

**TUGAS Modul I Pengantar Pemrograman Berorientasi Objek**

**MATA KULIAH: PEMBELAJAR BERBASIS OBJEK**

Dosen Pengampu:

**Adam Bachtiar, S.Kom., M.MT**

Acc f.  
12/16/25



Catatan :

(1) Perbaiki struk  
Penulisan laporan  
Kode - Output -  
Penjelasan.

(2) Judul - Cover

"PRAKTIKUM MODUL - 1"  
PBO

Dibuat oleh:

Ruhasdi (24241016)

(3) tidak ada margin  
di laporan .

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI**

**FAKULTAS SAINS, TEKNOLOGI DAN TERAPAN**

**UNIVERSITAS PENDIDIKAN MANDALIKA**

**TAHUN 2025**

## Materi: Praktikum 1 – Class dan Object

```
praktikum 1 - class dan objek.py
1  class Hero:
2      pass
3
4  # proses penciptaan objek
5  hero1 = Hero()
6  hero2 = Hero()
7
8  hero1.nama = "Goku"
9  hero2.nama = "Naruto"
10 |
11 hero1.power = 100
12 hero2.power = 200
13
14 print(hero1.nama)
15 print(hero1.power)
16
17 print(hero2.__dict__)
```

Output:

```
D:\Modul 1 PBO> & C:\Users\ASUS\AppData\Local\Programs\Python\Python313\python.exe "D:\Modul 1 PBO\praktikum 1 - Class d
Goku
100
{'nama': 'Naruto', 'power': 200}
D:\Modul 1 PBO>
```

Penjelasan baris:

Materi : Praktikum 1  $\Rightarrow$  Class dan Object

Basis 1  
Ini adalah konstruktur.

Basis 2 - 3  
Class Hero : digunakan untuk membuat sebuah class bersamaan Hero.

Class bisa kamu menggunakan seperti contoh di bawah ini atau template untuk membuat Object.

Pada intinya "tidak ada isi dulu"  $\Rightarrow$  ini hanya placeholder agar kode tidak error walau belum ada isi class-nya.

Basis 5  
konstruktur lagi. Mengandalkan bagian ini akan membuat Object dari class Hero.

Basis 6 - 7  
Di sini kita membuat dua Object dari class Hero yaitu :  
- hero\_1  
- hero\_2

Keduanya punya struktur yang sama karena dari class Hero, tapi masing-masing bisa punya atribut dan nilai berbeda.

Basis 9 - 11  
Kita memambahkan atribut baru bersamaan name ke masing-masing Object.

Object hero\_1 setara dengan punya atribut name dengan nilai "goku"

Object hero\_2 setara dengan punya atribut name dengan nilai "kyarotto".

Basis 13 - 15  
Sama seperti sebelumnya, kita tambahkan atribut baru bernama power :

- o hero\_1 Punya Power = 100
- o hero\_2 Punya Power = 200

Sebarang setting object (hero\_1 dan hero\_2) punya dua atribut :

- name
- power

Batas 17 - 19

Melampirkan 18<sup>0</sup> dari atribut name :

- Rauf (hero\_1.name) akan menampilkan Goku
- Rauf (hero\_2.name) akan menampilkan ~~Naruto~~

Batas 21 - 22

--dict-- adalah atribut bawaan Python yang menyimpan semua atribut milik suatu object dalam bentuk dictionary.

Jadi, --dict-- fungsinya adalah membuat daftar atribut apa saja yang dimiliki object, lengkap dengan nilainya (key = nama atribut, value = nilainya).

Pertikum 2 - Constructor --dict--()

...

