

# MODUL 5

## STATIC & FINAL

### 1.1 TUJUAN PRAKTIKUM

Mahasiswa dapat memahami konsep final dan static dalam pemrograman berorientasi objek dan menggunakannya dengan baik

### 1.2 MATERI

#### 1. Static

- Property atau method dalam sebuah Class dapat dinyatakan **static** yang mana nilainya akan sama untuk semua *instance* atau objek yang dibuat dan nilainya jarang diganti-ganti. Hal ini berguna untuk menampung nilai yang sama untuk semua objek yang akan dibuat.
- Property / Method yang dinyatakan static adalah milik Class dan untuk mengaksesnya tidak perlu menggunakan perintah **new** seperti contoh berikut:

```
System.print.out(); // static method di class System, tanpa new  
Math.PI; // static property PI di Class Math, tanpa new Math()
```

- Pada umumnya **static property** digunakan untuk menyimpan konstanta (nilai yang tetap) dan **static method** digunakan untuk fungsi yang cepat dan ringkas tanpa butuh bantuan method lain.

#### 2. Final

- Kata kunci final di property dan method digunakan untuk mencegah programmer lain mengubah definisi atau nilai dari property/method yang kita buat
- Final Class tidak bisa diturunkan, sedangkan final method bisa diturunkan ke Class anak tapi tidak bisa di Override
- Kata static yang ditambah final biasanya bernilai tetap, tidak akan berubah dan tidak bisa diubah, contohnya pada Math.PI, Math.pow(), dsb.

-

### 1.3 PRAKTIKUM

- Buat Class baru dengan mengklik kanan icon **Java Source Package** di jendela *Project* lalu pilih **New -> Java Class**. Di bagian Class Name diisi dengan nama **Prodi** dan kemudian tekan tombol **Finish**.
- Ketikkan kode berikut di Class Prodi untuk membuat beberapa constant kode Prodi:



```
package your.app.package;

// class dibuat final agar tidak bisa di extends
public final class Prodi {

    public static final int INFORMATIKA = 52;
    public static final int TEKNIK_SIPIL = 51;
    public static final int TEKNIK_INDUSTRI = 53;

    public static String getNamaProdi(int kode) {
        switch (kode) {
            case 52:
                return "INFORMATIKA";
            case 51:
                return "TEKNIK SIPIL";
            case 53:
                return "TEKNIK INDUSTRI";
        }
        return "";
    }
}
```

- Buat Class baru dengan nama class **Mahasiswa** dan ketikkan kode berikut:

```

package your.app.package;

public class Mahasiswa {

    private final String npm;
    private final String nama;
    private int prodi = 0;
    public static String universitas = "Universitas Madura";

    public Mahasiswa(String npm, String nama) {
        this.npm = npm;
        this.nama = nama;
    }

    public void setProdi(int prodi) {
        this.prodi = prodi;
    }

    public static void ubahKampus(String kampus) {
        universitas = kampus;
    }

    public void tampil() {
        System.out.println(npm + " - " + nama + " - " + Prodi.getNamaProdi(prodi) + " - " +
            universitas);
    }

}

```

- Untuk contoh penggunaan static property / method Anda ketikkan kode berikut di **Main.java**:

```

package your.app.package;

public class Main {

    public static void main(String[] args) {
        Mahasiswa ridho = new Mahasiswa("2019520001", "Ridho");
        ridho.setProdi(Prodi.INFORMATIKA);
        ridho.tampil();
        Mahasiswa najib = new Mahasiswa("2019520002", "Najeeb");
        najib.setProdi(Prodi.TEKNIK_SIPIL);
        najib.tampil();

        // ubah static property, maka semua objek langsung berubah
        Mahasiswa.universitas = "UNIRA";
        ridho.tampil();
        najib.tampil();

        // ubah dengan static method
        Mahasiswa.ubahKampus("Univ. Madura");
        ridho.tampil();
        najib.tampil();
    }

}

```

- Jalankan aplikasi dan Anda akan dapat melihat hasil penggunaan beragam Class yang telah kita buat

#### 1.4 TUGAS

Buat aplikasi sederhana dengan input dari user untuk memasukkan data mahasiswa dengan ketentuan:

1. NIM, Nama dan Prodi harus dibuat final
2. Alamat dan email tidak boleh final
3. Buat constant kode Prodi dengan data dari tabel berikut:

ID PRODI	NAMA PRODI
11	HUKUM
21	MANAJEMEN
22	AKUNTANSI
31	ADMINISTRASI PUBLIK
41	PETERNAKAN
51	TEKNIK SIPIL
52	INFORMATIKA
53	TEKNIK INDUSTRI
61	PENDIDIKAN BAHASA INDONESIA
62	PENDIDIKAN MATEMATIKA
63	PENDIDIKAN BAHASA INGGRIS

Setiap kali selesai input data mahasiswa langsung tampilkan sebagai output, kemudian kembali minta data mahasiswa lagi (perulangan).