

LAPORAN PRATIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Rakhel Cakra Kusumadinata Sera
NIM : 193030503062
MODUL : III
KELAS : A

JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

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

2. Landasan Teori

Data dalam database MySQL disimpan dalam tabel-tabel. Sebuah tabel adalah koleksi dari data yang berelasi dan mengandung kolom dan baris. Database sangat bermanfaat untuk menyimpan informasi secara kategori. Contoh yang akan diberikan pada modul praktikum ini adalah tabel yang mengandung data Employees (Pekerja), Products (Produk), Customers (Pelanggan) dan Orders (Pesanan).

Membuka Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL. Berikut adalah contoh kode program agar terhubung dengan server mySQL:

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

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

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

Gambar 1.1 Koneksi

Jika kode diatas tidak berhasil, kemungkinan variabel \$connect_error sudah tidak tersedia pada versi PHP yang kita gunakan, maka gunakan kode program berikut:

```
<?php

$servername = "localhost";
$username = "username";
$password = "password";

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

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

echo "Connected successfully";

?>
```

Gambar 1.2 Koneksi

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut: **mysqli_close(\$conn);**

Membuat Database

Database pada MySQL bisa juga dibuat menggunakan kode program PHP. Program tersebut akan berisi statement SQL “CREATE DATABASE”. Berikut adalah contoh program yang digunakan untuk membuat database “myDB”:

```

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    // Membuat Database
    $sql = "CREATE DATABASE myDB";
    if (mysqli_query($conn, $sql)) {

```

Gambar 1,3 Membuat Database

```

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

    mysqli_close($conn);
?>

```

Gambar 1.4 Membuat Database

Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

```

CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP )

```

Berikut adalah contoh program untuk membuat tabel diatas:

```

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

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// SQL untuk membuat tabel
$sql = "CREATE TABLE MyGuests (

```

Gambar 1.5 Membuat Tabel

```

        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
        KEY, firstname VARCHAR(30) NOT NULL, lastname
        VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
    );

    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.6 Membuat Tabel

Memasukkan Data ke Database

Untuk memasukkan data kedalam tabel di database, terdapat beberapa aturan sintaks yang harus diikuti:

- Query SQL harus diberikan kutip dalam PHP.
- Nilai string didalam query SQL harus diberikan kutip.
- Nilai numeris tidak harus diberikan kutip.
- Nilai NULL tidak harus diberikan kutip.

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:

INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)

adapun contoh kode program PHP untuk menambahkan data tersebut adalah sebagai berikut:

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

$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Gambar 1.7 Memasukan Data

Jika kita melakukan perintah INSERT pada tabel yang menggunakan field dengan opsi AUTO_INCREMENT, kita bisa mendapatkan ID dari baris yang terakhir diinputkan. Caranya menggunakan fungsi `mysqli_insert_id($conn)`; nilai kembalian dari fungsi ini akan memberikan id dari record atau baris terakhir yang diinputkan.

Untuk menambahkan record dengan jumlah yang banyak, kita bisa menggunakan fungsi `mysqli_multi_query($conn, $sql)`, berikut adalah contoh program untuk menambahkan record dengan jumlah yang banyak:

```

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

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){

```

Gambar 1.8 Memasukan Data

```

        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Mary', 'Moe', 'mary@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Julie', 'Dooley', 'julie@example.com');";

    if (mysqli_multi_query($conn, $sql)) {
        echo "New records created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

Gambar 1.9 Memasukan Data

Mengambil Data dari

Database Untuk mengambil data, statement SQL yang digunakan adalah SELECT nama_kolom() FROM nama_tabel, atau kita bisa menggunakan karakter * untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan statement WHERE setelah nama_tabel. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:

```

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

// Membuat koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

```

Gambar 1.10 Mengambil Data

```

$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```

Gambar 1.11 Mengambil Data

Menghapus Data

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

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


```

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

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

```

Gambar 1.12 Menghapus

```

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

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

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

mysqli_close($conn);
?>

```

Gambar 1.13 Menghapus

Update Data dalam Database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:

UPDATE nama_tabel

SET kolom1=nilai1, kolom2=nilai2, ...

WHERE kolom_penentu=nilai_penentu ...

Jika statement ini digunakan untuk update data ke dua pada tabel “MyGuest” diatas untuk mengganti nama belakang dari “Moe” menjadi “Doe”, maka kode program untuk update tersebut adalah sebagai berikut:

