

PRAKTIKUM 5

KONEKSI KE MYSQL

1. Tujuan Praktikum

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat memahami cara untuk:

- 1) Melakukan koneksi ke basis data MySQL
- 2) Membaca isi form dengan PHP
- 3) Melakukan validasi input data dengan PHP
- 4) Menampilkan isi kembali form
- 5) Menambahkan data dari form ke database.
- 6) Mengedit data dari form
- 7) Membuat halaman untuk CRUD (Create, Read, Update, Delete)
- 8) Melakukan query dengan prepare statement
- 9) Melakukan query dengan transaction

2. Tools

Tools yang dibutuhkan untuk melakukan praktikum ini adalah Apache web server, PHP, MySQL dan text editor.

3. Langkah-Langkah Praktikum

3.1. Praktikum 1 : Melakukan koneksi ke basis data

Secara umum langkah-langkah untuk melakukan query ke database adalah:

1. Membuka koneksi ke basis data
2. Memilih database yang akan digunakan
3. Menyusun query
4. Mengeksekusi query, hasilnya disimpan dalam
5. Menampilkan hasilnya
6. Menutup koneksi

PHP library yang digunakan untuk koneksi ke MySQL adalah `mysql`, namun `mysql` tidak dianjurkan sejak PHP 5.5 dan akan segera dihapus pada versi ke depannya, sebagai alternatifnya adalah library `mysqli` (i singkatan dari *improved*). Library `mysqli` dapat

digunakan dengan pendekatan prosedural maupun pendekatan beorientasi objek. Contoh fungsi-fungsi dalam mysqli dapat dilihat pada Tabel 3.1.

Tabel 3.1 Fungsi-fungsi untuk melakukan query ke database dengan mysqli

Pendekatan Prosedural	Pendekatan OO	Keterangan
<code>\$con = mysqli_connect(\$db_host, \$db_username, \$db_password);</code> Untuk pemilihan database sekaligus <code>\$con = mysqli_connect(\$db_host, \$db_username, \$db_password, \$db_database);</code>	<code>\$con = new mysqli(\$db_host, \$db_username, \$db_password);</code> Untuk pemilihan database sekaligus <code>\$con = new mysqli(\$db_host, \$db_username, \$db_password, \$db_database);</code>	Membangun koneksi ke database.
<code>mysqli_select_db(\$con, \$db_database)</code>	<code>\$con->select_db(\$db_database)</code>	Memilih database yang akan digunakan
<code>mysqli_connect_errno()</code>	<code>\$con->connect_errno</code>	Mengembalikan nilai 0 jika koneksi berhasil dan nilai bukan 0 jika gagal.
<code>mysqli_connect_error()</code>	<code>\$con->connect_error</code>	Mengembalikan pesan error jika koneksi gagal.
<code>mysqli_error(\$con)</code>	<code>\$con->error</code>	Mengembalikan pesan error.
<code>\$result = mysqli_query(\$con,\$query);</code>	<code>\$result = \$con->query(\$query)</code>	Mengeksekusi query yang tersimpan di variabel string \$query, hasilnya disimpan di \$result.
<code>\$row = mysqli_fetch_array(\$result)</code>	<code>\$row = \$result->fetch_object()</code>	Mengambil setiap baris pada \$result.
<code>mysqli_num_rows(\$result)</code>	<code>\$result->num_rows</code>	Mengembalikan jumlah baris hasil query.
<code>mysqli_close(\$con);</code>	<code>\$con->close();</code>	Menutup koneksi ke database.

Database yang digunakan untuk praktikum ini adalah database ‘db_book’ sintak untuk DDL dan contoh insert datanya sudah tersedia di dalam file db_book.sql. Buatlah database baru bernama ‘db_book’ lalu import file db_book.sql.

Sebelumnya simpan parameter yang dibutuhkan untuk koneksi ke basis data (nama host database server, username, password serta database yang akan digunakan) dalam sebuah file, misalnya, db_login.php.

```
<?php
//File : db_login.php
//Deskripsi : menyimpan parameter untuk koneksi ke database

$db_host='localhost';
```

```
$db_database='bookorama';
$db_username='root';
$db_password='';
?>
```