## Materi: Praktikum 2 – Constructor \_\_init\_\_()

```
PS D:\Modul 1 PBO> cd C:/Users/ASUS/AppData/Local/Programs/Python/Py  
thon3.8  
File: constructor_init_0.py  
1  class Mahasiswa:  
2      def __init__(self, name, nim, prodi):  
3          self.name = name  
4          self.nim = nim  
5          self.prodi = prodi  
6  
7      def __str__(self):  
8          return f"Nama: {self.name}, NIM: {self.nim}, Prodi: {self.prodi}"  
9  
10     def __dict__(self):  
11         return self.__dict__  
12  
13 mhs1 = Mahasiswa("Adam", "21051214005", "Informatika")  
14 mhs2 = Mahasiswa("Budi", "21051214006", "Sistem Informasi")  
15  
16 print("Nama:", mhs1.name)  
17 print("NIM:", mhs1.nim)  
18 print("Prodi:", mhs1.prodi)  
19 print(mhs1.__dict__)  
20 print()  
21 print("Nama:", mhs2.name)  
22 print("NIM:", mhs2.nim)  
23 print("Prodi:", mhs2.prodi)  
24 print(mhs2.__dict__)
```

Output:

```
PS D:\Modul 1 PBO> & C:/Users/ASUS/AppData/Local/Programs/Python/Py  
thon3.8\python.exe constructor_init_0.py  
Nama: Adam  
NIM: 21051214005  
Prodi: Informatika  
{'name': 'Adam', 'nim': '21051214005', 'prodi': 'Informatika'}  
  
Nama: Budi  
NIM: 21051214006  
Prodi: Sistem Informasi  
{'name': 'Budi', 'nim': '21051214006', 'prodi': 'Sistem Informasi'}  
PS D:\Modul 1 PBO> []
```

Penjelasan perbaris:

Sebaring setting object (hero-1 dan hero-2) punya dua atribut :

- name
- power

Batas 17 - 19

Melanjutkan 18<sup>o</sup> dari atribut name :

- print(hero-1.name) akan menampilkan Goku
- print(hero-2.name) akan menampilkan ~~Naruto~~ Naruto

Batas 21 - 22

--dict-- adalah attribut bawaan Python yang menyimpan semua attribut milik suatu object dalam bentuk dictionary.

Jadi, --dict-- fungsinya adalah membuat daftar atribut apa saja yang dimiliki Object, lengkap dengan nilainya (key = nama atribut, value = nilainya).

Praktikum 2 - Constructor --init--()

Batas 1

Ini adalah komentar

Batas 2

Membuat sebuah class bernama Mahasiswa.  
Class ini seperti contoh ~~base~~ biru untuk membuat Object mahasiswa.

Batas 3

Membuat method constructor bernama .. init ..

Method ini otomatis dijalankan saat object dibuat  
Digunakan untuk mengisi data awal ke dalam Object.

Method ini memerlukan Parameter :

- self → menulis object yang sedang dibuat,
- name, nim, prodi → data yang akan disimpan.

## Basis 4 - 6

Setiap basis ini menyimpan nilai ke atribut object :

- o Self.name = name → menyimpan nama mahasiswa,
- o Self.nim = nim → menyimpan NIM.
- o Self.prodi = prodi → menyimpan program studi.

Jadi, Setiap kali kita membuat object baru, atribut ini otomatis diisi sesuai nilai yang diberikan.

## Basis 6 - 9

Membuat dua object dari class Mahasiswa

- o mhs1 Punya :

- name = "Izum"
- nim = "2051219005"
- prodi = "Informatika"

- o mhs2 Punya :

- name = "Budi"
- nim = "2051219006"
- prodi = "Sistem Informasi"

Saat object dibuat, Python otomatis memanggil \_\_init\_\_ dan mengisi atribut dengan nilai yang diberikan.

## Basis 10 - 13

Menampilkan isi atribut milik object mhs1

## Basis 14

Menampilkan semua atribut milik object mhs1 dalam bentuk dictionary (key = nama atribut, value = nilainya).

## Basis 15

Hanya untuk membuat basis kosong di terminal agar output lebih rapi.

## Basis 16 - 18

Menampilkan atribut milik object mhs2.

## Basis 19

Menampilkan atribut object mhs2 dalam bentuk dictionary.

