Praktikum
Object Oriented Program
2022

EAD Laboratory
EAD

Modul 1

Method dan Class

ASISTEN


ADIS


AFHM


ARTI


CACA


DAPA


DEYA


DIKA


DRFZ


FATH


FDIL


FRQI


HILM


NICA


OLLA


RAIN


SKUY


TITY


WANN


WISH


XASH

Daftar Isi

Peraturan Praktikum

1. Setiap peserta praktikum harus datang tepat waktu sesuai dengan jadwal. Toleransi keterlambatan hadir 10 menit. Jika melebihi batas waktu tersebut dan tidak dapat memberikan bukti alasan keterlambatan maka praktikan tidak diperkenankan mengikuti praktikum.

2. Perizinan Praktikum :
    a. Izin berkaitan dengan Sakit atau Kemalangan , maka praktikan dapat memberikan surat perizinan kepada pihak Komisi Disiplin maksimal 3 hari setelah jadwal (shift) praktikum.
    b. Izin lomba atau penugasan institusi tidak berlaku apabila tidak terdapat bukti dispensasi dari Igracias. NB : screenshot dispensasi dari igracias wajib dilampirkan.

3. Seragam Praktikum :
    a. Mahasiswa wajib menggunakan celana bahan hitam (bukan chino atau jeans) pada saat praktikum.
    b. Mahasiswi wajib menggunakan rok hitam/biru gelap panjang tidak ketat pada saat praktikum
    c. Dresscode praktikum (Mengikuti Peraturan Telkom)
        - Senin : Menggunakan kemeja merah telkom atau kemeja putih polos
        - Selasa s/d Rabu : Menggunakan kemeja putih polos
        - Kamis s/d Sabtu : Menggunakan kemeja formal berkerah (bukan kerah sanghai dan bukan polo)
    d. Membuka sepatu saat memasuki ruangan lab.

4. Peraturan Pengerjaan
    a. Studi Kasus dikerjakan secara individu.
    b. Jawaban tidak boleh sama dengan setiap individu.
    c. Hasil pengerjaan di push ke repository github tiap individu
    d. Submit hasil pengerjaan berupa file format PDF dan link github ke LMS tiap individu
    e. Format Pengumpulan Hasil Pengerjaan
        - Format nama repository
          OOP_KODEASISTEN_NAMAPENDEK_NIM
          Contoh :
          OOP_ADIS_ADHISTY_1202204087
        - Format nama file PDF :
          OOP_KODEASISTEN_NAMAPENDEK_NIM_SSMODULX
          Contoh :
          OOP_ADIS_ADHISTY_1202204087_SSMODUL1

f. Salah penamaan pada file pengerjaan nilai modul akan dipotong sebesar 10%
g. Terlambat mengumpulkan file pengerjaan nilai modul akan dipotong sebesar 20%
h. Segala alat komunikasi dikumpulkan di loker yang tersedia dalam ruangan.
i. Jika ada perangkat praktikum yang bermasalah dapat menghubungi asprak yang bertugas.
j. Segala bentuk **kecurangan** dan **plagiarisme** akan diproses ke komisi disiplin dan nilai akhir modul menjadi 0.

Object Dan Class

Dua konsep terpenting dalam pemrograman berorientasi objek adalah kelas dan objek. Dalam arti luas, objek adalah sesuatu, baik yang berwujud maupun tidak berwujud, yang dapat kita bayangkan. Sebuah program yang ditulis dalam gaya berorientasi objek akan terdiri dari objek-objek yang saling berinteraksi. Contoh dari Objek itu sendiri seperti : Motor, Mobil, dll.

Di dalam program kita menulis instruksi untuk membuat objek. Agar komputer dapat membuat objek, kita harus memberikan definisi, yang disebut kelas. Kelas adalah sejenis cetakan atau template yang menentukan apa yang bisa dan tidak bisa dilakukan oleh objek. Sebuah objek disebut instance dari kelas.

Deklarasi Class

Pada pemrograman Java, pendeklarasian kelas dilakukan dengan menggunakan syntax sebagai berikut :

[modifier] class class_identifier

Keterangan :

1. *Modifier* : merupakan suatu kata kunci yang diberikan untuk mendefinisikan makna dari suatu *variable*, *method*, atau *class*. Terdapat dua jenis *modifier* pada Java yaitu *Access Control Modifier* dan *Non Access Control Modifier*.
    a. Contoh *Access Control Modifier :*
        i. *Default, merupakan Access Control Modifier* yang tidak ada pendeklarasian suatu modifier sehingga *variable*, *method*, atau *class* dapat langsung diakses di dalam *package* yang sama.
        ii. *Private*, merupakan akses yang hanya dapat dilakukan pada *class* yang sama.

