengan PHP*.

Tim Dosen Algoritma Pemrograman. *Modul Praktikum Pemrograman Web I*. 2021. Palangka Raya. Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya (UPR).

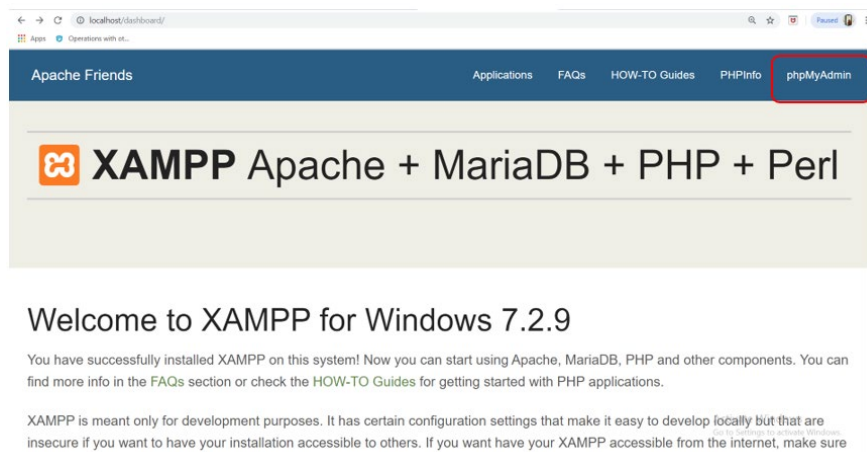
Andre. 2014. *Tutorial PHP MySQL Part 5: Cara Menjalankan query MySQL dari PHP (mysql\_query)*  
[https://www.duniailkom.com/tutorial-php-mysql-cara-menjalankan-query-mysql-dari-php-mysql\\_query/](https://www.duniailkom.com/tutorial-php-mysql-cara-menjalankan-query-mysql-dari-php-mysql_query/) (Diakses 11 April 2021).

Andre. 2014. *Tutorial Form PHP Part 5: Cara Membuat Validasi Form PHP (fungsi isset dan empty)*.  
<https://www.duniailkom.com/tutorial-form-php-cara-membuat-validasi-form-php-fungsi-isset-dan-empty/#:~:text=Untuk%20memeriksa%20apakah%20sebuah%20objek,jika%20variabel%20tersebut%20belum%20dibuat>  
(Diakses 11 April 2021)

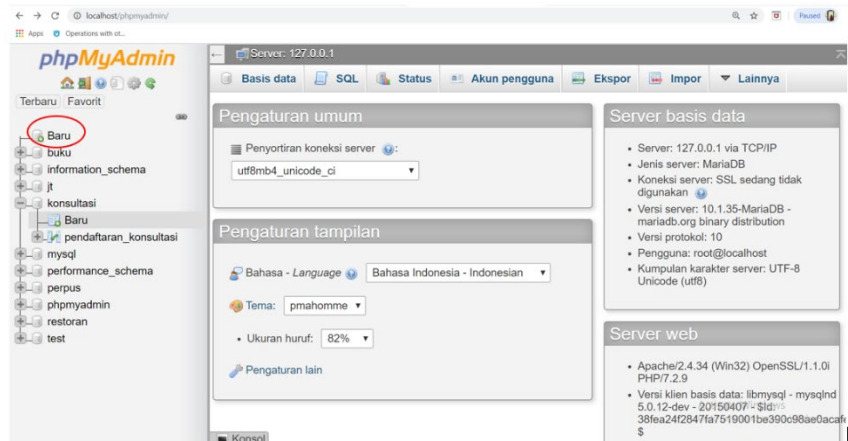
## LAMPIRAN



Gambar 1.1. Mengaktifkan Apache dan MySQL di XAMPP



Gambar 1.2 Membuka PHPMyAdmin



Gambar 1.3. Tampilan PHPMyAdmin

## Basis data

**Buat basis data**

rumahsakit latin1\_swedish\_ci **Buat**

Gambar 1.4. Membuat Database rumah sakit

**Buat tabel**

Nama: jadwal\_dokter Jumlah kolom: 4 **Kirim**

Gambar 1.5 Membuat Tabel

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran
id	INT		Tidak ada	
nama	VARCHAR	225	Tidak ada	
spesialis	VARCHAR	225	Tidak ada	
hari	VARCHAR	225	Tidak ada	

Gambar 1.6. Menentukan Kolom

Kolom	Jenis	Fungsi	Tak Ternilai	Nilai
id	int(11)			1
nama	varchar(225)			dr. Nyoman
spesialis	varchar(225)			Reumatologi
hari	varchar(225)			Senin, Selasa, Jumat

Kirim

Gambar 1.7. Mengisi Tabel

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor

Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0000 detik.)

