

Laporan Praktikum CRUD dan Login PHP Native



Nama Kelompok: Febriana Althafunisa
Haruki Kireina S.
Laecia Octavia

SMK TEXMACO SEMARANG

- Membuat Database di PHP myadmin

1. Buka Xampp dan jalankan mysql
2. Lalu klik admin di mysql
3. Pilih database dan buat baru sesuai yang di inginkan
- Membuat tabel “users” baru di dalam database
 1. Pilih database yang sudah dibuat
 2. Klik Create Table
 3. Buat tabel dengan struktur 3 colom dan dengan colom (id, username, dan password)
 4. Lalu atur Id: INT dengan A.I, Username:Varchar, Password:Varchar
 5. Klik “Insert” lalu masukan data apa saja yang ingin di tampilkan pada tabel
- Membuat penghubung antar visual code dan mysql
 1. Buka folder yang akan di simpan atau pakai
 2. Buat sebuah folder dengan nama “config”
 3. Buat file di dalam folder config dengan nama “database.php”
 4. Isi “database.php” dengan code

```

1  <?php
2  // config/Database.php
3  class Database {
4      private $host = "localhost";
5      private $uname = "root";
6      private $pass = "";
7      private $db = "motor";
8      public $connection;
9
10
11     function __getConnection() {
12         $this->connection = null;
13         try {
14
15             $this->connection = new PDO("mysql:host=" . $this->host . ";dbname=" .
16                 $this->db, $this->uname, $this->pass);
17             $this->connection->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
18         } catch(PDOException $exception) {
19             echo "Connection error:" . $exception->getMessage();
20         }
21         return $this->connection;
22         // check connection
23         if ($this->connection->connect_error) {
24             die("connection failed: " . $this->connection->connect_error);
25         }
26     }
27 }
28
29 ?>
30

```

○ Properti Kelas

- **Private** : Properti privat yang digunakan untuk menyimpan informasi koneksi ke database:
- **\$host**: Alamat server database (biasanya localhost untuk pengembangan lokal).
- **\$uname**: Nama pengguna untuk mengakses database (umumnya root).
- **\$pass**: Kata sandi untuk pengguna database
- **\$db**: Nama database yang akan dihubungkan (film).

- **public \$connection:** Variabel ini akan menyimpan objek koneksi setelah berhasil terhubung. Dideklarasikan menjadi publik untuk bisa diakses dari luar kelas .
- ◆ Metode Kelas
- **__getConnection():** Metode ini bertanggung jawab untuk membuat dan mengembalikan koneksi ke database.
- Pengaturan Mode Kesalahan: Kita mengatur mode kesalahan menjadi **PDO::ERRMODE_EXCEPTION**, yang berarti jika terjadi kesalahan saat melakukan operasi database, PDO akan melemparkan pengecualian (exception) yang dapat ditangani.

● Membuat Index

1. Buat file baru dengan nama “index.php”
2. Lalu isi file tersebut dengan

```

1  <?php
2  // index.php
3  include './config/database.php';
4  include './classes/user.php';
5
6  $database = new Database();
7  $db = $database->__getConnection();
8  $user = new user($db);
9
10 if($_SERVER["REQUEST_METHOD"] == "POST") {
11     $username = $_POST['username'];
12     $password = $_POST['password'];
13
14     if($user->login($username, $password)) {
15         header("Location: tampil.php");
16         exit();
17     } else {
18         $error_message = "username atau password salah!";
19     }
20
21 }
22 ?>
23

```

```

24 <!DOCTYPE html>
25 <html lang="en">
26 <head>
27 <meta charset="UTF-8">
28 <title>login</title>
29 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
30 </head>
31 <body>
32 <div class="container mt-5">
33 <div class="row justify-content-center">
34 <div class="col-md-4">
35 <h3 class="text-center">login</h3>
36 <?php if(isset($error_message)): ?>
37 <div class="alert alert-danger">
38 <?php echo $error_message; ?>
39 </div>
40 <?php endif; ?>
41 <form action="" method="POST">
42 <div class="form-group">
43 <label>Username</label>
44 <input type="text" name="username"
45 class="form-control" required>
46 </div>
47 <div class="form-group">
48 <label>password</label>
49 <input type="password" name="password"
50 class="form-control" required>
51 </div>
52 <button type="submit" class="btn btn-primary
53 btn-block">Login</button>
54 <a href="register.php" class="btn btn-warning btn-block">Registrasi</a>
55 </form>
56 </div>
57 </div>
58 </div>
59 </body>

