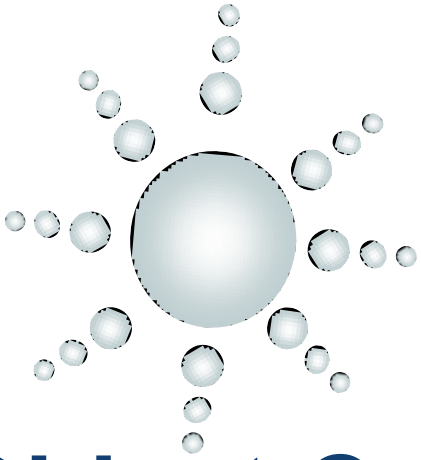
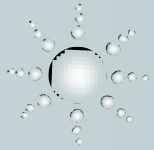


Object Oriented Programming

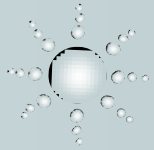
Apa itu OOP?



- Dalam Bahasa Indonesia, OOP diterjemahkan sebagai pemrograman berarah atau berorientasi objek.

- Sebuah metodologi dalam pemrograman yang diciptakan untuk memodelkan kasus-kasus nyata ke dalam sebuah objek.

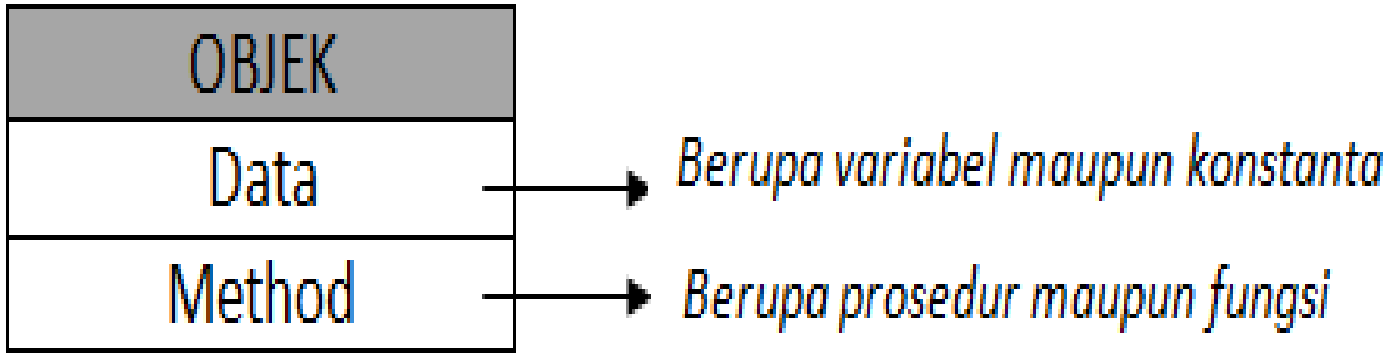
Apa itu Objek?



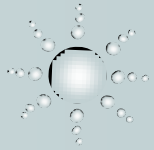
- Sesuatu yang dapat memodelkan atau menyederhanakan permasalahan-permasalahan yang terjadi di dalam dunia nyata.
- Dalam pemrograman, objek adalah sesuatu paket yang merupakan kumpulan data dan perilaku.

Lanjutan...

Dalam pemrograman, **data** dalam objek direpresentasikan dengan **variabel**, sedangkan perilaku direpresentasikan dengan **prosedur atau fungsi** yang disebut **method**.

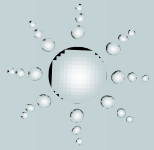


Apa itu Instance?



- Instance adalah contoh atau wujud nyata dari suatu objek.
- Sebagai contoh, apabila terdapat objek manusia, maka si Udin, Joko, Paijo merupakan instance dari objek manusia.

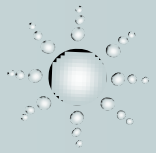
Konsep Dasar OOP



Karakteristik OOP :

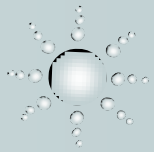
- Abstraksi
- Pembungkusan
- Pewarisan
- Polimorfisme

Abstraksi (Abstraction)



