

MODUL 2

CRUD KE DATABASE DALAM PHP OOP

A. MAKSUD DAN TUJUAN

1. Memahami konsep dasar enkapsulasi dalam pemrograman berorientasi objek (OOP).
2. Menerapkan CRUD (Create, Read, Update, Delete) menggunakan PHP OOP dan MySQL.
3. Menggunakan PDO untuk berinteraksi dengan database.

B. ALAT, BAHAN DAN PERANGKAT

Alat, bahan, perangkat lunak, perangkat keras yang digunakan adalah:

1. Sistem Operasi Windows.
2. Teks editor Visual Studio Code.
3. Server local Laragon.
4. Database MySQL.

C. DASAR TEORI

Enkapsulasi adalah salah satu prinsip dasar dalam OOP yang bertujuan untuk membatasi akses ke properti dan metode dalam suatu class agar tidak dapat diakses secara langsung dari luar class tersebut. Hal ini bertujuan untuk menjaga integritas data dan mencegah perubahan yang tidak diinginkan.

CRUD adalah operasi dasar dalam pengelolaan database yang terdiri dari:

Create → Menambahkan data ke database.

Read → Membaca dan menampilkan data dari database.

Update → Memperbarui data yang ada dalam database.

Delete → Menghapus data dari database.

PDO (PHP Data Objects) adalah ekstensi di PHP yang menyediakan antarmuka yang konsisten untuk mengakses berbagai database dengan cara yang lebih aman dan fleksibel. Dengan PDO, Anda bisa menggunakan berbagai jenis database seperti MySQL, PostgreSQL, SQLite, dan lainnya tanpa harus mengubah banyak kode.

D. PRE-TEST

1. Jelaskan konsep enkapsulasi dalam OOP dan mengapa penting?
2. Sebutkan dan jelaskan tiga jenis modifier akses dalam PHP OOP.
3. Bagaimana cara menghubungkan PHP dengan database menggunakan PDO?
4. Apa perbedaan antara private dan protected dalam PHP?
5. Sebutkan langkah-langkah dasar dalam membuat CRUD menggunakan PHP OOP.

E. PRAKTIKUM

Latihan 1

1. Buat file baru dengan nama latihan_02_01.php
2. Tambahkan kode program berikut

```
<?php
class Alumni {
    private $nama;
    private $email;
    private $password;
    public function __construct($nama, $email, $password) {
        $this->nama = $nama;
        $this->email = $email;
        $this->password = password_hash($password, PASSWORD_DEFAULT);
    }
    public function getNama() {
        return $this->nama;
    }
    public function getEmail() {
        return "Email alumni bersifat rahasia!";
    }
    public function verifyPassword($password) {
        return password_verify($password, $this->password);
    }
}
$alumni = new Alumni("Budi Santoso", "budi@example.com", "password123");
echo $alumni->getNama();
echo $alumni->getEmail();

if ($alumni->verifyPassword("password123")) {
    echo "Password benar!";
} else {
    echo "Password salah!";
}
?>
```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_01.php'**
4. Amati hasilnya

Latihan 2

1. Buat file baru dengan nama latihan_02_02.php
2. Tambahkan kode program berikut

```
<?php
class Alumni {
    protected $nama;
    protected $tahunLulus;
```

```

    public function __construct($nama, $tahunLulus) {
        $this->nama = $nama;
        $this->tahunLulus = $tahunLulus;
    }
    public function getInfo() {
        return "Nama: $this->nama, Tahun Lulus: $this->tahunLulus";
    }
}
class Admin extends Alumni {
    public function updateTahunLulus($tahunBaru) {
        $this->tahunLulus = $tahunBaru;
        return "Tahun lulus diperbarui menjadi: $this->tahunLulus";
    }
}
$admin = new Admin("Milchatun", 2015);
echo $admin->getInfo();
echo $admin->updateTahunLulus(2017);
?>

```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_02.php'**
4. Amati hasilnya

