

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



NAMA : ELSAN PENTANUGRAHA
NIM : 193030503047
KELAS : A
MODUL : III (PHP MySQL Database)

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

BAB I

LANDASAN TEORI

1.1 Tujuan

1.1.1. Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.

1.1.2. Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

1.2 Landasan Teori

1.2.1 MySQL

MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server. (Shah, 2020)

Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language). (Shah, 2020)

SQL (Structured Query Language) adalah sebuah bahasa permintaan database yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merelasikan antar database. (Shah, 2020)

Dua bentuk Query yang akan digunakan, yaitu :

1. DDL (Data Definition Language), yang digunakan untuk mendefinisikan data seperti tabel pada sebuah database. Query yang ada pada DDL yaitu, CREATE, DROP, dan ALTER.

2. DML (Data Manipulation Language), yang digunakan untuk memanipulasi data yang ada pada sebuah database seperti mengubah, menambahkan, dan menghapus pada sebuah tabel yang ada pada database. Query yang ada pada DML yaitu, INSERT, UPDATE, dan DELETE.

1.2.2 Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL.(Praktikum, no date). Berikut adalah contoh kode program agar terhubung dengan server mySQL:

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

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Gambar 1.2.2.1 kode program koneksi php dengan database.

1.2.3 Memasukan Data ke database

Berikut adalah query yang digunakan untuk memasukan data kedalam database melalui mysqli.

```
$sql = "INSERT INTO Students (name, lastname, email)
VALUES ('Thom', 'Vial', 'thom.v@some.com')";
```

Gambar 1.2.3.1 kode program masukan data melalui mysqli.

1.2.4 Menampilkan Data dari database

Berikut adalah query yang digunakan untuk menampilkan data kedalam database melalui mysqli.

```
10 echo <table>
11     <thead>
12         <tr>
13             <th>ID PRODUK</th>
14             <th>TGL TRANSAKSI</th>
15             <th>KUANTITAS</th>
16             <th>HARGA</th>
17             <th>TOTAL BYR</th>
18         </tr>
19     </thead>
20     <tbody>';
21
22 while ($row = mysqli_fetch_array($query))
23 {
24     echo ' <tr>
25         <td>'. $row['id_produk']. '</td>
26         <td>'. $row['tgl_transaksi']. '</td>
27         <td>'. $row['kuantitas']. '</td>
28         <td>'. $row['harga']. '</td>
29         <td>'. number_format($row['total_byr'], 0, ',', '.'). '</td>
30     </tr>';
31 }
```

Gambar 1.2.4.1 kode program Menampilkan data melalui mysqli.

1.2.5 Mengubah Data ke database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:(Praktikum, no date).

```
UPDATE nama_tabel
SET kolom1=nilai1, kolom2=nilai2, ...
WHERE kolom_penentu=nilai_penentu ...
```

Gambar 1.2.5.1 kode program mengubah data melalui mysqli.

1.2.6 Menghapus Data ke database

Statement DELETE digunakan untuk menghapus baris data atau records dari tabel. Clause WHERE digunakan untuk menspesifikasikan baris yang akan dihapus. Jika statement DELETE digunakan tanpa menggunakan clausa WHERE, maka semua record yang ada pada tabel akan dihapus.(Praktikum, no date).

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

Gambar 1.2.6.1 contoh tabel pada database.

Jika kita memiliki data seperti tabel “MyGuest” seperti terlihat diatas, kemudian kita ingin menghapus data dengan nama depan Julie, maka kode program untuk menghapus data tersebut adalah sebagai berikut:(Praktikum, no date).

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

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

mysqli_close($conn);
```

Gambar 1.2.6.2 kode program menghapus data melalui mysqli.

1.3 Tugas Praktikum

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

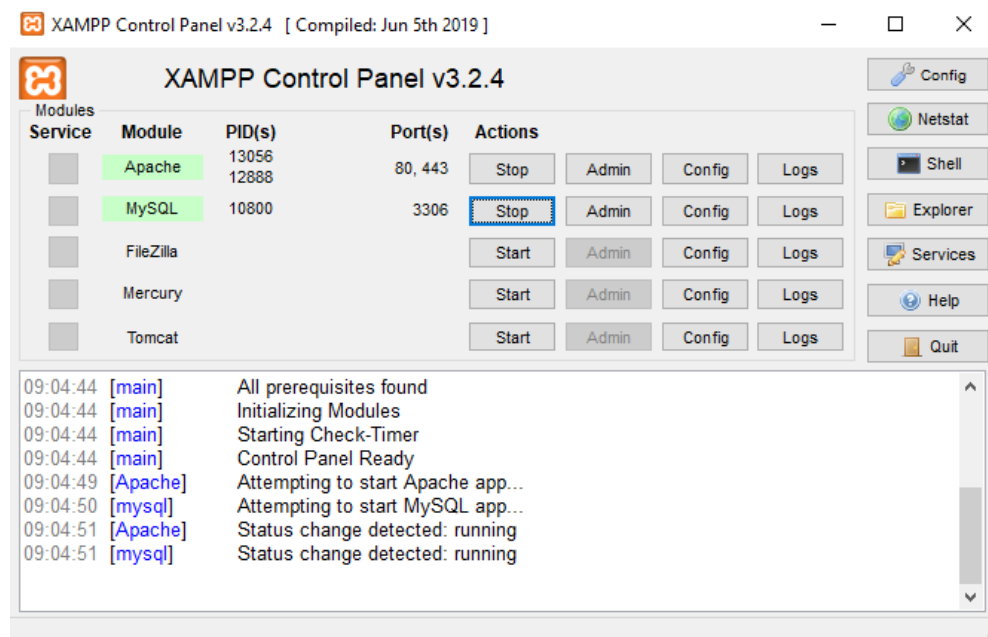
BAB II

PEMBAHASAN

Untuk menyelesaikan tugas yang ada pada modul 3 ini aplikasi yang digunakan adalah visual studio code yang digunakan sebagai code editor untuk membuat program login form dengan bahasa PHP. Kemudian XAMPP yang digunakan sebagai server yang berdiri sendiri (localhost), pada XAMPP ini terdapat MySQL yang digunakan sebagai database dan juga web browser Chrome yang digunakan untuk menjalankan program web login form yang telah dibuat.

2.1 Pembahasan soal pertama

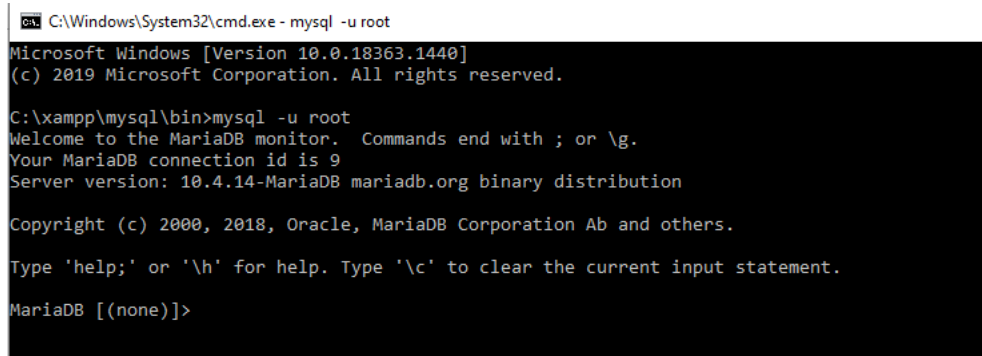
Langkah pertama yang dilakukan adalah mengaktifkan Apache dan MySQL pada aplikasi XAMPP dengan klik start. Apache digunakan sebagai web server dan MySQL digunakan sebagai database server. Berikut adalah hasil tangkapan layar pada aplikasi XAMPP dengan apache dan MySQL yang telah aktif.



Gambar 2.1.1 XAMPP.

Kemudian buka command prompt, lalu pindah directory pada command prompt ke `C:\xampp\mysql\bin`. Setelah itu ketikkan `mysql -u root` agar command prompt terkoneksi ke server MariaDB. Setelah terkoneksi maka ketikkan query `create`

database kepegawaian untuk melakukan pembuatan database yang diberi nama kepegawaian. Kemudian masukan query *use kepegawaian* yang digunakan untuk menggunakan database kepegawaian yang telah dibuat untuk membuat tabel didalamnya. Berikut adalah hasil dari pembuatan database kepegawaian.



