



OBJEK

TIM AJAR

ALGORITMA DAN STRUKTUR DATA

2022/2023

Konsep Pemrograman Berorientasi Objek #1



- Java → merupakan bahasa pemrograman yang secara penuh menerapkan konsep PBO
- Pemrograman Berorientasi Obyek (PBO) / *Object Oriented Programming* (OOP) → paradigma pemrograman yang memandang suatu **program** terdiri dari **kumpulan objek-objek** yang saling **berinteraksi**
- **Objek** menjadi perhatian utama
- Ketika hendak membuat suatu program dari suatu, yang harus diidentifikasi dulu adalah **objek-objek** apa yang ada di dalam sistem tersebut

Konsep Pemrograman Berorientasi Objek #2



	PBO	Prosedural
Sudut pandang	Melihat program sebagai kumpulan objek2 yg berinteraksi	Melihat program sebagai kumpulan proses2/prosedur2
Fokus utama	Objek	Proses/Prosedur/Fungsi
Contoh 1	Sistem Perkuliahan di Kelas: <ul style="list-style-type: none">• Mahasiswa• Dosen• Matakuliah• Form Absensi• Surat ijin• Soal	Sistem Perkuliahan di Kelas: <ul style="list-style-type: none">• Bertanya• Menjawab• Presentasi• Mengabsensi kehadiran• Mengerjakan soal

Konsep Pemrograman Berorientasi Objek #3



	PBO	Prosedural
Contoh 2	Sistem Perbankan: <ul style="list-style-type: none">• Akun/Rekening• Nasabah• Uang	Sistem Perbankan: <ul style="list-style-type: none">• Setor• Tarik• Transfer

Diskusi! Berikan contoh lain untuk konsep PBO dan prosedural.

Konsep Pemrograman Berorientasi Objek #4



- Konsep PBO yang paling mendasar adalah (1) **Class** dan (2) **Object**
- Selain kedua konsep dasar tersebut, ada beberapa konsep dasar lain:
 - Enkapsulasi
 - Inheritance
 - Polimorfisme
- Pada semester ini, konsep yang akan dominan digunakan adalah **Class** dan **Object**
- Konsep Enkapsulasi, Inheritance, dan Polimorfisme akan dijelaskan detail di matakuliah PBO semester 3

Lalu, apa itu objek?

