1. **MATERI PEMBELAJARAN DI KELAS**
   1. **Mengenal konsep Enkapsulasi (Class, Property, Method)**

Enkapsulasi adalah suatu cara untuk meyembunyikan implementasi detil dari class untuk mencegah akses yang ilegal. Singkatnya, Enkapsulasi digunakan untuk limit access antar class dalam suatu aplikasi.

Ada 3 jenis enkapsulasi, yaitu :

1. **Public**

Merupakan sintaks dalam java yang memungkinkan setiap variabel juga method yang di deklarasikan dalam class public dapat di akses oleh semua class, subclass dan non subclass di package yang sama maupun berbeda.

1. **Private**

Merupakan kata kunci yang di deklarasikan untuk memproteksi suatu class agar setiap variabel dan method yang dideklarasikan tidak dapat di akses oleh non subclass di dalam package yang sama dengan sebelumnya.

1. **Protected**

Merupakan kata kunci yang di deklarasikan dalam suatu class agar setiap variabel dan method yang berada dalam class private ini hanya dapat diakses oleh class yang sama. Dalam bahasa pemgrograman java, class yang bersifat private memproteksi agar semua variabel dan methodnya tidak dapat di akses oleh semua subclass dan nonsubclass di package yang sama, dan subclass dan nonsubclass di package yang berbeda dengan sebelumnya.

* 1. **Mengenal konsep OOP pada Java**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi.

* 1. **Membuat aplikasi Desktop berbasis OOP seperti :**
* Aplikasi Warteg (Aplikasi Manajemen Warung Makan)
* Aplikasi Hitung Berat Badan Ideal
* Aplikasi Konversi Kecepatan
  1. **Mempelajari konsep *Inheritance* dan *Polymorphism***
* **Inheritance (Pewarisan)**

Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.

* **Polymorphism (Polimorfisme)**

Polymorphism merupakan kemampuan objek-objek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.

* 1. **Mempelajari konsep pemrograman berbasis graphics dan mengenal beberapa library Java yang berhubungan dengan materi tersebut seperti :**
* Line 2 Dimensi
* Draw & Fill
* Shapes (Rectangle, Oval, Rounded Rectangle, Polygon)
  1. **Mempelajari konsep pemrograman Multimedia seperti :**
* Animasi Sederhana
* Media Player
* Audio Recorder
  1. **Mengenal konsep MVC (Model, View, Controller) pada Java**
* Mengenal fungsi setiap komponen MVC (Model, Events, dll)
* Menerapkan konsep MVC pada aplikasi data siswa
* Menerapkan konsep MVC pada aplikasi CRUD
* Membuat aplikasi CRUD berbasis MVC

1. **MATERI PEMBELAJARAN PENTING YANG TIDAK DIPELAJARI KELAS**
2. **Penerapan Pustaka (Library) pada bahasa pemrograman Java**

**Pustaka** (atau ***Library*** dalam *Bahasa Inggris*), dalam ilmu pemrograman adalah kumpulan sintaks-sintaks [program](https://id.wikipedia.org/wiki/Program_komputer" \o "Program komputer) yang digunakan untuk membangun dan mengembangkan [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak" \o "Perangkat lunak). Seperti :

- Apache Xerces

- IText PDF

- MySQL JDBC Driver

1. **Penerapan UI yang baik dan benar**

**User Interface** merupakan komponen yang paling dekat dengan User, karena interaktifitas suatu aplikasi adalah yang paling pertama diakses oleh user. Oleh karenanya, user interface yang baik sangat diperlukan guna menghasilkan user experience yang baik pula.

1. **MATERI PEMBELAJARAN TERSULIT**
2. **MVC**

Karena kurangnya pemahaman mengenai fungsi dari Overriding dan EventListener, maka agak sulit untuk memahami alur kerja dari aplikasi berbasis MVC.

1. **Graphics (Gambar Koordinat)**

Sulitnya menentukan koordinat untuk membuat sebuah objek dengan bentuk abstrak seperti polygon.

1. **Polymorphism**

Kurangnya pemahaman mengenai sifat inheritance dari Parent Class untuk Sub class di dalamnya membuat sulitnya pengambilan keputusan untuk menentukan method yang akan digunakan.

1. **Multimedia**

Kurangnya wawasan mengenai penerapan penggunaan Library pada bahasa pemrograman Java khususnya dalam materi Multimedia terlebih lagi pada modul merekam suara membuat sulitnya memahami alur kerja dari aplikasi multimedia tersebut.