```
C:\Windows\System32\cmd.exe - mysql -u root
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

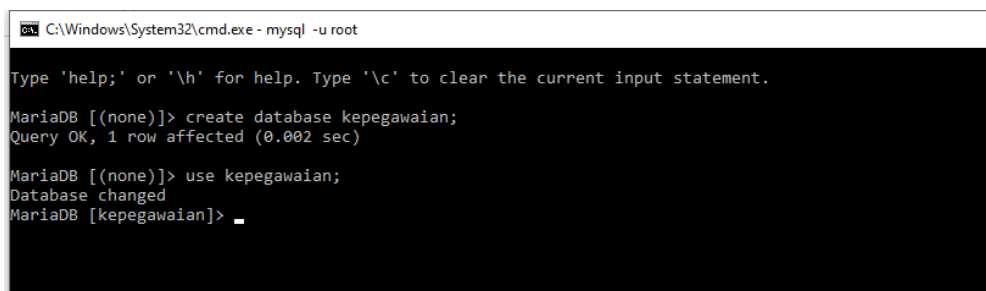
C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Gambar 2.1.2 koneksi ke server MariaDB.



```
C:\Windows\System32\cmd.exe - mysql -u root

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database kepegawaian;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use kepegawaian;
Database changed
MariaDB [kepegawaian]> _
```

Gambar 2.1.3 Pembuatan database.

Langkah selanjutnya adalah pembuatan tabel. Pada modul ini saya membuat dua buah tabel yaitu tabel gaji dan tabel pegawai dimana tabel pegawai ini memiliki relasi dengan tabel gaji dengan data dari kolom golongan. Kemudian setiap kolom yang dibuat saya memiliki tiga tipe data yaitu INT, BIGINT, dan VARCHAR. Pada tabel gaji, kolom golongan saya menggunakan tipe data integer dengan panjang maksimal lima karakter yang dijadikan sebagai primary key. Kemudian kolom gaji dan kolom tunjangan saya menggunakan tipe data BIGINT dengan panjang maksimal dua puluh lima karakter yang digunakan untuk menginputkan data gaji dan tunjangan.

Dan pada tabel pegawai, kolom idpegawai saya menggunakan tipe data integer dengan panjang maksimal sebelas karakter yang digunakan sebagai primary key. Kemudian kolom nama saya menggunakan tipe data varchar dengan panjang

maksimal lima puluh karakter yang digunakan untuk menampung data nama pegawai. Kemudian kolom golongan yang merupakan foreign key dari tabel gaji menggunakan tipe data yang sama dengan kolom golongan yang berada pada tabel gaji yaitu integer dengan panjang maksimal lima karakter. Berikut adalah hasil dari pembuatan tabel gaji dan tabel pegawai.

```
C:\Windows\System32\cmd.exe - mysql -u root

Empty set (0.001 sec)

MariaDB [kepegawaian]> create table gaji(
  -> golongan INT(5) NOT NULL PRIMARY KEY,
  -> gaji BIGINT(25),
  -> tunjangan BIGINT(25)
  -> );
Query OK, 0 rows affected (0.294 sec)

MariaDB [kepegawaian]> show tables;
+-----+
| Tables_in_kepegawaian |
+-----+
| gaji                   |
+-----+
1 row in set (0.001 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.4 pembuatan tabel gaji.

```
C:\Windows\System32\cmd.exe - mysql -u root

MariaDB [kepegawaian]>
MariaDB [kepegawaian]> create table pegawai(
  -> idpegawai INT(11) NOT NULL PRIMARY KEY,
  -> nama VARCHAR(50),
  -> golongan INT(5),
  -> FOREIGN KEY (golongan) REFERENCES gaji(golongan)
  -> );
Query OK, 0 rows affected (0.294 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.5 pembuatan tabel pegawai.

Setelah tabel dibuat, maka selanjutnya adalah memasukan data kedalam tabel gaji dan tabel pegawai. Pada tabel pegawai, kolom golongan harus diinputkan data yang sesuai dengan data pada kolom golongan yang ada pada tabel gaji oleh karena

itu yang pertama dimasukan data adalah tabel gaji. Berikut adalah hasil tangkapan layar command prompt dari query yang digunakan untuk memasukan data kedalam tabel gaji dan tabel pegawai.

```
cmd Select C:\Windows\System32\cmd.exe - mysql -u root
MariaDB [kepegawaian]>
MariaDB [kepegawaian]> insert into gaji values
-> ('101','1500000','500000'),
-> ('102','2000000','750000'),
-> ('103','3000000','1200000'),
-> ('104','3500000','1550000');
Query OK, 4 rows affected (0.117 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [kepegawaian]> select *from gaji;
+-----+-----+-----+
| golongan | gaji    | tunjangan |
+-----+-----+-----+
| 101      | 1500000 | 500000    |
| 102      | 2000000 | 750000    |
| 103      | 3000000 | 1200000   |
| 104      | 3500000 | 1550000   |
+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.6 input data kedalam tabel gaji.

```
cmd C:\Windows\System32\cmd.exe - mysql -u root
MariaDB [kepegawaian]> insert into pegawai values
-> ('2021001','elsan','103'),
-> ('2021119','penta','102'),
-> ('2021862','nugraha','104');
Query OK, 3 rows affected (0.119 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [kepegawaian]> select *from pegawai;
+-----+-----+-----+
| idpegawai | nama    | golongan |
+-----+-----+-----+
| 2021001   | elsan   | 103      |
| 2021119   | penta   | 102      |
| 2021862   | nugraha | 104      |
+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.7 input data kedalam tabel pegawai.

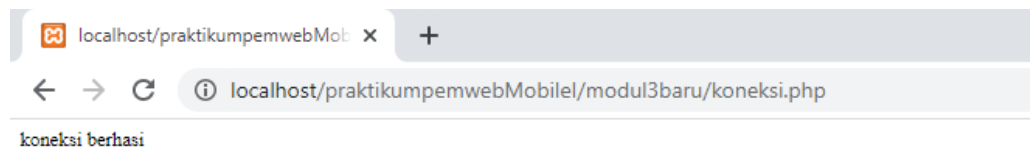
2.2 Pembahasan Soal Kedua

Pada tugas kedua diminta untuk membuat sebuah program web untuk menampilkan data yang ada pada database kepegawaian berserta program yang dapat menyimpan, mengubah dan menghapus datanya.

Yang pertama dilakukan adalah membuat koneksi ke database tersebut. Pada koneksi.php ini dideklarasikan variabel yang bernama koneksi yang akan digunakan untuk menyimpan hasil dari `mysqli_connect` ("localhost", "root", "", "kepegawaian"); dengan server localhost, username root dan database kepegawaian. Kemudian juga dideklarasikan kondisi if yang dimana jika percobaan koneksi gagal maka akan menampilkan pesan “koneksi database gagal” kemudian jika berhasil maka akan menampilkan pesan “koneksi berhasil”. Berikut adalah kode program yang digunakan untuk melakukan koneksi ke database kepegawaian dan pengujiannya.

```
modul3baru > koneksi.php
1  <?php
2  $koneksi = mysqli_connect("localhost","root","", "kepegawaian");
3
4  // Check connection
5  if (mysqli_connect_errno()){
6      echo "Koneksi database gagal : " . mysqli_connect_error();
7  } else {
8      echo "koneksi berhasil";
9  }
10 ?>
```

Gambar 2.2.1 kode program koneksi.



Gambar 2.2.2 pesan koneksi berhasil.

Setelah pengujian berhasil, langkah selanjutnya adalah pembuatan halaman form yang digunakan untuk menampilkan data dari tabel gaji dan tabel pegawai. Pada kedua program halaman ini harus juga dideklarasikan koneksi.php dengan kode program include 'koneksi.php'; agar program laman ini juga dapat terkoneksi dengan database kepegawaian. Berikut adalah hasil tangkapan layar dari kode program pegawai.php dan gaji.php.

Pegawai.php

```
modul3baru > pegawai.php
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6          <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha
7          <title>Pegawai</title>
8      </head>
9      <body>
10         <?php
11             include "koneksi.php";
12         ?>
13         <br/>
14         <div class="container">
15             <div class="row">
16                 <div class="col mt-5">
17                     <a href="tambah_pegawai.php">
18                         <div class="div btn btn-primary mb-4">
19                             Tambah Data
20                         </div>
21                     </a>
22                     <a href="gaji.php">
23                         <div class="div btn btn-primary mb-4">
24                             Ke Halaman Gaji
25                         </div>
26                     </a>
27                 </div>
28             </div>
29         </div>
```

Gambar 2.2.3 baris program pegawai.php

Pada gambar diatas, terdapat kode program:

```
<?php
    include "koneksi.php";
?>
```

Yang digunakan agar program pegawai.php dapat terkoneksi ke database kepegawaian.

```

30 <table border="1" class="table" cellpadding="15" cellspacing="0">
31 <tr>
32 <th>NO</th>
33 <th>NIP</th>
34 <th>Nama</th>
35 <th>golongan</th>
36 <th>Action</th>
37 </tr>
38 <?php
39 $tpegawai = mysqli_query($koneksi, "SELECT * FROM pegawai");
40 $i = 1;
41 while($datapegawai = mysqli_fetch_array($tpegawai)){
42 ?>
43 <tr>
44 <td><?php echo $i++; ?></td>
45 <td><?php echo $datapegawai['idpegawai'] ; ?></td>
46 <td><?php echo $datapegawai['nama'] ; ?></td>
47 <td><?php echo $datapegawai['golongan'] ; ?></td>
48 <td>
49 <a class="edit" href="edit_pegawai.php?id=<?php echo $datapegawai['idpegawai'] ; ?>">Edit</a> |
50 <a class="hapus" href="hapus_pegawai.php?id=<?php echo $datapegawai['idpegawai'] ; ?>"
51 onclick="return confirm('yakin?');">hapus</a>
52 </td>
53 </tr>
54 <?php } ?>
55 </table>
56 </body>
57 </html>

