



# Laporan Praktikum

*Mata pelajaran Konsentrasi Keahlian RPL*

Materi Ajar :

**PHP: OOP PHP**

| No. | Nama         | NIS   | Kelas  |
|-----|--------------|-------|--------|
| 1.  | Gilar Faizal | 20363 | XI RPL |

Pengembangan Perangkat Lunak dan Gim (PPLG)

SMK Negeri 1 Bawang

2026

---

# LEMBAR KERJA SISWA (JOB SHEET) PRAKTIKUM

**Mata Pelajaran:** Pemrograman Web dan Perangkat Bergerak (RPL)

**Materi:** Pemrograman Berorientasi Objek (OOP) PHP

**Alokasi Waktu:** 2 x 45 Menit

---

## 1. Tujuan Pembelajaran

Setelah menyelesaikan praktikum ini, siswa diharapkan mampu:

1. Membuat `class`, `property`, dan `method` di PHP dengan benar.
  2. Menerapkan konsep `inheritance` (pewarisan) untuk menghubungkan dua class.
  3. Menerapkan `encapsulation` (enkapsulasi) menggunakan `getter` dan `setter`.
  4. Melakukan instansiasi `object` dan menampilkan datanya pada browser.
- 

## 2. Alat dan Bahan

- Komputer/Laptop dengan OS Windows/Linux/macOS.
  - Aplikasi Web Server Lokal: **XAMPP / Laragon** (Pastikan Apache service *Running*).
  - Kode Editor: **VS Code / Sublime Text**.
  - Browser: **Google Chrome / Mozilla Firefox**.
- 

## 3. Studi Kasus & Skenario Tugas

Siswa diminta membangun sistem backend mini untuk **Manajemen Data Sekolah**.

- Terdapat class utama bernama `Manusia` (sebagai Parent Class).
  - Terdapat class turunan bernama `Siswa` (sebagai Child Class) yang mewarisi class `Manusia`.
  - Properti nilai ujian siswa harus dilindungi menggunakan enkapsulasi (`private`) agar tidak bisa diubah sembarangan tanpa validasi.
-

## 4. Langkah-Langkah Praktikum

### Langkah 1: Persiapan Folder Kerja

1. Buka folder `htdocs` di dalam direktori instalasi XAMPP Anda (biasanya di `C:\xampp\htdocs`).
2. Buat folder baru dengan nama `praktikum_oop_php`.
3. Buka folder tersebut menggunakan VS Code.
4. Buat file baru di dalamnya dengan nama `index.php`.

### Langkah 2: Menulis Struktur Code Parent Class (`Manusia`)

Ketiklah kode berikut di dalam `index.php` untuk membuat kerangka dasar manusia.

```
<?php
// 1. Membuat Parent Class
class Manusia {
    // Properti dengan visibility public dan protected
    public $nama;
    protected $gender;

    // Constructor untuk mengisi nama dan gender saat objek dibuat
    public function __construct($namaAwal, $genderAwal) {
        $this->nama = $namaAwal;
        $this->gender = $genderAwal;
    }

    // Method untuk mengenalkan diri
    public function kenalan() {
        return "Halo, nama saya {$this->nama}, berjenis kelamin
{$this->gender}.";
    }
}
```

### Langkah 3: Menerapkan Inheritance & Encapsulation pada Child Class (`Siswa`)

Tambahkan kode di bawah ini tepat di bawah kode Langkah 2. Kita akan membuat class `Siswa` yang mewarisi `Manusia`, serta mengunci properti `nilai`.

```
// 2. Membuat Child Class dengan Inheritance (extends)
class Siswa extends Manusia {
    public $nisn;
```

```

        private $nilai; // Enkapsulasi: Properti diproteksi agar tidak bisa
        diakses langsung

        // Method Setter untuk mengisi nilai dengan validasi keamanan
        public function setNilai($angka) {
            if ($angka >= 0 && $angka <= 100) {
                $this->nilai = $angka;
            } else {
                $this->nilai = 0; // Jika nilai tidak masuk akal, otomatis set
ke 0
            }
        }

        // Method Getter untuk mengambil data nilai
        public function getNilai() {
            return $this->nilai;
        }

        // Method untuk menentukan kelulusan siswa
        public function cekKelulusan() {
            if ($this->nilai >= 75) {
                return "LULUS";
            } else {
                return "REMEDIAL";
            }
        }
    }
}