Selanjutnya buatlah file `view_books.php` seperti berikut ini untuk menampilkan data buku menggunakan pendekatan prosedural.

```
<!-- File: view_books.php
Deskripsi : menampilkan data buku menggunakan mysqli dengan
pendekatan prosedural
-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html401/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Displaying in an HTML table</title>
</head>
<body>
<table border="1">
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Price</th>
</tr>
<?php
// Include our login information
require_once('db_login.php');
// Connect
$con = mysqli_connect($db_host, $db_username, $db_password,$db_database);
if (mysqli_connect_errno()){
    die ("Could not connect to the database: <br />".
mysqli_connect_error( ));
}
//Assign a query
$query = " SELECT * FROM books ";
// Execute the query
$result = mysqli_query($con,$query);
if (!$result){
    die ("Could not query the database: <br />". mysqli_error($con));
}
// Fetch and display the results
while ($row = mysqli_fetch_array($result)){
    echo '<tr>';
    echo '<td>'.$row['isbn'].'</td>';
    echo '<td>'.$row['author'].'</td> ';
    echo '<td>'.$row['title'].'</td> ';
    echo '<td>'.$row['price'].'</td>';
    echo '</tr>';
}
echo '</table>';
echo '<br />';
echo 'Total Rows = '.mysqli_num_rows($result).'<br />';
mysqli_close($con);
?>
</table>
</body>
</html>
```

Mengapa saat menampilkan pesan error digunakan `die()` bukan `echo`? Apa perbedaannya?

Latihan:

1. Ubahlah kode program untuk menampilkan hasil menjadi seperti berikut:

```
while ($row = mysqli_fetch_row($result)){
    echo 'ISBN: '.$row[0] . '<br />';
    echo 'Author: '.$row[1] . '<br /> ';
    echo 'Title: '.$row[2] . '<br /> ';
    echo 'Price: '.$row[3] . '<br /><br />';
}
```

Simpan file tersebut dengan nama `view_books1.php`. Jalankan dan amati perbedaannya.

2. Ubahlah query menjadi:

```
$query = " SELECT b.isbn AS book_isbn, b.author AS book_author,
b.title AS book_title FROM books b ";
```

Lalu ubah pula kode program untuk menampilkan hasil menjadi seperti berikut:

```
while ($row = mysqli_fetch_array($result)){
    echo 'ISBN: '.$row['book_isbn'] . '<br />';
    echo 'Author: '.$row['book_author'] . '<br /> ';
    echo 'Title: '.$row['book_title'] . '<br /><br /> ';
}
```

Simpan file tersebut dengan nama `view_books2.php`. Jalankan dan amati perbedaannya.

Bagaimana kesimpulan Anda?

3. Buatlah program yang sama dengan file `view_books.php` namun menggunakan pendekatan berbasis objek. Simpan file tersebut dengan nama `view_books3.php`. Jalankan dan amati perbedaannya.
4. Buatlah program untuk menampilkan semua data buku beserta reviewnya, sehingga tampilannya seperti berikut:

```
ISBN : 0-672-31509-2
Title : Teach Yourself GIMP in 24 Hours
Author : Pruitt, et al.
Price : 24.99
Review : No review

ISBN : 0-672-31697-8
Title : Java 2 for Professional Developers
Author : Michael Morgan
Price : 34.99
Review : Morgan's book is clearly written and goes well beyond most of the basic Java books out there.

ISBN : 0-672-31745-1
Title : Installing Debian GNU/Linux
Author : Thomas Down
Price : 24.99
Review : No review

ISBN : 0-672-31769-9
Title : Caldera OpenLinux System Administration Unleashed
Author : Thomas Schenk
Price : 49.99
Review : No review

Total Rows = 4
```

Simpan file tersebut dengan nama `view_reviews.php`.

5. Buatlah program untuk menampilkan semua data order beserta informasi pelanggan dan item yang diorder, sehingga tampilannya seperti gambar berikut:

Name : Michelle Arthur					
Address : 357 North Road / Yarraville					
Email : michelle@gmail.com					
No.	ISBN	Title	Price	Qty	Price X Qty
1	0-672-31509-2	Teach Yourself GIMP in 24 Hours	24.99	1	24.99
1	0-672-31697-8	Java 2 for Professional Developers	34.99	2	69.98
1	0-672-31769-9	Caldera OpenLinux System Administration Unleashed	49.99	2	99.98
				Total	194.95

Date : 2015-11-02					
Name : Julie Smith					
Address : 25 Oak Street / Airport West					
Email : julie@gmail.com					
No.	ISBN	Title	Price	Qty	Price X Qty
1	0-672-31769-9	Caldera OpenLinux System Administration Unleashed	49.99	1	49.99
				Total	49.99

Date : 2015-10-14					
Name : Alan Wong					
Address : 1/47 Haines Avenue / Box Hill					
Email : alan@gmail.com					
No.	ISBN	Title	Price	Qty	Price X Qty
1	0-672-31509-2	Teach Yourself GIMP in 24 Hours	24.99	1	24.99
1	0-672-31769-9	Caldera OpenLinux System Administration Unleashed	49.99	1	49.99
				Total	74.98

Simpan file tersebut dengan nama `view_orders.php`.