```

○ Import File

- **Include './config/database.php';, include './classes/user.php';** : Kode ini mengimpor dua file: satu untuk mengatur koneksi database (database.php) dan satu lagi untuk kelas pengguna (user.php).

○ Membuat koneksi

- **\$database = new Database(); \$db = \$database->__getConnection(); \$user = new user(\$db);** : Membuat objek dari kelas Database untuk mendapatkan koneksi ke database.
- Membuat objek user dengan mengoper koneksi database sebagai argumen.

● Membuat halaman awal

1. Buat file baru dengan nama “register.php”
2. Isi file “register.php” dengan

```

1  <?php
2  //register.php
3  require_once 'config/database.php';
4  require_once 'classes/user.php';
5
6  $database = new Database();
7  $db = $database->__getConnection();
8  $user = new User($db);
9
10 if($_SERVER["REQUEST_METHOD"] == "POST") {
11     $username = $_POST['username'];
12     $password = $_POST['password'];
13
14     if($user->register($username, $password)) {
15         $success_mesasage = "Register berhasil silahkan login";
16     } else {
17         $error_message = "Register gagal. Username mungkin sudah digunakan.";
18     }
19 }
20
21
22 <!DOCTYPE html>
23 <html lang="en">
24     <head>
25         <meta charset="UTF-8">
26         <title>login</title>
27         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.c
28 ss" rel="stylesheet">
29     </head>
30     <body>
31         <div class="container mt-5">
32             <div class="row justify-content-center">
33                 <div class="col-md-4">
34                     <h3 class="text-center">login</h3>
35                     <?php if(isset($error_message)): ?>
36                         <div class="alert alert-danger">
37                             <?php echo $error_message; ?>
38                         </div>
39                     <?php endif; ?>
40                     <form action="" method="POST">
41                         <div class="form-group">
42                             <label>Username</label>
43                             <input type="text" name="username" class="form-control" required>
44                         </div>
45                         <div class="form-group">
46                             <label>password</label>
47                             <input type="password" name="password" class="form-control" required>
48                         </div>
49                         <button type="submit" class="btn btn-primary btn-block">Daftar</button>
50                     </form>
51                     <p class="text-center mt-3">Sudah punya akun?
52                         <a href="index.php">Login di sini</a></p>
53                     </div>
54                 </div>
55             </div>
56         </body>

```

- ◆ Halaman register
- **require_once 'config/database.php'; require_once 'classes/user.php';** Untuk mengimport file
- **\$database = new Database(); \$db = \$database->__getConnection(); \$user = new User(\$db);** Untuk mengkonekan file dengan database

- **if(\$_SERVER['REQUEST_METHOD'] == "POST") { \$username = \$_POST['username']; \$password = \$_POST['password'];** Memeriksa method jika formulir telah di submit, jika ya data dari input formulir username dan password akan di ambil