```

Gambar 2.2.4 baris program pegawai.php

Pada gambar diatas, terdapat kode program:

```

<?php

$tpegawai = mysqli_query($koneksi, "SELECT * FROM pegawai");
$i = 1;
while($datapegawai = mysqli_fetch_array($tpegawai)){
?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $datapegawai['idpegawai'] ; ?></td>

<td><?php echo $datapegawai['nama'] ; ?></td>

<td><?php echo $datapegawai['golongan'] ; ?></td>

<td>

<a class="edit" href="edit_pegawai.php?id=<?php echo
$datapegawai['idpegawai'] ; ?>">Edit</a> |

<a class="hapus" href="hapus_pegawai.php?id=<?php echo
$datapegawai['idpegawai'] ; ?>" onclick="return
confirm('yakin?');">hapus</a>

</td>

</tr>

<?php } ?>

```

Kode program ini digunakan untuk menampilkan data pada laman pegawai.php dengan data yang diambil dari tabel pegawai menggunakan query `mysqli_query($koneksi, "SELECT * FROM pegawai");` yang kemudian disimpan kedalam variabel `tpegawai`.

Gaji.php

```
modul3baru > gaji.php
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha
8
9 <title>Gaji</title>
10 </head>
11 <body>
12 <?php
13     include "koneksi.php";
14
15 >
16 <br/>
17 <div class="container">
18     <div class="row">
19         <div class="col mt-5">
20             <a href="tambah_gaji.php">
21                 <div class="div btn btn-primary mb-4">
22                     Tambah Data
23                 </div>
24             </a>
25             <a href="pegawai.php">
26                 <div class="div btn btn-primary mb-4">
27                     Ke Halaman Pegawai
28                 </div>
29             </a>
30         </div>
31     </div>
</br>
```

Gambar 2.2.5 baris program gaji.php

Pada gambar diatas, baris program gaji juga dideklarasikan kode program:

```
<?php
    include "koneksi.php";
?>
```

Yang berarti bahwa laman gaji.php ini juga akan dapat terkoneksi ke database kepegawaian.

```

32     <table border="1" class="table" cellpadding="15" cellspacing="0">
33         <tr>
34             <th>NO</th>
35             <th>Golongan</th>
36             <th>Gaji</th>
37             <th>Tunjangan</th>
38             <th>Action</th>
39         </tr>
40         <?php
41             $tgaji = mysqli_query($koneksi, "SELECT * FROM gaji");
42             $i = 1;
43             while($datagaji = mysqli_fetch_array($tgaji)){
44                 ?>
45                 <tr>
46                     <td><?php echo $i++; ?></td>
47                     <td><?php echo $datagaji['golongan'] ; ?></td>
48                     <td><?php echo $datagaji['gaji'] ; ?></td>
49                     <td><?php echo $datagaji['tunjangan'] ; ?></td>
50                     <td>
51                         <a href="hapus_gaji.php?id=<?php echo $datagaji['golongan']; ?>"
52                           onclick="return confirm('yakin?');">hapus</a>
53                     </td>
54                 </tr>
55             <?php } ?>
56         </table>
57     </body>
58 </html>

```

Gambar 2.2.6 baris program gaji.php

Sama seperti laman pegawai.php, laman gaji.php juga akan menampilkan data yang ada pada database kepegawaian. Pada laman ini tabel yang akan ditampilkan adalah tabel gaji maka pada laman ini query yang digunakan adalah *mysqli_query(\$koneksi, "SELECT * FROM gaji");* kemudian disimpan kedalam variabel *\$tgaji*.

Berikut adalah baris program yang digunakan untuk menampilkan data dari tabel gaji yang berada pada database kepegawaian.

```

<?php
    $tgaji = mysqli_query($koneksi, "SELECT * FROM gaji");
    $i = 1;
    while($datagaji = mysqli_fetch_array($tgaji)){
        ?>
        <tr>
            <td><?php echo $i++; ?></td>
            <td><?php echo $datagaji['golongan'] ; ?></td>
            <td><?php echo $datagaji['gaji'] ; ?></td>
            <td><?php echo $datagaji['tunjangan'] ; ?></td>
            <td>

```

```

        <a href="hapus_gaji.php?id=<?php echo $datagaji["golongan"];
?>" onclick="return confirm('yakin?');">hapus</a>

    </td>

</tr>

<?php } ?>

```

Kemudian yang dilakukan selanjutnya adalah pembuatan laman program *functionpegawai.php* dan *functiongaji.php* yang akan menyimpan fungsi untuk melakukan tambah data, mengubah data, dan menghapus data pada laman *pegawai.php* dan *gaji.php*. Pada kedua program halaman ini harus juga dideklarasikan koneksi.php dengan kode program include '*koneksi.php*'; agar program laman ini juga dapat terkoneksi dengan database kepegawaian. Berikut adalah baris program yang menyimpan fungsi-fungsi tersebut.

Functionpegawai.php

```

<?php
    include 'koneksi.php';

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

    // Fungsi menambahkan data
    function tambah($data){
        global $koneksi;
        $idpegawai = htmlspecialchars($data["idpegawai"]);

```

```

$nama = htmlspecialchars($data["nama"]);
$golongan = htmlspecialchars($data["golongan"]);

$query = "INSERT INTO pegawai
        VALUES
        ('$idpegawai', '$nama', '$golongan')
        ";

mysqli_query($koneksi, $query);

return mysqli_affected_rows($koneksi);
}

//Fungsi mengubah data
function ubah($data){
    global $koneksi;
    $idpegawai = $data["idpegawai"];
    $nama = htmlspecialchars($data["nama"]);
    $golongan = htmlspecialchars($data["golongan"]);
    $query = "UPDATE pegawai SET
            nama = '$nama',
            golongan = '$golongan'
            WHERE idpegawai = $idpegawai
            ";

    mysqli_query($koneksi, $query);

    return mysqli_affected_rows($koneksi);
}

function hapus($id){
    global $koneksi;
    mysqli_query($koneksi, "DELETE FROM pegawai WHERE idpegawai
= $id");

```



```
        return mysqli_affected_rows($koneksi);  
    }  
  
?>
```

Dari baris program diatas, fungsi tambah pada laman ini digunakan untuk memasukan data kedalam tabel pegawai dan juga dideklarasikan variabel koneksi yang merupakan variabel global, kemudian ada juga variabel lokal yaitu idpegawai, nama, dan golongan yang menyimpan data data yang akan dimasukan. Untuk menambah data query yang digunakan adalah *INSERT INTO pegawai VALUES* dengan nilai-nilai yang akan dimasukan diambil dari variabel lokal yang telah dideklarasikan.

```
// Fungsi menambahkan data  
function tambah($data){  
    global $koneksi;  
    $idpegawai = htmlspecialchars($data["idpegawai"]);  
    $nama = htmlspecialchars($data["nama"]);  
    $golongan = htmlspecialchars($data["golongan"]);  
  
    $query = "INSERT INTO pegawai  
              VALUES  
              ('$idpegawai', '$nama', '$golongan')  
              ";  
    mysqli_query($koneksi, $query);  
  
    return mysqli_affected_rows($koneksi);  
}
```

Gambar 2.2.7 fungsi tambah pada laman functionpegawai.php

Kemudian fungsi ubah pada laman ini digunakan untuk mengubah data tabel pegawai dan juga dideklarasikan variabel koneksi yang merupakan variabel global, kemudian ada juga variabel lokal yaitu idpegawai, nama, dan golongan yang menyimpan data data yang akan dimasukan. Untuk mengubah data query yang digunakan adalah *UPDATE pegawai SET* dengan nilai-nilai yang akan diubah diambil dari variabel lokal yang telah dideklarasikan.

```

//Fungsi mengubah data
function ubah($data){
    global $koneksi;
    $idpegawai = $data["idpegawai"];
    $nama = htmlspecialchars($data["nama"]);
    $golongan = htmlspecialchars($data["golongan"]);
    $query = "UPDATE pegawai SET
        nama = '$nama',
        golongan = '$golongan'
        WHERE idpegawai = $idpegawai
    ";
    mysqli_query($koneksi, $query);

    return mysqli_affected_rows($koneksi);
}

```

Gambar 2.2.8 fungsi ubah pada laman functionpegawai.php

Kemudian fungsi hapus pada laman ini digunakan untuk menghapus data tabel pegawai dan juga dideklarasikan variabel koneksi yang merupakan variabel global. Query yang digunakan adalah `mysqli_query($koneksi, DELETE FROM pegawai WHERE idpegawai = $id)`; data yang dihapus akan berdasarkan idpegawai dengan id tersebut akan diambil dari url.

```

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

