

# Rencana Pelaksanaan Pembelajaran (RPP)

Mata Pelajaran: Pemrograman Berorientasi Objek

Kelas: 12 PPLG

Fase: F

Semester: Ganjil

Topik: Inheritance (Pewarisan): Konsep pewarisan, kelas dasar (base class) dan kelas turunan (derived class), kata kunci Inherits dan MyBase.

Alokasi Waktu: 3 Jam Pelajaran (1 x pertemuan)

---

## A. Kompetensi Awal

Peserta didik telah memiliki pemahaman tentang konsep dasar PBO, dasar-dasar bahasa pemrograman Visual Basic .NET, serta telah mampu membuat dan menggunakan **kelas**, **objek**, dan menerapkan **enkapsulasi**.

## B. Profil Pelajar Pancasila

1. **Bernalar Kritis:** Peserta didik mampu menganalisis kebutuhan program dan menentukan kapan sebuah kelas harus mewarisi kelas lain untuk menghindari duplikasi kode.
2. **Mandiri:** Peserta didik mampu membuat kelas turunan dan mengimplementasikan pewarisan secara mandiri.
3. **Kreatif:** Peserta didik mampu merancang hierarki kelas yang efektif untuk memodelkan objek-objek di dunia nyata.

## C. Sarana & Prasarana

- Komputer atau laptop yang terinstal **Visual Studio 2013** atau versi yang relevan.
- Proyektor atau layar interaktif.
- Papan tulis atau *whiteboard*.
- Materi presentasi (PowerPoint, slide, atau video tutorial).

- Akses internet.

#### D. Target Peserta Didik

Peserta didik reguler, tidak ada kesulitan dalam mengikuti pembelajaran.

#### E. Model Pembelajaran

- **Pendekatan:** *Deep Learning* (Pembelajaran Mendalam).
- **Model:** *Discovery Learning* atau Pembelajaran Penemuan.

#### F. Tujuan Pembelajaran

Setelah mengikuti serangkaian kegiatan pembelajaran, peserta didik diharapkan mampu:

1. Menjelaskan konsep *inheritance* dan manfaatnya dalam PBO.
2. Mendefinisikan **kelas dasar** (*base class*) dan **kelas turunan** (*derived class*) menggunakan kata kunci *Inherits*.
3. Menggunakan kata kunci *MyBase* untuk memanggil *constructor* atau metode dari kelas dasar.
4. Mengakses atribut dan metode yang diwarisi oleh kelas turunan.

#### G. Pemahaman Bermakna

*Inheritance* adalah pilar PBO yang memungkinkan kita untuk membangun hierarki kelas, menciptakan kode yang efisien, dan menggunakan kembali kode yang sudah ada (*reuse code*). Konsep ini mengurangi duplikasi kode dan mempermudah pemeliharaan program.

#### H. Pertanyaan Pemantik

- Jika kalian punya kelas Karyawan dengan atribut nama dan alamat, dan kalian juga butuh kelas Manajer yang memiliki atribut yang sama ditambah atribut divisi, apakah kalian akan menyetik ulang atribut nama dan alamat di kelas Manajer?
  - Bagaimana cara kita menciptakan hubungan "adalah seorang..." (misalnya, Manajer adalah seorang Karyawan) dalam kode program?
  - Mengapa kita perlu memanggil *constructor* dari kelas induk ketika membuat objek dari kelas anak?
- 

## I. Kegiatan Pembelajaran

### Pendahuluan (15 Menit)

1. Guru memulai pembelajaran dengan salam dan doa, kemudian memeriksa kehadiran peserta didik.
2. Guru melakukan *ice-breaking* untuk mempersiapkan peserta didik.
3. Guru mengajukan pertanyaan pemantik untuk memicu pemikiran kritis tentang duplikasi kode dan pentingnya hierarki kelas.
4. Guru menjelaskan tujuan pembelajaran dan relevansi *inheritance* dalam pengembangan perangkat lunak.