- **Membuat classes**

1. Buat folder baru dengan nama “classes”
2. Buat 3 file dengan nama “user.php”, “film.php”, “logout.php”
3. Isi file “user.php dengan

```
1  <?php
2  // class/User.php
3  require_once './config/database.php';
4
5  class User {
6      private $conn;
7      private $table_name = "user";
8
9      public function __construct($db) {
10         $this->conn = $db;
11     }
12
13     /**
14      *Mengecek kredensial login
15      *@param string $username
16      *@param string $password
17      *@return boolean
18      */
19     public function login($username,$password) {
20         //Query untuk mengambil data user yang sesuai dengan username yang diinput
21         $query = "SELECT *FROM ". $this->table_name . " WHERE username = :username LIMIT
22         1";
23
24         // Mempersiapkan statment untuk query di atas
25         $stmt = $this->conn->prepare($query);
26         $stmt->bindParam(":username", $username);
27         $stmt->execute();
28
29         //Jika ada data user yang sesuai dengan username yang diinput
30         if($stmt->rowCount() > 0) {
31             // Ambil data user yang sesuai dengan user yang diinput
32             $row = $stmt->fetch(PDO::FETCH_ASSOC);
```



```

31         //Jika password yang diinput sesuai dengan password yang tersimpan di databas
e
32         if(password_verify($password, $row['password'])) {
33             //kembalikan true
34             session_start();
35             $_SESSION['username'] = $username;
36             return true;
37         }
38     }
39     // kembalikan false jika tidak ada data user yang sesuai dengan username yang dii
nput atau password yang tersimpan di database
40     return false;
41 }
42 }
43
44 // Fungsi untuk menambahkan user baru
45 public function register($username,$password) {
46     $query = "INSERT INTO " . $this->table_name . " (username, password) VALUES (:use
rname, :password)";
47     $stmt = $this->conn->prepare($query);
48
49     //Enkripsi password sebelum disimpan ke database
50     $hashed_password = password_hash($password, PASSWORD_DEFAULT);
51
52     $stmt->bindParam(":username", $username);
53     $stmt->bindParam(":password", $hashed_password);
54
55     if($stmt->execute()) {
56         return true;
57     }
58     return false;
59 }
60 }
61 ?>

```

4. Isi file “motor.php” dengan’

```

1  <?php
2  require_once "../config/database.php";
3
4  class motor {
5      private $connection;
6      private $table_name = "motor";
7
8      public function __construct($db) {
9          $this->connection = $db;
10     }
11
12     public function get_motor_by_id($id) {
13         $stmt = $this->connection->prepare("SELECT * FROM motor WHERE id = :id");
14         $stmt->bindParam(':id', $id, PDO::PARAM_INT); // Use bindParam for PDO
15         $stmt->execute();
16         return $stmt->fetch(PDO::FETCH_ASSOC); // Correct constant here
17     }
18
19     public function delete_data($id) {
20         $stmt = $this->connection->prepare("DELETE FROM motor WHERE id = :id");
21         $stmt->bindParam(':id', $id, PDO::PARAM_INT); // Use bindParam for PDO

```

```

22     return $stmt->execute();
23 }
24
25 public function update_motor($id, $nama, $merk, $warna) {
26     $stmt = $this->connection->prepare("UPDATE motor SET nama = ?, merk = ?, warna =
? WHERE id = ?");
27     $stmt->execute([$nama, $merk, $warna, $id]); // Execute with an array
28     return $stmt->rowCount(); // Optional: return number of affected rows
29 }
30
31 function tampil_data() {
32     $hasil = [];
33     $data = $this->connection->query("SELECT * FROM " . $this->table_name);
34
35     while ($d = $data->fetch(PDO::FETCH_ASSOC)) { // Correct constant here
36         $hasil[] = $d;
37     }
38
39     return $hasil;
40 }
41 }
42

```

5. Isi file “logout.php” dengan

```

1  <?php
2  session_start();
3  session_unset();
4  session_destroy();
5
6  // Redirect ke halaman login setelah logout
7
8  header("Location: index.php");
9  exit();

```

◆ Logout

- **Session_start();** : Memulai atau melanjutkan sesi yang ada, di panggil sebelum output
 - **Session_unset();** :Menghapus semua variabel yang disimpan, untuk membersihkan data sebelum sesi dihancurkan
 - **Session_destroy();** :Menghancurkan seluruh data
- Header(“Location: index.php”); Exit();** :untuk kembali ke halaman login. Fungsi “exit()” untuk memastikan tidak ada kode yang dieksekusi untuk mencegah masalah.

