

JOBSITE II

OBJECT



Disusun oleh:

Nama : Galih Candra Kirana
No. Absen : 10
Kelas : TI 1G
NIM : 254107020080

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

- Mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
- Mendeklarasikan class, atribut dan method
- Membuat objek (instansiasi)
- Mengakses atribut dan method dari suatu objek
- Menerapkan konstruktor

2. Praktikum

2.1. Percobaan 1: Deklarasi Class, Atribut dan Method

2.1.1. Langkah-langkah

```
1  public class Mahasiswa0 {  
2  
3      String nama;  
4      String nim;  
5      String kelas;  
6      double ipk;  
7  
8      void tampilanInformasi() {  
9          System.out.println("Nama: " + nama);  
10         System.out.println("NIM: " + nim);  
11         System.out.println("IPK: " + ipk);  
12         System.out.println("Kelas: " + kelas);  
13     }  
14  
15     void ubahKelas(String kelasBaru) {  
16         kelas = kelasBaru;  
17     }  
18  
19     void updateIpk(double ipkBaru) {  
20         ipk = ipkBaru;  
21     }  
22  
23     String nilaiKinerja() {  
24         if (ipk < 0.0 && ipk > 4.0) {  
25             return "IPK tidak valid. Harus antara 0.0 dan 4.0";  
26         } else if (ipk >= 3.5) {  
27             return "Kinerja sangat baik";  
28         } else if (ipk >= 3.0) {  
29             return "Kinerja baik";  
30         } else if (ipk >= 2.0) {  
31             return "Kinerja cukup";  
32         } else {  
33             return "Kinerja kurang";  
34         }  
35     }  
36 }
```

2.1.2. Verifikasi Hasil Percobaan

✖ Error: Main method not found in the file, please define the main method as: public static void main(String[] args)
Source: Debugger for Java

2.1.3. Pertanyaan:

- 1) Sebutkan dua karakteristik class atau object!

Jawaban:

1. Mempunyai sesuatu.
2. Melakukan sesuatu.

- 2) Perhatikan class **Mahasiswa** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!

Jawaban:

Ada 4 atribut, yaitu: nama, NIM, kelas, IPK.

- 3) Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!

Jawaban:

Ada 4 method, yaitu: tampilkanInformasi, ubahKelas, updateIpk, nilaiKerja.

- 4) Perhatikan method **updateIpk()** yang terdapat di dalam class **Mahasiswa**.

Modifikasi isi method tersebut sehingga IPK yang dimasukkan valid yaitu terlebih dahulu dilakukan pengecekan apakah IPK yang dimasukkan di dalam rentang 0.0 sampai dengan 4.0 ($0.0 \leq \text{IPK} \leq 4.0$). Jika IPK tidak pada rentang tersebut maka dikeluarkan pesan: "IPK tidak valid. Harus antara 0.0 dan 4.0".

Modifikasi:

```
23 ~ String nilaiKinerja() {
24 ~     if (ipk < 0.0 && ipk > 4.0) {
25 ~         return "IPK tidak valid. Harus antara 0.0 dan 4.0";
26 ~     } else if (ipk >= 3.5) {
27 ~         return "Kinerja sangat baik";
28 ~     } else if (ipk >= 3.0) {
29 ~         return "Kinerja baik";
30 ~     } else if (ipk >= 2.0) {
31 ~         return "Kinerja cukup";
32 ~     } else {
33 ~         return "Kinerja kurang";
34 ~     }
35 }
```

- 5) Jelaskan bagaimana cara kerja method **nilaiKinerja()** dalam mengevaluasi kinerja mahasiswa, kriteria apa saja yang digunakan untuk menentukan nilai kinerja tersebut, dan apa yang dikembalikan (di-return-kan) oleh method **nilaiKinerja()** tersebut?

Jawaban:

Cara kerja method **nilaiKerja** adalah melakukan pengecekan kondisi dari yang pertama, apakah $\text{IPK} < 0.0$ dan $\text{IPK} > 4.0$? Jika iya, return "IPK tidak valid". Jika tidak, maka dilanjut apakah $\text{IPK} \geq 3.5$? Jika iya, return "Kinerja sangat baik". Jika tidak, maka dilanjut apakah $\text{IPK} \geq 3.0$? Jika iya, return "Kinerja baik". Jika tidak, maka dilanjut apakah $\text{IPK} \geq 2.0$? Jika iya, return "Kinerja cukup". Jika tidak dari kondisi tadi, maka akan di return "Kinerja kurang".

- 6) Commit dan push kode program ke Github



2.2. Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

2.2.1. Langkah-langkah

```
1 public class MahasiswaMain10 {
2     public static void main(String[] args) {
3
4         Mahasiswa10 mhs1 = new Mahasiswa10();
5         mhs1.nama = "Muhammad Ali Fathan";
6         mhs1.nim = "2241720171";
7         mhs1.kelas = "SI 2J";
8         mhs1.ipk = 3.55;
9
10        mhs1.tampilkanInformasi();
11        mhs1.ubahKelas(kelasBaru: "SI 2K");
12        mhs1.updateIpk(ipkBaru: 3.60);
13        mhs1.tampilkanInformasi();
14    }
15 }
```

2.2.2. Verifikasi hasil percobaan

```
Nama: Muhammad Ali Fathan
NIM: 2241720171
IPK: 3.55
Kelas: SI 2J
Nama: Muhammad Ali Fathan
NIM: 2241720171
IPK: 3.6
Kelas: SI 2K
```

2.2.3. Pertanyaan

- 1) Pada class **MahasiswaMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?