```

Gambar 2.2.9 fungsi hapus pada laman functionpegawai.php

Functiongaji.php

```

<?php
    include 'koneksi.php';

    function query($query){
        global $koneksi;

```

```
$result = mysqli_query($koneksi, $query);  
$rows = [];  
while($row = mysqli_fetch_assoc($result)){  
    $rows[] = $row;  
}  
return $rows;  
}
```

```
function tambah($data){  
global $koneksi;  
// ambil data dari tiap elemen dalam form  
$golongan = htmlspecialchars($data["golongan"]);  
$gaji = htmlspecialchars($data["gaji"]);  
$tunjangan = htmlspecialchars($data["tunjangan"]);
```

```
$query = "INSERT INTO gaji  
        VALUES  
        ('$golongan', '$gaji', '$tunjangan')  
        ";
```

```
// query insert data  
mysqli_query($koneksi, $query);
```

```
return mysqli_affected_rows($koneksi);  
}
```

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

?>

Dari baris program diatas, fungsi tambah pada laman ini digunakan untuk memasukan data kedalam tabel gaji dan juga dideklarasikan variabel koneksi yang merupakan variabel global, kemudian ada juga variabel lokal yaitu golongan, gaji, dan tunjangan yang menyimpan data data yang akan dimasukan. Untuk menambah data query yang digunakan adalah *INSERT INTO gaji VALUES* dengan nilai-nilai yang akan dimasukan diambil dari variabel lokal yang telah dideklarasikan.

```
function tambah($data){  
    global $koneksi;  
    $golongan = htmlspecialchars($data["golongan"]);  
    $gaji = htmlspecialchars($data["gaji"]);  
    $tunjangan = htmlspecialchars($data["tunjangan"]);  
  
    $query = "INSERT INTO gaji  
              VALUES  
              ('$golongan', '$gaji', '$tunjangan')  
              ";  
    mysqli_query($koneksi, $query);  
    return mysqli_affected_rows($koneksi);  
}
```

Gambar 2.2.10 fungsi tambah pada laman functiongaji.php

Kemudian fungsi hapus pada laman ini digunakan untuk menghapus data tabel gaji dan juga dideklarasikan variabel koneksi yang merupakan variabel global. Query yang digunakan adalah *mysqli_query(\$koneksi, DELETE FROM gaji WHERE golongan = \$id)*; data yang dihapus akan berdasarkan golongan dengan id tersebut akan diambil dari url.

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

Gambar 2.2.11 fungsi hapus pada laman functiongaji.php

Kemudian dibuat laman-laman untuk membuat form tambah data pegawai, form ubah data pegawai, dan laman yang menyimpan program pesan yang akan ditampilkan ketika data dihapus. Berikut adalah baris program dari laman tambah_pegawai.php, edit_pegawai.php, dan hapus_pegawai.php.

Tambah_pegawai.php

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

    $gaji = query("SELECT * FROM gaji");

    if(isset($_POST["submit"])) {
        if (tambah($_POST)>0){
            echo "

                                <script>

                                    alert('data berhasil ditambahkan!');

                                    document.location.href =
'tambah_pegawai.php';

                                </script>

                                ";
        } else {
            echo "

                                <script>

                                    alert('data gagal ditambahkan!');

                                    document.location.href = 'tambah_pegawai.php';

                                </script>

                                ";
        }
    }
?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
```

```

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<title>Tambah Pegawai</title>
</head>
<body>
<div class="container">
<div class="row">
<div class="col">
<h2>Tambah Data Pegawai</h2>
<a href="pegawai.php">
<div class="div btn btn-primary mb-4">
kembali
</div>
</a>
<form action="" method="post">
<ul>
<li class="mb-2">
<label for="idpegawai">NIP&ensp;: </label>
<input type="text" name="idpegawai" id="idpegawai"
required>
</li>
<li class="mb-2">
<label for="nama">Nama : </label>
<input type="text" name="nama" id="nama" required>
</li>
<li class="mb-2">
<div class="form-group">
<label for="golongan">Golongan</label>
<select name="golongan" class="form-control"
id="golongan">
<option selected>Pilih Golongan</option>
<?php foreach($gaji as $dtgaji) : ?>

```

```

                                <option value="<?= $dtgaji["golongan"] ?>"><?=
$dtgaji["golongan"] ?></option>
                                <?php endforeach; ?>
                                </select>
                                </div>
                            </li>
                            <li class="mt-2">
                                <button type="submit" name="submit">Simpan</button>
                            </li>
                        </ul>
                    </form>
                </div>
            </div>
        </div>
    </body>
</html>

```

Dari baris program diatas, baris program yang berada didalam tag body adalah merupakan baris program untuk pembuatan form yang akan digunakan untuk menampung data tambah data pegawai. Kemudian baris program yang berada dalam syntax php diatas digunakan untuk mendeklarasikan require 'functionpegawai.php'; yang digunakan untuk mendeklarasikan bahwa laman ini akan menggunakan fungsi yang berada pada laman functionpegawai.php.

Kemudian juga dideklarasikan kondisi yang akan menampilkan pesan, dengan kondisi yang dideklarasikan jika tombol simpan telah ditekan maka program akan mengecek apakah data telah dimasukan. Jika data telah diisi maka akan menampilkan pesan data berhasil ditambahkan. Tetapi akan menampilkan pesan data gagal ditambahkan jika data yang dimasukan tidak sesuai.

```

<?php
    require 'functionpegawai.php';

    $gaji = query("SELECT * FROM gaji");

    if(isset($_POST["submit"])) {
        if (tambah($_POST)>0){
            echo "
                <script>
                    alert('data berhasil ditambahkan!');
                    document.location.href = 'tambah_pegawai.php';
                </script>
            ";
        } else {
            echo "
                <script>
                    alert('data gagal ditambahkan!');
                    document.location.href = 'tambah_pegawai.php';
                </script>
            ";
        }
    }
}

```

Gambar 2.2.12 baris program pada tambah_pegawai.php

Edit_pegawai.php

```

<?php

require 'functionpegawai.php';

// mengambil data id yang ada pada url
$id = $_GET["id"];

//query select
$pegawai = query("SELECT * FROM pegawai INNER JOIN gaji ON
pegawai.golongan=gaji.golongan
WHERE pegawai.idpegawai = $id")[0];
$gaji = query("SELECT * FROM gaji");

//Syntax untuk mengecek data setelah submit setelah mengedit data pegawai
if(isset($_POST["submit"])) {
    if (ubah($_POST)>0){
        echo "

```



```

        <script>
            alert('data berhasil diubah!');
            document.location.href = 'pegawai.php';
        </script>

        ";
    } else {
        echo "
        <script>

            alert('data gagal diubah!');
            document.location.href = 'pegawai.php';

        </script>

        ";
    }
}
?>

<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
        <title>Ubah Data Pegawai</title>
    </head>
    <body>
        <div class="container">
            <div class="row">
                <div class="col">
                    <h2>Tambah Data Employe</h2>
                    <a href="pegawai.php">

```

```

        <div class="div btn btn-primary mb-4">
            kembali
        </div>
    </a>
    <form action="" method="post">
        <ul>
            <li>
                <label for="nama">Nama Pegawai : </label>
                <input type="text" name="nama" id="nama" required
value="<?php echo $pegawai["nama"]; ?>"

                <input type="hidden" name="idpegawai" id="nama" required
value="<?php echo $pegawai["idpegawai"]; ?>"

            </li>
            <li class="mb-2">
                <div class="form-group">
                    <label for="golongan ">Golongan</label>
                    <select      name="golongan"      class="form-control"
id="golongan">
                        <option value="<?php echo $pegawai["golongan"]; ?>"
selected><?php echo $pegawai["golongan"]; ?></option>
                        <?php foreach($gaji as $dtgaji) : ?>
                            <option value="<?= $dtgaji["golongan"] ?>"><?=
$dtgaji["golongan"] ?></option>
                        <?php endforeach; ?>
                    </select>
                </div>
            </li>
            <li class="mt-2">
                <button type="submit" name="submit">Simpan</button>
            </li>
        </ul>
    </form>
</div>
</div>

```

```
</div>

</body>

</html>
```

Dari baris program diatas, baris program yang berada didalam tag body adalah merupakan baris program untuk pembuatan form yang akan digunakan untuk menampung data data pegawai yang akan diubah. Kemudian baris program yang berada dalam syntax php diatas digunakan untuk mendeklarasikan require 'functionpegawai.php'; yang digunakan untuk mendeklarasikan bahwa laman ini akan menggunakan fungsi yang berada pada laman functionpegawai.php dan juga dideklarasikan variabel id yang akan mengambil data id yang ada pada url laman.

Kemudian juga dideklarasikan kondisi yang akan menampilkan pesan, dengan kondisi yang dideklarasikan jika tombol simpan telah ditekan maka program akan mengecek apakah data telah diubah. Jika data telah diubah maka akan menampilkan pesan data berhasil diubah. Tetapi akan menampilkan pesan data gagal diubah jika data yang dimasukan tidak sesuai.

