Fakultas Ilmu Terapan
School of Applied Science
Diploma of Informatics Management

CLASS & OBJECT

Disusun Oleh:
Reza Budiawan

Untuk:
Tim Dosen Algoritma & Pemrograman Lanjut

Kelas dan Objek

- Object adalah gambaran dari entitas
- Kelas mendeskripsikan suatu objek.
- Kelas memiliki
  - variabel yang disebut sebagai atribut
  - subrutin yang biasa disebut method.

Kelas Dan Objek

NIP: -
Nama: -
Divisi: -

NIP: 11-xx-76
Nama: ERIK
Divisi: SDM

NIP: 11-XX-83
Nama: ERIK
Divisi: Keuangan

ENKAPSULASI

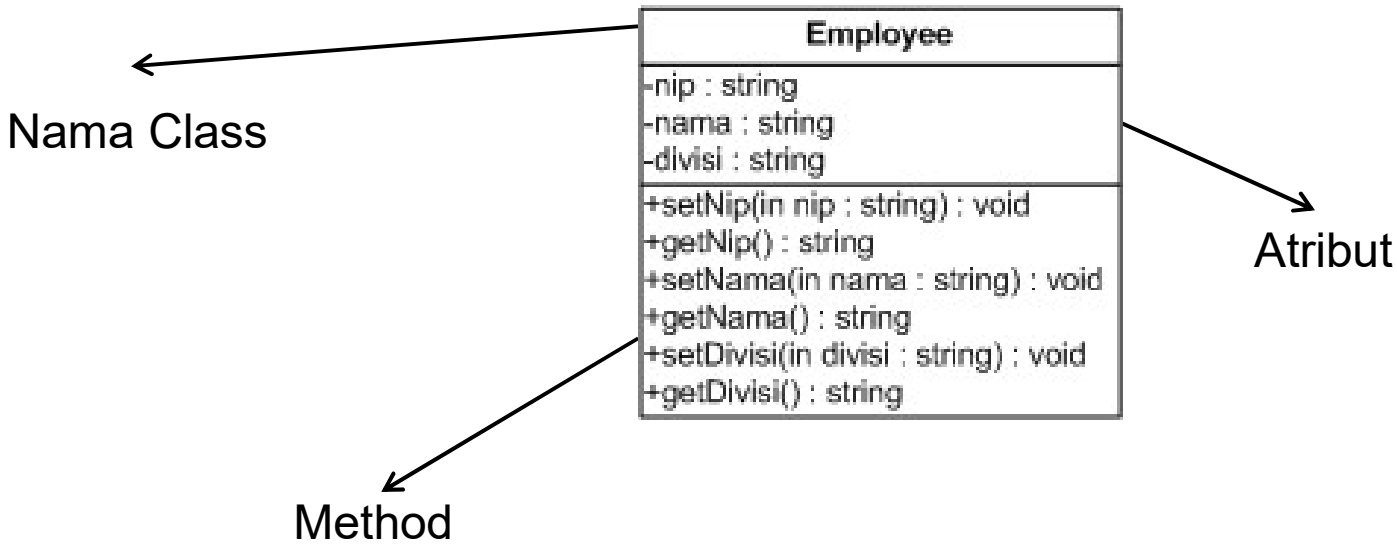
- Enkapsulasi merupakan proses pemaketan objek beserta methodnya
- Dimaksudkan untuk menyembunyikan rincian implementasi dari pemakai/objek lainnya.
- Inti dari enkapsulasi adalah ketidaktahuan apa yang ada dalam suatu objek dan bagaimana pengimplementasiannya.

Note: Berikan modifier private untuk atribut

Class & Object

- Object dibentuk dari Class melalui konstruktor
- Setiap atribut dan method dipanggil melalui object-nya
- Method static dan atribut static dipanggil melalui class-nya

CLASS


Employee
-nip : string
-nama : string
-divisi : string
+setNip(in nip : string) : void
+getNip() : string
+setNama(in nama : string) : void
+getNama() : string
+setDivisi(in divisi : string) : void
+getDivisi() : string
Nama Class
Atribut
Method

```
public class Employee{
    private String nip;
    private String nama;
    private String divisi;
    public void setNip(String nip){
        this.nip = nip;
    }
    public String getNip(){
        return this.nip;
    }
    ...
    ...
}
```

SILAHKAN DILANJUTKAN!!!

Note:
Perhatikan modifier pada attrb/method
(-) → private
(+) → public
(#) → protected

PEMBENTUKAN OBJEK

Konstruktor

- Constructor atau konstruktor digunakan untuk melakukan inisialisasi variable-variabel instan class
- Bentuk lain dari enkapsulasi adalah memasukkan nilai atribut dengan menggunakan konstruktor

Konstruktor = Membentuk Objek

Konstruktor

- Pendefenisian konstruktor:
  - Nama constructor sama dengan nama class.
  - Pengaturan visibility constructor.

```
public class Employee{
    private String nama;
    public Employee(){

    }
}
```

Konstruktor Kosong
(secara default akan dibentuk jika tidak ada konstruktor didefenisikan)

```
public class Employee{
    private String nama;
    public Employee(String nama){
        this.nama = nama;
    }
}
```

Konstruktor dengan parameter
(konstruktor kosong tidak akan dibentuk jika ada konstruktor yang didefenisikan)

Overloading Konstruktor

- Konstruktor dapat dituliskan >1x dalam sebuah class
- Penulisan kembali konstruktor disebut “overloading”

Apa syarat dari overloading?

Bagaimana cara penggunaan konstruktor overloading?

Penggunaan Konstruktor

```
public class Main{
    public static void main(String args[]){
        Employee e1 = new Employee("XYZ");
    }
}
```

Pemanggilan Konstruktor

Tipe Object

Nama Object

Keyword
Instansiasi

```
public class Employee{
    private String nama;
    public Employee(String nama){
        this.nama = nama;
    }
}
```

Prosedur dan Fungsi

- Prosedur → tidak mengembalikan nilai
- Fungsi → mengembalikan nilai
- Pemanggilan prosedur dan fungsi non-static dilakukan melalui objeknya

```
public class HitungAritmatik{
    private int hasil;
    public int hitungTambah(int a, int b){
          hasil = a+b;
    }
    public int hitungKali(int a, int b){
          System.out.println("Hasil Kali:"+(a*b));
    }

}
```

```
public class Main{
      public static void main(String args[]){
            HitungAritmatik h1 = new HitungAritmatik();
            h1.hitungKali(3,2);
            int x = h1.hitungTambah(4,17);
            System.out.println("Hasil Operasi:"+x);
      }
}
```

Prosedur dan Fungsi

NOTE:
Perhatikan cara pemanggilan dari prosedur
Perhatikan cara pemanggilan dari fungsi

} Apakah bedanya??

Perhatikan jumlah parameter masukan dari tiap pemanggilan

END OF SLIDE…