3.2. Praktikum 2 : Membaca isi form dengan PHP

Berikut ini adalah file untuk membuat form customer. Pemrosesan form dilakukan pada file itu sendiri, sehingga pada atribut `action` diisi dengan `$_SERVER["PHP_SELF"]`. (Pada tabel customer hanya ada 3 field, namun pada praktikum ini ditambahkan beberapa field input, yaitu gender, email dan hobby, untuk mendemonikan tipe input form)

```
<!-- File : form_customer.php
Deskripsi : Form untuk menerima input data customer -->
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<h2>User Input</h2>
<form method="POST" autocomplete="on" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
<table>
  <tr>
    <td valign="top">Name</td>
    <td valign="top">:</td>
    <td valign="top"><input type="text" name="name" size="30"
maxlength="50" placeholder="Name (max 50 characters)"></td>
  </tr>
  <tr>
    <td valign="top">Address</td>
    <td valign="top">:</td>
    <td valign="top"><textarea name="address" rows="5" cols="30"
placeholder="Address (max 100 characters)"></textarea></td>
  </tr>
  <tr>
    <td valign="top">Gender</td>
    <td valign="top">:</td>
    <td valign="top">
      <input type="radio" name="gender" value="male">Male <br />
      <input type="radio" name="gender" value="female">Female
    </td>
  </tr>
</table>
</form>
</body>
</html>
```


3.3. Praktikum 3 : Validasi form dengan PHP

Form pada file form_customer.php belum ditambahkan kode program untuk penanganan validasi. Berikut ini adalah contoh penanganan validasi untuk field nama. Tambahkan file form_customer.php dengan tambahan kode yang dicetak tebal, lalu simpan dengan nama baru form_customer1.php dan jalankan file tersebut. Bagaimana hasilnya?

Tambahkan pula validasi untuk field lainnya dengan aturan semua field harus diisi, untuk email formatnya harus benar sesuai dengan format email.

```
...
<body>
<?php
if (isset($_POST["submit"])){
    $name = test_input($_POST['name']);
    if ($name == ''){
        $error_name = "Name is required";
    }elseif (!preg_match("/^[a-zA-Z ]*$/",$name)) {
        $error_name = "Only letters and white space allowed";
    }
}

//kode untuk validasi field lainnya ....

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
...
    <tr>
        <td valign="top">Name</td>
        <td valign="top">:</td>
        <td valign="top"><input type="text" name="name" size="30"
maxlength="50" placeholder="Name (max 50 characters)"></td>
        <td valign="top"><span class="error">* <?php echo
$error_name;?></span></td>
    </tr>
    ...

</form>

<?php
echo "<h2>Your Input:</h2>";
echo 'Name = '.$name.'<br />';
...
?>
...
```

3.4. Praktikum 4 : Redisplaying form

Pada file form_customer2.php jika terjadi kesalahan maka isian form akan kembali kosong lagi, untuk menampilkannya kembali tambahkan kode yang dicetak tebal berikut pada field nama dan gender untuk menampilkan kembali isi form. Tambahkan pula untuk field lainnya, kemudian simpan dengan nama form_customer3.php dan jalankan. Bagaimana hasilnya?

```

...
<td><input type="text" name="name" size="30" maxlength="50"
placeholder="Name (max 50 characters)" value="<?php echo $name;?>"></td>
...
<td>
<input type="radio" name="gender" value="male" <?php if
(isset($gender) && $gender=="male") echo "checked";?>>Male <br />
<input type="radio" name="gender" value="female" <?php if
(isset($gender) && $gender=="female") echo "checked";?>>Female
</td>
...