## Penjelasan Tumbuhan :

Apa itu ~~Self~~ & --init--() ?

- o --init--() adalah method khusus (constructor) di Python.
- o Di Panggil Otomatis Setiap kali Object dibuat dari class.
- o Fungsinya untuk menginisiasi atribut awal milik object

Apa itu Self ?

- o Self mewakili Object itu sendiri
- o Dengan Self, kita bisa membuat atribut milik Object (bukan Variabel Lokal).
- o Misalnya :

Self.name = name

Berarti : buat atribut bernama name didalam Object, dan isi dengan nilai dari Parameter name

Jalan tanpa Self, maka Variabel hanya hidup di dalam Function Saja dan tidak disimpan di Object.

Apa itu --dict-- ?

- o --dict-- adalah atribut bawaan Python yang menyimpan semua atribut Object dalam bentuk dictionary.
- o Isinya Pasangan key : Value → nama- atribut : nilainya.

## Pertikum 3 - Class dan Instance Atribut

Soal 1

## Materi: Praktikum 3 – class dan instance atribut

```
praktikum_3 - class dan instance atribut > ...
  File: Mahasiswa.py

  jumlah_mahasiswa = 0

  def __init__(self, name, nim, prodi):
    self.name = name
    self.nim = nim
    self.prodi = prodi
    Mahasiswa.jumlah_mahasiswa += 1
    print("Membuat Object Mahasiswa dengan nama " + self.name)

  mhs1 = Mahasiswa("Adam", "21051214005", "Teknik Informatika")
  print("Total Mahasiswa : " + str(mhs1.jumlah_mahasiswa))
  mhs2 = Mahasiswa("Budi", "21051214006", "Teknik Industri")
  print("Total Mahasiswa : " + str(mhs2.jumlah_mahasiswa))

  print()
18  print("Total Mahasiswa : " + str(Mahasiswa.jumlah_mahasiswa))
```

Output:

```
Membuat Object Mahasiswa dengan nama Adam
Total Mahasiswa : 1
Membuat Object Mahasiswa dengan nama Budi
Total Mahasiswa : 2

Total Mahasiswa : 2
PS D:\Modul 1 PBO>
```

Penjelasan Perbaris:

## Penjelasan Tambahan :

Apa itu ~~self~~ & `--init--()` ?

- o `--init--()` adalah method khusus (constructor) di Python.
- o Di panggil otomatis setiap kali object dibuat dari class.
- o Fungsinya untuk mengisi atribut awal untuk object

Apa itu `self` ?

- o `self` merupakan object itu sendiri
- o Dengan `self`, kita bisa membuat atribut milik object (bukan variabel lokal).
- o Misalnya :

`self.name = name`

berarti : buat atribut bernama `name` didalam object, dan isi dengan nilai dari parameter `name`

Jika tanpa `self`, maka variabel hanya hidup di dalam fungsi function saja dan tidak disimpan di object.

Apa itu `--dict--` ?

- o `--dict--` adalah atribut bawaan Python yang menyimpan semua atribut object dalam bentuk dictionary.
- o Contoh Pasangan key : Value → nama-atribut : nilainya.

## Pertemuan 3 - Class dan Instance Atribut

Basis 1

ini adalah konsep

Basis 2

Membuat class bernama Mahasiswa  
class ini berfungsi sebagai template untuk membuat object mahasiswa.

Basis 3

ini adalah class attribut, artinya :

- o Nilainya milik bersama semua object.
- o Di simpan di class, bukan dimasing-masing object.

- o bisa diakses melalui :
  - o Mahasiswa, jumlah\_mahasiswa (cara umum)
  - o atau Object. jumlah\_mahasiswa (cara alternatif)

Nilai awalnya adalah 0.

### Basis 6

Membuat constructor -\_\_init\_\_. yang otomatis di jalankan setiap kali membuat object baru.

