



# OBJECT

TIM AJAR

ALGORITMA DAN STRUKTUR DATA

2022/2023

# Capaian Pembelajaran

Setelah mempelajari materi Object, mahasiswa diharapkan mampu

- Memahami konsep object
- Mendeklarasikan class, atribut dan method
- Membuat object (instansiasi)
- Mengakses atribut dan method dari sebuah object
- Memanggil konstruktor
- Memahami konsep object dan menuliskan ke dalam bentuk class diagram

# Konsep Pemrograman Berorientasi Objek (1)

- Pada MK Dasar Pemrograman, pemrograman dilakukan menggunakan **prosedural** dengan menulis prosedur atau **method/fungsi** yang melakukan operasi pada data, sedangkan **PBO membuat objek yang berisi data dan method/fungsi**.
- **Java** merupakan bahasa pemrograman yang secara penuh menerapkan konsep Pemrograman Berorientasi Obyek (PBO) / *Object Oriented Programming* (OOP)
- PBO merupakan paradigma pemrograman yang memandang suatu **program** terdiri dari **kumpulan objek-objek** yang saling **berinteraksi**
- Ketika hendak membuat suatu program, yang harus diidentifikasi dulu adalah **objek-objek** apa yang ada di dalam sistem tersebut

# Konsep Pemrograman Berorientasi Objek (2)

	PBO	Prosedural
<b>Sudut pandang</b>	Program dianggap sebagai kumpulan objek-objek yang berinteraksi	Program dianggap sebagai kumpulan proses atau prosedur
<b>Fokus utama</b>	Objek	Proses/Prosedur/Fungsi
<b>Contoh</b>	Sistem Perbankan: <ul style="list-style-type: none"> <li>• Nasabah</li> <li>• Akun rekening</li> <li>• Transaksi</li> <li>• Teller</li> <li>• Uang</li> </ul>	Sistem Perbankan: <ul style="list-style-type: none"> <li>• Ubah PIN</li> <li>• Transfer</li> <li>• Tarik tunai</li> <li>• Setor</li> </ul>

# Konsep Pemrograman Berorientasi Objek (3)

- **Class** dan **object** adalah dua aspek utama dalam PBO
- Selain kedua aspek tersebut, ada beberapa aspek dasar lain, seperti Enkapsulasi, Inheritance, Polimorfisme, dll yang akan dijelaskan lebih detail pada MK PBO di Semester 3
- Pada MK ini, aspek utama yang digunakan adalah Class dan Object

# Object

**Object** adalah representasi dari **benda nyata**

Contoh object di dalam kamar tidur:

- Kasur
- Meja belajar
- Bantal1, Bantal2, dst

Contoh object di dalam kelas:

- Mahasiswa1, Mahasiswa2, Mahasiswa3, dst
- Papan tulis
- PC1, PC2, PC3, dst

# Karakteristik Object

## Istilah

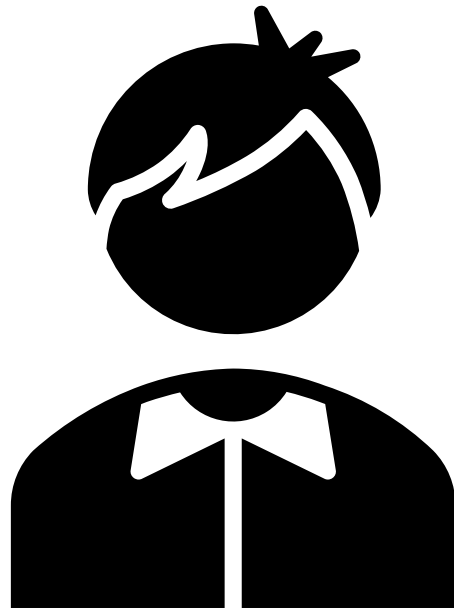
Mempunyai  
sesuatu

- Data
- Properti
- Variabel
- *State*
- Atribut

Melakukan  
sesuatu

- Tingkah laku
- *Behaviour*
- Fungsi
- Method

# Contoh Object (1)



Mahasiswa1

## Atribut

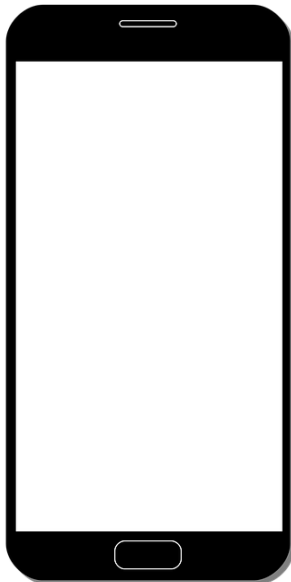
- NIM → 2414210045
- Nama → Will Dafoe
- IPK → 3.8
- Alamat → Malang