```

3.5. Praktikum 5 : Menambahkan isi form ke database

Pada file form_customer3.php hapus kode program yang berkaitan dengan field gender, email dan hobby karena tidak akan disimpan di database. Perhatikan kode yang dicetak tebal, tambahkan kode program untuk insert data customer ke database pada file form_customer3.php, lalu simpan dengan nama form_customer4.php dan jalankan. Bagaimana hasilnya?

```

<?php
if (isset($_POST["submit"])){
    $name = test_input($_POST['name']);
    if ($name == ''){
        $error_name = "Name is required";
        $valid_name = FALSE;
    }elseif (!preg_match("/^[a-zA-Z ]*$/",$name)) {
        $error_name = "Only letters and white space allowed";
        $valid_name = FALSE;
    }else{
        $valid_name = TRUE;
    }
}

...

//insert data into database
if ($valid_name && $valid_address && $valid_city){
    // Include our login information
    require_once('db_login.php');
    // Connect
    $db = new mysqli($db_host, $db_username,
$db_password,$db_database);
    if ($db->connect_errno){
        die ("Could not connect to the database: <br />". $db-
>connect_error);
    }
    //Assign a query
    $query = " INSERT INTO customers (name, address, city)
VALUES('$name', '$address', '$city') ";
    // Execute the query
    $result = $db->query( $query );
    if (!$result){
        die ("Could not query the database: <br />". $db->error);
    }else{
        echo '1 record added. <br />';
    }
    //close db connection}

...
?>

```


Coba jalankan kembali file form_customer4.php dan isi field address dengan 'Wall's Street', kemudian perhatikan hasilnya. Apakah terjadi error? Mengapa?

Tambahkan kode yang dicetak tebal berikut sebelum perintah query pada file form_customer4.php, lalu simpan kembali.

```
//escape input data
$name = $db->real_escape_string($name);
$address = $db->real_escape_string($address);
$city = $db->real_escape_string($city);
//Assign a query
$query = " INSERT INTO customers (name, address, city)
VALUES('$name', '$address', '$city') ";
```

Jalankan kembali file form_customer4.php dan isi field address dengan 'Wall's Street', kemudian perhatikan hasilnya. Apakah terjadi error? Mengapa?

3.6. Praktikum 6 : Edit data dari form

Berikut ini adalah file untuk menampilkan data customer dan ditambahkan link untuk edit data. Buatlah sebuah folder bernama 'customer' setara dengan lokasi file db_login.php. tulis kode program berikut dan simpan dengan nama view_customer.php di dalam folder customer.

```
<!--File      : view_customer.php
Deskripsi    : menampilkan data customers
-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html401/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Displaying in an HTML table</title>
</head>
<body>
<h2>Customers Data</h2>
<table border="1">
<tr>
<th>No</th>
<th>Name</th>
<th>Address</th>
<th>City</th>
<th>Action</th>
</tr>

<?php
// connect database
require_once('../db_login.php');
$db = new mysqli($db_host, $db_username, $db_password, $db_database);
if ($db->connect_errno){
    die ("Could not connect to the database: <br />". $db->connect_error);
}
//Assign a query
$query = " SELECT * FROM customers ORDER BY customerid ";
// Execute the query
$result = $db->query( $query );
if (!$result){
    die ("Could not query the database: <br />". $db->error);
}
```

```
// Fetch and display the results
$i = 1;
while ($row = $result->fetch_object()){
    echo '<tr>';
    echo '<td>'.$i.'</td>';
    echo '<td>'.$row->name.'</td> ';
    echo '<td>'.$row->address.'</td> ';
    echo '<td>'.$row->city.'</td>';
    echo '<td><a href="edit_customer.php?id='.$row->customerid.'">
Edit </a></td>';
    echo '</tr>';
    $i++;
}
echo '</table>';
echo '<br />';
echo 'Total Rows = '.$result->num_rows;
$result->free();
$db->close();
?>
</table>
</body>
</html>
```