Mencantum Parameter :

- o self - mengacu pada object yang sedang dibuat.
- o name, nim, Prodi - data yang akan disimpan ke object.

### Basis 7 - 9

Basis ini membuat instance attribut, yaitu :

- o self.name
- o self.nim
- o self.Prodi

Instance attribut adalah data yang spesifik untuk setiap object. Jadi wks1 dan wks2 bisa punya nilai berbeda.

### Basis 10

Setiap kali object baru dibuat, nilai class attribut jumlah\_mahasiswa akan bertambah 1.

Ini sebab setiap kali -\_\_init\_\_() dipanggil, jumlah mahasiswa bertambah.

### Basis 11

Menampilkan pesan setiap kali object baru dibuat.

Contoh output :

"Membuat object Mahasiswa dengan nama Adam"

### Basis 13

Membuat object Person berulang wks1.

Jika basis ini diaksess :

- o -\_\_init\_\_() dijalankan,
- o name, nim, Prodi disimpan ke object
- o jumlah\_mahasiswa naik menjadi 1.

### Batas 1A

Menampilkan jumlah mahasiswa saat ini lewat object mhs1. Karena jumlah-mahasiswa tidaklah class attribute, maka nilainya tetap sama seperti di class (1)

### Batas 1B

Membuat object kedua bersemaunya mhs2. Sekarang jumlah-mahasiswa bertambah menjadi 2.

### Batas 1C

Menampilkan jumlah mahasiswa melalui object kedua (mhs2). nilainya tetap sama 2

### Batas 1B - 1C

Batas kosong untuk memisalkan tampilan, lalu mencetak jumlah mahasiswa lewat nama class langsung. Nilainya juga 2, karena class attribute milik class-nya bukan objectnya.

### PENJELASAN ISTILAH :

Jenis Atribut contoh

Class Atribut Jumlah-mahasiswa

Instance scf.name, scf.nim  
Atribut scf.Prodi

### Penjelasan

Dimiliki bersama oleh semua object. nilainya sama di seluruh class. Dimiliki oleh masing-masing object. nilainya bisa berbeda.

### PERBEDAAN :

mhs1. jumlah-mahasiswa

- Mengakses class attribute merupakan object mhs1
- Python akan mencari di object dulu, karena tidak ada baru ke class.
- Jadi nilainya sama seperti mahasiswa. jumlah-mahasiswa

Mahasiswa jumlah-mahasiswa

- Mengakses class attribute langsung dari class.
- ini cara lebih disarankan karena lebih dekat bahwa bahwa itu atribut milik class.

Keduaanya menampilkan nilai yang sama, tapi cara aksesnya berbeda :

## Materi: Praktikum 4 – instance method

```
1. class Hero:
2.     def __init__(self, name, power):
3.         self.name = name
4.         self.power = power
5.         self.health = 100
6.         self.jumlah_hero += 1
7.
8.     def siapa_hero(self):
9.         return f"Hero {self.name}, memiliki power {self.power}"
10.
11.     def setHealth(self, newHealth):
12.         self.health = newHealth
13.
14.     def getHealth(self):
15.         return self.health
16.
17.     def attack(self, hero):
18.         hero.setHealth(hero.getHealth() - self.power)
19.
20.         print(f"\n{self.name} menyerang {hero.name} dengan power {self.power}")
21.
22.         if hero.getHealth() <= 0:
23.             print(f"\n{hero.name} mati")
24.             Hero.jumlah_hero -= 1
25.
26.             print(f"\nJumlah Hero : {Hero.jumlah_hero}")
27.
28. hero1 = Hero("Asep", 50, 100)
29. hero2 = Hero("Agus", 70, 100)
30.
31. print(hero1.siapa_hero())
32.
33. hero1.setHealth(100)
34. print(hero1.getHealth())
35.
36. print("Jumlah Hero : ", Hero.jumlah_hero)
37.
38. hero3 = Hero("sandy", 90, 20)
39. print(hero3.siapa_hero())
40.
41. print("jumlah Hero : ", Hero.jumlah_hero)
```

```
42.
43. print(hero1.getHealth())
44.
45. hero1.health_pp(10)
46. print(hero1.getHealth())
47.
48. print("Jumlah Hero : ", Hero.jumlah_hero)
49.
50. hero3 = Hero("sandy", 200, 90, 20)
51. print(hero3.siapa_hero())
52.
53. print("jumlah Hero : ", Hero.jumlah_hero)
```