```
1  <?php
2
3  require 'functionpegawai.php';
4
5  // mengambil data id yang ada pada url
6  $id = $_GET["id"];
7
8  //query select
9  $pegawai = query("SELECT * FROM pegawai INNER JOIN gaji ON pegawai.golongan=gaji.golongan
10 WHERE pegawai.idpegawai = $id")[0];
11 $gaji = query("SELECT * FROM gaji");
12
13 //Syntax untuk mengecek data setelah submit setelah mengedit data pegawai
14 if(isset($_POST["submit"])) {
15     if (ubah($_POST)>0){
16         echo "
17         <script>
18             alert('data berhasil diubah!');
19             document.location.href = 'pegawai.php';
20         </script>
21         ";
22     } else {
23         echo "
24         <script>
25             alert('data gagal diubah!');
26             document.location.href = 'pegawai.php';
27         </script>
28         ";
29     }
30 }
31 ?>
```

Gambar 2.2.13 baris program pada edit_pegawai.php

Hapus_pegawai.php

```
<?php

    require 'functionpegawai.php';

    $id = $_GET["id"];

    if(hapus($id) >0 ){
        echo "
            <script>
                alert('data berhasil dihapus!');
                document.location.href = 'pegawai.php';
            </script>
        ";
    } else {
        echo "
            <script>
                alert('data gagal dihapus!');
                document.location.href = 'pegawai.php';
            </script>
        ";
    }
?>
```

Dari baris program diatas, syntax php diatas digunakan untuk mendeklarasikan *require 'functionpegawai.php';* yang digunakan untuk mendeklarasikan bahwa laman ini akan menggunakan fungsi yang berada pada laman *functionpegawai.php* dan juga dideklarasikan variabel *id* yang akan mengambil data id yang ada pada url laman.

Pada syntax diatas juga dideklarasikan kondisi yang akan menampilkan pesan, dengan kondisi yang dideklarasikan jika tombol hapus telah ditekan maka program akan mengecek apakah data telah dihapus. Jika data telah dihapus maka akan menampilkan pesan data berhasil dihapus. Tetapi akan menampilkan pesan data gagal dihapus jika data tidak dapat dihapus.

Kemudian dibuat laman-laman untuk membuat form tambah data gaji dan laman yang menyimpan program pesan yang akan ditampilkan ketika data gaji dihapus. Berikut adalah baris program dari laman tambah_gaji.php dan hapus_gaji.php.

Tambah_gaji.php

```
<?php

require 'functiongaji.php';

// cek tombol submit sdh ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
            <script>
                alert('data berhasil ditambahkan!');
                document.location.href = 'tambah_gaji.php';
            </script>
        ";
    } else {
        echo "
            <script>
                alert('data gagal ditambahkan!');
                document.location.href = 'tambah_gaji.php';
            </script>
        ";
    }
}
?>

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
```

```

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"

<title>Tambah Gaji</title>
</head>
<body>
<div class="container">
<div class="row">
<div class="col">
<h2>Tambah Data Gaji</h2>
<a href="gaji.php">
<div class="div btn btn-primary mb-4">
kembali
</div>
</a>
<form action="" method="post">
<ul>
<li class="mb-2">
<label for="golongan">golongan : </label>
<input type="text" name="golongan" id="golongan"
required>
</li>
<li class="mb-2">
<label for="gaji">gaji &emsp; &emsp;: </label>
<input type="text" name="gaji" id="gaji" required>
</li>
<li class="mb-2">
<label for="tunjangan">Tunjangan : </label>
<input type="text" name="tunjangan" id="tunjangan"
required>
</li>
<li class="mt-2">
<button type="submit" name="submit">Simpan</button>

```

```
        </li>
    </ul>
</form>
</div>
</div>
</div>
</body>
</html>
```

Dari baris program diatas, baris program yang berada didalam tag body adalah merupakan baris program untuk pembuatan form yang akan digunakan untuk menampung data tambah data pegawai. Kemudian baris program yang berada dalam syntax php diatas digunakan untuk mendeklarasikan require 'functiongaji.php'; yang digunakan untuk mendeklarasikan bahwa laman ini akan menggunakan fungsi yang berada pada laman functiongaji.php.

Kemudian juga dideklarasikan kondisi yang akan menampilkan pesan, dengan kondisi yang dideklarasikan jika tombol simpan telah ditekan maka program akan mengecek apakah data telah dimasukan. Jika data telah diisi maka akan menampilkan pesan data berhasil ditambahkan. Tetapi akan menampilkan pesan data gagal ditambahkan jika data yang dimasukan tidak sesuai.

```

<?php

require 'functiongaji.php';

// cek tombol Simpan(submit) sudah ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
            <script>
                alert('data berhasil ditambahkan!');
                document.location.href = 'tambah_gaji.php';
            </script>
        ";
    } else {
        echo "
            <script>
                alert('data gagal ditambahkan!');
                document.location.href = 'tambah_gaji.php';
            </script>
        ";
    }
}

```

Gambar 2.2.14 baris program pada tambah_pegawai.php

Hapus_gaji.php

```

<?php

require 'functiongaji.php';

$id = $_GET['id'];

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

```



```
        <script>
            alert('data gagal dihapus!');
            document.location.href = 'gaji.php';
        </script>
        ";
    }