Form untuk edit data customer selanjutnya ditampilkan pada file edit_customer.php. Pada file tersebut juga terdapat kode untuk memproses isi form dan perintah query untuk update data ke database.

```
<?php
//File      : edit_customer.php
//Deskripsi : menampilkan form edit data customer dan mengupdate data ke
database

$id = $_GET['id'];
// connect database
require_once('../db_login.php');
$db = new mysqli($db_host, $db_username, $db_password, $db_database);
if ($db->connect_errno){
    die ("Could not connect to the database: <br />". $db-
>connect_error);
}

if (!isset($_POST["submit"])){
    $query = " SELECT * FROM customers WHERE customerid=".$id." ";
    // Execute the query
    $result = $db->query( $query );
    if (!$result){
        die ("Could not query the database: <br />". $db->error);
    }else{
        while ($row = $result->fetch_object()){
            $name = $row->name;
            $address = $row->address;
            $city = $row->city;
        }
    }
}

}else{
    $name = test_input($_POST['name']);
    if ($name == ''){
        $error_name = "Name is required";
        $valid_name = FALSE;
    }elseif (!preg_match("/^[a-zA-Z ]*$/",$name)) {
        $error_name = "Only letters and white space allowed";
        $valid_name = FALSE;
    }else{
```

```

        $valid_name = TRUE;
    }

    $address = test_input($_POST['address']);
    if ($address == ''){
        $error_address = "Address is required";
        $valid_address = FALSE;
    }else{
        $valid_address = TRUE;
    }

    $city = $_POST['city'];
    if ($city == '' || $city == 'none'){
        $error_city = "City is required";
        $valid_city = FALSE;
    }else{
        $valid_city = TRUE;
    }

    //update data into database
    if ($valid_name && $valid_address && $valid_city){
        //escape inputs data
        $name = $db->real_escape_string($name);
        $address = $db->real_escape_string($address);
        $city = $db->real_escape_string($city);
        //Assign a query
        $query = " UPDATE customers SET name='".$name."',
address='".$address."', city='".$city.'" WHERE customerid='".$id.'" ";
        // Execute the query
        $result = $db->query( $query );
        if (!$result){
            die ("Could not query the database: <br />". $db->error);
        }else{
            echo 'Data has been updated.<br /><br />';
            echo '<a href="view_customer.php">Back to customers
data</a>';

            $db->close();
            exit;
        }
    }
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<h2>User Input</h2>
<form method="POST" autocomplete="on" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]).'?id='.$id;?>">
<table>
    <tr>
        <td valign="top">Name</td>
        <td valign="top">:</td>
        <td valign="top"><input type="text" name="name" size="30"
maxlength="50" placeholder="Name (max 50 characters)" autofocus
value="<?php echo $name;?>"></td>

```

```

        <td valign="top"><span class="error">* <?php echo
$error_name;?></span></td>
    </tr>
    <tr>
        <td valign="top">Address</td>
        <td valign="top">:</td>
        <td valign="top"><textarea name="address" rows="5" cols="30"
placeholder="Address (max 100 characters)"><?php echo
$address;?></textarea></td>
        <td valign="top"><span class="error">* <?php echo
$error_address;?></span></td>
    </tr>
    <tr>
        <td valign="top">City</td>
        <td valign="top">:</td>
        <td valign="top"><select name="city" required>
            <option value="none" <?php if (!isset($city)) echo
'selected="true"';?>>--Select a city--</option>
            <option value="Airport West" <?php if (isset($city) &&
$city=="Airport West") echo 'selected="true"';?>>Airport West</option>
            <option value="Box Hill" <?php if (isset($city) &&
$city=="Box Hill") echo 'selected="true"';?>>Box Hill</option>
            <option value="Yarraville" <?php if (isset($city) &&
$city=="Yarraville") echo 'selected="true"';?>>Yarraville</option>
        </select>
        </td>
        <td valign="top"><span class="error">* <?php echo
$error_city;?></span></td>
    </tr>
    <tr>
        <td colspan="3" valign="top"><br><input type="submit"
name="submit" value="Submit">
    </td>
    </tr>
</table>
</form>
</body>
</html>
<?php
$db->close();
?>

```

Tambahkan link 'Delete' pada file view_customer.php dan sebuah file PHP untuk menghapus data customer, sebelumnya berikan konfirmasi saat akan menghapus data. Konfirmasi penghapusan data dibuat dengan kode PHP.

Tambahkan link 'Detail' pada file view_customer.php dan sebuah file PHP untuk menampilkan detail data customer.

Tambahkan link 'Add Customer' untuk menambah data customer. File untuk add data customer dapat menggunakan file form_customer.php pada praktikum sebelumnya atau memodifikasi file edit_customer.php sehingga file tersebut dapat digunakan untuk menambah atau mengubah data customer.

