

## **Membuat Toko Buku Sederhana dengan PHP - MySQL**




Pada kasus contoh pengimplementasian MySQL di PHP kali ini, akan dibuat aplikasi sederhana Toko Buku yang berisikan book list (daftar buku) dan pemrosesan belanja. Kemampuan dasar PHP yang harus dipahami adalah variabel, array, looping, HTML form, dan fungsi-fungsi MySQL, serta perintah SQL sederhana seperti SELECT dan INSERT. Selain itu, diperlukan kemampuan dasar HTML, serta pengetahuan CSS untuk perapihan tampilan.

Silahkan kamu pelajari kembali dasar-dasar PHP dari halaman <http://doku.saung-foss.web.id/id-php>, agar memudahkan dalam membuat aplikasi ini. Dengan contoh kasus kali ini, diharapkan aplikasi sederhana ini dapat dikembangkan menjadi aplikasi Toko Buku yang lebih kompleks.


### ***Alur Aplikasi***

Aplikasi sederhana ini terdiri dari 3 alur sederhana, yaitu daftar buku, form order dan proses order. Pada alur pertama menampilkan halaman berisikan sejumlah daftar buku yang diambil dari tabel buku yang tersimpan di database. Pada form order, menampilkan form yang berisikan informasi belanja buku dari pengunjung. Dan proses order merupakan alur terakhir yang di dalamnya memproses informasi dari form order yang dimasukkan oleh pengunjung.


**Books List**

-  Book 1: Rp. 25000
-  Book 2: Rp. 10000
-  Book 3: Rp. 13000

Alur 1 - Daftar Buku


**Order**

Nama:

Alamat:

Buku:

Jumlah:

Alur 2 - Form Order / Pembelian

**Terima kasih !!**

Anda telah berhasil melakukan pembelian, berikut data pembelian Anda:

Nama : Ryan

Alamat : Jl. A. Yani no 10 Jakarta

Buku : Book 2

Harga : 5 x 10000 = 50000

Alur 3 - Proses Order / Pembelian

Berikut penjelasan setiap alur dalam aplikasi yang akan dibuat:

- **Alur 1 - Daftar Buku**

Pada alur pertama, ditampilkan daftar buku yang diambil dari tabel buku yang tersimpan di database. Pada halaman ini, terdapat daftar buku beserta harganya.

- **Alur 2 - Form Order**

Halaman ini memunculkan web form yang berisikan kolom-kolom data pembelian seperti nama pembeli, alamat, buku yang dibeli, dan jumlahnya.

- **Alur 3 - Proses Order**

Bagian ini memproses data-data yang dimasukkan dari Form Order pada alur 2, hasil yang diperoleh berupa halaman konfirmasi serta memasukkan hasil pembelian ke dalam tabel.

## ***Membuat Database***

Sebelum melakukan pengkodean aplikasi, persiapkan terlebih dahulu database untuk menyimpan data-data yang diperlukan oleh aplikasi. Kita dapat menggunakan phpMyAdmin untuk mengatur database seperti membuat database, membuat tabel dan field, memasukkan data dan lain sebagainya. Berikut struktur database Aplikasi Toko Buku Sederhana:

Nama database: **bookstore**

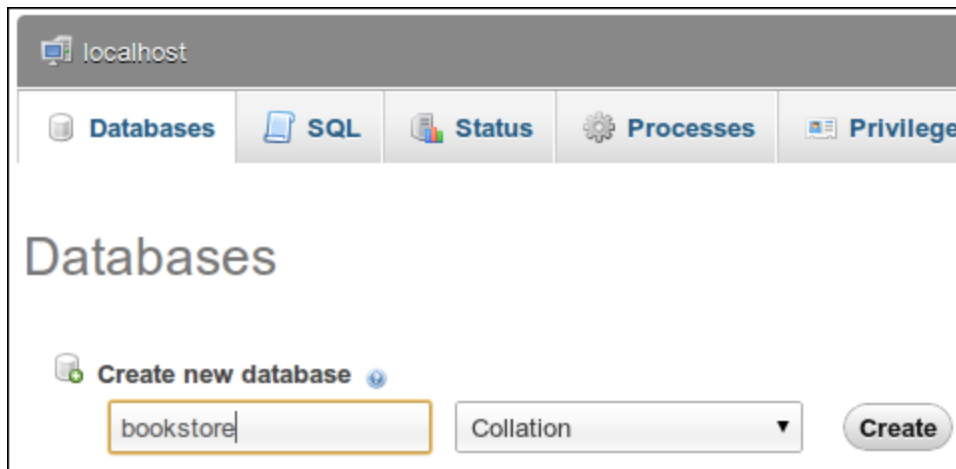
- Tabel **books**: berisikan daftar buku