Objek adalah representasi dari **benda nyata**

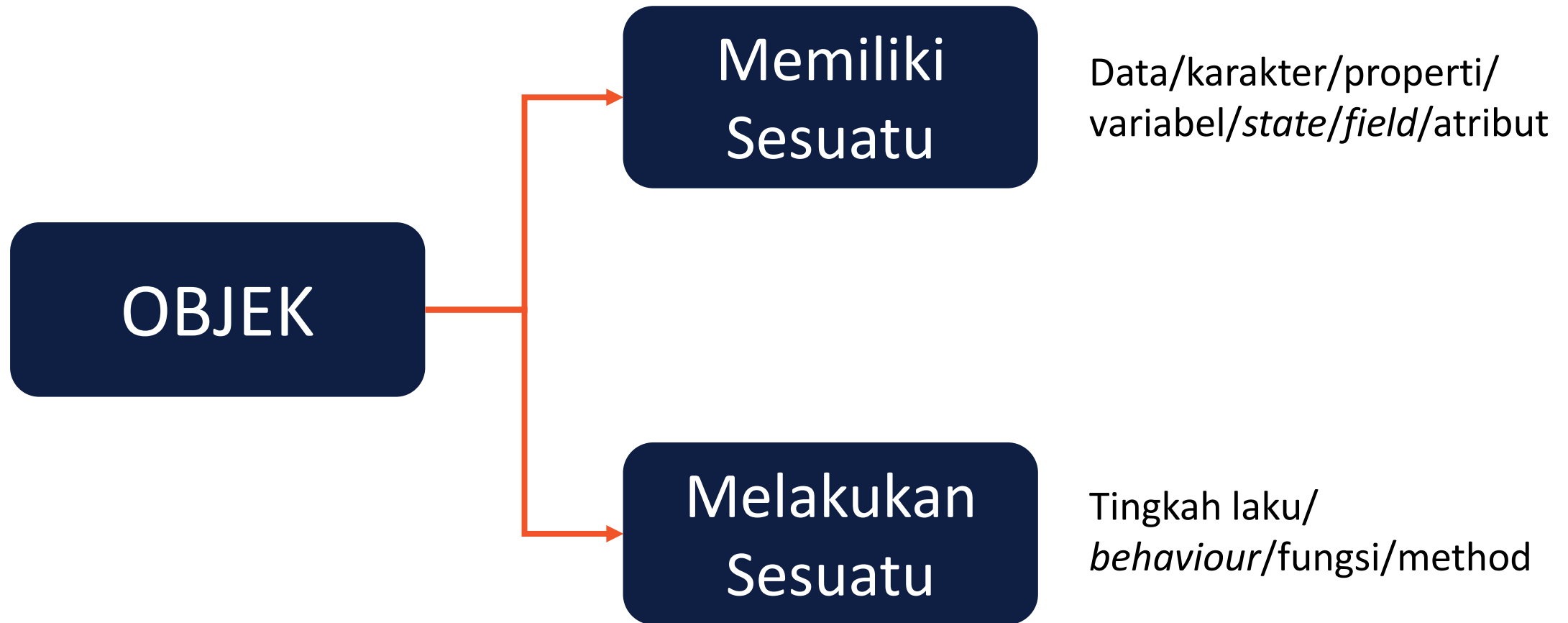
Contoh → Object di dalam kelas

- ❖ Meja 1, Meja 2, Meja 3, ..., Meja 30
- ❖ Mahasiswa 1, Mahasiswa 2, ..., Mahasiswa 30
- ❖ Proyektor
- ❖ Papan tulis
- ❖ Router
- ❖ Dan seterusnya

Objek



Bagaimana sebuah “benda” dapat dikatakan sebagai objek?