```

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

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){

```

Gambar 1.4 Update Data

```

        die("Connection failed: " . mysqli_connect_error());
    }

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

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

    mysqli_close($conn);
?>

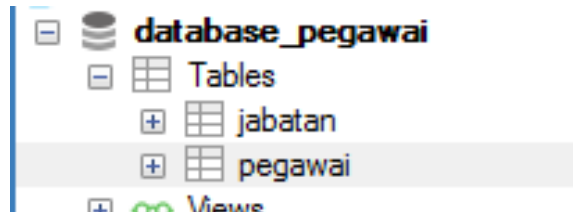
```

Gambar 1.5 Update Data

BAB II

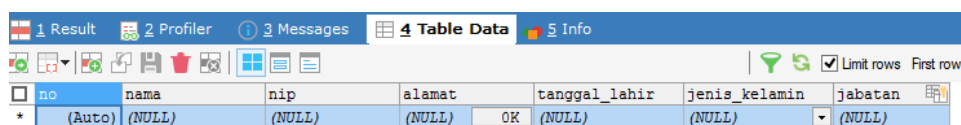
PEMBAHASAN

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

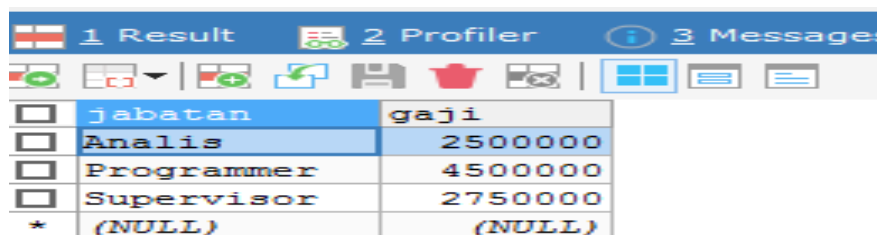


Gambar 2. 1. Pembuatan Database

Untuk database yang dibuat yaitu *database_pegawai*. Database tersebut terdiri dari dua tabel yaitu *jabatan* dan *pegawai*. Nantinya kedua tabel akan saling berkaitan satu sama lain.



Gambar 2. 2. Tampilan data tabel pegawai



Gambar 2. 3. Tampilan data tabel jabatan

Data pada tabel *pegawai* disini masih dibuat kosong. Field yang digunakan yaitu *no*, *nama*, *nip*, *alamat*, *tanggal_lahir*, *jenis_kelamin*, dan *jabatan*. Field *no* digunakan sebagai penentu posisi data. Untuk tipe data yang digunakan pada *nama* dan *nip* yaitu *varchar*, lalu *alamat* yaitu *text*, *tanggal_lahir* yaitu *text*, *jenis_kelamin* yaitu *enum* yang berisi pilihan laki atau wanita, dan terakhir *jabatan* bertipe *varchar*.

Pada gambar 2.3, data yang diisikan pada field *jabatan* yaitu Analis, Programmer, dan Supervisor. Dan untuk field *gaji* telah diisikan masing-masing gaji sesuai jabatan mereka.

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

Index.php

```
<?php
$koneksi = mysqli_connect('localhost', 'root', '', 'database_pegawa
i');
if(!$koneksi){
    exit('File tidak ada');
}
?>
```

Sebelum membuat program untuk menyimpan, menghapus, dan mengubah data-data pegawai. Perlu untuk mengkoneksikan database yang sebelumnya dengan program. Buatlah variable seperti *\$koneksi* yang berisi koneksi database yang telah dikoneksikan menggunakan fungsi *mysqli* yaitu *mysqli_connect(nama_host, username, password, nama_database)*. Lalu dilakukan pengecekan apakah data tersedia. Jika tidak ada, maka akan menampilkan tulisan “File tidak ada”.

Show_tabel.php

```
<?php
include('index.php');
$query = mysqli_query($koneksi, "SELECT pegawai.no,
                                pegawai.nama,
                                pegawai.nip,
                                pegawai.alamat,
                                pegawai.tanggal_lahir,
                                pegawai.jenis_kelamin,
                                pegawai.jabatan,
                                jabatan.gaji
                                FROM jabatan, pegawai
                                where pegawai.jabatan = jabatan.jabatan");
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>
```

Untuk mengkoneksikan data pada database, dapat membuat *include('index.php')*. Setelah itu masukkan query ke variable *\$query* untuk memilih data pada tabel pegawai menggunakan *mysqli_query()*. Setelah *\$query* akan dimasukkan *mysqli_fetch_all()* yang berfungsi untuk mengubah data pegawai pada *\$query* menjadi array.

Show_tabel.php