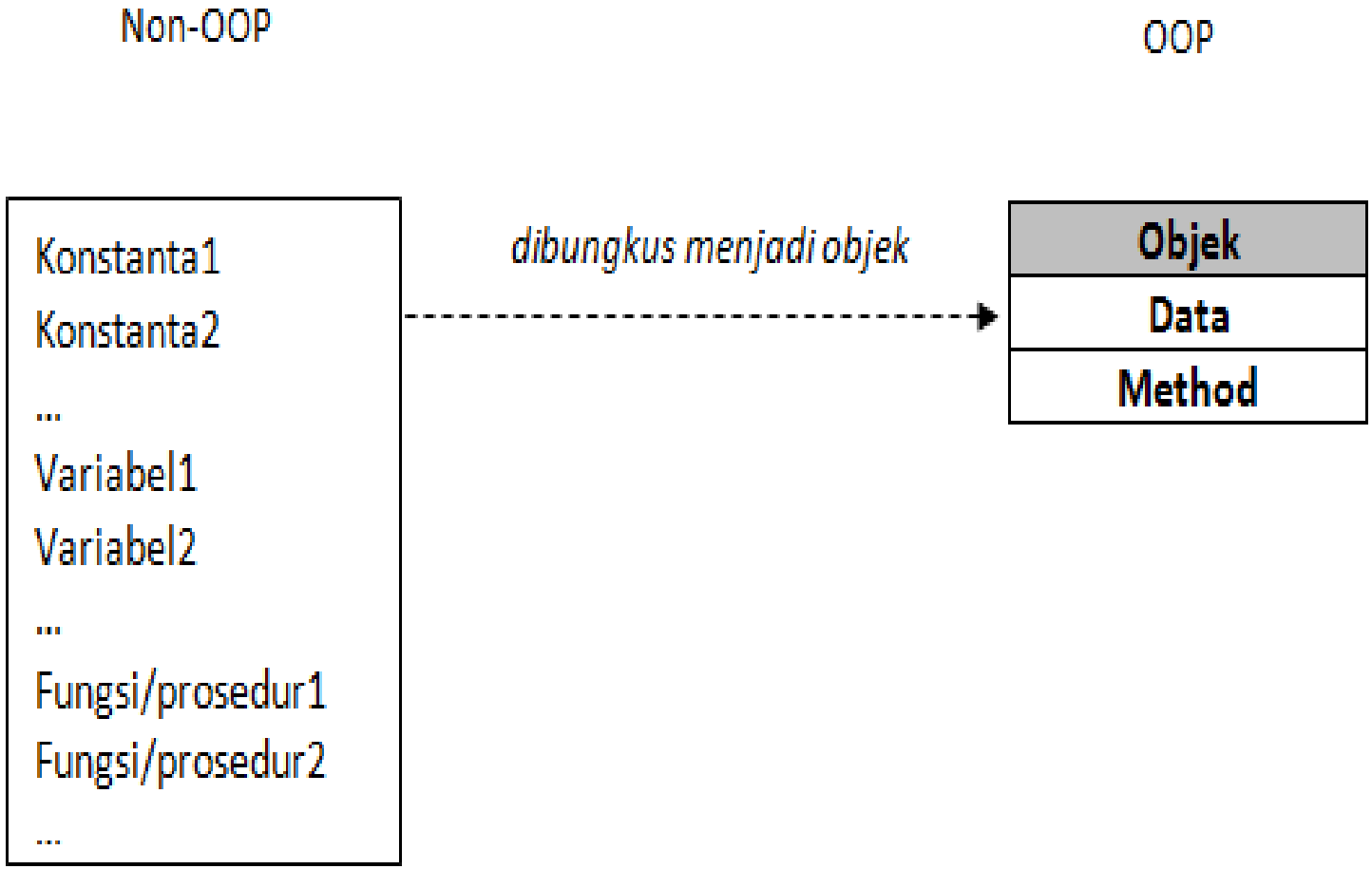
- Abstraksi merupakan ciri yang paling mendasar dari OOP.
- Suatu proses penyembunyian kerumitan (pengabstrakan) yang terjadi dalam suatu objek sehingga pengguna objek tidak perlu untuk mengetahui detail proses yang dilakukan.

Pembungkusan (Encapsulation)

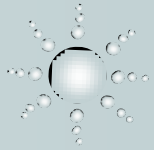


- Ciri kedua dari OOP adalah adanya pembungkusan.
- Artinya data-data dan method akan dibungkus menjadi paket objek yang merupakan satu-kesatuan sehingga dapat bekerjasama dalam melaksanakan tugas-tugas pemrograman tertentu

Lanjutan...


Non-OOP
OOP
Konstanta1
Konstanta2
...
Variabel1
Variabel2
...
Fungsi/prosedur1
Fungsi/prosedur2
...
dibungkus menjadi objek
Objek
Data
Method

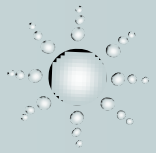
Pewarisan (Inheritance)



- Sebuah objek dapat diturunkan menjadi objek baru lainnya, sehingga objek baru tersebut akan mewarisi sifat dari objek induknya

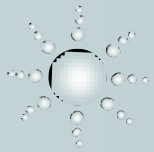
- Objek induk disebut ***base class*** atau ***anchestor class***, objek turunannya disebut ***derived class*** atau ***descendent class***

Polimorfisme (Polymorphism)



- Kegiatan mengungkap beberapa hal yang berbeda melalui satu cara yang sama.
- Misalkan terdapat sebuah objek manusia, yang diturunkan menjadi dua buah objek baru yaitu tentara dan mahasiswa.