● Membuat table film

1. Pilih database yang sudah dibuat
 2. Klik Create Table
 3. Isi dengan nama motor
 4. Buat tabel dengan struktur 4 colom dan dengan colom (id, nama, merk, warna)
 5. Lalu atur Id: INT dengan A.I, nama:Varchar, merk:Varchar, dan warna: Varchar
 6. Klik "Insert" lalu masukan data apa saja yang ingin di tampilkan pada table
- Membuat tampilan tabel
 1. Buat file baru di visual code dengan nama "tampil.php"
 2. Isi file dengan

```

1  <?php
2  include './config/database.php';
3  include './classes/motor.php';
4  $db = new Database();
5  $motor = new motor($db-> __getConnection());
6
7  session_start();
8  if (!isset($_SESSION['username'])) {
9      echo "<script>alert('anda belum login.
10      silahkan login terlebih dahulu.');" window.location =
11      'index.php';</script>";
12      exit();
13  }
14  ?>
15
16  <h1>CRUD OOP PHP</h1>
17  <link rel="stylesheet" href="tampilan.css">
18  <a href="input.php">Input Data</a>
19  <table border="3">
20      <tr>
21          <th>Id</th>
22          <th>nama</th>
23          <th>merk</th>
24          <th>warna</th>
25      </tr>
26      <?php
27      $id = 1;
28      foreach($motor->tampil_data() as $x) {
29          ?>
30          <tr>
31              <td><?php echo $id++; ?></td>
32              <td><?php echo htmlspecialchars($x['nama']); ?></td>
33              <td><?php echo htmlspecialchars($x['merk']); ?></td>
34              <td><?php echo htmlspecialchars($x['warna']); ?></td>
35              <td>
36                  <a href="edit.php?id=<?php echo $x['id']; ?>&aksi=edit" onclick="return confir
37                  m('Apakah Anda yakin ingin mengedit data ini?');">Edit</a>
38                  <a href="delete.php?id=<?php echo $x['id']; ?>&aksi=delete" onclick="return co
39                  nfirm('Apakah Anda yakin ingin menghapus data ini?');">Delete</a>
40              </td>
41          </tr>
42          <?php
43      }
44      ?>
45  </table>

```

◆ Code untuk bisa mengedit dan menghapusnya

- `<a href="edit.php?id=<?php echo $x['id']; ?>&aksi=edit" onclick="return confirm('Apakah Anda yakin ingin mengedit data ini?');">Edit`: Untuk tombol edit dan menghubungkan ke halaman edit
 - `<a href="delete.php?id=<?php echo $x['id']; ?>&aksi=delete" onclick="return confirm('Apakah Anda yakin ingin menghapus data ini?');">Delete`: Untuk tombol hapuss dan menghubungkan untuk hapus data
- Membuat halaman Input atau untuk menambahkan data
 1. Buat file baru di visual code dengan nama “input.php”
 2. Isi file dengan

```

1  <?php
2  // Konfigurasi database
3  $host = 'localhost';
4  $username = 'root';
5  $password = '';
6  $database = 'motor';
7
8  $conn = new mysqli($host, $username, $password, $database);
9
10 if ($conn->connect_error) {
11     die("Koneksi gagal: " . $conn->connect_error);
12 }
13
14 if ($_SERVER["REQUEST_METHOD"] == "POST") {
15     $nama = $_POST['nama'];
16     $merk = $_POST['merk'];
17     $warna = $_POST['warna'];
18
19     $sql = "INSERT INTO motor (nama, merk, warna) VALUES (?, ?, ?)";
20     $stmt = $conn->prepare($sql);
21     $stmt->bind_param("sss", $nama, $merk, $warna);
22
23     if ($stmt->execute()) {
24         echo "Data berhasil!";
25     } else {
26         echo "Error: " . $stmt->error;
27     }
28
29     $stmt->close();
30 }
31
32 $conn->close();
33 ?>
34