Output:

```
Hero Asep, memiliki power 50
Hero Agus, memiliki power 70
100
110
Jumlah Hero : 2
Hero sandy, memiliki power 90
jumlah Hero : 3
PS D:\Modul 1 PBO>
```

Penjelasan Perbaris:

Kedua nya merupakan nilai yang sama, tetapi cara aksesnya berbeda:

- o Mahasiswa. jumlah - Mahasiswa → tidak jelas berasal dari attribute mana saja, dan tidak mudah ditebak.
- o Mahasiswa. jumlah - Mahasiswa → bisa diambil juga, tetapi kurang disarankan karena bisa mengbingungkan

## ~~Praktikum 3~~ Praktikum 3 - Instance Method

### Poin 1

Membuat class bernama Hero.

### Poin 2

Class Variable  
Komunitas Penjelasan.

### Poin 3

Jumlah adalah class variable, milik bersama semua object fungsi nya untuk menghitung berapa banyak hero yang dibuat

### Poin 5

--init-- adalah fungsi khusus yang otomatis dipeluncurkan saat kamu membuat hero baru.

Fungsi ini seperti konstruktor.

### Poin 6 - 10

Ini adalah instance Variable

Aktivitas, Setiap hero punya data Scud10 - Scud10 :

- o self.name → nama hero
- o self.health → darah hero
- o self.power → kekuatan Serangan
- o self.damage → kekuatan Pertahanan

### Poin 11

Setiap kali object hero baru dibuat, Hero. jumlah akan berubah :

### Poin 13 - 14

Method ini menggunakan teknik (return) yang menyebutkan nama hero dan kekuatannya.

Rumus : Panjang x lebar  
Hasil di tambahkan pada satuan satuan

Rumus 13 - 14

Membuat Method untuk menghitung keliling Persegi

Panjang

Rumus :  $2 \times (\text{Panjang} + \text{lebar})$

Rumus 17 - 18

Membuat 2 object dari class :

- o Pp 1 dengan Panjang 10 dan lebar 5
- o Pp 2 dengan Panjang 7 dan lebar 3

Setiap Object Punya dulu Scoping

Rumus 21 - 26

Menampilkan hasil dari Object Pp 1 :

- o Cetak nilai Panjang dan lebar
- o Panggil fungsi hitung\_luas()  $\rightarrow$  tampilkan hasilnya
- o Panggil fungsi hitung\_keliling()  $\rightarrow$  tampilkan hasilnya

Sama seperti di atas, tapi untuk object Pp 2  
In aktivitas bukan bukan supaya tampilan rapi

### Kesimpulan

- o Class Persegi Panjang:  $\rightarrow$  membuat catatan object
- o --luas--()  $\rightarrow$  mengisi dulu nilai (Panjang & lebar)
- o Scop. Panjang, Scop. lebar  $\rightarrow$  instance attribute (milik tiap object)
- o hitung\_luas() & hitung\_keliling()  $\rightarrow$  method (fungsi dalam class)
- o Pp 1, Pp 2  $\rightarrow$  object (bentuk nyata dari class)

Bruis 37 - Bruis 38

Menzurah menampilkan nilai rasio terbaik, dan jumlah total hasil yang sudah dibuat.

Bruis 39 - 41

Menzurah dari hasil 1. Siswa yang 20, lalu menampilkan dari hasil kedua hasil.