Mendeklarasikan Objek



- Untuk mendeklarasikan objek dalam Pascal digunakan kata kunci ***object***.

- Pendeklarasian objek dilakukan di bagian ***type***.

```
NamaObjek = object
   Deklarasi_data;
   …
   Deklarasi_prosedur_atau_fungsi;
   …
end.
```

Lanjutan...

- Method berupa prosedur

```
procedure NamaObjek.NamaMethod (daftar_parameter);
begin
    {Kode yang akan dituliskan}
end;
```

- Method berupa fungsi

```
function NamaObjek.NamaMethod (daftar_parameter):
tipe data;
begin
    {Kode yang akan dituliskan}
    NamaMethod := nilai_balik;
end;
```

Lanjutan...

```
type
   TTitik = object
      x, y: real;
      procedure SetKoordinat(absis, ordinat: real);
      procedure GetKoordinat(var absis, ordinat: real);
   end;

{bagian implementasi}
procedure TTitik.SetKoordinat(absis, ordinat: real);
begin
   x := absis;
   y := ordinat;
end;

procedure TTitik.GetKoordinat(var absis, ordinat: real);
begin
   absis := x;
   ordinat := y;
end;

{program utama}
begin
   ...
end.
```

Lanjutan...

Dari kode diatas terlihat bahwa method *SetKoordinat* dan *GetKoordinat* merupakan milik dari objek TTitik sehingga saat implementasi harus dituliskan :

```
procedure TTitik.SetKoordinat(absis, ordinat: real);
procedure TTitik.GetKoordinat(var absis, ordinat: real);
```

Lanjutan...

Cara mengakses data atau method yang terdapat dalam sebuah objek sama seperti mengakses field dalam sebuah record, yaitu menggunakan operator titik.

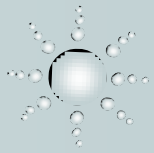
```
var
  A: TTitik;    {Mendeklarasikan instance A yang bertipe TTitik}

begin
  A. SetKoordinat (5.0, 3.0);    {Mengakses method SetKoordinat}
  …
end.
```

Tingkat Akses

- Tingkat akses digunakan untuk memberikan batasan-batasan pihak luar untuk mengakses data-data yang terdapat di dalam sebuah objek.

- Pihak luar disini adalah berupa objek lain maupun bagian lain program yang berada di luar objek. Dalam OOP dikenal tiga buah jenis tingkat akses yaitu private, protected dan public.

Private



- Data maupun method yang berada dalam tingkat akses ini hanya dapat diakses oleh kelas itu sendiri.
- Ini artinya objek lain dan juga lingkungannya tidak diperkenankan untuk mengakses data atau method tersebut.

Lanjutan...

```
program ContohPrivate;

uses crt;

type
   TTitik = object
   private
     x, y: integer;
     procedure SetXY(absis, ordinat: integer);
     procedure GetXY(var absis, ordinat: integer);
   end;

procedure TTitik.SetXY(absis, ordinat: integer);
begin
   x:= absis;
   y:= ordinat;
end;

{Program utama}
var
   A: TTitik;
   bil1, bil2: integer;
begin
   A.SetXY(2, 5)          {BENAR, akses private dari objek yang berada dalam 1 unit}
   A.GetXY(bil1, bil2);  {BENAR, akses private dari objek yang berada dalam 1 unit

   writeln('Nilai x di dalam objek A :', bil1);
   writeln('Nilai y di dalam objek A :', bil2);
   readln;
end.
```

Protected

- Data maupun method pada tingkat akses ini dapat diakses oleh objek itu sendiri dan juga oleh objek-objek turunannya.
- Namun di dalam pascal belum mendukung adanya tingkat akses protected.

Lanjutan...

```
type
   TTitik = class        {Dalam Delphi kata kunci object diganti dengan class}

   private
     x, y: integer;
   protected
     procedure SetXY(absis, ordinat: integer);
     procedure GetXY(var absis, ordinat: integer);
   end;
...
```

Public

- Pada tingkat akses ini data dan method dapat diakses secara publik oleh bagian program manapun, baik oleh objek itu sendiri, objek turunannya maupun oleh lingkungan luar objek yang berada di dalam program.

Lanjutan...

```
type
   TTitik = object
   private
      x, y: integer;
   public
      procedure SetXY(absis, ordinat: integer);
      procedure GetXY(var absis, ordinat: integer);
   end;
...
```

Thank You!