Contoh Objek #1



SEPEDA

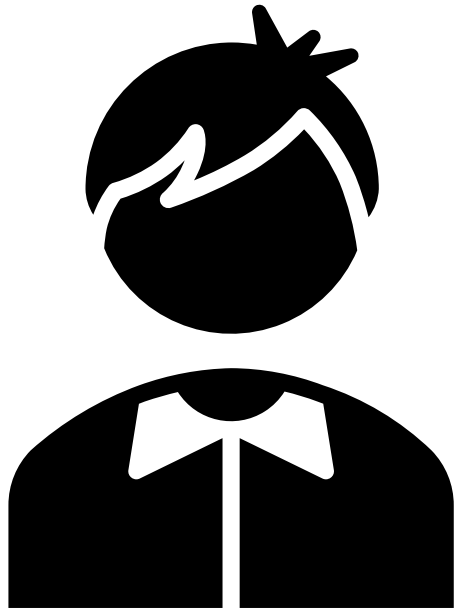
Atribut

- Gear → 5
- Kecepatan → 18 mph
- RPM → 90

Method

- Ubah gear
- Mengerem
- Ubah ritme

Contoh Objek #2



MAHASISWA

Atribut

- NIM → 1614210045
- Nama → Will Dafoe
- IPK → 3.8
- Alamat → Malang

Method

- Mengikuti ujian
- Melihat KHS
- Melihat jadwal
- Melakukan absensi

Class

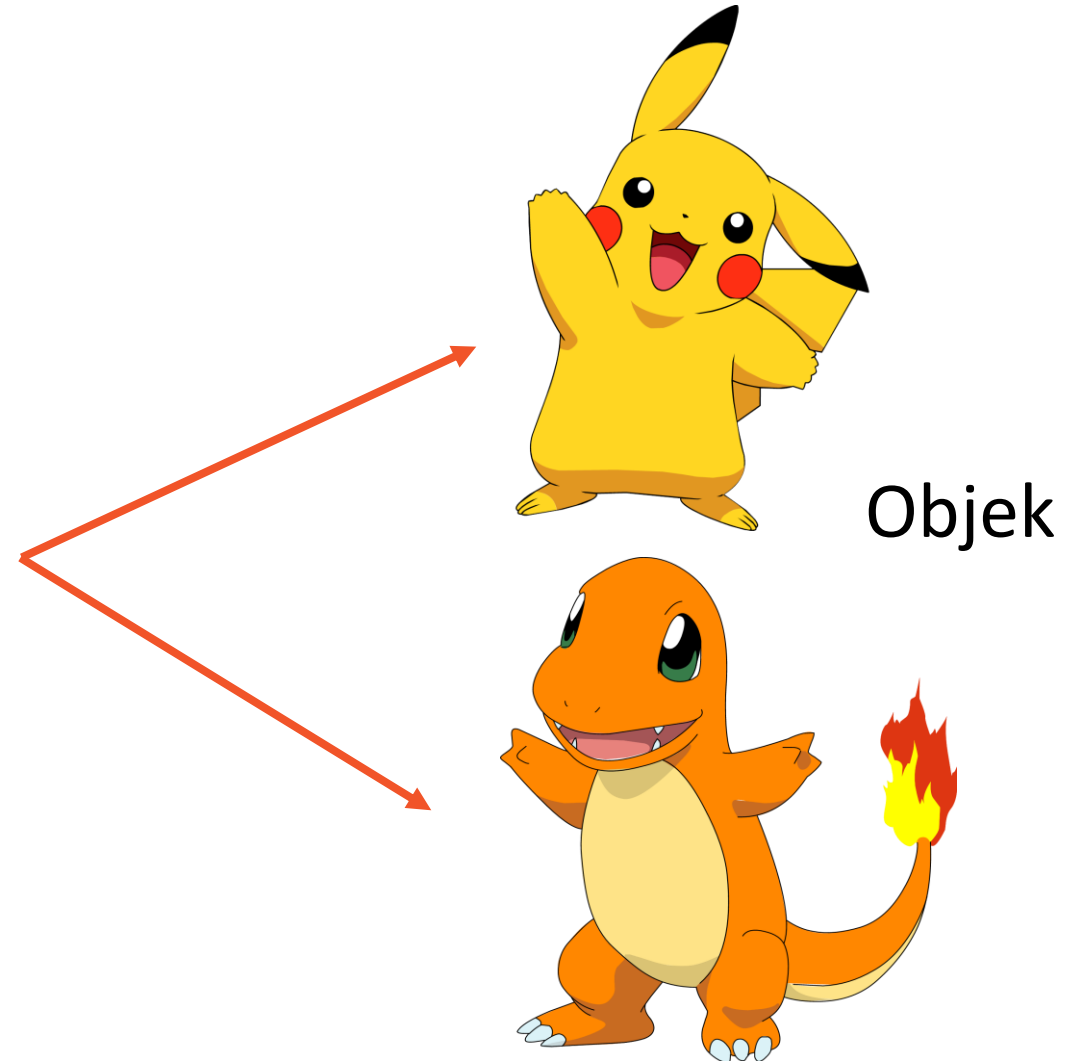


- Setiap **object** pasti berawal/dibuat dari suatu rancangan/ desain/ template/ class
- Class merupakan **template** untuk membuat object
- Proses pembuatan objek dari suatu class disebut **instansiasi**
- **Objek tidak akan ada jika tidak ada class**, dan class tidak akan bisa digunakan jika belum dibuat objek nya.

Analogi class → Who's that Pokémon?