Latihan 3

1. Buat file baru dengan nama latihan_02_03.php
2. Tambahkan kode program berikut

```

<?php
class Alumni {
    public $nama;
    public $jurusan;

    public function __construct($nama, $jurusan) {
        $this->nama = $nama;
        $this->jurusan = $jurusan;
    }

    public function tampilkanProfil() {
        return "Nama: $this->nama <br> Jurusan: $this->jurusan";
    }
}

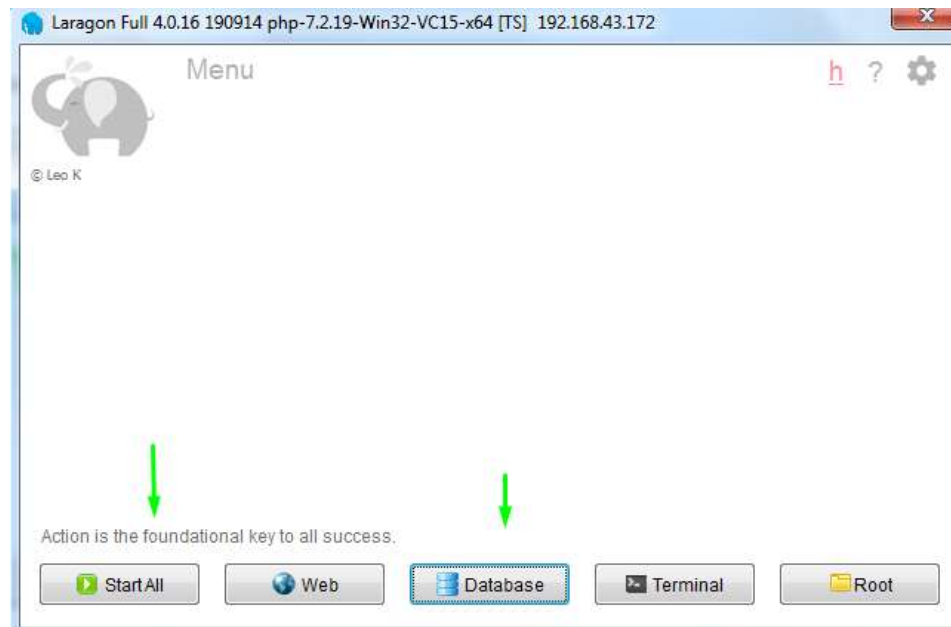
$alumni = new Alumni("Asgevara Maqarav Zarba", "Teknik Informatika");
echo $alumni->tampilkanProfil();
?>

```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_03.php'**
4. Amati hasilnya

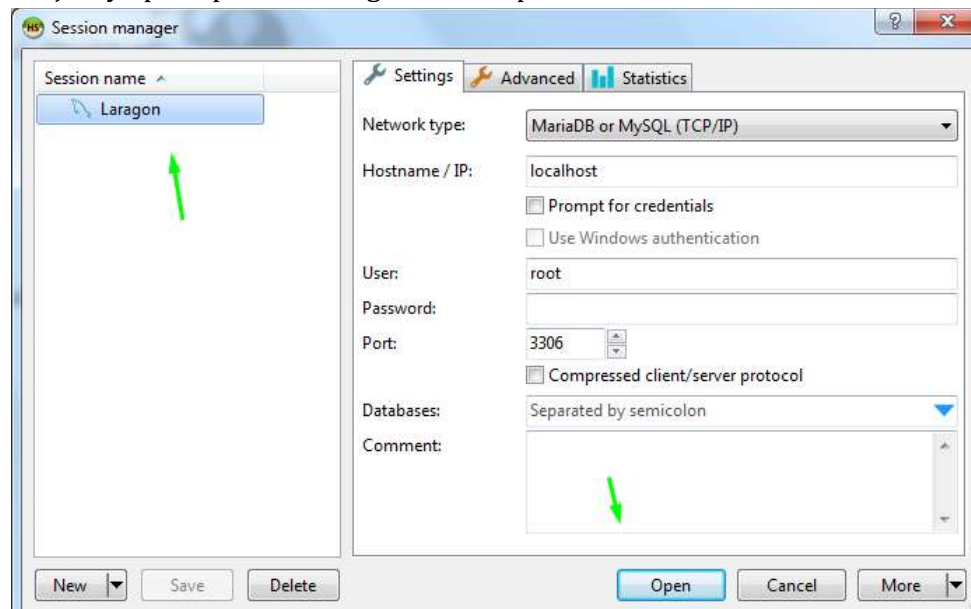
Latihan 4

1. Buka aplikasi laragon yang sudah kamu instal > Klik “ Start All” jika muncul peringatan klik Allow Access > Selanjutnya klik “ Database”



