

Nama & NPM	Topik:	Tanggal:
Zahra Sari Fhadilah G1F024025 Iqbal Tri Agustian G1F024043 Rivan Alfatoni G1F024047	Class , Object and Method	17 September 2024

### [No. 1] Identifikasi Masalah

1. Apabila diketahui kelas induk adalah Mahasiswa dan Kelas anak adalah turunan dari mahasiswa maka:
  - (a) Analisa atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
  - (b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
  - (c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!
  - (d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

### [No. 1] Analisis dan Argumentasi

1. Analisa atribut, system, dan constructor dari kelas induk Mahasiswa dan kelas anak( turunan dari Mahasiswa)!

1. Kelas Induk Mahasiswa
  - a. Atribut
    - String ruangan Menyimpan informasi ruangan.
    - String belajar Menyimpan informasi pelajaran.
  - b. Method
    - void sedangBelajar( String belajar) Mencetak aktivitas belajar mahasiswa.
    - int jamBelajar( int sks) Menghitung waktu belajar berdasarkan SKS( 50 menit per SKS).
  - c. Constructor
    - Menggunakan constructor dereliction.
2. Kelas Turunan MahasiswaSI
  - a. Method
    - void sedangBelajar( String belajar, String ruangan) Overloading, mencetak aktivitas belajar dengan tambahan informasi ruangan.
    - int jamBelajar( int sks) Overriding, menghitung waktu belajar berdasarkan SKS ( sama dengan kelas induk).
  - b. Constructor
    - Menggunakan constructor dereliction dari kelas induk.

2. Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak( turunan dari mahasiswa)!

1. Kelas Induk( Mahasiswa)
  - Memiliki atribut ruangan dan belajar.
  - Memiliki metode
    - sedangBelajar( String belajar) – mencetak pesan kegiatan belajar dengan parameter belajar.
    - jamBelajar( int sks) – menghitung jam belajar berdasarkan SKS( 1 SKS = 50 menit).
  - Digunakan untuk menunjukkan kegiatan belajar standar mahasiswa tanpa tambahan informasi seperti ruangan.
2. Kelas Anak( MahasiswaSI)

Mewarisi atribut dan metode dari kelas induk Mahasiswa. Overloading metode sedangBelajar menambahkan versi metode sedangBelajar( String belajar, String ruangan) yang mencetak pesan kegiatan belajar serta ruangan. Overriding metode jamBelajar metode ini menggantikan metode yang sama di kelas induk, namun dengan implementasi yang tetap sama(

- 1 SKS = 50 menit). Menyediakan lebih banyak informasi( seperti ruangan belajar) dibandingkan kelas induk.
3. Rekomendasi atribut, system, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!
- Atribut
    - a. Kelas Induk( Mahasiswa) String belajar
    - b. Kelas Anak( MahasiswaSI) String ruangan
  - Method
    - a. Kelas Induk
      - a. sedangBelajar( String belajar)( bisa dioverride di anak)
      - b. jamBelajar( int sks)( umum untuk semua)
    - b. Kelas Anak
      - a. Stamp sedangBelajar( String belajar, String ruangan)
    - c. Constructor
      - Kelas Induk Constructor untuk inisialisasi belajar.
      - Kelas Anak Constructor dengan super( belajar) dan inisialisasi ruangan.

#### **[No. 1]Penyusunan Algoritma dan Kode Program**

##### **1. Algoritma**

Algoritma adalah langkah- langkah menyelesaikan masalah

Algoritma

1. Deklarasi kelas Mahasiswa.
2. Deklarasi atribut String ruangan dan String belajar.
3. Deklarasi system sedangBelajar( String belajar) untuk menampilkan pesan aktivitas belajar.
4. Deklarasi system jamBelajar( int sks) untuk menghitung total waktu belajar dengan mengalikan SKS dengan 50.
5. Deklarasi system main
6. Deklarasi kelas MahasiswaSI yang mewarisi kelas Mahasiswa.
7. Override sedangBelajar( String belajar, String ruangan) untuk menampilkan pesan belajar dengan informasi ruangan.
8. Override jamBelajar( int sks) untuk menghitung waktu belajar.
9. Deklarasi method main
10. Cetak pesan