Cetakan/Template/Class



Class vs. Objek



	Class	Object
Penjelasan	Masih berupa rancangan/template/desain/blueprint	Objek nyata/riil yang sudah dibentuk dari suatu class
Sifat	Umum	Spesifik
Contoh 1	<ul style="list-style-type: none">• Mahasiswa• Dosen• Matakuliah	<ul style="list-style-type: none">• Mahasiswa 1, Mahasiswa 2, dst• Dosen A, dosen B, dst• Matakuliah PBO, matakuliah basis data

Apakah class memiliki atribut dan method?

- Ya, Class juga memiliki atribut dan method, sebagaimana objek juga memilikinya.
- Bedanya, jika atribut dan method yang di objek sudah riil, sudah bisa diberi nilai tertentu. Sedangkan atribut dan method di class masih berupa rancangan.

Implementasi Class (Java)

- Format Deklarasi Class:

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

- Contoh:

```
class Sepeda{  
  
}
```

Atribut

- Atribut : data yang dimiliki oleh object atau class
- Nama atribut biasanya ditandai dengan **kata benda**
- Contoh : Atribut yang dimiliki oleh mahasiswa antara lain nim, nama, IPK, alamat

Implementasi Atribut (Java)

- Format Deklarasi Atribut :

```
tipeData namaAtribut;
```

- Contoh:

```
String nama;  
float ipk;  
int umur;  
boolean statusLulus;
```


Method

- Method : merupakan proses/tingkah laku/fungsi yang bisa dilakukan oleh suatu objek
- Method digunakan oleh objek untuk berinteraksi dengan objek yang lain
- Nama method biasanya ditandai dengan **kata kerja**
- Contoh : method melihat KHS, mengerjakan ujian dari objek mahasiswa

Implementasi Method (Java)

- Format Deklarasi Method:

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

- Contoh:

```
void tampilStatus(boolean st) {  
    if(st==true)  
        System.out.println("Lulus");  
    else  
        System.out.println("Belum Lulus");  
}
```

Instansiasi

- Instansiasi → Proses pembuatan objek dari suatu class
- Ditandai dengan kata kunci **new**
- Contoh:

```
Sepeda spd = new Sepeda ();
```

Mengakses Atribut dan Method Suatu Objek

- Setelah objek terbentuk, selanjutnya atribut dan method yang dimiliki objek tersebut bisa mulai digunakan atau diakses.
- Cara mengakses atribut:

```
namaObjek.namaAtribut = nilai;
```

- Cara mengakses method:

```
namaObjek.namaMethod();
```

Contoh:

```
spd.kecepatan = 10;  
System.out.println(spd.kecepatan);  
spd.tampil();
```

Contoh Pembuatan Class: Sepeda

```
1 public class Sepeda { | Deklarasi Class
2     float kecepatan;
3     int gear; | Atribut
4
5     void tambahKecepatan(float a) { | Method
6         kecepatan=kecepatan+a;
7         sesuaikanGear();
8     }
9     void kurangiKecepatan(float a) {
10        kecepatan=kecepatan-a;
11        sesuaikanGear();
12    }
13    void sesuaikanGear() {
14        if(kecepatan<10){
15            gear=1;
16        }else if(kecepatan>=10 && kecepatan<20){
17            gear=2;
18        }else if(kecepatan>=20 && kecepatan<40){
19            gear=3;
20        }else{
21            gear=4;
22        }
23    }
24    void tampil() {
25        System.out.println("Kecepatan="+kecepatan+", Gear="+gear);
26    }
27 }
```

Contoh Pembuatan Objek: Sepeda



```
1 public class SepedaMain {  
2     public static void main(String[] args) {  
3         Sepeda spd = new Sepeda();  
4         spd.kecepatan = 10;  
5         spd.tambahKecepatan(5);  
6         spd.tampil();  
7     }  
8 }
```

Instansiasi

Memberikan nilai
atribut dari objek spd

Menjalankan method dari objek spd

Konstruktor (*Constructor*) #1

- Konstruktor → Method **istimewa** yang digunakan saat **pembuatan objek (instansiasi)**
- Istimewa:
 - Nama method sama dengan nama class
 - Tidak memiliki tipe data method
 - Hanya bisa dijalankan/dipanggil pada proses instansiasi
 - Bisa memiliki parameter