```
</form>
<table border="1">
  <thead>
    <th>Nama</th>
    <th>NIP</th>
    <th>Alamat</th>
    <th>Tanggal Lahir</th>
    <th>Jenis Kelamin</th>
    <th>Jabatan</th>
    <th>Gaji</th>
    <th>Option</th>
  </thead>
  <tbody>
    <?php foreach ($result as $value) { ?>
      <tr>
        <td><?php echo $value['nama'] ?></td>
        <td><?php echo $value['alamat'] ?></td>
        <td><?php echo $value['nip']?></td>
        <td><?php echo $value['tanggal_lahir'] ?></td>
        <td><?php echo $value['jenis_kelamin'] ?></td>
        <td><?php echo $value['jabatan']?></td>
        <td><?php echo $value['gaji']?></td>
        <td>
          <a href="edit.php?no=<?php echo $value['no']?>"
>Edit</a>
          <br>
          <a href="delete.php?no=<?php echo $value['no']?>"
>">Hapus</a>
        </td>
      </tr>
    <?php } ?>
  </tbody>
</table>
```

Terdapat tabel yang berisi kolom dengan no, nama, nip, alamat, tanggal lahir, jenis kelamin, dan jabatan. Lalu terdapat perulangan foreach dimana variabel \$result sebagai \$value.

Variabel \$value memiliki nilai sesuai dengan nama kolom yang dipanggil, contoh \$value['nama'] mengambil nilai dari kolom nama dan nilai-nilai tersebut diletakkan sesuai dengan kolom-kolom yang telah ditentukan pada tabel sebelumnya.

Lalu terdapat tag <a> yang mana <a> element digunakan untuk mendefinisikan sebuah hyperlink. Dimana hyperlink tersebut terdapat tiga, yakni menuju ke file ubah.php, hapus.php dan terakhir input.php

input.php

```
<form action="transfer.php" method="POST">
  <label>Nama</label><br/>
  <input type="text" name="nama"/>
  <br/><br/>

  <label>Alamat</label><br/>
  <textarea name="alamat" cols="30" rows="10"/></textarea>
  <br/><br/>

  <label>NIP</label>
  <input type="text" name="nip">
  <br/><br/>

  <label>Tanggal Lahir</label><br/>
  <input type="date" name="tanggal_lahir"/>
  <br/><br/>

  <label>Jenis Kelamin</label><br/>
  <select name="jenis_kelamin">
    <option value="Pria">Pria</option>
    <option value="Wanita">Wanita</option>
  </select>
  <br/><br/>
  <label>Jabatan</label><br/>
  <select name="jabatan">
    <option value="Analisis">Analisis</option>
    <option value="Programmer">Programmer</option>
    <option value="Supervisor">Supervisor</option>
  </select>
  <br/><br/>
  <button type="submit">Tambah</button>
</form>
```

Edit.php