?>
```

Dari baris program diatas, syntax php diatas digunakan untuk mendeklarasikan *require 'functiongaji.php'*; yang digunakan untuk mendeklarasikan bahwa laman ini akan menggunakan fungsi yang berada pada laman *functiongaji.php* dan juga dideklarasikan variabel *id* yang akan mengambil data id yang ada pada url laman.

Pada syntax diatas juga dideklarasikan kondisi yang akan menampilkan pesan, dengan kondisi yang dideklarasikan jika tombol hapus telah ditekan maka program akan mengecek apakah data telah dihapus. Jika data telah dihapus maka akan menampilkan pesan data berhasil dihapus. Tetapi akan menampilkan pesan data gagal dihapus jika data tidak dapat dihapus.

2.3 Pengujian

Setelah program telah dibuat maka yang dilakukan selanjutnya adalah pengujian fungsi-fungsi yang telah dibuat. Berikut adalah hasil tangkapan layar dari pengujian program yang meliputi fungsi tampil, tambah, edit dan hapus data dari tabel pegawai dan tabel gaji.

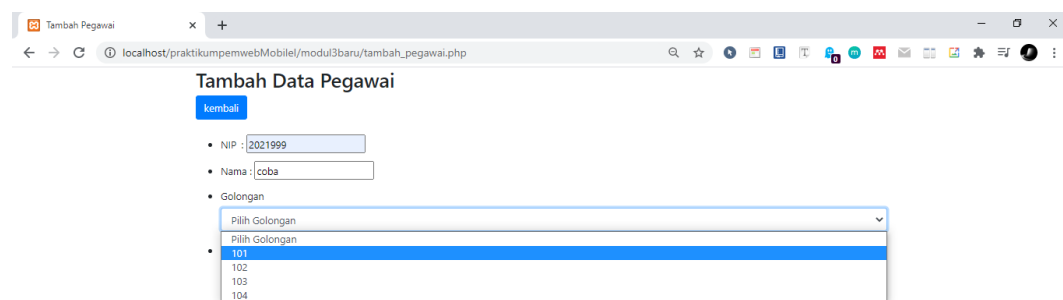
Menampilkan data dari tabel pegawai:



NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus

Gambar 2.3.1 menampilkan data dari tabel pegawai.

Pengujian menambahkan data dengan nilai yang sesuai. Berikut hasil tangkapan layar dari data yang ditambahkan pada tabel pegawai.

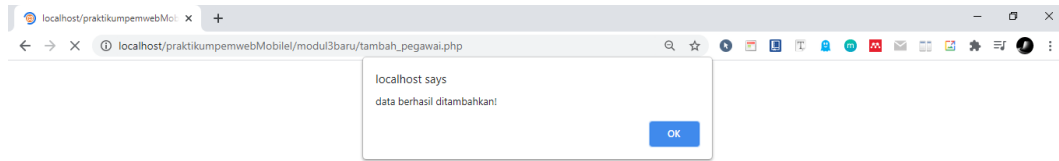


Tambah Data Pegawai

[kembali](#)

- NIP : 2021999
- Nama : coba
- Golongan
 - Pilih Golongan
 - Pilih Golongan
 - 101
 - 102
 - 103
 - 104

Gambar 2.3.2 form tambah data pegawai.



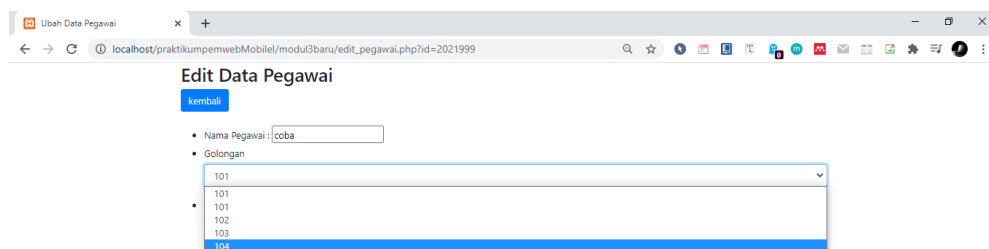
Gambar 2.3.3 pesan data berhasil ditambahkan.

[Tambah Data](#) [Ke Halaman Gaji](#)

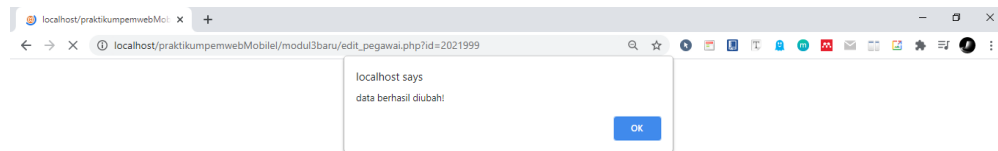
NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus
4	2021999	coba	101	Edit hapus

Gambar 2.3.4 data setelah ditambahkan.

Pengujian edit data pegawai. Berikut hasil tangkapan layar dari data yang telah diubah pada tabel pegawai.



Gambar 2.3.5 form edit data pegawai.



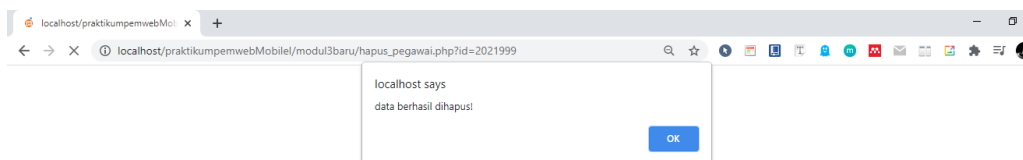
Gambar 2.3.6 pesan data berhasil diubah.

[Tambah Data](#) [Ke Halaman Gaji](#)

NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus
4	2021999	coba	104	Edit hapus

Gambar 2.3.7 data berhasil diubah.

Pengujian penghapusan data pegawai. Berikut hasil tangkapan layar penghapusan data pada tabel pegawai.



Gambar 2.3.8 pesan berhasil dihapus.

Tambah Data		Ke Halaman Gaji		
NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus

Gambar 2.3.9 data berhasil dihapus.

Pengujian menambahkan data dengan nilai yang sesuai. Berikut hasil tangkapan layar dari data yang ditambahkan pada tabel gaji.

Tambah Data		Ke Halaman Pegawai		
NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus

Gambar 2.3.10 data dari tabel gaji.

Tambah Gaji

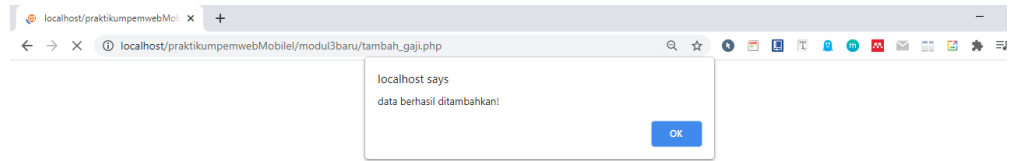
localhost/praktikumpemwebMobile/modul3baru/tambah_gaji.php

Tambah Data Gaji

kembali

- golongan :
- gaji :
- Tunjangan :
-

Gambar 2.3.11 form tambah data gaji.



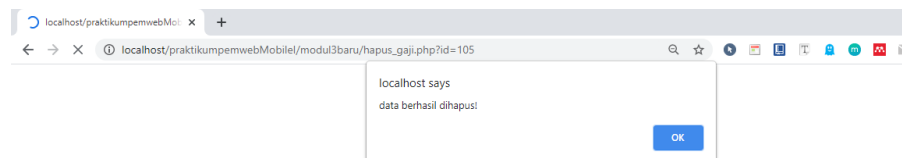
Gambar 2.3.12 pesan data berhasil ditambahkan.

Tambah Data Ke Halaman Pegawai

NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus
5	105	5000000	3500000	hapus

Gambar 2.3.13 data gaji yang telah ditambahkan.

Pengujian penghapusan data pegawai. Berikut hasil tangkapan layar penghapusan data pada tabel gaji.



Gambar 2.3.14 pesan data berhasil dihapus

Tambah Data Ke Halaman Pegawai

NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus

Gambar 2.3.15 data yang telah dihapus.

KESIMPULAN

Kesimpulan yang didapat dari praktikum modul tiga ini adalah CRUD adalah singkatan dari Create, Read, Update, dan Delete. Program CRUD ini dibuat agar user dapat melakukan manipulasi data yang ada pada database melalui aplikasi atau laman web. Create digunakan untuk menambahkan data kedalam database. Read digunakan untuk menampilkan data yang berada pada database dan ditampilkan ke aplikasi. Update digunakan untuk mengubah data dari sebuah kolom yang memiliki id ataupun kode unik tertentu. Dan Delete digunakan untuk menghapus data dari suatu database berdasarkan id ataupun kode unik tertentu. Kemudian ada juga kode program yang digunakan untuk menghubungkan aplikasi atau web yang dibuat dengan database.

DAFTAR PUSTAKA

Praktikum, K. (no date) 'MODUL PRAKTIKUM PEMROGRAMAN WEB I
Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya'.

Shah, C. (2020) 'MySQL', *A Hands-On Introduction to Data Science*, pp. 187–206.
doi: 10.1017/9781108560412.008.

LAMPIRAN

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

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Gambar 1.2.2.1 kode program koneksi php dengan database.

```
$sql = "INSERT INTO Students (name, lastname, email)
VALUES ('Thom', 'Vial', 'thom.v@some.com')";
```

Gambar 1.2.3.1 kode program masukan data melalui mysqli.

```
10 echo <table>
11     <thead>
12         <tr>
13             <th>ID PRODUK</th>
14             <th>TGL TRANSAKSI</th>
15             <th>KUANTITAS</th>
16             <th>HARGA</th>
17             <th>TOTAL BYR</th>
18         </tr>
19     </thead>
20     <tbody>';
21
22 while ($row = mysqli_fetch_array($query))
23 {
24     echo '<tr>
25         <td>'.$row['id_produk'].'</td>
26         <td>'.$row['tgl_transaksi'].'</td>
27         <td>'.$row['kuantitas'].'</td>
28         <td>'.$row['harga'].'</td>
29         <td>'.number_format($row['total_byr'], 0, ',', '.').'</td>
30     </tr>';
31 }
```

Gambar 1.2.4.1 kode program Menampilkan data melalui mysqli.

```
UPDATE nama_tabel
SET kolom1=nilai1, kolom2=nilai2, ...
WHERE kolom_penentu=nilai_penentu ...
```

Gambar 1.2.5.1 kode program mengubah data melalui mysqli.

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

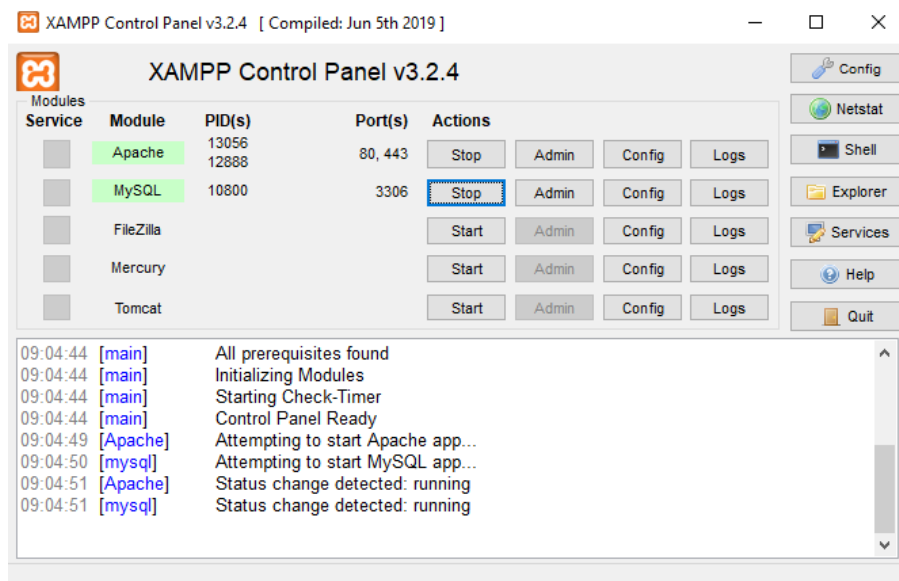
Gambar 1.2.6.1 contoh tabel pada database.

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

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

mysqli_close($conn);
```

Gambar 1.2.6.2 kode program menghapus data melalui mysqli.