Jawaban:

```
Mahasiswa10 mhs1 = new Mahasiswa10();
```

Object yang dihasilkan dari instansiasi di atas adalah object mhs1 yang memiliki atribut dan method dari class Mahasiswa10.

- 2) Bagaimana cara mengakses atribut dan method dari suatu objek?

Jawaban:

Cara mengaksesnya dengan menambahkan “.” Di belakang object yang akan diakses dan menambahkan atribut/methodnya langsung agar bisa diakses.

- 3) Mengapa hasil output pemanggilan method **tampilkanInformasi()** pertama dan kedua berbeda

Jawaban:

Karena telah di set ulang atribut kelas dan ipk, yang ditandai dengan pemanggilan method ubahKelas serta updateIpk.

2.3. Percobaan 3: Membuat Konstruktor

2.3.1. Langkah-langkah

```
8 |     Mahasiswa10 () {
9 | }
10 |
11 |     Mahasiswa10(String nm, String nim, double ipk, String kls) {
12 |         nama = nm;
13 |         this.nim = nim;
14 |         this.ipk = ipk;
15 |         kelas = kls;
16 |     }
17 |
15 |     Mahasiswa10 mhs2 = new Mahasiswa10(nm: "Annisa Nabila", nim: "2141720160", ipk: 3.25, kls: "TI 2L");
16 |
17 |     mhs2.updateIpk(ipkBaru: 3.30);
|     mhs2.tampilkanInformasi();
```

2.3.2. Verifikasi hasil percobaan

```
Nama: Muhammad Ali Fathan
NIM: 2241720171
IPK: 3.55
Kelas: SI 2J
Nama: Muhammad Ali Fathan
NIM: 2241720171
IPK: 3.6
Kelas: SI 2K
Nama: Annisa Nabila
NIM: 2141720160
IPK: 3.3
Kelas: TI 2L
```

2.3.3. Pertanyaan

- 1) Pada class **Mahasiswa** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!

Jawaban:

```
Mahasiswa10(String nm, String nim, double ipk, String kls) {
    nama = nm;
    this.nim = nim;
```

```

        this.ipk = ipk;
        kelas = kls;
    }
}

```

- 2) Perhatikan class **MahasiswaMain**. Apa sebenarnya yang dilakukan pada baris program berikut?

```
Mahasiswa mhs2 = new Mahasiswa("Annisa Nabila", "2141720160", 3.25, "TI 2L");
```

Jawaban:

Yang dilakukan baris tersebut adalah menginstansiasi class Mahasiswa yang memiliki konstruktor nama, Nim, IPK dan kelas menjadi sebuah object yang memiliki nama mhs2.

- 3) Hapus konstruktor default pada class **Mahasiswa**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

Jawaban:

Setelah konstruktor default dihapus, maka terjadilah error di MahasiswaMain. Hal itu terjadi karena ada object mhs1 yang masih memakai konstruktor default.

- 4) Setelah melakukan instansiasi object, apakah method di dalam class Mahasiswa harus diakses secara berurutan? Jelaskan alasannya!

Jawaban:

Tidak harus diakses secara berurutan karena bisa disesuaikan sesuai kebutuhan.

- 5) Buat object baru dengan nama mhs<NamaMahasiswa> menggunakan konstruktor berparameter dari class Mahasiswa!

```

19 |     Mahasiswa10 mhsGalih = new Mahasiswa10(nm: "Galih Candra K", nim: "254107020080", ipk: 3.65, kls: "TI 1G");
20 |     mhsGalih.tampilanInformasi();

```

- 6) Commit dan push kode program ke Github

modifikasi percobaan 3

 galihcandraa committed 2 minutes ago

2.4 Latihan Praktikum

1) Class Diagram Matkul

- a. Class Matkul Main:

```

1  public class MataKuliahMain10 {
2      Run | Debug
3
4      public static void main(String[] args) {
5
6          MataKuliah10 matkul1 = new MataKuliah10();
7          matkul1.nama = "Algoritma Struktur Data";
8          matkul1.kodeMk = "ASD";
9          matkul1.sks = 4;
10         matkul1.jumlahJam = 8;
11         matkul1.tampilanInformasi();
12         matkul1.ubahSKS(sksBaru: 3);
13         matkul1.tambahJam(jam: 4);
14         matkul1.kurangJam(jam: 2);
15
16         MataKuliah10 matkul2 = new MataKuliah10(kodeMk: "Konsep Teknologi
17             Informatasi", nama: "KTI", sks: 2, jumlahJam: 4);
18         matkul2.tampilanInformasi();
19         matkul2.ubahSKS(sksBaru: 3);
20         matkul2.tambahJam(jam: 4);
21         matkul2.kurangJam(jam: 2);
22     }
}