```

```

35 <!DOCTYPE html>
36 <html lang="en">
37 <head>
38   <meta charset="UTF-8">
39   <meta name="viewport" content="width=device-width, initial-scale=1.0">
40   <title>Motor</title>
41   <link rel="stylesheet" href="input.css">
42 </head>
43 <body>
44   <h1 align="center">Motor</h1>
45   <form method="POST">
46     <label for="nama">Nama:</label>
47     <input type="text" name="nama" id="nama" required>
48     <label for="merk">Merk:</label>
49     <input type="text" name="merk" id="merk" required>
50     <label for="warna">Warna:</label>
51     <input type="text" name="warna" id="warna" required>
52     <input type="submit" value="Simpan">
53   </form>
54 </body>
55 </html>
56

```

◆ Halaman Input

- **if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST['nama']; \$merk = \$_POST['merk']; \$warna = \$_POST['warna'];**
Untuk mengambil data dari input formulir
- **\$sql = "INSERT INTO motor (nama, merk, warna) VALUES (?, ?, ?)";**
\$stmt = \$conn->prepare(\$sql); \$stmt->bind_param("sss", \$nama, \$merk,
\$warna); Untuk memasukan data ke table
- **if (\$stmt->execute()) { echo "Data berhasil!"; } else { echo "Error: " . \$stmt->error; }** Untuk memberi tahu jika data berhasil di tambah dan akan muncul notifikasi eror jika penambahanya gagal
- **Pengubahan pada file**
 1. Buat file baru di visual code dengan nama "edit.php"
 2. Isi file dengan

```

1  <?php
2  include 'database.php';
3  $db = new Database(); // Ensure the class name matches
4
5  // Check if ID is set in the URL
6  if (isset($_GET['id'])) {
7      $id = $_GET['id'];
8
9      // Fetch existing motor data
10     $motor = $db->get_motor_by_id($id); // Ensure this method is defined in your database
11     class
12     if (!$motor) {
13         die("motor tidak ditemukan.");
14     }
15     } else {
16         die("ID motor tidak ditentukan.");
17     }
18
19     // Handle form submission
20     if ($_SERVER["REQUEST_METHOD"] == "POST") {
21         $nama = $_POST['nama'];
22         $merk = $_POST['merk'];
23         $warna = $_POST['warna'];
24
25         if ($db->update_motor($id, $nama, $merk, $warna)) {
26             header("Location: tampil.php"); // Redirect back to the list
27             exit();
28         } else {
29             echo "Error: Gagal mengupdate data.";
30         }
31     }
32     ?>
33

```

```

34 <!DOCTYPE html>
35 <html lang="en">
36 <head>
37     <meta charset="UTF-8">
38     <meta name="viewport" content="width=device-width, initial-scale=1.0">
39     <title>Edit Data Motor</title>
40     <link rel="stylesheet" href="edit.css">
41 </head>
42 <body>
43     <h1>Edit Data Motor</h1>
44     <form method="POST" action="">
45         <label for="nama">Nama:</label><br />
46         <input type="text" id="nama" name="nama" value="php echo htmlspecialchars($motor
47 ['nama']); ?&gt;" required&gt;&lt;br /&gt;&lt;br /&gt;
48         &lt;label for="merk"&gt;Merk:&lt;/label&gt;&lt;br /&gt;
49         &lt;input type="text" id="merk" name="merk" value="<?php echo htmlspecialchars($motor
50 ['merk']); ?&gt;" required&gt;&lt;br /&gt;&lt;br /&gt;
51         &lt;label for="warna"&gt;Warna:&lt;/label&gt;&lt;br /&gt;
52         &lt;input type="text" id="warna" name="warna" value="<?php echo htmlspecialchars($mot
53 or['warna']); ?&gt;" required&gt;&lt;br /&gt;&lt;br /&gt;
54         &lt;input type="submit" value="Update Motor"&gt;
55     &lt;/form&gt;
56
57     &lt;div class="ta"&gt;
58         &lt;a href="tampil.php"&gt;Kembali ke Daftar Motor&lt;/a&gt;
59     &lt;/div&gt;
60 &lt;/body&gt;
61 &lt;/html&gt;
62
</pre