```
<?php
include('index.php');
$no = $_GET['no'];
$query = mysqli_query($koneksi, "SELECT * FROM pegawai WHERE pegawai.no
= '$no' LIMIT 1");
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>
<html>
<form action="update.php" method="POST">
    <input type="hidden" name="no" value="<?php echo $result[0]['no']?>"
    >
    <label>Nama</label><br />
    <input type="text" name="nama" value="<?php echo $result[0]['nama'
?>" />
    <br /><br />
    <label>Alamat</label><br />
    <textarea name="alamat" cols="30" rows="10"><?php echo $result[0]['a
alamat'] ?></textarea>
    <br /><br />
    <label>NIP</label>
    <input type="text" name="nip" value="<?php echo $result[0]['nip']?>"
    >
    <br /><br />
    <label>Tanggal Lahir</label><br />
    <input type="date" name="tanggal_lahir" value="<?php echo $result[0]
['tanggal_lahir']?>" />
    <br /><br />
    <label>Jenis Kelamin</label><br />
    <select name="jenis_kelamin">
        <option value="Pria" <?php echo ($result[0]['jenis_kelamin'] ==
'Pria') ? 'selected' : '' ;?>>Pria</option>
        <option value="Wanita" <?php echo ($result[0]['jenis_kelamin'] =
'Wanita') ? 'selected' : '' ;?>>Wanita</option>
    </select>
    <br /><br />
    <label>Jabatan</label><br />
    <select name="jabatan">
        <option value="Analisis" <?php echo ($result[0]['jabatan'] == 'Ana
lisis') ? 'selected' : '' ;?>>Analisis</option>
        <option value="Programmer" <?php echo ($result[0]['jabatan'] ==
'Programmer') ? 'selected' : '' ;?>>Programmer</option>
        <option value="Supervisor" <?php echo ($result[0]['jabatan'] ==
```

Dapat dilihat terdapat bentuk algoritma yang kurang lebih sama, yaitu file input.php dan edit.php. File ini adalah sebagai tempat atau wadah untuk menginputkan atau memasukan data ke dalam kolom-kolom dari table pegawai. Dapat dilihat dalam sintaks tersebut terdapat tag form, yang mana form merupakan bagian pada HTML yang dapat digunakan untuk membuat elemen Form pada halaman Web, form menjadi tempat penginputan data sebelum diproses oleh system dan tempat mengganti penginputan data.

Yang membedakannya adalah tindakan yang akan dilakukan setelah penginputan. Dapat dilihat pada input.php yaitu hanya menginputkan data dan tinggal mengirimkan menggunakan POST menuju file transfer.php. Sedangkan edit.php harus menggunakan value *id* yang didapatkan menggunakan GET dan digunakan untuk key mencari data tersebut. Sehingga data sebelumnya dapat dilihat dengan menfilter menggunakan *id*.

Dan didalam form tersebut terdapat action, yang mana action digunakan untuk mendefinisikan tindakan yang dilakukan oleh form dikirimkan. Disitu tindakan yang dilakukan adalah memproses file dengan nama insert.php.

Hapus.php

```
<?php
include('index.php');
$no = $_GET['no'];
$delete = mysqli_query($koneksi, "DELETE FROM pegawai
                                WHERE pegawai.no = '$no'");

if ($delete) {
    header('Location:show_tabel.php');
} else {
    echo "File tidak ada";
}
?>
```

Fungsi utama dari file tersebut adalah untuk menghapus data pada tabel pegawai. Terdapat variabel \$delete dengan query DELETE FROM yang berfungsi untuk menghapus data dari tabel yang diinginkan dengan kondisi pegawai.no = '\$no'.

Lalu terdapat percabangan if, dimana jika program berjalan dengan benar maka program akan masuk ke file show_tabel.php, jika tidak maka program akan mengeluarkan output “File tidak ada”.

update.php

```
include('index.php');
$no = $_POST['no'];
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$nip = $_POST['nip'];
$tanggallahir = $_POST['tanggal_lahir'];
$jenisKelamin = $_POST['jenis_kelamin'];
$jabatan = $_POST['jabatan'];

$query = mysqli_query($koneksi, "    UPDATE pegawai
    SET
    nama='$nama',
    alamat='$alamat',
    nip='$nip',
    tanggal_lahir='$tanggallahir',

    jenis_kelamin='$jenisKelamin',

    jabatan='$jabatan'
    WHERE pegawai.no = '$no'");

if ($query) {
    header('Location:show_tabel.php');
}else {
    echo 'Input gagal';
}
```

transfer.php

```
<?php

include('index.php');
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$nip = $_POST['nip'];
$tanggalLahir = $_POST['tanggal_lahir'];
$jenisKelamin = $_POST['jenis_kelamin'];
$jabatan = $_POST['jabatan'];

$query = mysqli_query($koneksi, "INSERT INTO pegawai
set
nama='$nama',
alamat='$alamat',
nip='$nip',
tanggal_lahir='$tanggalLahir',
jenis_kelamin='$jenisKelamin',
jabatan='$jabatan' ");

if ($query) {
    header('Location:show_tabel.php');
}else {
    echo 'Input gagal';
}