Field	Type	Deskripsi	Keterangan
book_id	int	ID Buku	Primary Key & AUTO_INCREMENT
book_name	varchar(10)	Nama Buku	-
book_price	decimal(10,2)	Harga Buku	-

- Tabel **orders**: berisikan order / pembelian yang dilakukan oleh pengunjung

Field	Type	Deskripsi	Keterangan
order_id	int	ID Order	Primary Key & AUTO_INCREMENT
order_name	varchar(10)	Nama Pembeli	-
order_address	text	Alamat Pembeli	-
book_id	int	ID Buku	Foreign Key dari tabel books
order_amount	decimal(10,2)	Jumlah Pembelian	-

Pastikan phpMyAdmin telah terinstal, dan buka di peramban web dengan alamat <http://localhost/phpmyadmin>. Lalu pada halaman "Home" phpMyAdmin klik tab "Databases", dan masukkan "bookstore" (tanpa tanda kutip) pada bagian "Create new database".



Membuat database dengan nama "bookstore"

Database dengan nama "bookstore" akan disimpan di MySQL. Pilih database "bookstore" yang telah dibuat sebelumnya, lalu pada bagian "Create table on database bookstore", masukkan nama tabel yang ingin dibuat, tabel pertama yang dibuat adalah "books" dan masukkan banyaknya kolom / field pada tabel tersebut.



Membuat tabel "books" di database "bookstore"

Setelah mendefinisikan nama tabel beserta banyaknya field, langkah selanjutnya adalah mengatur struktur tabel. Di bagian ini, kita menentukan tipe data pada setiap field, menentukan "Storage engine", dan pengaturan lainnya. Untuk tipe field, silahkan lihat pada tabel struktur database di atas. Lakukan langkah yang sama pada pembuatan tabel "orders".

Create Table

Structure

Column	book_id	book_name	book_price
Type	INT	VARCHAR	DECIMAL
Length/Values1		100	10,2
Default2	None	None	None
Collation			
Attributes			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	PRIMARY	---	---
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments			

Table comments:

Storage Engine:

Collation:

Cancel

Mengatur struktur tabel "books"

Isikan beberapa data contoh pada tabel "books". Caranya dengan memilih tabel "books", lalu klik tab "Insert". Masukkan beberapa data di dalamnya.

localhost > bookstore > books

Browse Structure SQL Search Insert Export Import

Column	Type	Function	Null	Value
book_id	int(11)			
book_name	varchar(100)			Buku 1
book_price	decimal(10,2)			50000

Go

☐ Ignore

Column	Type	Function	Null	Value
book_id	int(11)			
book_name	varchar(100)			Buku 2
book_price	decimal(10,2)			25000

Go

Memasukkan data buku di tabel "books"

## ***Mengelola File-file yang Dibutuhkan***

File-file aplikasi terdiri dari file PHP, CSS dan gambar yang dibutuhkan untuk tampilan. Berikut struktur file aplikasi:

Folder images      berisikan gambar-gambar yang dibutuhkan untuk tampilan

list.php              file PHP yang berisikan script untuk menampilkan daftar buku

order.php            file PHP yang berisikan script untuk menampilkan Form Order

order\_process.php file PHP yang berisikan script yang memproses data-data dari Form Order

style.css            file CSS yang digunakan untuk merapihkan tampilan

Name	Size	Type
images	2 items	folder
cart.png	504 bytes	PNG image
list.png	287 bytes	PNG image
list.php	563 bytes	PHP script
order.php	1.0 kB	PHP script
order_process.php	876 bytes	PHP script
style.css	715 bytes	CSS stylesheet

Struktur File Aplikasi

Buat file-file serta folder yang dibutuhkan di [lokasi dokumen web server](#) agar dapat dieksekusi oleh web server. Untuk file-file gambar, silahkan unduh [di sini](#)

### ***Alur 1 - Membuat Daftar Buku***

Semua skrip yang digunakan untuk menampilkan data buku disimpan pada file list.php. Dua langkah dalam menampilkan daftar buku, pertama mengambil data yang tersimpan di dalam database, selanjutnya menampilkan dalam bentuk HTML.