## Method

- Mengerjakan ujian
- Melihat KHS
- Melihat jadwal
- Melakukan absensi
- Mengumpulkan tugas



# Contoh Object (2)



HP1

## Atribut

- Merk → Samsung
- Tipe → S23 Ultra
- UkuranLayar → 6.8
- Harga → 20.000.000

## Method

- Mengirim pesan
- Menerima panggilan
- Membuka dokumen
- Connect bluetooth

# Class

**Class** merupakan **template** untuk membuat object

- Setiap **object** pasti berawal dari sebuah rancangan/ desain/ template class
- Proses pembuatan object dari suatu class disebut **instansiasi**
- **Object tidak akan ada jika tidak ada class**, dan class tidak akan bisa digunakan jika belum dibuat object-nya.

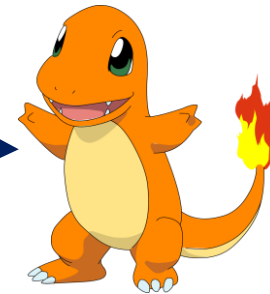
# Analogi Class



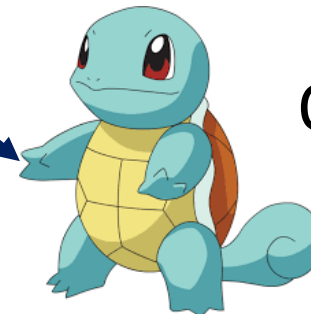
Class: **Pokemon**



Object: **Pokemon1**



Object: **Pokemon2**



Object: **Pokemon3**

# Class VS Object

	Class	Object
<b>Penjelasan</b>	Masih berupa <b>rancangan/ template/ desain/ blueprint</b>	<b>Objek nyata</b> yang sudah dibentuk dari suatu class
<b>Sifat</b>	Umum	Spesifik
<b>Contoh</b>	<ul style="list-style-type: none"> <li>• Mahasiswa</li> <li>• Dosen</li> <li>• Matakuliah</li> </ul>	<ul style="list-style-type: none"> <li>• Mahasiswa 1, Mahasiswa 2, dst</li> <li>• Dosen A, Dosen B, dst</li> <li>• Matakuliah Sistem Operasi, Matakuliah Basis Data</li> </ul>

# Apakah class memiliki atribut dan method?

- Karena **class pasti dibuat terlebih dahulu** sebelum object, tentunya class memiliki atribut dan method, yang kemudian juga dimiliki oleh object setelah dibuat.
- Namun, atribut dan method pada **object sudah nyata** (bisa diisi dengan nilai tertentu), sementara atribut dan method pada **class masih berupa rancangan** (tidak bisa diisi nilai)

# Implementasi Class pada Java

- Format Deklarasi Class:

```
class NamaClass{  
    //deklarasi atribut  
    //deklarasi method  
}
```

- Contoh:

```
class HP{  
  
  
}
```

# Atribut

**Atribut** merupakan **data** yang dimiliki oleh object atau class

- Nama atribut biasanya ditandai dengan **kata benda**
- Contoh atribut yang dimiliki oleh:
  - Mahasiswa → nim, nama, IPK, alamat
  - HP → merk, tipe, ukuranLayar, harga
  - Buku → judul, pengarang, halaman, penerbit

# Implementasi Atribut pada Java

- Format Deklarasi Atribut:

```
tipeData namaAtribut;
```

- Contoh:

```
String merk;  
String tipe;  
float ukuranLayar;  
int harga;
```



# Method

**Method** merupakan **proses/tingkah laku/fungsi** yang bisa dilakukan oleh object

- Method digunakan oleh object untuk berinteraksi dengan object yang lain
- Nama method biasanya ditandai dengan **kata kerja**
- Contoh method yang dimiliki oleh mahasiswa: mengerjakan ujian, melihat KHS, melihat jadwal, melakukan absensi, mengumpulkan tugas

# Implementasi Method pada Java

- Format Deklarasi Method:

```
tipeData namaMethod(tipeData parameter){  
    //isi method  
}
```

- Contoh:

```
void cekKondisi(boolean c){  
    if(c==true)  
        System.out.println("HP ini second\n");  
    else  
        System.out.println("HP ini masih baru\n");  
}
```

# Instansiasi

**Instansiasi** merupakan proses **pembuatan object** dari class

- Instansiasi ditandai dengan kata kunci **new**
- Format Instansiasi:

```
namaClass namaObject = new konstruktor();
```

- Contoh:

```
HP phone1 = new HP();
```

Konstruktor mempunyai nama yang sama dengan class, akan dibahas lebih detail pada slide berikutnya