?>
?>
```

Dapat dilihat terdapat bentuk algoritma yang kurang lebih sama, yaitu file transfer.php dan update.php. File ini adalah sebagai tempat atau wadah setelah menginputkan atau memasukan data ke dalam kolom-kolom dari table pegawai. Dapat dilihat dalam terdapat variable yang menyimpan data dari file *input.php* dan *edit.php*. Seperti contohnya variabel *\$nama* yang menyimpan *\$_POST(nama)*. Tapi yang membedakan variable tranfer.php dan update.php adalah terdapat variable *\$id*.

Lalu terdapat variabel *\$query* dimana query yang digunakan adalah Update pegawai Set nama kolom yang diubah = nama sesuai dengan form, lalu kondisinya adalah pegawai.no = variabel *\$no*.

Lalu terdapat percabangan if, dimana jika program berjalan dengan benar maka program akan masuk ke file show_tabel.php, jika tidak maka program akan mengeluarkan output Input Gagal.

BAB III

KESIMPULAN

Untuk menghubungkan database ke website dapat menggunakan mysqli dan PHP. Dan terdapat bermacam fungsi-fungsi sesuai kondisi yang diperlukan pada suatu program. Perintah pada query seperti SELECT, UPDATE, DELETE, dll dapat digunakan dengan menggunakan *mysqli_query(nama_koneksi, perintah_query)*.

DAFTAR PUSTAKA

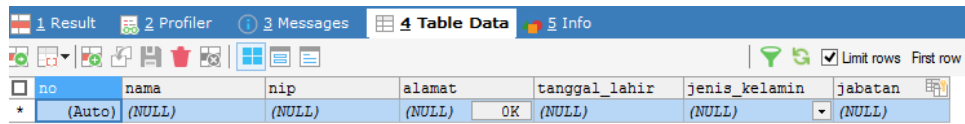
Dosen Teknik Informatika. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya. MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

Hadi, D. A. (n.d.). *Membuat CRUD Dengan PHP Dan MySQL – Menampilkan Data Dari Database.* Malasngoding.
<https://www.malasngoding.com/membuat-crud-dengan-php-dan-mysql-menampilkan-data-dari-database/>

Unknown. (2021). *Mengenal Properti dan Fungsi MySQLi.* Eplusgo.
<https://www.eplusgo.com/mengenal-properti-dan-fungsi-mysqli/>

Yukawa, K. (2019). *Tutorial Mendesain Form Login Menggunakan HTML&CSS _ by Kimiyuki Yukawa _ Medium.* HTML dan CSS.

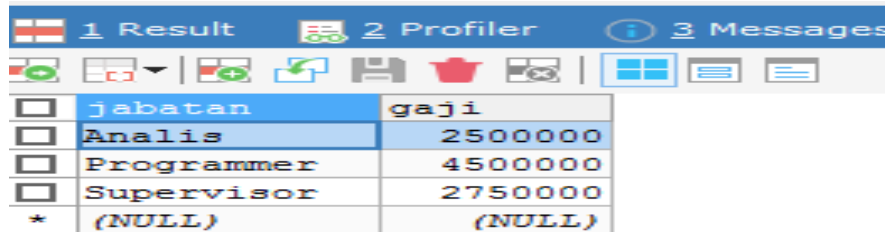
LAMPIRAN



The screenshot shows a database management interface with a table named 'pegawai'. The table has the following columns: 'no', 'nama', 'nip', 'alamat', 'tanggal_lahir', 'jenis_kelamin', and 'jabatan'. The first row is highlighted in blue and contains the values: '(Auto)', '(NULL)', '(NULL)', '(NULL)', 'OK', '(NULL)', and '(NULL)'. The interface includes a toolbar with various icons and a status bar at the bottom.

no	nama	nip	alamat	tanggal_lahir	jenis_kelamin	jabatan
*	(Auto)	(NULL)	(NULL)	OK	(NULL)	(NULL)

Gambar 2. 4. Tampilan data tabel pegawai



The screenshot shows a database management interface with a table named 'jabatan'. The table has the following columns: 'jabatan' and 'gaji'. The first row is highlighted in blue and contains the values: 'Analisis' and '2500000'. The second row contains 'Programmer' and '4500000'. The third row contains 'Supervisor' and '2750000'. The fourth row is highlighted in blue and contains the values: '*' and '(NULL)'. The interface includes a toolbar with various icons and a status bar at the bottom.

jabatan	gaji
Analisis	2500000
Programmer	4500000
Supervisor	2750000
*	(NULL)

Gambar 2. 5. Tampilan data tabel jabatan