Berikut pengkodean untuk mengambil data daftar buku yang tersimpan dalam database:

```
mysql_connect('localhost', 'root', '');
mysql_select_db('bookstore');
$result = mysql_query("SELECT * FROM books");
```

Pada baris pertama, fungsi mysql\_connect() melakukan koneksi ke MySQL dan dilanjutkan dengan baris berikutnya memilih database 'bookstore' dengan menggunakan fungsi mysql\_select\_db(). Ketika koneksi ke MySQL dan memilih database berhasil, dengan menggunakan perintah "SELECT" untuk mengambil data dari tabel "books" dan hasil dari query yang dilakukan fungsi mysql\_query() dimasukkan ke dalam variabel \$result.

Selanjutnya, data yang diambil dari kode di atas kemudian ditampilkan dalam bentuk HTML. Berikut kode untuk menampilkan data daftar buku:

```
echo "<h1>Books List</h1>";
echo "<ul>";
while($data = mysql_fetch_array($result)) {
    echo "<li>$data[book_name] : Rp. $data[book_price]</li>";
}
echo "</ul>";
```

Fungsi echo digunakan untuk menampilkan keluaran dalam bentuk HTML. Struktur kendali perulangan while digunakan untuk melakukan perulangan dari data-data yang berhasil diperoleh dari query sebelumnya. Fungsi mysql\_fetch\_array() digunakan untuk mengambil data dari variabel \$result yang dikembalikan dalam bentuk array.

Secara utuh, skrip file list.php adalah sebagai berikut:

[list.php](#)

```
<html>
<head>
  <title>Book Store</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<?php
// koneksi ke database
mysql_connect('localhost', 'root', '');
// pilih database
mysql_select_db('bookstore');
// melakukan query
$result = mysql_query("SELECT * FROM books");

// menampilkan ke browser
echo "<h1>Books List</h1>";
echo "<ul>";
// mengambil data
while($data = mysql_fetch_array($result)) {
  echo "<li>$data[book_name] : Rp. $data[book_price]</li>";
}
echo "</ul>";
?>
<a href="order.php">[order]</a>
</body>
</html>
```

## ***Alur 2 - Membuat Form Order***

Pada bagian ini sebenarnya hanya menampilkan bentuk form order yang terdiri dari sintak HTML. Hanya saja, option di dalam tag select membutuhkan data-data dari tabel books, dan memerlukan perintah PHP untuk melakukan pengambilan tersebut. Form order disimpan dengan nama file order.php

Berikut kode form order:

```
<form method="post" action="order_process.php">
  <p>
    Nama: <br />
    <input type="text" name="nama">
  </p>
  <p>
    Alamat: <br />
    <textarea name="alamat"></textarea>
  </p>
  <p>
    Buku: <br />
    <select name="buku">
  <?php
```



```

        // mengambil data dari tabel books
        ?>
    </select>
</p>
<p>
    Jumlah: <br />
    <input type="text" name="jumlah" size="5">
</p>
<p>
    <input type="submit" value="Beli" class="button">
</p>
</form>

```

Pada bagian pemilihan buku, sintak pengambilan datanya ditambahkan menjadi sebagai berikut:

```

<p>
    Buku: <br />
    <select name="buku">
    <?php
    mysql_connect('localhost', 'root', '');
    mysql_select_db('bookstore');
    $result = mysql_query("SELECT * FROM books");
    while($data = mysql_fetch_array($result)) {
        echo "<option value='$data[book_id]'$data[book_name] : Rp. $data[book_price]</option>";
    }
    ?>
    </select>
</p>

```

Secara lengkap kode form order sebagai berikut:

<order.php>

```

<html>
<head>
    <title>Book Store</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h1>Order</h1>
<form method="post" action="order_process.php">
    <p>
        Nama: <br />
        <input type="text" name="nama">
    </p>
    <p>
        Alamat: <br />
        <textarea name="alamat"></textarea>
    </p>
    <p>
        Buku: <br />
        <select name="buku">
        <?php
        //koneksi ke database

```

```

mysql_connect('localhost', 'root', '');
// pilih database
mysql_select_db('bookstore');
// melakukan query
$result = mysql_query("SELECT * FROM books");
while($data = mysql_fetch_array($result)) {
    echo "<option value='$data[book_id]'$data[book_name] : Rp. $data[book_price]</option>";
}
?>
</select>
</p>
<p>
    Jumlah: <br />
    <input type="text" name="jumlah" size="5">
</p>
<p>
    <input type="submit" value="Beli" class="button">
</p>
</form>
</body>
<html>

```

### ***Alur 3 - Membuat Proses Order***

Seperti yang telah didefinisikan melalui atribut "action" yang ada di tag "form" untuk order di atas, file php yang digunakan adalah order\_process.php. Begitu pula dengan metode pengiriman yang didefinisikan dari atribut "method" pada tag "form", metode post digunakan sebagai pengirimannya, dan PHP mengambil data dengan menggunakan variabel global \$\_POST untuk mengambil data tersebut.