##### **2. Kode program**

Screenshot Capture potongan kode dan hasil luaran

```

1 //Kelas Induk Mahasiswa
2 public class Mahasiswa {
3     //Atribut
4     String ruangan; //ruangan belajar mahasiswa
5     String belajar;
6     void sedangBelajar(String belajar) { //method untuk sedang belajar
7         System.out.println("Sedang belajar :"+ belajar); //cetak pesan sedang belajar
8     }
9     int jamBelajar(int sks) { //method untuk menghitung jam belajar
10        return sks * 50; //hitung total jam belajar dengan mengalikan sks dengan 50
11    }
12    public static void main(String[] args) { //method main
13        System.out.println("Kegiatan Mahasiswa UNIB"); //cetak pesan kegiatan mahasiswa UNIB
14        MahasiswaSI mahasiswa = new MahasiswaSI();
15        mahasiswa.sedangBelajar("Pemrograman Java"); //sedang belajar Pemrograman Java
16        mahasiswa.jamBelajar(3); //jumlah jam belajar 3 jam
17    }
18 }
19
20 class MahasiswaSI extends Mahasiswa {
21     void sedangBelajar(String belajar, String ruangan) { //method untuk sedang belajar dengan ruangan
22         System.out.println("Sedang belajar :"+ belajar + " di Ruangan : " + ruangan); //cetak pesan sedang belajar dengan ruangan
23     }
24     int jamBelajar(int sks) { //method untuk menghitung jam belajar
25         return sks * 50; //hitung total jam belajar dengan mengalikan sks dengan 50
26     }
27     public static void main(String[] args) { //method main
28         System.out.println("Kegiatan Mahasiswa UNIB"); //cetak pesan kegiatan mahasiswa UNIB
29         Mahasiswa mahasiswa = new Mahasiswa();
30         mahasiswa.sedangBelajar("Pemrograman Java"); //sedang belajar Pemrograman Java
31         System.out.println("Total jam belajar: " + mahasiswa.jamBelajar(3)+" Menit"); //total jam belajar 3 jam
32         System.out.println();
33         System.out.println("Kegiatan Mahasiswa SISTEM INFORMASI"); //cetak pesan kegiatan mahasiswa SISTEM INFORMASI
34         MahasiswaSI mahasiswa2 = new MahasiswaSI();
35         mahasiswa2.sedangBelajar("Pemrograman Berorientasi Objek", "RK 3.7FT"); //sedang belajar Pemrograman Berorientasi Objek di ruangan RK 3.7FT
36         System.out.println("Total jam belajar: " + mahasiswa2.jamBelajar(2)+" Menit"); //total jam belajar 2 jam
37     }
38 }
39
40

```

## Kode

*//Kelas Induk Mahasiswa*

*public class Mahasiswa {*

*//Atribut*

*String ruangan; //ruangan belajar mahasiswa*

*String belajar;*

*void sedangBelajar(String belajar) { //method untuk sedang belajar*

*System.out.println("Sedang belajar :"+ belajar); //cetak pesan sedang belajar*

*}*

*int jamBelajar(int sks) { //method untuk menghitung jam belajar*

*return sks \* 50; //hitung total jam belajar dengan mengalikan sks dengan 50*

*}*

*public static void main(String[] args) { //method main*

*System.out.println("Kegiatan Mahasiswa UNIB"); //cetak pesan kegiatan mahasiswa UNIB*

*MahasiswaSI mahasiswa = new MahasiswaSI();*

*mahasiswa.sedangBelajar("Pemrograman Java"); //sedang belajar Pemrograman Java*

*mahasiswa.jamBelajar(3); //jumlah jam belajar 3 jam*

*}*

*}*

*class MahasiswaSI extends Mahasiswa {*

*void sedangBelajar(String belajar, String ruangan) { //method untuk sedang belajar dengan ruangan*

*System.out.println("Sedang belajar :"+ belajar + " di Ruangan : " + ruangan); //cetak pesan*

*sedang belajar dengan ruangan*

*}*

*int jamBelajar(int sks) { //method untuk menghitung jam belajar*

*return sks \* 50; //hitung total jam belajar dengan mengalikan sks dengan 50*

*}*

*public static void main(String[] args) { //method main*

*System.out.println("Kegiatan Mahasiswa UNIB"); //cetak pesan kegiatan mahasiswa UNIB*

*Mahasiswa mahasiswa = new Mahasiswa();*

*mahasiswa.sedangBelajar("Pemrograman Java"); //sedang belajar Pemrograman Java*

```

        System.out.println("Total jam belajar: " + mahasiswa.jamBelajar(3)+" Menit"); //total jam
        belajar 3 jam
        System.out.println();
        System.out.println("Kegiatan Mahasiswa SISTEM INFORMASI"); //cetak pesan kegiatan
        mahasiswa SISTEM INFORMASI
        MahasiswaSI mahasiswa2 = new MahasiswaSI();
        mahasiswa2.sedangBelajar("Pemrograman Berorientasi Objek", "RK 3.7FT"); //sedang belajar
        Pemrograman Berorientasi Objek di ruangan RK 3.7FT
        System.out.println("Total jam belajar: " + mahasiswa2.jamBelajar(2)+" Menit"); //total jam
        belajar 2 jam
    }
}

```

#### **Luaran**

```

Kegiatan Mahasiswa UNIB
Sedang belajar :Pemrograman Java
Total jam belajar: 150 Menit

Kegiatan Mahasiswa SISTEM INFORMASI
Sedang belajar :Pemrograman Berorientasi Objek di Ruangan : RK 3.7FT
Total jam belajar: 100 Menit

```

#### **No. 1) Kesimpulan**

Kode ini dibuat untuk menurunkan Kelas Mahasiswa sebagai kelas induk dan MahasiswaSI sebagai kelas turunan. Bahwa MahasiswaSI adalah turunan dari Mahasiswa karena Mahasiswa mencakup 1 universitas sedangkan MahasiswaSI hanya mencakup 1 prodi saja.

#### **Refleksi**

Praktikum minggu ini cukup ada peningkatan mulai dari kelas, objek dan system. Untuk praktikum selanjutnya mungkin dapat ditingkatkan lagi dalam belajar kelas, objek dan system.