iii. *Public*, merupakan akses yang dapat dilakukan oleh seluruh *package*.
iv. *Protected*, merupakan akses yang **tidak** dapat dilakukan dari luar *class*, namun bisa diakses dalam *package* yang sama dan seluruh sub-*class*nya.

b. Contoh *Non Access Control Modifier :*
   i. *Static*, merupakan salah satu jenis modifier di Java yang digunakan agar suatu atribut ataupun method dapat diakses oleh kelas atau *object* tanpa harus melakukan instansiasi terhadap kelas tersebut.
   ii. *Final*, merupakan proses finalisasi pada *method*, *class* dan *variable* agar nilai didalamnya tidak dapat berubah.
   iii. *Abstract*, merupakan pendeklarasian terhadap suatu *class* / *method* yang bersifat *abstract*.

2. *Class :* merupakan kata kunci untuk deklarasi sebuah kelas.
3. *Class_Identifier :* merupakan nama kelas yang dideklarasikan.

Contoh pendeklarasian Class :

```
public class barang {

}
```

Deklarasi Variabel

Deklarasi variabel dilakukan di dalam kelas. Variabel yang didefinisikan di kelas merupakan atribut dari kelas tersebut (yang otomatis merupakan atribut dari objek). Penugasan terhadap variabel merupakan pemberian nilai kepada nilai. Karena dalam deklarasi kelas, penugasan variabel merupakan penugasan pertama kali, maka penugasan ini dapat juga disebut inisialisasi variabel. Syntax umum deklarasi dan inisialisasi variabel adalah sebagai berikut :

[modifier] data_type identifier [=value];

Keterangan :

1. Modifier : Merupakan kata kunci untuk mendefinisikan makna dari suatu variable, method, atau class.

2. Data_type : Merupakan kata kunci tipe data dari variabel, misalnya: int, float, double, string. dll
3. Identifier : Merupakan nama variabel.
4. Value : Merupakan nilai awal dari variabel, bersifat opsional

Contoh pendeklarasian variable:

```
public class barang {
     public int jumlah;
     public int hargaBeli;
     public Date tanggalKadaluarsa;
     public int hargaJual;
     public String idBarang;
     public double diskon=0.0;
}
```

Method

Method adalah satu container pada kelas yang memuat baris-baris kode. Method biasanya digunakan untuk mengenkapsulasi proses-proses yang diperlukan untuk membentuk suatu fungsi/tugas tertentu. Misalnya ada object kucing, dan kucing tersebut dapat makan. Method secara singkat dibagi menjadi dua yaitu method void dan method return

Method Void

Method void adalah method yang digunakan untuk memberikan nilai pada suatu attribute, object, dan lain - lain. Method void memiliki ciri ciri memiliki kata void dan this.

Contoh Method void:

```
public class Animal {

    String name;
    String kind;
    String food;

    public void setName(String name){
        this.name = name;
    }

    public void setKind(String kind){
        this.kind = kind;
    }

    public void setFood(String food){
        this.food = food;
    }
```

Method Return

Method return adalah method yang pengembalian nilai dari suatu object atau attribute yang sudah berisi nilai. Method return memakai tipe data dan tidak menggunakan void melainkan memiliki kata kunci return.

Contoh Method Return:

```
public class Animal {

    String name;
    String kind;
    String food;

    public String getName(){
        return name;
    }

    public String getKind(){
        return kind;
    }

    public String getfood(){
        return food;
    }
```

Method Static

Method Static merupakan method yang dapat berdiri sendiri tanpa perlu instansiasi dari kelas. [modifiers] static return_type method_identifier (parameter) { method_body; }

Contoh Method Static:

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            new Antrian();
            System.out.println("Nomor Antri ke - " + Antrian.nomorAntrian);
        }

        System.out.println("Nomor Antri terakhir " + Antrian.getAntrian());
    }
}
```

```
public class Antrian {
    public static int nomorAntrian = 0;

    protected static int getAntrian() {
        return nomorAntrian;
    }

    Antrian() {
        nomorAntrian++;
    }
}
```

```
run:
Nomor Antri ke - 1
Nomor Antri ke - 2
Nomor Antri ke - 3
Nomor Antri ke - 4
Nomor Antri ke - 5
Nomor Antri terakhir 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Daftar Pustaka

[1] Chuck Cavaness, Geoff Friesen, Special Edition Using JavaTM

[2] C. Thomas Wu, An Introduction to Object-Oriented Programming with Java, 5th Edition, The McGraw-Hill Companies, Inc, 2010