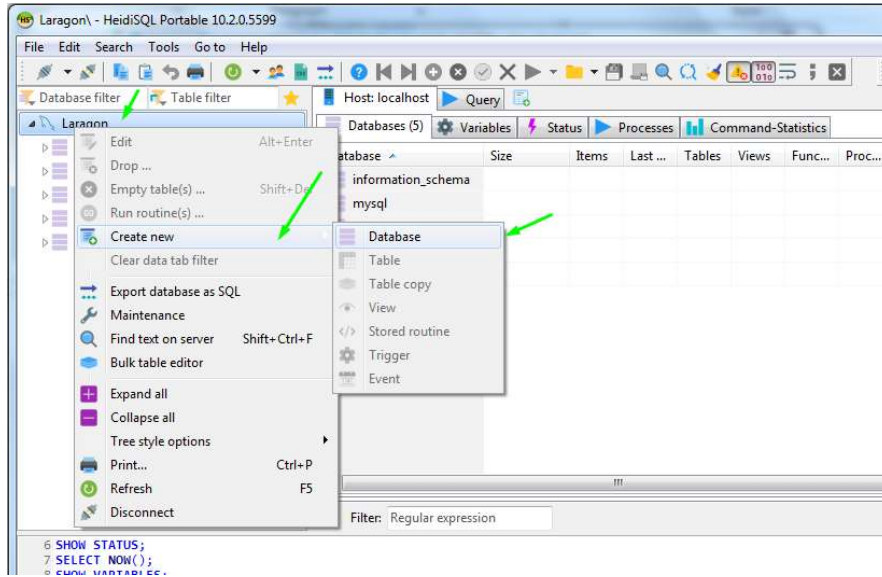
Gambar 2. Laragon Control Panel

2. Selanjutnya pada pilihan Laragon “ Klik Open”.



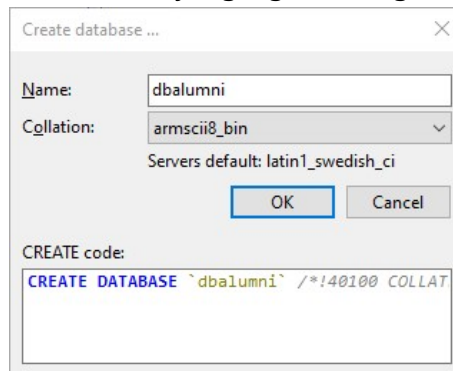
Gambar 3. Laragon Session Manager

3. Setelah klik open > Klik kanan pada "laragon" > Create new > Database.



Gambar 4. Laragon Session Manager

4. Masukan atau buat nama database yang ingin kamu gunakan > klik "OK".



Gambar 5. Laragon Session Manager

5. Klik tab "SQL", lalu masukkan kode berikut:

```
CREATE TABLE alumni (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nama VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL,  
  jurusan VARCHAR(50) NOT NULL,  
  tahun_lulus YEAR NOT NULL,  
  password VARCHAR(255) NOT NULL  
);
```

6. Klik "Run" untuk menjalankan perintah.
7. Masukkan perintah berikut untuk menambahkan data kedalam table alumni:

```
INSERT INTO alumni (nama, email, jurusan, tahun_lulus, password)  
VALUES  
(  
'Dede Irawan', 'dede@example.com', 'Teknik Informatika', 2025, 'hashed_password'),  
(  
'Milchatun', 'mila@example.com', 'Sistem Informasi', 2025, 'hashed_password');
```

8. Klik "Run" untuk menjalankan perintah.

Latihan 5

1. Buat file baru dengan nama latihan_02_04.php
2. Tambahkan kode program berikut

```
<?php
class Database {
    private $host = "localhost";
    private $db_name = "dbalumni";
    private $username = "root";
    private $password = "";
    public $conn;

    public function getConnection() {
        $this->conn = null;
        try {
            $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->db_name,
            $this->username, $this->password);
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        } catch (PDOException $exception) {
            echo "Koneksi gagal: " . $exception->getMessage();
        }
        return $this->conn;
    }
}
?>
```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_04.php'**
4. Amati hasilnya, jika ada yang perlu diperbaiki maka perbaiki.