Tampilan file view_customer.php kurang lebih seperti berikut:

Customers Data

[Add Customer Data](#)

No	Name	Address	City	Action
1	Julie Smith	25 Oak Street	Airport West	Detail Edit Delete
2	Alan Wong	1/47 Haines Avenue	Box Hill	Detail Edit Delete
3	Michelle Arthur	357 North Road	Yarraville	Detail Edit Delete

Total Rows = 3

3.7. Praktikum 7 : Melakukan Query dengan Prepare Statement

Berikut ini adalah contoh file yang melakukan insert data ke table books melauai prepare statement dengan pendekatan berbasis objek.

```
<?php
//Nama File: query_oo_prepare1.php
//Melakukan insert data buku dengan prepare statement

// Include our login information
require_once('db_login.php');
// Connect
$db = new mysqli($db_host,$db_username,$db_password,$db_database);
if ($db->connect_errno){
    die ("Could not connect to the database: <br />". $db->connect_error);
}
// prepare and bind
if($stmt = $db->prepare("INSERT INTO books (isbn,author,title,price)
VALUES (?, ?, ?, ?)")){
    $stmt->bind_param("sssd", $isbn, $author, $title, $price);
    // set parameters and execute
    $isbn = "0-672-31509-3";
    $author = "John Doe";
    $title = "PHP Tutorial";
    $price = 15.78;
    $stmt->execute();

    $isbn = "0-672-31509-4";
    $author = "Mary Moe";
    $title = "MySQL Tutorial";
    $price = 15.90;
    $stmt->execute();

    $isbn = "0-672-31509-5";
    $author = "Dooley";
    $title = "Java Tutorial";
    $price = 10.78;
    $stmt->execute();

    echo "New records created successfully";

    $stmt->close();
}
$db->close();
?>
```

Berikut ini adalah contoh program yang menampilkan data dari table books dengan kriteria ISBN tertentu. Hasil dari query tersebut di-bind ke beberapa variabel.

```

<?php
//Nama file: query_oo_prepare2.php
//Menampilkan data buku dengan isbn tertentu

// Include our login information
require_once('db_login.php');

// Connect
$db = new mysqli($db_host,$db_username,$db_password,$db_database);
if ($db->connect_errno){
    die ("Could not connect to the database: <br />". $db->connect_error);
}
//prepare statement
if ($stmt = $db->prepare("SELECT * FROM books WHERE isbn=?")) {
    //bind parameter
    $stmt->bind_param("s",$keyword);

    //execute prpeapred statement
    $keyword = "0-672-31509-2";
    $stmt->execute();

    /* bind variables to prepared statement */
    $stmt->bind_result($isbn,$author,$title,$price);

    /* fetch values */
    while ($stmt->fetch()) {
        echo 'ISBN: '.$isbn.<br />';
        echo 'Author: '.$author.<br /> ' ';
        echo 'Title: '.$title.<br /> ' ';
        echo 'Price: '.$price.<br /><br />';
    }

    /* close statement */
    $stmt->close();
}
/* close connection */
$db->close();
?>

```

3.8. Praktikum 8 : Transaction

Transaksi digunakan untuk menjalankan beberapa query sebagai satu kesatuan, jika terdapat salah satu eksekusi query yang gagal, maka eksekusi query yang lainnya akan dibatalkan. Transaksi menjamin kondisi basis data agar tetap konsisten.

Berikut ini adalah contoh transaksi untuk insert data pemesanan ke basis data db_book. Input data pemesanan terdiri atas input ke dua tabel, yaitu tabel orders dan order_items. Jika salah satu perintah insert tersebut salah, maka perintah insert yang satunya akan dibatalkan (roll back), sehingga kondisi basis data tetap konsisten.

```

<?php
//Nama file: query_transact_oo.php
//Melakukan insert data pemesanan ke tabel orders dan order_items
menggunakan transaction dengan object oriented style

// Include our login information
require_once('db_login.php');

// Connect
$db = new mysqli($db_host,$db_username,$db_password,$db_database);
if ($db->connect_errno){
    die ("Could not connect to the database: <br />". $db->connect_error);
}

//set autocommit to false
$db->autocommit(false);
$flag = true;

// Assign the query
$query1 = "INSERT INTO orders (orderid,customerid,date) VALUES
(4,1,'2016-10-18')";
$query2 = "INSERT INTO order_items (orderid,isbn,quantity) VALUES (6,'0-
672-31745-1',2)";

$result1 = $db->query($query1);
if (!$result1) {
    $flag = false;
    echo "Error details: " . $db->error . ".";
}
$result2 = $db->query($query2);
if (!$result2) {
    $flag = false;
    echo "Error details: " . $db->error . ".";
}

if ($flag) {
    $db->commit();
    echo "All queries were executed successfully";
} else {
    $db->rollback();
    echo "All queries were rolled back";
}

//close connection
$db->close();
?>

```