SELECT \* FROM `jadwal\_dokter`

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode]

Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini

+ Opsi

id	nama	spesialis	hari
1	dr. Nyoman	Reumatologi	Senin, Selasa, Jumat

Ubah Salin Hapus

Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

Gambar 1.8. Data yang Berhasil Diinput

Jadwal Dokter x Inferensi Rantai Markov

localhost/RS/

Apps Operations with ot...

## Jadwal Dokter RS Sehat Selalu

Masukkan Jadwal Baru

No	Nama	Spesialis	Hari	Ubah?
1	dr. Nyoman	Reumatologi	Senin, Selasa, Jumat	Hapus Edit

Gambar 1.9 tampilan awal table jadwal dokter

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.1. tampilan program

[home](#)

- namapegawai dani
- alamat j.marini
- iddept 1
- insert

Gambar 2.2 tampilan input insert data table pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE
15	dani	j.marini	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.3. setelah dilakukan input insert data pegawai

[home](#)

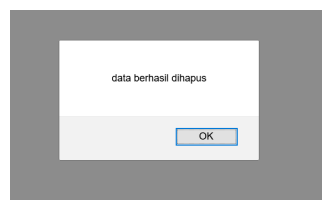
- idpegawaibaru 4
- idpegawailama 15
- namapegawai dani
- alamat jl.bandung
- iddept 2
- update

Gambar 2.4. tampilan input update data pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE
4	dani	jl.bandung	2	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.5 setelah dilakukan input update data table pegawai



Gambar 2.6 tampilan delete data table pegawai

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE

Gambar 2.7 setelah dilakukan delete data table pegawai

[home](#)

- namadep
- lokasi
- 

Gambar 2.8 tampilan input insert data table departemen

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	andi	Jl. Dara	2	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	raden	jl.lima	1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

INSERT			
iddept	namadep	lokasi	aksi
1	perhubungan	jl.satria	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
2	administrasi	jl.amanah1	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>
3	pabrik	jl.perkutu	<input type="button" value="DELETE"/> <input type="button" value="UPDATE"/>

Gambar 2.9 setelah dilakukan insert data table departemen

[home](#)

- iddept
- namadep
- lokasi
- 

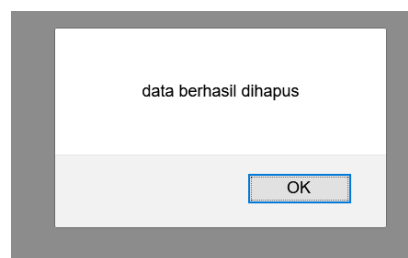
Gambar 2.10. tampilan input update data table departemen



INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	DELETE UPDATE
2	andi	Jl. Dara	2	DELETE UPDATE
3	raden	jl.lima	1	DELETE UPDATE

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	DELETE UPDATE
2	administrasi	jl.amanah1	DELETE UPDATE
3	pabrik	jl.merpati	DELETE UPDATE

Gambar 2.11 setelah dilakukan update data table departemen



Gambar 2.12. tampilan delete data table departemen

INSERT				
id pegawai	nama pegawai	alamat	iddept	aksi
1	nathan	jl. delima	1	<div>DELETE</div> <div>UPDATE</div>
2	andi	Jl. Dara	2	<div>DELETE</div> <div>UPDATE</div>
3	raden	jl.lima	1	<div>DELETE</div> <div>UPDATE</div>

INSERT			
iddept	namadept	lokasi	aksi
1	perhubungan	jl.satria	<div>DELETE</div> <div>UPDATE</div>
2	administrasi	jl.amanah1	<div>DELETE</div> <div>UPDATE</div>

Gambar 2.14 setelah dilakukan delete data table departemen