### Kegiatan Inti (105 Menit)

1. **Stimulasi (15 menit):**
  - Guru memberikan studi kasus: "Kita ingin membuat dua kelas, Karyawan dan Manajer. Keduanya memiliki nama dan alamat. Kelas Manajer memiliki tambahan divisi."
  - Peserta didik diminta untuk memikirkan cara paling efisien untuk membuat kedua kelas tersebut.
2. **Identifikasi Masalah (15 menit):**
  - Guru memandu peserta didik untuk mengidentifikasi masalah duplikasi kode. Guru memperkenalkan konsep *inheritance* sebagai solusi.
  - Guru mengenalkan istilah **kelas dasar** (*base class*) dan **kelas turunan** (*derived class*).
3. **Pengumpulan Data (45 menit):**

- Guru menjelaskan sintaks untuk membuat kelas turunan menggunakan kata kunci *Inherits*. Guru memandu peserta didik untuk membuat kelas *Karyawan* sebagai kelas dasar dan kelas *Manajer* sebagai kelas turunan.
  - Guru menjelaskan bahwa kelas turunan secara otomatis mewarisi semua anggota *public* dan *protected* dari kelas dasar.
  - Guru memperkenalkan kata kunci *MyBase* yang digunakan untuk mengakses anggota kelas dasar dari kelas turunan. Guru memberikan contoh penggunaan *MyBase.New()* untuk memanggil *constructor* dari kelas dasar.
  - Peserta didik mencoba mengimplementasikan kode pewarisan ini di Visual Studio dan menguji cara mengakses atribut nama dan alamat dari objek *Manajer*.
4. **Pembuktian dan Verifikasi (15 menit):**
- Peserta didik secara individu mencoba membuat hierarki kelas lain, misalnya *Hewan* (kelas dasar) dan *Kucing* (kelas turunan).
  - Peserta didik diminta untuk membuat objek dari kedua kelas dan menguji kode untuk memastikan pewarisan bekerja dengan benar.
5. **Menarik Kesimpulan (15 menit):**
- Guru membimbing peserta didik untuk menyimpulkan manfaat *inheritance* dalam mengurangi duplikasi kode.
  - Peserta didik menyimpulkan bahwa *Inherits* digunakan untuk membuat hubungan pewarisan, dan *MyBase* untuk mengakses anggota kelas dasar.

## Penutup (15 Menit)

1. Peserta didik bersama guru menyimpulkan poin-poin penting yang telah dipelajari tentang *inheritance*.
  2. Guru memberikan penugasan individu untuk membuat hierarki kelas lain yang lebih kompleks.
  3. Guru menginformasikan materi untuk pertemuan berikutnya.
  4. Pembelajaran ditutup dengan doa dan salam.
- 

## J. Asesmen (Penilaian)

- **Asesmen Diagnostik (Sebelum Pembelajaran):**
  - Guru memberikan pertanyaan lisan di awal pertemuan untuk mengetahui pemahaman awal tentang duplikasi kode.
- **Asesmen Formatif (Selama Pembelajaran):**
  - **Penilaian Kinerja:** Mengamati kemampuan peserta didik dalam menulis kode pewarisan dan penggunaan *Inherits* serta *MyBase*.

- **Penilaian Diskusi:** Mengamati keaktifan peserta didik dalam merancang hierarki kelas.
  - **Asesmen Sumatif (Setelah Pembelajaran):**
    - **Penugasan Individu:** Penilaian terhadap kode yang dibuat, mencakup:
      - Kebenaran sintaks dalam mendefinisikan kelas dasar dan turunan.
      - Penerapan Inherits yang benar.
      - Penggunaan MyBase untuk memanggil *constructor* kelas dasar.
- 

## K. Pengayaan & Remedial

- **Pengayaan:** Peserta didik yang telah menguasai materi dapat diberikan tantangan tambahan, seperti mempelajari tentang *multiple inheritance* (meskipun tidak didukung secara langsung di VB.NET, konsepnya bisa dibahas), atau konsep overriding metode.
  - **Remedial:** Peserta didik yang belum mencapai tujuan pembelajaran akan diberikan bimbingan khusus oleh guru dan sumber belajar tambahan (video tutorial atau panduan langkah demi langkah) untuk membantu mereka memahami kembali konsep kelas dasar dan kelas turunan.
- 

## L. Refleksi

- **Refleksi Guru:**
  - Apakah peserta didik mampu membedakan dengan jelas antara kelas dasar dan kelas turunan?
  - Apakah contoh kasus yang diberikan cukup relevan dan mudah dipahami?
  - Bagian mana dari materi (Inherits atau MyBase) yang paling sulit dipahami?
  - Apa yang perlu saya perbaiki untuk sesi pembelajaran berikutnya?
- **Refleksi Peserta Didik:**
  - Apa hal baru yang saya pelajari tentang PBO hari ini?
  - Bagaimana saya bisa menjelaskan konsep *inheritance* kepada orang lain?
  - Apakah saya merasa lebih mudah untuk membuat program yang terstruktur setelah belajar tentang *inheritance*?
  - Apa yang akan saya lakukan untuk melatih kemampuan saya dalam membuat hierarki kelas?