# Cara Mengakses Atribut dan Method Suatu Object

- Setelah object terbentuk, selanjutnya atribut dan method yang dimiliki object tersebut dapat digunakan atau diakses

- Cara mengakses atribut:

```
namaObject.namaAtribut = nilai;
```

- Cara mengakses method:

```
namaObject.namaMethod();
```

- Contoh:

```
phone1.merk = "Samsung";  
phone1.ukuranLayar = 6.8f;  
phone1.cekKondisi(false);
```

# Kode Program Pembuatan Class HP

```
public class HP {  
    String merk;  
    String tipe;  
    float ukuranLayar;  
    int harga;  
  
    void cekKondisi(boolean c) {  
        if (c == true)  
            System.out.println(x:"HP ini second\n");  
        else  
            System.out.println(x:"HP ini masih baru\n");  
    }  
  
    void tampilInformasi() {  
        System.out.printf(format:"HP merk %s tipe %s dengan ukuran layar %.1f\n", merk, tipe, ukuranLayar);  
    }  
  
    void mengirimPesan(String pesan, String penerima, String paketData) {  
        if (paketData.equalsIgnoreCase(anotherString:"ada")) {  
            System.out.printf(format:"Pesan %s berhasil dikirim ke %s", pesan, penerima);  
        } else {  
            System.out.printf(format:"Pengirim pesan ke %s gagal", penerima);  
        }  
    }  
}
```

**Deklarasi Class**

**Atribut**

**Method**

# Kode Program Pembuatan Object phone1 dari Class HP

```
public class HPMain {  
    Run | Debug  
    public static void main(String[] args) {  
        HP phone1 = new HP();  
        phone1.merk = "Samsung";  
        phone1.tipe = "S23 Ultra";  
        phone1.ukuranLayar = 6.8f;  
        phone1.tampilInformasi();  
        phone1.cekKondisi(c:false);  
    }  
}
```

Memberikan nilai  
atribut dari object phone1

Menjalankan method dari object phone1

# Konstruktor

**Konstruktor** merupakan **method istimewa** yang digunakan saat pembuatan object (instansiasi)

Istimewa karena:

- Nama method sama dengan nama class
- Tidak memiliki tipe data method
- Hanya bisa dijalankan/dipanggil pada proses instansiasi
- Bisa memiliki parameter

# Jenis Konstruktor

## Konstruktor Default

Konstruktor yang tidak mempunyai parameter

- Contoh penulisan:

```
public HP(){  
}
```

## Konstruktor Berparameter

Konstruktor yang mempunyai parameter

- Contoh penulisan:

```
public HP(String mr, String tp, float ukuran){  
}
```



# Penggunaan Konstruktor Saat Instansiasi

- Konstruktor digunakan pada saat instansiasi
- Contoh penggunaan konstruktor default:

```
HP phone1 = new HP();
```

- Contoh penggunaan konstruktor berparameter:

```
HP phone1 = new HP("Samsung", "S23 Ultra", 6.8f);
```

# Kode Program Penggunaan Konstruktor

```
public class HP {  
    String merk;  
    String tipe;  
    float ukuranLayar;  
    int harga;  
  
    public HP(String mr, String tp, float ukuran) {  
        merk = mr;  
        tipe = tp;  
        ukuranLayar = ukuran;  
    }  
  
    void cekKondisi(boolean c) {  
        if (c == true)  
            System.out.println(x:"HP ini second\n");  
        else  
            System.out.println(x:"HP ini masih baru\n");  
    }  
  
    void tampilInformasi() {  
        System.out.printf(format:"HP merk %s tipe %s dengan ukuran layar %.1f\n", merk, tipe, ukuranLayar);  
    }  
  
    void mengirimPesan(String pesan, String penerima, String paketData) {  
        if (paketData.equalsIgnoreCase(anotherString:"ada")) {  
            System.out.printf(format:"Pesan %s berhasil dikirim ke %s", pesan, penerima);  
        } else {  
            System.out.printf(format:"Pengirim pesan ke %s gagal", penerima);  
        }  
    }  
}
```

**Konstruktor  
berparameter**

# Kode Program Penggunaan Konstruktor Saat Instansiasi

```
public class HPMain {  
    Run | Debug  
    public static void main(String[] args) {  
        HP phone1 = new HP();  
        phone1.merk = "Samsung";  
        phone1.tipe = "S23 Ultra";  
        phone1.ukuranLayar = 6.8f;  
        phone1.tampilInformasi();  
        phone1.cekKondisi(c:false);  
        HP phone2 = new HP(mr:"iPhone",tp:"15 Plus", ukuran:6.69f);  
    }  
}
```