```

## Langkah 4: Proses Instansiasi Objek dan Output

Tambahkan kode implementasi ini di bagian paling bawah file `index.php` Anda untuk melihat hasilnya.

```

// 3. Proses Instansiasi Objek dari Class Siswa
$siswaRPL = new Siswa("Ahmad Fauzi", "Laki-laki");
$siswaRPL->nisn = "0045671239";

// Mengisi nilai menggunakan Setter (Aman)
$siswaRPL->setNilai(85);

// 4. Menampilkan Output ke Browser
echo "<h2>--- DATA SISWA RPL ---</h2>";
echo $siswaRPL->kenalan() . "<br>"; // Memanggil method warisan dari class
Manusia
echo "NISN : " . $siswaRPL->nisn . "<br>";

```

```
echo "Nilai Ujian : " . $siswaRPL->getNilai() . "<br>"; // Memanggil lewat  
Getter  
echo "Status Kelulusan : <strong>" . $siswaRPL->cekKelulusan() .  
"</strong>";  
?>
```

## Langkah 5: Pengujian

1. Buka browser Anda.
2. Akses URL: <http://localhost/nis-oop1/index.php>.
3. Pastikan data muncul di layar dengan format yang benar.

---

## 5. Tantangan Mandiri (Tugas Siswa)

Untuk menguji pemahaman, minta siswa melakukan modifikasi mandiri pada kode di atas:

1. **Validasi Tambahan:** Ubah logika pada `setNilai()`. Jika nilai yang diinput di bawah 0 atau di atas 100, selain mengubah nilai menjadi 0, tampilkan pesan peringatan teks berwarna merah: *"Input nilai tidak valid!"*.
2. **Class Baru:** Buat satu Child Class lagi bernama `Guru` yang juga meng-extends `Manusia`. Berikan properti khusus `$nip` dan method `mengajar()` yang mengembalikan teks *"Guru [Nama] sedang mengajar pemrograman."*. Instansiasi objeknya di bagian bawah halaman.

## Tampilan Layar

The screenshot displays a web browser window on the right and a code editor on the left. The browser shows a warning message and two sections of student and teacher data. The code editor shows the PHP code that generates this output.

**Browser Window:**

- Address bar: localhost/20363-oop1/index.php
- Warning: [PERINGATAN] Input nilai tidak valid untuk Azzam!
- Section: --- DATA SISWA RPL ---
- Text: Halo, nama saya Gilar Faizal, berjenis kelamin Laki-laki.  
NISN : 20363  
Nilai Ujian : 85  
Status Kelulusan : LULUS
- Section: --- DATA GURU RPL ---
- Text: Halo, nama saya Pak Aril, berjenis kelamin Laki-laki.  
NIP : 12345  
Aktivitas : Guru **Pak Aril** sedang mengajar pemrograman.

**Code Editor (index.php):**

```
C: > xampp > htdocs > 20363-oop1 > index.php > ...
1 reference | 0 implementations
46 class Guru extends Manusia {
47     public $nip;
48
49     // Method khusus untuk guru
50     public function mengajar() {
51         return "Guru <strong>{$this->nama}</strong> sedang mengajar pemrograman.";
52     }
53 }
54
55 // 3. Proses Instansiasi Objek dari Class Siswa
56 $siswaRPL = new Siswa("Gilar Faizal", "Laki-laki");
57 $siswaRPL->nis = "20363";
58 $siswaRPL->setNilai(85); // Nilai valid
59
60 // Contoh menguji validasi error tantangan 1 (Bisa dihapus jika tidak diperlukan)
61 $siswaError = new Siswa("Azzam", "Laki-laki");
62 $siswaError->setNilai(111); // Memicu pesan error merah
63
64
65 // 4. Proses Instansiasi Objek dari Class Guru (Tantangan 2)
66 $guruRPL = new Guru("Pak Aril", "Laki-laki");
67 $guruRPL->nip = "12345";
68
69
70 // 5. Menampilkan Output ke Browser
71 echo "<h2>--- DATA SISWA RPL ---</h2>";
72 echo $siswaRPL->kenalan() . "<br>";
73 echo "NISN : " . $siswaRPL->nis . "<br>";
74 echo "Nilai Ujian : " . $siswaRPL->getNilai() . "<br>";
75 echo "Status Kelulusan : <strong>" . $siswaRPL->cekKelulusan() . "</strong><br>";
76
77 echo "<h2>--- DATA GURU RPL ---</h2>";
```