**Job Interview 1**

**Outcome**

* **Mampu menjelaskan dan mengimplementasikan dasar OOP**
* Mempu menganalisis permasalahan dunia nyata dan menerjemahkannya dalam solusi berbasis OOP

**Deskripsi**

1. Pilih produk aplikasi / game digital yang Kamu suka (IG / Tiktok / BCA Mobile / Gojek / Grab / Mobile Legend / Photoshop / Office / dsb.)
2. Buat list use case prioritas dari produk digital tersebut (tabel MD)
3. Desain dan implementasikan dalam bentuk program sederhana berbasis Object Oriented Programming dengan ketentuan
   * Buat class diagram sederhananya
   * Mengimplementasikan 4 pilar OOP dengan tepat
     1. Encapsulation
     2. Abstraction
     3. Inheritance
     4. Polymorphism
        + Method overriding
        + Method overloading
   * Mengimplementasikan keseluruhan fitur OOP
   * Diberi komentar untuk tiap class / method [tips](https://stackoverflow.blog/2021/12/23/best-practices-for-writing-code-comments/)
4. Beri interface minimal dalam bentuk Command Line interface
5. Buat video presentasi dari produk yang dibuat (audience publik)
6. Mampu menjelaskan OOP dari pertemuan 1 - 7

**Aspek Penilaian**

* Mampu mendemonstrasikan penyelesaian masalah dengan pendekatan matematika dan algoritma pemrograman secara tepat
* Mampu menjelaskan algoritma dari solusi yang dibuat
* Mampu menjelaskan konsep dasar OOP
* Mampu mendemonstrasikan penggunaan Encapsulation secara tepat
* Mampu mendemonstrasikan penggunaan Abstraction secara tepat
* Mampu mendemonstrasikan penggunaan Inheritance dan Polymorphism secara tepat
* Mampu menerjemahkan proses bisnis ke dalam skema OOP secara tepat
* Mampu menjelaskan rancangan dalam bentuk Class Diagram, dan Use Case table
* Mampu memberikan gambaran umum aplikasi kepada publik menggunakan presentasi berbasis video
* Inovasi UX

**Contoh**

/\*

Instagram

1. Bisa posting Konten

2. Bisa respon Konten: like, comment, bookmark, share

3. Bisa kirim pesan personal

4. Bisa follow

InstagramApp

- contents: Content[]

- id

- media: Media[]

- responses: Response[]

- users: User[]

- id

- username

- displayName

- photoURL

- followings: User[]

- followers: User[]

- contents: Content[]

- post(Content content)

- respond(Response response)

\*/

import java.util.ArrayList;

class User {

private String username;

public User(String username) {

this.username = username;

}

public String getUsername() {

return this.username;

}

}

class InstagramApp {

private String version = "0.0.1";

public void showVersion() {

System.out.println("Version " + this.version);

}

public void createUser(User newUser) {

this.users.add(newUser);

System.out.println("User " + newUser.getUsername() + " berhasil ditambahkan, jumlah user Instagram saat ini ada " + this.users.size());

}

private ArrayList<User> users = new ArrayList<User>();

}

public class Main

{

public static void main(String[] args) {

InstagramApp app = new InstagramApp();

app.showVersion();

User user1 = new User("syamil\_mr");

app.createUser(user1);

User user2 = new User("the\_dian");

app.createUser(user2);

}

}

# Job Interview 1 fix

## Outcome

* **Mampu menjelaskan dan mengimplementasikan dasar OOP**
* Mempu menganalisis permasalahan dunia nyata dan menerjemahkannya dalam solusi berbasis OOP

## Deskripsi

1. Pilih produk aplikasi / game digital yang Kamu suka (IG / Tiktok / BCA Mobile / Gojek / Grab / Mobile Legend / Photoshop / Office / dsb.)
2. Buat list use case prioritas dari produk digital tersebut
3. Desain dan implementasikan dalam bentuk program sederhana berbasis Object Oriented Programming dengan ketentuan
   * Buat class diagram sederhananya
   * Mengimplementasikan 4 pilar OOP dengan **tepat**
     1. Encapsulation
     2. Abstraction
        + The why
        + Abstract class
        + Interface
     3. Inheritance
     4. Polymorphism
        + Overriding
        + Overloading
   * Mengimplementasikan keseluruhan fitur OOP
     1. Constructor
     2. Final
     3. Static
     4. Association, Aggregation, Composition
     5. Modifier
4. Beri interface minimal dalam bentuk Command Line interface
5. Buat video presentasi dari produk yang dibuat (audience publik)
6. Wawancara:
   * Mampu menjelaskan OOP secara komprehensif menggunakan project yang dibuat