```

- ◆ **\$motor = \$db->get_motor_by_id(\$id);** Mengambil data motor
- ◆ **if (\$db->update_motor(\$id, \$nama, \$merk, \$warna)) { header("Location: tampil.php"); exit(); } else { echo "Error: Gagal mengupdate data."; }:**
Membuat pembaruan motor dalam database dan mengarahkan kembali ke halaman table atau tampil jika pembaruan berhasil
- **Penghapusan file**
 1. Buat file baru di visual code dengan nama “delete.php”
 2. Isi file dengan

```

1  <?php
2  include 'database.php';
3  $db = new database();
4
5  if (isset($_GET['id'])) {
6      $id = $_GET['id'];
7
8      // Call the method to delete the data
9      if ($db->delete_data($id)) {
10         // Redirect back to the main page with a success message
11         header("Location: tampil.php?message=Data berhasil dihapus");
12         exit();
13     } else {
14         // Redirect back with an error message
15         header("Location: tampil.php?message=Gagal menghapus data");
16         exit();
17     }
18 } else {
19     // Redirect back with an error message if no ID is set
20     header("Location: tampil.php?message=ID tidak valid");
21     exit();
22 }
23 ?>

```

- ◆ Menghapus Data pada table
- **if (\$db->delete_data(\$id)) {** Untuk mengeksekusi data pada table agar terhapus
- **header("Location: tampil.php?message=Data berhasil dihapus"); exit();**
Untuk memunculkan notifikasi “Data berhasil dihapus” jika data berhasil di hapus
- **header("Location: tampil.php?message=Gagal menghapus data"); exit();**
Untuk memunculkan notifikasi “gagal menghapus data” jika data tidak berhasil di hapus
- **header("Location: tampil.php?message=ID tidak valid"); exit();** Untuk mengembalikan ke halaman awal jika id tidak ada atau tidak ditemukan dan memunculkan notifikasi “ID tidak Valid”

- Tampilan Pada halaman agar Rapi
 1. Buat file baru di visual code dengan nama “tampil.css”
 2. Isi file dengan

```
1  body {
2    font-family: Arial, sans-se
rif;
3    background-color: #f2eee9;
4    margin: 0;
5    padding: 20px;
6  }
7
8  h1 {
9    text-align: center;
10   color: #333333;
11 }
12
13 a {
14   display: inline-block;
15   margin: 10px;
16   padding: 10px 15px;
17   background-color: #bdd3ce;;
18   color: white;
19   text-decoration: none;
20   border-radius: 5px;
21 }
22
23 a:hover {
24   background-color: #f2f2f2;
25 }
```

```
26
27 table {
28   width: 80%;
29   border-collapse: collapse;
30   margin-top: 20px;
31 }
32
33 table, th, td {
34   border: 1px solid #e85234;;
35 }
36
37 th, td {
38   padding: 12px;
39   text-align: left;
40 }
41
42 tr:nth-child(even) {
43   background-color: #f9b8af;;
44 }
45
46
47 tr:hover {
48   background-color: #f2d7d3;
49 }
50
```

3. Buat file baru di visual code dengan nama "Input.css"
4. Isi file dengan

```
1  body {
2    font-family: Arial, sans-serif;
3    background-color: #f2eeee9;
4    margin: 0;
5    padding: 20px
6  }
7
8  h1 {
9    color: #333;
10   }
11
12  form {
13    background-color: #e85234;
14    border-radius: 5px;
15    box-shadow: 2px 5px rgba(0, 0, 0, 0.1);
16    padding: 20px;
17    max-width: 400px;
18    margin: auto;
19  }
20  label {
21    display: block;
22    margin-bottom: 8px;
23    font-weight: bold;
24  }
```

```
25
26  input[type="text"],
27  input[type="number"] {
28    width: 50%;
29    padding: 10px;
30    margin-bottom: 15px;
31    border: 1px solid #ccc;
32    border-radius: 4px;
33  }
34  input[type="submit"] {
35    background-color: #f9b8af;
36    color: white;
37    padding: 10px;
38    border: none;
39    border-radius: 4px;
40    cursor: pointer;
41    width: 50%;
42  }
43
44  input[type="submit"]:hover {
45    background-color: #f9b8af;
46  }
```