**Konstruktor default**

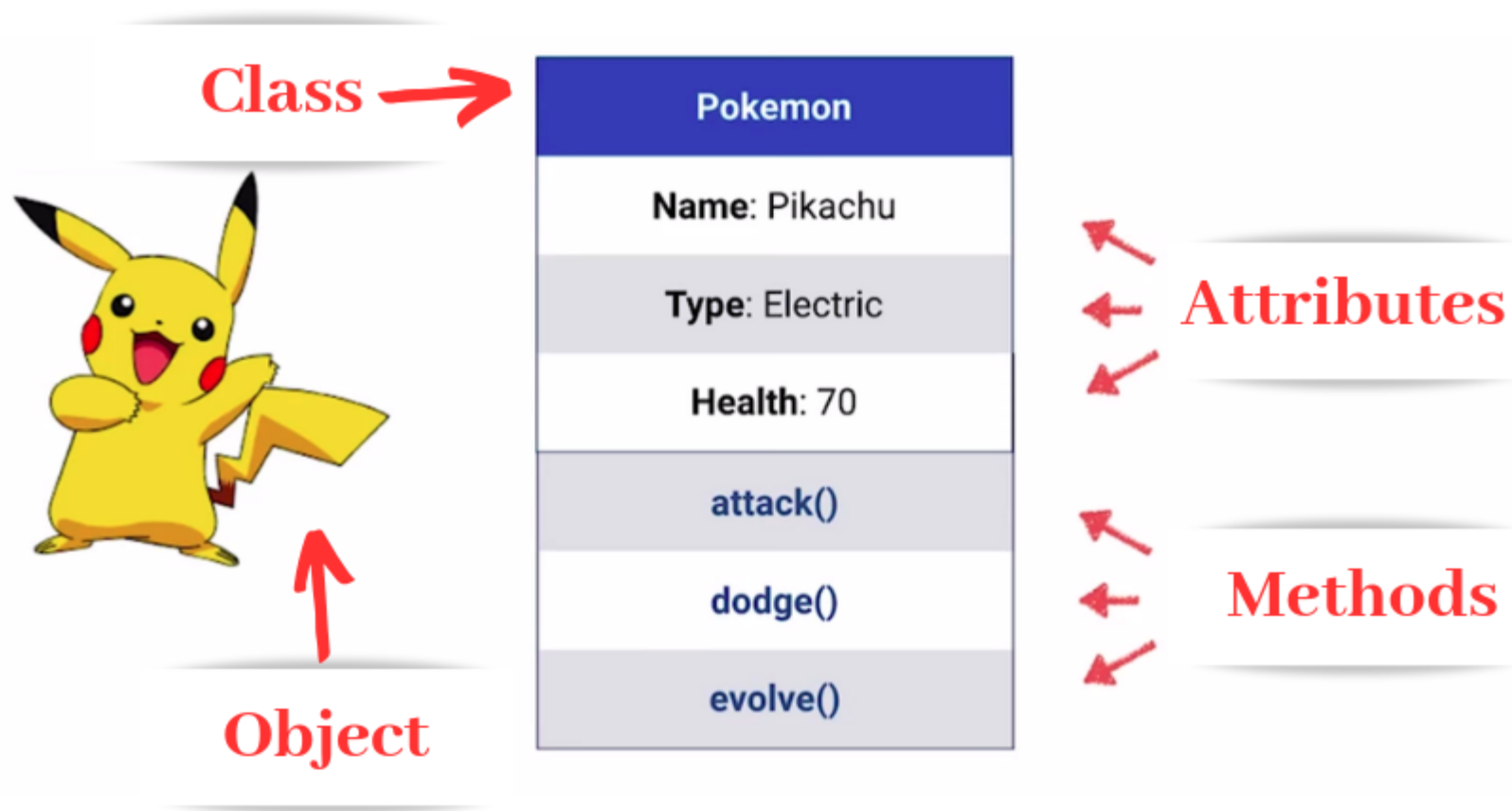
**Konstruktor berparameter**

# Class Diagram (1)

- Representasi dari class yang mempunyai atribut dan object
- Contoh:

HP
merk: String tipe: String ukuranLayar: float harga: int
cekKondisi(c: boolean): void tampilkanInformasi(): void mengirimPesan(psn: String, penerima: String, pkt: String): void

# Class Diagram (2)



<https://blog.glugmvit.com/oops/>

# Latihan

1. Tentukan sebuah object yang ada di sekitar Anda, serta tentukan atribut (minimal 4) dan method (minimal 3) yang dimiliki oleh object tersebut!
2. Pada Sistem Informasi Manajemen Peminjaman Ruang Kelas JTI, tentukan apa saja object-nya!
3. Terdapat sebuah class bernama **BangunRuang**, dengan dua object **PrismaSegitiga** dan **Balok** seperti pada gambar berikut. Buatlah class diagram dari **BangunRuang** tersebut!