Latihan 6

1. Buat file baru dengan nama latihan_02_05.php
2. Tambahkan kode program berikut

```
<?php
require_once "latihan_02_04.php";
class Alumni {
    private $conn;
    private $table_name = "alumni";

    public $id;
    public $nama;
    public $email;
    public $jurusan;
    public $tahun_lulus;

    public function __construct() {
        $database = new Database();
        $db = $database->getConnection();
    }
}
```

```

        $this->conn = $db;
    }

    public function getAll() {
        $query = "SELECT * FROM " . $this->table_name . " ORDER BY id DESC";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        return $stmt;
    }

    public function getById() {
        $query = "SELECT * FROM " . $this->table_name . " WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":id", $this->id);
        $stmt->execute();
        return $stmt;
    }

    public function add() {
        $query = "INSERT INTO " . $this->table_name . " (nama, email, jurusan, tahun_lulus)
VALUES (:nama, :email, :jurusan, :tahun_lulus)";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":nama", $this->nama);
        $stmt->bindParam(":email", $this->email);
        $stmt->bindParam(":jurusan", $this->jurusan);
        $stmt->bindParam(":tahun_lulus", $this->tahun_lulus);
        return $stmt->execute();
    }

    public function update() {
        $query = "UPDATE " . $this->table_name . " SET nama = :nama, email = :email, jurusan =
:jurusan, tahun_lulus = :tahun_lulus WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":id", $this->id);
        $stmt->bindParam(":nama", $this->nama);
        $stmt->bindParam(":email", $this->email);
        $stmt->bindParam(":jurusan", $this->jurusan);
        $stmt->bindParam(":tahun_lulus", $this->tahun_lulus);
        return $stmt->execute();
    }

    public function delete() {
        $query = "DELETE FROM " . $this->table_name . " WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":id", $this->id);
        return $stmt->execute();
    }
}
?>

```

Latihan 7

1. Buat file baru dengan nama latihan_02_06.php
2. Tambahkan kode program berikut

```
<?php
require_once "latihan_02_05.php";

$alumni = new Alumni();
$stmt = $alumni->getAll();
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Daftar Alumni</title>
</head>
<body>
  <h1>Daftar Alumni</h1>
  <a href="latihan_02_07.php">Tambah Alumni</a>
  <table border="1">
    <tr>
      <th>Nama</th>
      <th>Email</th>
      <th>Jurusan</th>
      <th>Tahun Lulus</th>
      <th>Aksi</th>
    </tr>
    <?php while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) { ?>
      <tr>
        <td><?php echo $row["nama"]; ?></td>
        <td><?php echo $row["email"]; ?></td>
        <td><?php echo $row["jurusan"]; ?></td>
        <td><?php echo $row["tahun_lulus"]; ?></td>
        <td>
          <a href="latihan_02_08.php?id=<?php echo $row['id']; ?>">Edit</a> |
          <a href="latihan_02_09.php?id=<?php echo $row['id']; ?>">Hapus</a>
        </td>
      </tr>
    <?php } ?>
  </table>
</body>
</html>
```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_06.php'**
4. Amati hasilnya, jika ada yang perlu diperbaiki maka perbaiki.