Berikut kode pengambilan data dengan \$\_POST:

```

$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$buku = $_POST['buku'];
$jumlah = $_POST['jumlah'];

```

Setiap order yang dilakukan oleh pengunjung disimpan ke dalam tabel "orders", berikut adalah sintak untuk memasukkan data ke dalam tabel orders:

```

mysql_connect('localhost', 'root', '');
mysql_select_db('bookstore');
mysql_query("INSERT INTO orders (order_name, order_address, book_id, order_amount)
    VALUES ('$nama', '$alamat', '$buku', '$jumlah')");

```

Cara memasukkan data hampir sama dengan pengambilan data pada kode-kode sebelumnya, yaitu melakukan koneksi ke MySQL dan memilih database, hanya saja perintah SQL yang digunakan adalah "INSERT" pada fungsi mysql\_query().

Langkah terakhir adalah menampilkan konfirmasi order dengan menggunakan fungsi echo. Tetapi, sebelum menampilkan konfirmasi order, perlu diperhatikan bahwa variabel \$buku yang

dikirim berisikan ID Buku, maka untuk memperoleh informasi nama buku dan harganya dibutuhkan pengambilan data dari tabel books sebelum diolah menjadi tampilan konfirmasi order. Berikut sintaknya:

```
// mengambil data dari tabel books
$result = mysql_query("SELECT * FROM books WHERE book_id = '$buku'");
$data = mysql_fetch_array($result);
$total = $jumlah * $data['book_price'];
// menampilkan konfirmasi order
echo "<h2>Terima kasih !!</h2>";
echo "<p>Anda telah berhasil melakukan pembelian, berikut data pembelian Anda:</p>";
echo "<p>Nama : $nama</p>";
echo "<p>Alamat : $alamat</p>";
echo "<p>Buku : $data[book_name]</p>";
echo "<p>Harga : $jumlah x $data[book_price] = $total</p>";
```

Kode lengkap memproses order sebagai berikut:

[order\\_process.php](#)

```
<html>
<head>
    <title>Book Store</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<?php
// mengambil data yang telah di submit
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$buku = $_POST['buku'];
$jumlah = $_POST['jumlah'];

// koneksi ke database
mysql_connect('localhost', 'root', '');
// pilih database
mysql_select_db('bookstore');
// memasukkan data ke dalam tabel orders
mysql_query("INSERT INTO orders (order_name, order_address, book_id, order_amount)
VALUES ('$nama', '$alamat', '$buku', '$jumlah')");

// mengambil data dari tabel books
$result = mysql_query("SELECT * FROM books WHERE book_id = '$buku'");
$data = mysql_fetch_array($result);
$total = $jumlah * $data['book_price'];
// menampilkan konfirmasi order
echo "<h2>Terima kasih !!</h2>";
echo "<p>Anda telah berhasil melakukan pembelian, berikut data pembelian Anda:</p>";
echo "<p>Nama : $nama</p>";
echo "<p>Alamat : $alamat</p>";
echo "<p>Buku : $data[book_name]</p>";
echo "<p>Harga : $jumlah x $data[book_price] = $total</p>";
?>
</body>
```

<html>

## ***Merapihkan Halaman Web***

Langkah terakhir adalah merapihkan tampilan web dengan CSS. File yang berisikan kode-kode CSS disimpan dengan nama file style.css. Berikut adalah isi dari kode CSS:

[style.css](#)

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 9pt;
}

h1 {
    font-size: 11pt;
    padding-left: 20px;
    background: url(images/cart.png) no-repeat;
}

h2 {
    font-size: 11pt;
    padding-left: 0px;
}

ul li {
    padding-bottom: 3px;
    list-style-image: url(images/list.png);
}

input {
    font-size: 9pt;
    background-color: #F5F5F5;
    border: 1px solid #666666;
}

textarea {
    font-size: 9pt;
    background-color: #F5F5F5;
    border: 1px solid #666666;
}

input.button
{
    font-family : Helvetica, Arial, sans-serif;
    font-size  : 11px;
    font-weight : bold;
    color      : black;
    background-color  : #dddddd;
    border      : 1px solid black;
}
```