



Laporan Praktikum

Mata pelajaran Konsentrasi Keahlian RPL

Materi Ajar :

PHP: OOP PHP CRUD

No.	Nama	NIS	Kelas
1.	Najwan Uqosah Fawwaz	20371	XI RPL

Pengembangan Perangkat Lunak dan Gim (PPLG)

SMK Negeri 1 Bawang

2026

JOBSHEET PRAKTIKUM OOP MVC

Mata Pelajaran: Pemrograman Web dan Perangkat Bergerak (PWPB)

Materi Pokok: Implementasi PHP OOP dengan Arsitektur MVC & MySQLi

Kelas / Semester: XI atau XII / Ganjil

Alokasi Waktu: 2 x 45 Menit (1 Kali Pertemuan)

I. KOMPETENSI DASAR & INDIKATOR KETERCAPAIAN

- **Kompetensi Dasar:** Menerapkan aplikasi web stateful dengan arsitektur Model-View-Controller (MVC).
- **Indikator:**
 1. Siswa mampu membuat struktur folder standar arsitektur MVC.
 2. Siswa mampu menghubungkan database MySQL menggunakan MySQLi Driver dengan pendekatan OOP.
 3. Siswa mampu memisahkan logika data (Model), antarmuka (View), dan pengendali (Controller).

II. ALAT DAN BAHAN

1. Komputer/Laptop dengan Sistem Operasi Windows/Linux/macOS.
 2. Web Server Lokal: **XAMPP**, **FlyEnv** (Apache & MySQL aktif).
 3. Text Editor: **VS Code**, Sublime Text, atau sejenisnya.
 4. Web Browser: Google Chrome atau Mozilla Firefox.
-

III. LANGKAH KERJA & KODE PROGRAM

Langkah 1: Persiapan Database

1. Buka browser, akses <http://localhost/phpmyadmin/>.
2. Buat database baru dengan nama [db_perpustakaan](#).

3. Masuk ke menu **SQL**, salin dan eksekusi perintah berikut untuk membuat tabel dan data uji coba:

sql

```
CREATE TABLE tb_buku (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    judul VARCHAR(100) NOT NULL,  
    penulis VARCHAR(50) NOT NULL,  
    penerbit VARCHAR(50) NOT NULL  
);  
  
INSERT INTO tb_buku (judul, penulis, penerbit) VALUES  
('Pemrograman PHP OOP', 'Eko Kurniawan', 'Informatika'),  
('Lahirnya Arsitektur MVC', 'Sandhika Galih', 'Web Programming UNPAS'),  
('Dasar Basis Data untuk SMK', 'Rian Adi', 'Erlangga');
```

Gunakan kode dengan hati-hati.

Langkah 2: Membuat Struktur Folder Proyek

Buatlah folder baru di dalam direktori `C:/xampp/htdocs/` dengan nama `perpus-mvc`. Susun struktur folder di dalamnya tepat seperti bagan di bawah ini:

text

```
perpus-mvc/  
├── app/  
│   ├── controllers/  
│   ├── core/  
│   ├── models/  
│   └── views/  
│       └── buku/  
└── public/
```

Gunakan kode dengan hati-hati.

Langkah 3: Membuat Komponen Core (Inti Aplikasi)

Masuk ke folder `app/core/`, buat dua file utama sistem berikut:

File 1: `app/core/Database.php` (Menangani Koneksi DB OOP)**php**

```
<?php
class Database {
    private $host = "localhost";
    private $user = "root";
    private $pass = "";
    private $db_name = "db_perpustakaan";
    protected $conn;

    public function __construct() {
        $this->conn = new mysqli($this->host, $this->user, $this->pass,
        $this->db_name);
        if ($this->conn->connect_error) {
            die("Koneksi Database Gagal: " . $this->conn->connect_error);
        }
    }
}
```

Gunakan kode dengan hati-hati.

File 2: `app/core/Controller.php` (Base Controller / Induk)**php**

```
<?php
class Controller {
    public function view($view, $data = []) {
        if (file_exists('../app/views/' . $view . '.php')) {
            require_once '../app/views/' . $view . '.php';
        } else {
            die("View {$view} tidak ditemukan.");
        }
    }

    public function model($model) {
        if (file_exists('../app/models/' . $model . '.php')) {
            require_once '../app/models/' . $model . '.php';
            return new $model();
        } else {
            die("Model {$model} tidak ditemukan.");
        }
    }
}
```

Gunakan kode dengan hati-hati.

Langkah 4: Membuat Model, Controller, dan View

Buat file spesifik untuk mengelola data buku pada masing-masing folder yang sesuai:

File 3: `app/models/Buku_model.php` (Logika Data)

php

```
<?php
class Buku_model extends Database {
    public function getAllBuku() {
        $query = "SELECT * FROM tb_buku";
        $result = $this->conn->query($query);
        $rows = [];
        while ($row = $result->fetch_assoc()) {
            $rows[] = $row;
        }
        return $rows;
    }
}
```

Gunakan kode dengan hati-hati.

File 4: `app/controllers/Buku.php` (Pengendali / Jembatan)

php

```
<?php
class Buku extends Controller {
    public function index() {
        $data['judul'] = 'Daftar Buku Perpustakaan';
        $data['buku'] = $this->model('Buku_model')->getAllBuku();
        $this->view('buku/index', $data);
    }
}
```

Gunakan kode dengan hati-hati.