Gambar 2.1.1 XAMPP.

```

C:\Windows\System32\cmd.exe - mysql -u root
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

```

Gambar 2.1.2 koneksi ke server MariaDB.

```

C:\Windows\System32\cmd.exe - mysql -u root

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database kepegawaian;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use kepegawaian;
Database changed
MariaDB [kepegawaian]> _

```

Gambar 2.1.3 Pembuatan database.

```

C:\Windows\System32\cmd.exe - mysql -u root

Empty set (0.001 sec)

MariaDB [kepegawaian]> create table gaji(
    -> golongan INT(5) NOT NULL PRIMARY KEY,
    -> gaji BIGINT(25),
    -> tunjangan BIGINT(25)
    -> );
Query OK, 0 rows affected (0.294 sec)

MariaDB [kepegawaian]> show tables;
+-----+
| Tables_in_kepegawaian |
+-----+
| gaji                   |
+-----+
1 row in set (0.001 sec)

MariaDB [kepegawaian]>

```

Gambar 2.1.4 pembuatan tabel gaji.

```
C:\Windows\System32\cmd.exe - mysql -u root

MariaDB [kepegawaian]>
MariaDB [kepegawaian]> create table pegawai(
-> idpegawai INT(11) NOT NULL PRIMARY KEY,
-> nama VARCHAR(50),
-> golongan INT(5),
-> FOREIGN KEY (golongan) REFERENCES gaji(golongan)
-> );
Query OK, 0 rows affected (0.294 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.5 pembuatan tabel pegawai.

```
Select C:\Windows\System32\cmd.exe - mysql -u root

MariaDB [kepegawaian]>
MariaDB [kepegawaian]> insert into gaji values
-> ('101','1500000','500000'),
-> ('102','2000000','750000'),
-> ('103','3000000','1200000'),
-> ('104','3500000','1550000');
Query OK, 4 rows affected (0.117 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [kepegawaian]> select *from gaji;
+-----+-----+-----+
| golongan | gaji      | tunjangan |
+-----+-----+-----+
| 101      | 1500000   | 500000    |
| 102      | 2000000   | 750000    |
| 103      | 3000000   | 1200000   |
| 104      | 3500000   | 1550000   |
+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.6 input data kedalam tabel gaji.

```
C:\Windows\System32\cmd.exe - mysql -u root
MariaDB [kepegawaian]> insert into pegawai values
  -> ('2021001','elsan','103'),
  -> ('2021119','penta','102'),
  -> ('2021862','nugraha','104');
Query OK, 3 rows affected (0.119 sec)
Records: 3  Duplicates: 0  Warnings: 0

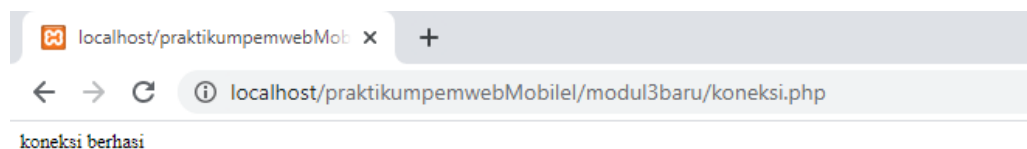
MariaDB [kepegawaian]> select *from pegawai;
+-----+-----+-----+
| idpegawai | nama   | golongan |
+-----+-----+-----+
| 2021001   | elsan  | 103      |
| 2021119   | penta  | 102      |
| 2021862   | nugraha| 104      |
+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [kepegawaian]>
```

Gambar 2.1.7 input data kedalam tabel pegawai.

```
modul3baru > koneksi.php
1  <?php
2  $koneksi = mysqli_connect("localhost","root","","kepegawaian");
3
4  // Check connection
5  if (mysqli_connect_errno()){
6      echo "Koneksi database gagal : " . mysqli_connect_error();
7  } else {
8      echo "koneksi berhasil";
9  }
10 ?>
```

Gambar 2.2.1 kode program koneksi.



Gambar 2.2.2 pesan koneksi berhasil.

```

modul3baru > pegawai.php
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6          <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha
7          <title>Pegawai</title>
8
9      </head>
10     <body>
11         <?php
12             include "koneksi.php";
13         ?>
14         <br/>
15         <div class="container">
16             <div class="row">
17                 <div class="col mt-5">
18                     <a href="tambah_pegawai.php">
19                         <div class="div btn btn-primary mb-4">
20                             Tambah Data
21                         </div>
22                     </a>
23                     <a href="gaji.php">
24                         <div class="div btn btn-primary mb-4">
25                             Ke Halaman Gaji
26                         </div>
27                     </a>
28                 </div>
29             </div>

```

Gambar 2.2.3 baris program pegawai.php

```

30     <table border="1" class="table" cellpadding="15" cellspacing="0">
31         <tr>
32             <th>NO</th>
33             <th>NIP</th>
34             <th>Nama</th>
35             <th>golongan</th>
36             <th>Action</th>
37         </tr>
38         <?php
39             $tpegawai = mysqli_query($koneksi, "SELECT * FROM pegawai");
40             $i = 1;
41             while($datapegawai = mysqli_fetch_array($tpegawai)){
42                 ?>
43                 <tr>
44                     <td><?php echo $i++; ?></td>
45                     <td><?php echo $datapegawai['idpegawai'] ; ?></td>
46                     <td><?php echo $datapegawai['nama'] ; ?></td>
47                     <td><?php echo $datapegawai['golongan'] ; ?></td>
48                     <td>
49                         <a class="edit" href="edit_pegawai.php?id=<?php echo $datapegawai['idpegawai'] ; ?>">Edit</a> |
50                         <a class="hapus" href="hapus_pegawai.php?id=<?php echo $datapegawai['idpegawai'] ; ?>"
51                             onclick="return confirm('yakin?');">hapus</a>
52                     </td>
53                 </tr>
54             <?php } ?>
55         </table>
56     </body>
57 </html>

```

Gambar 2.2.4 baris program pegawai.php

```

modul3baru > gaji.php
1
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha
8
9     <title>Gaji</title>
10  </head>
11  <body>
12    <?php
13      include "koneksi.php";
14
15    ?>
16    <br/>
17    <div class="container">
18      <div class="row">
19        <div class="col mt-5">
20          <a href="tambah_gaji.php">
21            <div class="div btn btn-primary mb-4">
22              Tambah Data
23            </div>
24          </a>
25          <a href="pegawai.php">
26            <div class="div btn btn-primary mb-4">
27              Ke Halaman Pegawai
28            </div>
29          </a>
30        </div>
31      </div>

```

Gambar 2.2.5 baris program gaji.php

```

32   <table border="1" class="table" cellpadding="15" cellspacing="0">
33     <tr>
34       <th>NO</th>
35       <th>Golongan</th>
36       <th>Gaji</th>
37       <th>Tunjangan</th>
38       <th>Action</th>
39     </tr>
40     <?php
41       $tgaji = mysqli_query($koneksi, "SELECT * FROM gaji");
42       $i = 1;
43       while($datagaji = mysqli_fetch_array($tgaji)){
44
45         <tr>
46           <td><?php echo $i++; ?></td>
47           <td><?php echo $datagaji['golongan'] ; ?></td>
48           <td><?php echo $datagaji['gaji'] ; ?></td>
49           <td><?php echo $datagaji['tunjangan'] ; ?></td>
50           <td>
51             <a href="hapus_gaji.php?id=<?php echo $datagaji['golongan']; ?>"
52               onclick="return confirm('yakin?');">hapus</a>
53           </td>
54         </tr>
55       <?php } ?>
56     </table>
57   </body>
58 </html>

```

Gambar 2.2.6 baris program gaji.php

```

// Fungsi menambahkan data
function tambah($data){
    global $koneksi;
    $idpegawai = htmlspecialchars($data["idpegawai"]);
    $nama = htmlspecialchars($data["nama"]);
    $golongan = htmlspecialchars($data["golongan"]);

    $query = "INSERT INTO pegawai
VALUES
('$idpegawai', '$nama', '$golongan')
";
    mysqli_query($koneksi, $query);

    return mysqli_affected_rows($koneksi);
}

```

Gambar 2.2.7 fungsi tambah pada laman functionpegawai.php

```

//Fungsi mengubah data
function ubah($data){
    global $koneksi;
    $idpegawai = $data["idpegawai"];
    $nama = htmlspecialchars($data["nama"]);
    $golongan = htmlspecialchars($data["golongan"]);
    $query = "UPDATE pegawai SET
nama = '$nama',
golongan = '$golongan'
WHERE idpegawai = $idpegawai
";
    mysqli_query($koneksi, $query);

    return mysqli_affected_rows($koneksi);
}

```

Gambar 2.2.8 fungsi ubah pada laman functionpegawai.php

```

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