Konstruktor (*Constructor*) #2

- Jenis konstruktor
 - Konstruktor Default :
 - konstruktor yang tidak memiliki parameter
 - konstruktor default
 - Konstruktor Berparameter : konstruktor yang memiliki parameter
- Contoh konstruktor default:

```
public Sepeda() {  
}
```

- Contoh konstruktor berparameter

```
public Sepeda(float kecp, int ge) {  
}
```


Konstruktor (*Constructor*) #3

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

```
Sepeda spd = new Sepeda();
```

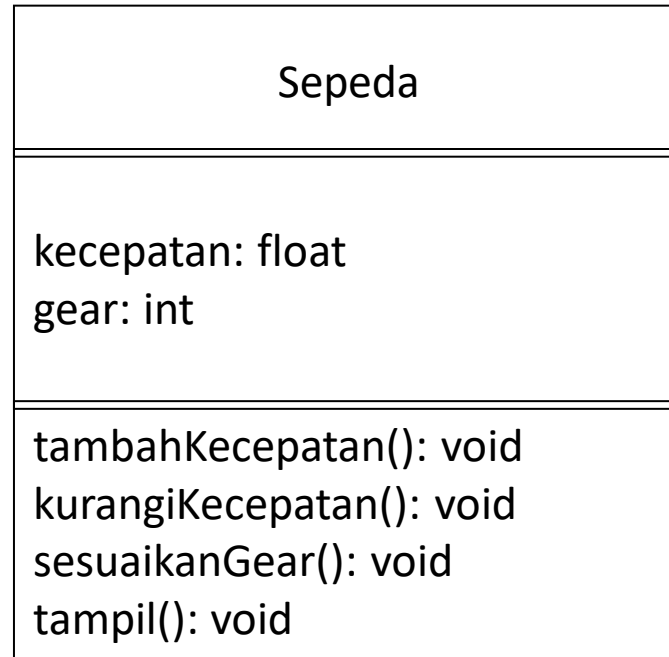
- Contoh penggunaan konstruktor berparameter:

```
Sepeda spd = new Sepeda(15.5, 2);
```