Latihan 8

1. Buat file baru dengan nama latihan_02_07.php
2. Tambahkan kode program berikut

```
<?php
require_once "latihan_02_05.php";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $alumni = new Alumni();
    $alumni->nama = $_POST["nama"];
    $alumni->email = $_POST["email"];
    $alumni->jurusan = $_POST["jurusan"];
    $alumni->tahun_lulus = $_POST["tahun_lulus"];

    if ($alumni->add()) {
        header("Location: latihan_02_06.php");
    }
}
?>

<!DOCTYPE html>
<html lang="id">
<head>
    <title>Tambah Alumni</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2>Tambah Alumni</h2>
    <form method="POST">
        <input type="text" name="nama" class="form-control mb-2" placeholder="Nama"
required>
        <input type="email" name="email" class="form-control mb-2" placeholder="Email"
required>
        <input type="text" name="jurusan" class="form-control mb-2" placeholder="Jurusan"
required>
        <input type="number" name="tahun_lulus" class="form-control mb-2"
placeholder="Tahun Lulus" required>
        <button type="submit" class="btn btn-success">Simpan</button>
    </form>
</div>
</body>
</html>
```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_07.php'**.
4. Amati hasilnya, jika ada yang perlu diperbaiki maka perbaiki.

Latihan 9

1. Buat file baru dengan nama latihan_02_08.php
2. Tambahkan kode program berikut

```
<?php
require_once "latihan_02_05.php";
if (isset($_GET["id"])) {
    $alumni = new Alumni();
    $alumni->id=$_GET["id"];
    $stmt = $alumni->getById();
    $data = $stmt->fetch(PDO::FETCH_ASSOC);
    if (!$data) {
        echo "Data tidak ditemukan!";
        exit;
    }
} else {
    echo "ID tidak diberikan!";
    exit;
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $alumni->id = $_POST["id"];
    $alumni->nama = $_POST["nama"];
    $alumni->email = $_POST["email"];
    $alumni->jurusan = $_POST["jurusan"];
    $alumni->tahun_lulus = $_POST["tahun_lulus"];
    if ($alumni->update()) {
        header("Location: latihan_02_06.php");
        exit;
    } else {
        echo "Gagal memperbarui data.";
    }
}
?>
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Alumni</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2>Edit Alumni</h2>
    <form method="POST">
        <input type="hidden" name="id" value="<?php echo $data['id']; ?>">
        <div class="mb-3">
            <label>Nama</label>
            <input type="text" name="nama" class="form-control" value="<?php echo
$data['nama']; ?>" required>
```

```

</div>
<div class="mb-3">
    <label>Email</label>
    <input type="email" name="email" class="form-control" value="<?php echo
$data['email']; ?>" required>
</div>
<div class="mb-3">
    <label>Jurusan</label>
    <input type="text" name="jurusan" class="form-control" value="<?php echo
$data['jurusan']; ?>" required>
</div>
<div class="mb-3">
    <label>Tahun Lulus</label>
    <input type="number" name="tahun_lulus" class="form-control" value="<?php echo
$data['tahun_lulus']; ?>" required>
</div>
<button type="submit" class="btn btn-success">Update</button>
<a href="latihan_02_06.php" class="btn btn-secondary">Batal</a>
</form>
</div>
</body>
</html>

```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_08.php'**.
4. Amati hasilnya, jika ada yang perlu diperbaiki maka perbaiki.

Latihan 10

1. Buat file baru dengan nama latihan_02_09.php
2. Tambahkan kode program berikut

```

<?php
require_once "latihan_02_05.php";
$alumni = new Alumni();
if (isset($_GET["id"])) {
    $alumni->id = $_GET["id"];
    if ($alumni->delete()) {
        header("Location: latihan_02_06.php");
        exit;
    } else {
        echo "Gagal menghapus data.";
    }
} else {
    echo "ID tidak diberikan!";
}
?>

```

3. Untuk menjalankan kode program diatas maka lakukan dengan cara pada terminal vs code ketik: **start 'http://localhost/web 2/latihan_02_09.php'**.
4. Amati hasilnya, jika ada yang perlu diperbaiki maka perbaiki.

F. POST-TEST

1. Mengapa kita tidak boleh langsung mengakses properti private dari luar class?
2. Bagaimana cara mengubah nilai properti private dalam class PHP?
3. Apa langkah-langkah utama dalam membuat CRUD menggunakan PHP OOP?
4. Jelaskan cara kerja PDO dalam menghubungkan PHP dengan database.
5. Mengapa penting menggunakan prepared statement dalam operasi database?

G. LATIHAN/TUGAS

1. Tambahkan table pengumuman pada database alumni.
2. Dengan PHP OOP buat program untuk mengelola pengumuman pada web alumni.