```

Gambar 2.2.9 fungsi hapus pada laman functionpegawai.php


```

function tambah($data){
    global $koneksi;
    $golongan = htmlspecialchars($data["golongan"]);
    $gaji = htmlspecialchars($data["gaji"]);
    $tunjangan = htmlspecialchars($data["tunjangan"]);

    $query = "INSERT INTO gaji
VALUES
('$golongan', '$gaji', '$tunjangan')
";
    mysqli_query($koneksi, $query);
    return mysqli_affected_rows($koneksi);
}

```

Gambar 2.2.10 fungsi tambah pada laman functiongaji.php

```

function hapus($id){
    global $koneksi;
    mysqli_query($koneksi, "DELETE FROM gaji WHERE golongan = $id");
    return mysqli_affected_rows($koneksi);
}

```

Gambar 2.2.11 fungsi hapus pada laman functiongaji.php

```

<?php
    require 'functionpegawai.php';

    $gaji = query("SELECT * FROM gaji");

    if(isset($_POST["submit"])) {
        if (tambah($_POST)>0){
            echo "
                <script>
                    alert('data berhasil ditambahkan!');
                    document.location.href = 'tambah_pegawai.php';
                </script>
            ";
        } else {
            echo "
                <script>
                    alert('data gagal ditambahkan!');
                    document.location.href = 'tambah_pegawai.php';
                </script>
            ";
        }
    }
}
?>

```

Gambar 2.2.12 baris program pada tambah_pegawai.php

```

1  <?php
2
3  require 'functionpegawai.php';
4
5  // mengambil data id yang ada pada url
6  $id = $_GET["id"];
7
8  //query select
9  $pegawai = query("SELECT * FROM pegawai INNER JOIN gaji ON pegawai.golongan=gaji.golongan
10 WHERE pegawai.idpegawai = $id")[0];
11 $gaji = query("SELECT * FROM gaji");
12
13 //Sintax untuk mengecek data setelah submit setelah mengedit data pegawai
14 if(isset($_POST["submit"])) {
15     if (ubah($_POST)>0){
16         echo "
17             <script>
18                 alert('data berhasil diubah!');
19                 document.location.href = 'pegawai.php';
20             </script>
21         ";
22     } else {
23         echo "
24             <script>
25                 alert('data gagal diubah!');
26                 document.location.href = 'pegawai.php';
27             </script>
28         ";
29     }
30 }
31 ?>

```

Gambar 2.2.13 baris program pada edit_pegawai.php

```

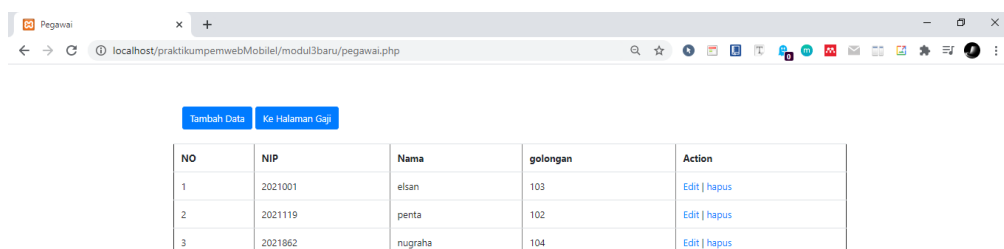
<?php

require 'functiongaji.php';

// cek tombol Simpan(submit) sudah ditekan atau belum
if(isset($_POST["submit"])) {
    // cek apakah data berhasil di tambahkan atau tidak
    if (tambah($_POST)>0){
        echo "
        <script>
            alert('data berhasil ditambahkan!');
            document.location.href = 'tambah_gaji.php';
        </script>
        ";
    } else {
        echo "
        <script>
            alert('data gagal ditambahkan!');
            document.location.href = 'tambah_gaji.php';
        </script>
        ";
    }
}

```

Gambar 2.2.14 baris program pada tambah_pegawai.php



NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus

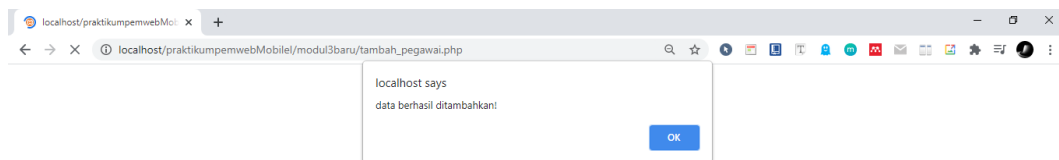
Gambar 2.3.1 menampilkan data dari tabel pegawai.

Tambah Data Pegawai

[kembali](#)

- NIP :
- Nama :
- Golongan
 - Pilih Golongan
 - Pilih Golongan
 - 101
 - 102
 - 103
 - 104

Gambar 2.3.2 form tambah data pegawai.



Gambar 2.3.3 pesan data berhasil ditambahkan.

[Tambah Data](#)
[Ke Halaman Gaji](#)

NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus
4	2021999	coba	101	Edit hapus

Gambar 2.3.4 data setelah ditambahkan.

Ubah Data Pegawai

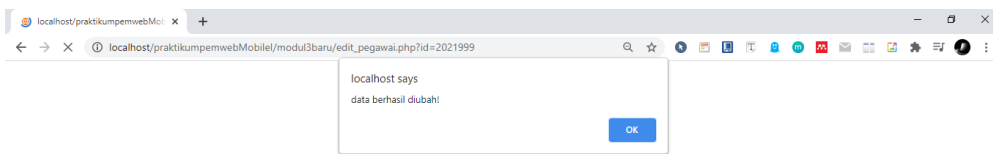
localhost/praktikumpemwebMobile/modul3baru/edit_pegawai.php?id=2021999

Edit Data Pegawai

kembali

- Nama Pegawai :
- Golongan
 - 101
 - 101
 - 102
 - 103
 - 104

Gambar 2.3.5 form edit data pegawai.

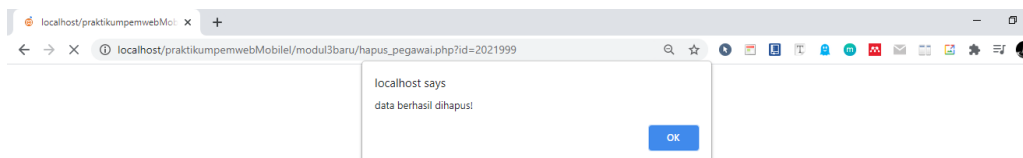


Gambar 2.3.6 pesan data berhasil diubah.

Tambah Data Ke Halaman Gaji

NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus
4	2021999	coba	104	Edit hapus

Gambar 2.3.7 data berhasil diubah.



Gambar 2.3.8 pesan berhasil dihapus.

Tambah Data Ke Halaman Gaji

NO	NIP	Nama	golongan	Action
1	2021001	elsan	103	Edit hapus
2	2021119	penta	102	Edit hapus
3	2021862	nugraha	104	Edit hapus

Gambar 2.3.9 data berhasil dihapus.

Gaji

localhost/praktikumpemwebMobile/modul3baru/gaji.php

Tambah Data Ke Halaman Pegawai

NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus

Gambar 2.3.10 data dari tabel gaji.

Tambah Gaji

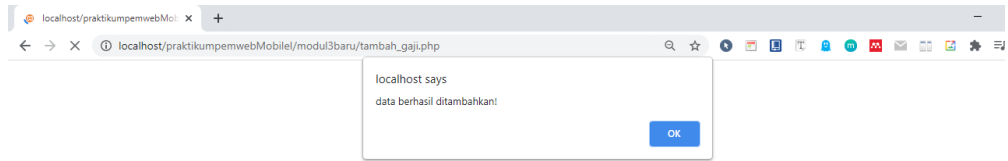
localhost/praktikumpemwebMobile/modul3baru/tambah_gaji.php

Tambah Data Gaji

kembali

- golongan :
- gaji :
- Tunjangan :
-

Gambar 2.3.11 form tambah data gaji.

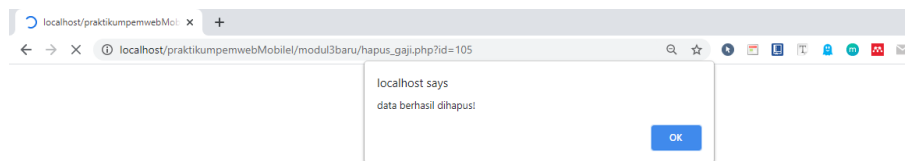


Gambar 2.3.12 pesan data berhasil ditambahkan.

Tambah Data Ke Halaman Pegawai

NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus
5	105	5000000	3500000	hapus

Gambar 2.3.13 data gaji yang telah ditambahkan.



Gambar 2.3.14 pesan data berhasil dihapus

Tambah Data Ke Halaman Pegawai

NO	Golongan	Gaji	Tunjangan	Action
1	101	1500000	500000	hapus
2	102	2000000	750000	hapus
3	103	3000000	1200000	hapus
4	104	3500000	1550000	hapus

Gambar 2.3.15 data yang telah dihapus.