```

b. Class Matkul

```
1  public class MataKuliah10 {
2
3      String kodeMK;
4      String nama;
5      int sks;
6      int jumlahJam;
7
8      MataKuliah10() {
9      }
10
11     MataKuliah10(String kodeMK, String nama, int sks, int jumlahJam) {
12         this.kodeMK = kodeMK;
13         this.nama = nama;
14         this.sks = sks;
15         this.jumlahJam = jumlahJam;
16     }
17
18     void tampilInformasi() {
19         System.out.println("Kode MK: " + kodeMK);
20         System.out.println("Nama: " + nama);
21         System.out.println("SKS: " + sks);
22         System.out.println("Jumlah Jam: " + jumlahJam);
23     }
24
25     void ubahSKS(int sksBaru) {
26         sks = sksBaru;
27         System.out.println("SKS telah berubah");
28     }
29
30     void tambahJam(int jam) {
31         jumlahJam += jam;
32     }
33
34     void kurangJam(int jam) {
35         if (jam >= jumlahJam) {
36             System.out.println("Pengurangan tidak dapat dilakukan!");
37         } else {
38             jumlahJam -= jam;
39             System.out.println("Jumlah Jam baru: " + jumlahJam);
40         }
41     }
42 }
```

c. Output

```
Kode MK: ASD
Nama: Algoritma Struktur Data
SKS: 4
Jumlah Jam: 8
SKS telah berubah
Jumlah Jam baru: 10
Kode MK: Konsep Teknologi Informasi
Nama: KTI
SKS: 2
Jumlah Jam: 4
SKS telah berubah
Jumlah Jam baru: 6
```

2) Class diagram Dosen

a. Class Dosen Main

```
1  public class DosenMain10 {
2      public static void main(String[] args) {
3
4          Dosen10 dosen1 = new Dosen10();
5          dosen1.idDosen = "22314";
6          dosen1.nama = "Eko";
7          dosen1.statusAktif = true;
8          dosen1.tahunBergabung = 2010;
9          dosen1.bidangKeahlian = "Information Technology";
10
11         dosen1.tampilInformasi();
12         dosen1.setStatusAktif(status: false);
13         dosen1.hitungMasaKerja(thnSkrg: 2026);
14         dosen1.ubahKeahlian(bidang: "IT");
15         dosen1.tampilInformasi();
16         System.out.println();
17
18         Dosen10 dosen2 = new Dosen10(idDosen: "22315", nama: "Sandhika",
19         statusAktif: false, tahunBergabung: 2015, bidang: "IT");
20         dosen2.tampilInformasi();
21         dosen2.setStatusAktif(status: true);
22         dosen2.hitungMasaKerja(thnSkrg: 2026);
23         dosen2.ubahKeahlian(bidang: "Information Technology");
24         dosen2.tampilInformasi();
25     }
}
```

b. Class Dosen

```
1 < public class Dosen10 {  
2  
3     String idDosen;  
4     String nama;  
5     boolean statusAktif;  
6     int tahunBergabung;  
7     String bidangKeahlian;  
8  
9     Dosen10() {  
10 }  
11  
12     Dosen10(String idDosen, String nama, boolean statusAktif, int tahunBergabung, String bidangKeahlian) {  
13         this.idDosen = idDosen;  
14         this.nama = nama;  
15         this.statusAktif = statusAktif;  
16         this.tahunBergabung = tahunBergabung;  
17         this.bidangKeahlian = bidangKeahlian;  
18     }  
19  
20     void tampilInformasi() {  
21         System.out.println("Id Dosen: " + idDosen);  
22         System.out.println("Nama: " + nama);  
23         System.out.println("Status aktif: " + statusAktif);  
24         System.out.println("Tahun bergabung: " + tahunBergabung);  
25         System.out.println("Bidang keahlian: " + bidangKeahlian);  
26     }  
27  
28     boolean setStatusAktif(boolean status) {  
29         return statusAktif = status;  
30     }  
31  
32     int hitungMasaKerja(int thnSkrg) {  
33         int masakerja = thnSkrg - tahunBergabung;  
34         System.out.println("Lama bekerja: " + masakerja + " tahun");  
35         return masakerja;  
36     }  
37  
38     String ubahKeahlian(String bidang) {  
39         return bidangKeahlian = bidang;  
40     }  
41 }
```

c. Output

```
Id Dosen: 22314  
Nama: Eko  
Status aktif: true  
Tahun bergabung: 2010  
Bidang keahlian: Information Technology  
Lama bekerja: 16 tahun  
Id Dosen: 22314  
Nama: Eko  
Status aktif: false  
Tahun bergabung: 2010  
Bidang keahlian: IT  
  
Id Dosen: 22315  
Nama: Sandhika  
Status aktif: false  
Tahun bergabung: 2015  
Bidang keahlian: IT  
Lama bekerja: 11 tahun  
Id Dosen: 22315  
Nama: Sandhika  
Status aktif: true  
Tahun bergabung: 2015  
Bidang keahlian: Information Technology
```