File 5: `app/views/buku/index.php` (User Interface / Tampilan)

html

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title><?= $data['judul']; ?></title>
    <link href="https://jsdelivr.net" rel="stylesheet">
</head>
<body class="bg-light">
<div class="container mt-5">
    <div class="card shadow-sm">
        <div class="card-header bg-primary text-white">
            <h3 class="mb-0"><?= $data['judul']; ?></h3>
        </div>
        <div class="card-body">
            <table class="table table-striped mt-3">
                <thead class="table-dark">
                    <tr>
                        <th>No</th><th>Judul
Buku</th><th>Penulis</th><th>Penerbit</th>
                    </tr>
                </thead>
                <tbody>
                    <?php $no = 1; foreach ($data['buku'] as $buku) : ?>
                        <tr>
                            <td><?= $no++; ?></td>
                            <td><?= htmlspecialchars($buku['judul']);
?></td>
                            <td><?= htmlspecialchars($buku['penulis']);
?></td>
                            <td><?= htmlspecialchars($buku['penerbit']);
?></td>
                        </tr>
                    <?php endforeach; ?>
                </tbody>
            </table>
        </div>
    </div>
</div>
</body>
</html>
```

Gunakan kode dengan hati-hati.

Langkah 5: Membuat Single Entry Point

Buat file index di luar folder `app` sebagai gerbang utama jalannya aplikasi.

File 6: `public/index.php`

php

```
<?php
require_once '../app/core/Database.php';
require_once '../app/core/Controller.php';
require_once '../app/controllers/Buku.php';

$app = new Buku();
$app->index();
```

Gunakan kode dengan hati-hati.

Langkah 6: Pengujian Aplikasi

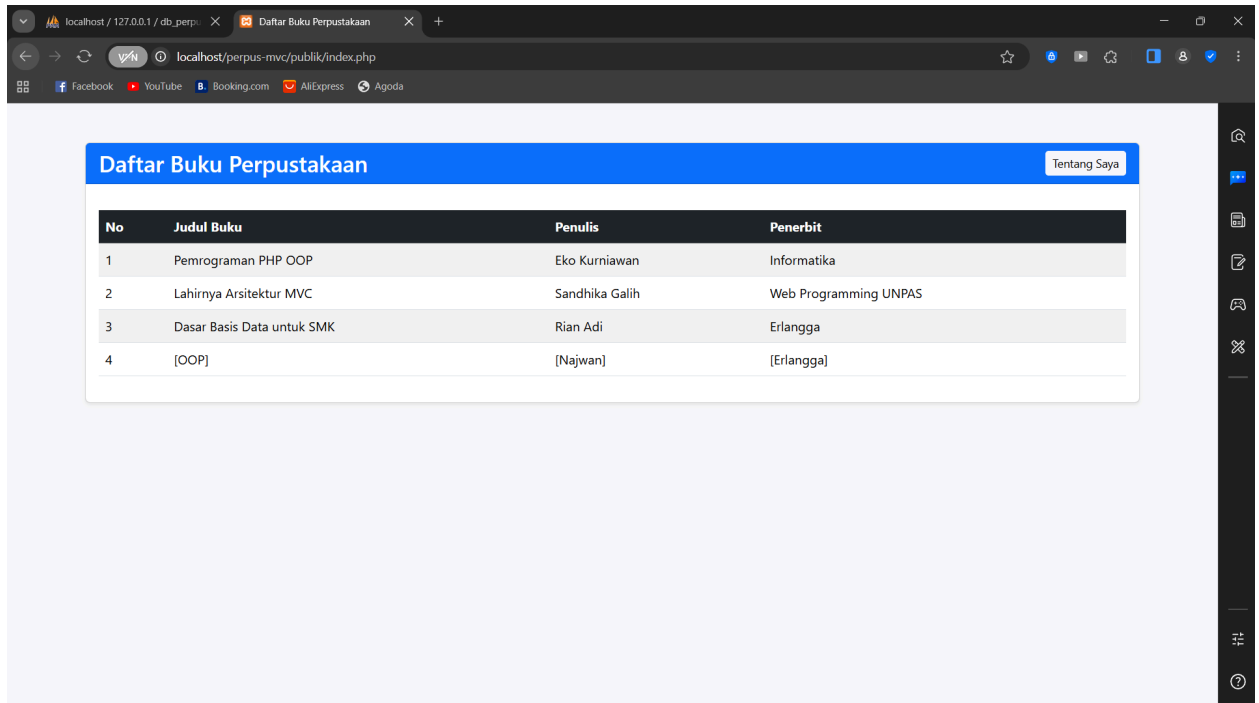
1. Buka Browser Anda.
2. Akses alamat URL berikut: `http://localhost/perpus-mvc/public/`
3. Amati hasilnya. Jika data buku dari database berhasil tampil dalam tabel Bootstrap, maka praktikum Anda berhasil.

IV. TUGAS MANDIRI (CHALLENGE SISWA)

Untuk memperdalam pemahaman mengenai alur data MVC, kembangkan aplikasi di atas dengan ketentuan:

1. Tambahkan data dummy baru langsung ke database sebanyak 2 baris data. Lihat apakah data otomatis bertambah di halaman web tanpa mengubah kode PHP.
2. Buatlah halaman view baru bernama `about.php` di dalam folder `app/views/buku/`. Modifikasi controller `Buku.php` dengan menambah *method* baru bernama `public function about()` untuk menampilkan halaman biodata

diri Anda (Nama dan Kelas).



V. LEMBAR PENILAIAN PRAKTIKUM

No	Komponen Penilaian	Skor Maksimal	Skor Capaian	Kesimpulan/Catatan
1	Persiapan & Struktur Folder (Kerapian susunan folder MVC)	20		
2	Koneksi Database (Koneksi OOP MySQLi berhasil tanpa error)	25		
3	Alur Coding MVC (Ketepatan penempatan kode Model, View, Controller)	35		
4	Hasil Akhir & UI (Data tampil rapi menggunakan framework Bootstrap)	20		

Total	100
Skor	

Kriteria Kelulusan: Minimal skor Capaian adalah **75**.