[3] Seno Adi Putra, Pemrograman Beriorientasi Obyek Dengan Teknologi Java, Edisi Pertama, Universitas Telkom, 2019

[3] Horstmann, Gary cornel, Cay S, Core Java 2 Volume I, Fundamentals 7th Edition, Prentice

hall, 2005 2 SE, QUE, 2000

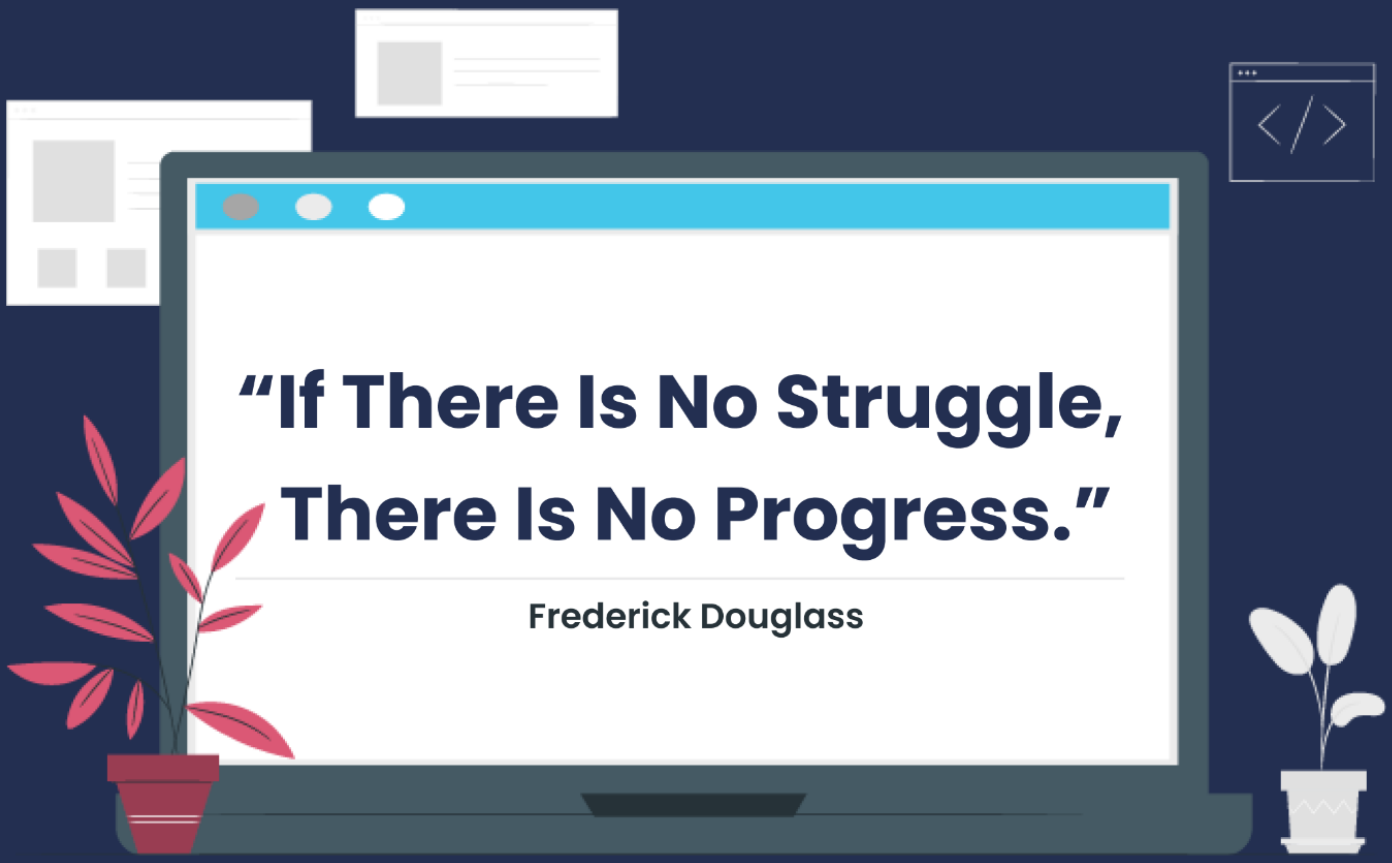
[4] Herbert Schildt, The Complete Reference Java J2SETM 5th

[5] James Gosling, The Java Edition, Mc Graw-Hill, Osborne, 2005.

[6] Laura Lemay, Sams Teach Yourself Java 2 in 21 Days, Second Edition, SAMS, 2000

Language Specification, Third Edition – Sun Mycrosystems, Addison-Wesley, 2005

EAD Laboratory
EAD



Find us on :

@eadlaboratory

@798bdlxl