Bruis 43 - 48

Hasil 1 mengacu hasil 2, lalu ditampilkan hasil sebagian dan sisanya dihasil hasil 2

Tugas Laporan

## Materi: Latihan soal modul 1

```
Latihan Soal Modul 1 > ...
  1 class PersegiPanjang:
  2     def __init__(self, panjang, lebar):
  3         self.panjang = panjang
  4         self.lebar = lebar
  5
  6     def luas(self):
  7         return self.panjang * self.lebar
  8
  9     def keliling(self):
 10         return 2 * (self.panjang + self.lebar)
 11
 12 # Program utama
 13 if __name__ == "__main__":
 14     # Input dari pengguna
 15     panjang = float(input("Masukkan panjang persegi panjang: "))
 16     lebar = float(input("Masukkan lebar persegi panjang: "))
 17
 18     # Membuat objek
 19     persegi = PersegiPanjang(panjang, lebar)
 20
 21     # Menghitung dan menampilkan hasil
 22     print(f"Luas persegi panjang: {persegi.luas()}")
 23     print(f"Keliling persegi panjang: {persegi.keliling()}")
```

## Output:

```
Masukkan panjang persegi panjang: 100
Masukkan lebar persegi panjang: 50
Luas persegi panjang: 5000.0
Keliling persegi panjang: 300.0
PS D:\Modul 1 PBO> █
```

## Penjelasan perbaris:

### Ruang 37 - Ruang 38

Mengambil data dari screen dan menulis ke dalam file, dan jumlah total hasil yang sudah dibuat.

### Ruang 39 - 41

Mengambil data dari hasil 1 dan menyatukannya dengan hasil yang sudah dibuat.

### Ruang 43 - 45

Hasil 1 mengacu pada hasil 2, lalu ditambahkan hasil sebelumnya dan siapa yang membuat hasil 2

### Tugas Laporan

#### Ruang 1

Ini adalah komentar, fungsi ini hanya untuk mempermudah keterangannya pada kode. Tidak dijalankan oleh program

#### Ruang 2

Membuat class berdasarkan persyaratan  
class adalah cetakan / blueprint untuk membuat object.  
Jika ada class ini akan menyimpan data persyaratan.  
Isinya, serta rumus luas dan keliling

#### Ruang 4 - 6

Ini adalah constructor ( --init-- ) yang fungsi yang akan otomatis jikalau saat kita buat object dari class

- Self : merupakan object itu sendiri
- Panjang dan lebar : Parameter yang kita gunakan saat membuat object
- Self.Panjang dan Self.lebar - instance attribute, artinya variable milik masing-masing object

Contoh

Kalau kita buat PR 1      • Persyaratan (10,5)  
Maka PR 1. Panjang      • 10 dan PR 1.lebar = 5

#### Ruang 9 - 10

Method (fungsi dalam class) untuk menhitung luas persegi panjang

Pjars 16 - 17

Method ini mengambil dari health Schausatz nilai up.  
Contoh : hero 1. healthUp(20) → darah berjumlah 20.

Pjars 19 - 20

Method ini mengambil return jumlah darah saat ini

Pjars 22

Method baru dengan parameter dan return.  
Komutasi Parameter Method berikutnya.

Pjars 23 - 28

Method ini diperlukan untuk menyerang hero lain, mencampur  
Parameter musuh (yang juga object Hero)

- menghitung damage = Self.Power - musuh.armor,
- jika damage negatif, di-set 0 (ygnt tidak  
jadi healing),
- mengurangi musuh.health sebesar damage  
(Side-effect)
- (ini return string yang menjelaskan hasil serangan).

Pjars 30

Membuat object  
Komutasi

Pjars 31 - 32

hero 1 = Hero(...); hero 2 = Hero(...)

Membuat dua object Hero. Saat dibuat..-nya.. dipanggil dan  
hero. jumlah berjumlah tipe kali.

Pjars 34 -

Menulis method

Komutasi

Pjars 35

Menampilkan size hero 1 dan berapa Power-nya

Pjars 36

Mengubah nilai armor hero 1 menjadi 10