## Registrasi Repo Job Interview

[Registrasi di sini: nama, nim, kelas, alamat repo interview](https://gitlab.com/marchgis/march-ed/2023/courses/if214005-praktikum-pemrograman-berorientasi-objek/-/issues/1)

## Contoh Pengerjaan

### Pilih Aplikasi

Instagram

### Use Case

| **User** | **Bisa** | **Nilai prioritas** |
| --- | --- | --- |
| user | posting post | 100 |
| user | komentar post user lain | 90 |
| user | kirim message | 80 |
| user | like post | 70 |
| public | registrasi | 80 |
| user | login | 70 |
| manajemen ig | melihat statistik penggunaan | 90 |
| manajemen ig | menindak lanjut konten buruk | 70 |
| dsb. |  |  |

### Class Diagram

* [Contoh](https://glints.com/id/lowongan/class-diagram-adalah/)

### Coding

/\*

Instagram

1. Bisa posting Konten

2. Bisa respon Konten: like, comment, bookmark, share

3. Bisa kirim pesan personal

4. Bisa follow

InstagramApp

- contents: Content[]

- id

- media: Media[]

- responses: Response[]

- users: User[]

- id

- username

- displayName

- photoURL

- followings: User[]

- followers: User[]

- contents: Content[]

- post(Content content)

- respond(Response response)

\*/

import java.util.ArrayList;

class User {

private String username;

public User(String username) {

this.username = username;

}

public String getUsername() {

return this.username;

}

}

class InstagramApp {

private String version = "0.0.1";

public void showVersion() {

System.out.println("Version " + this.version);

}

public void createUser(User newUser) {

this.users.add(newUser);

System.out.println("User " + newUser.getUsername() + " berhasil ditambahkan, jumlah user Instagram saat ini ada " + this.users.size());

}

private ArrayList<User> users = new ArrayList<User>();

}

public class Main

{

public static void main(String[] args) {

InstagramApp app = new InstagramApp();

app.showVersion();

User user1 = new User("syamil\_mr");

app.createUser(user1);

User user2 = new User("the\_dian");

app.createUser(user2);

}

}

**IF214004 - Pemrograman Berorientasi Objek**

**Learning Outcome Outline**

* Mampu menjelaskan pemanfaatan OOP dalam industri
* Mampu menjelaskan dan mendemonstrasikan konsep dasar OOP **- Class, Object, Attribute and Method**
  + Basic Object Oriented Programming (OOP)
    - Class
    - Object
    - Attribute
    - Method
  + Attribute
    - Data structure
  + Method
    - Parameter
    - Return
    - Overloading
    - Overriding
    - Control
    - Loop
  + Inheritance
  + Encapsulation
  + Abstraction
  + Polymorphism
  + Interface Implementation
  + Relation
    - Association
    - Aggregation
    - Composition
* Mampu membaca dokumentasi dan mendemonstrasikan penggunaan library bahasa pemrograman berbasis paradigma OOP
  + Big Data : Apache Spark Java
  + Game : Godot C#
  + Machine Learning : scikit-learn Python
  + Frontend Development : Browser API Javascript
  + Mobile Development : Flutter Dart, Android SDK Java
  + Backend Development : Fiber Go, Laravel PHP

**Pertemuan**

| **No** | **Learning Outcome** | **Materi & Asesmen** |
| --- | --- | --- |
| 1 | Pengantar OOP | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/tree/main/pertemuan-1) |
| 2 | Dasar OOP (Class, Object, Attribute and Method) | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/tree/main/pertemuan-2) |
| 3 | Class (Attribute, Modifier, Constructor, Static) | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/blob/main/pertemuan-3) |
| 4 | Method (Algoritma dan Struktur Data) | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/tree/main/pertemuan-4) |
| 5 | 4 Pilar OOP Abstraction (Abstraction Class, Interface) dan Encapsulation | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/tree/main/pertemuan-5) |
| 6 | 4 Pilar OOP Inheritance dan Polymorphism (Overloading, Overriding) | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/blob/main/pertemuan-6) |
| 7 | Relation (Association, Aggregation, Composition) dan Packaging | [Materi](/marchgis/march-ed/2023/courses/if214004-pemrograman-berorientasi-objek/-/tree/main/pertemuan-7) |