Contoh Konstrukt Pada Class Sepeda

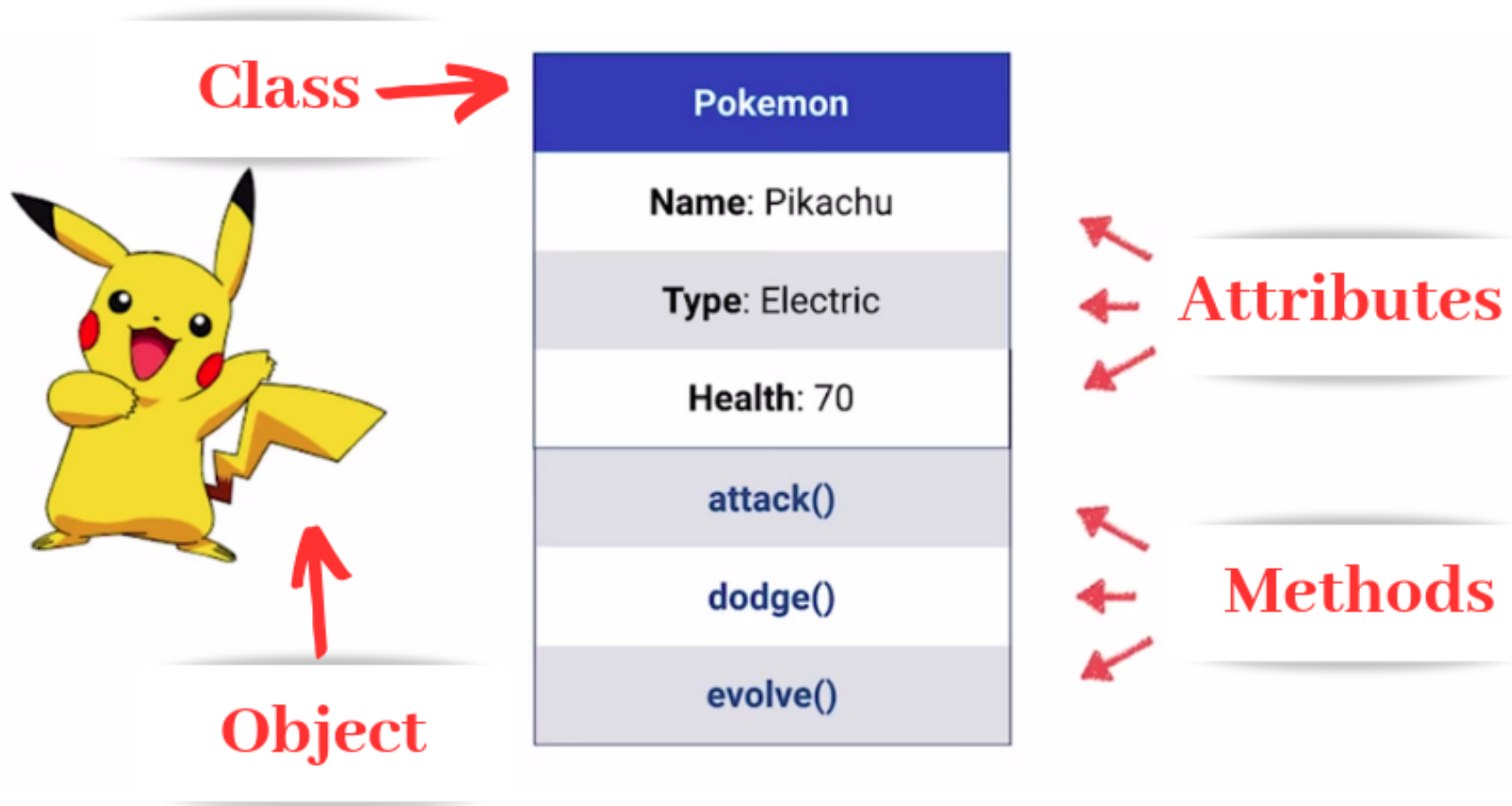
```
1 public class Sepeda {  
2     float kecepatan;  
3     int gear;  
4  
5     public Sepeda(float speed, int gr) {  
6         kecepatan = speed;  
7         gear = gr;  
8     }  
9  
10    void tambahKecepatan(float a) {  
11        kecepatan = kecepatan + a;  
12        sesuaikanGear();  
13    }  
14  
15    void kurangiKecepatan(float a) {  
16        kecepatan = kecepatan - a;  
17        sesuaikanGear();  
18    }  
19  
20    void sesuaikanGear() {  
21        if(kecepatan < 10) {  
22            gear = 1;  
23        } else if (kecepatan >= 10 && kecepatan < 20) {  
24            gear = 2;  
25        } else if (kecepatan >= 20 && kecepatan < 40) {  
26            gear = 3;  
27        } else {  
28            gear = 4;  
29        }  
30    }  
31  
32    void tampil() {  
33        System.out.println("Kecepatan="+kecepatan+", Gear="+gear);  
34    }  
35 }  
36
```

Class → Representasi dalam sebuah desain



CLASS DIAGRAM

Quick Recap



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

Latihan



1. Tentukan sebuah objek yang ada di sekitar anda, tentukan atribut dan method yang dimiliki oleh objek tersebut. Minimal class memiliki 2 atribut dan 2 method selain constructor.
2. Sebuah piramid memiliki alas berbentuk persegi. Buatlah class diagram untuk menggambarkan piramid tersebut (minimal 2 method, misal untuk menghitung volume)
3. Rancanglah class diagram untuk pasien rumah sakit.

