



# **DASAR PEMROGRAMAN**

**INF1004AA**

**LAPORAN PRAKTIKUM 11:**

## **Dasar Pemrograman Berorientasi Objek**

Oleh :

*INDRIAN*

*1911102441003*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
Samarinda, 21 Desember 2019

# Laporan Praktikum 11: Dasar Pemrograman Berorientasi Objek

---

## **Pokok Bahasan:**

- ❖ Kelas
- ❖ Object
- ❖ Method

## **Tujuan Pembelajaran:**

- ✓ Mengimplementasikan dasar-dasar pemrograman berorientasi objek pada Python

## 1. Kelas

### Percobaan & Latihan 11.1:

```
praktikum11 > code > latihan11.py > ...
1  # Class Worker
2  # Merepentasikan Object Perkerja
3  class Worker:
4
5      # Total Object
6      totalWorkerCount = 0
7
8      # Constuctor
9      # Inititialize Object
10     # name mendefisikan nama perkerja
11     # salaray mendefisikan gaji gaji
12     def __init__(self, name, salary):
13
14         self.name = name
15         self.salary = salary
16         Worker.totalWorkerCount += 1
17
18     # Print Worker Object Count
19     def printTotalWorker(self):
20
21         print("Total worker: %d" % Worker.totalWorkerCount)
22
23     # Print Detail Worker, name salary
24     def printPekerja(self):
25
26         print("Name: ", self.name, ", Salary: $", self.salary)
27
```

## 2. Objek

### Percobaan & Latihan 11.2:

```
praktikum11 > code > latihan11.py > ...
1  # Class Worker
2  # Merepentasikan Object Perkerja
3  class Worker:
4
5      # Total Object
6      totalWorkerCount = 0
7
8      # Constuctor
9      # Inititiallize Object
10     # name mendefisikan nama perkerja
11     # salaray mendefisikan gaji gaji
12     def __init__(self, name, salary):
13
14         self.name = name
15         self.salary = salary
16         Worker.totalWorkerCount += 1
17
18     # Print Worker Object Count
19     def printTotalWorker(self):
20
21         print("Total worker: %d" % Worker.totalWorkerCount)
22
23     # Print Detail Worker, name salary
24     def printPekerja(self):
25
26         print("Name: ", self.name, ", Salary: $", self.salary)
27
28     budi = Worker("Budi Setiawan", 1000)
29     alanSuryajana = Worker("Alan Suryajana", 1000)
30
31     budi.printPekerja()
32     alanSuryajana.printPekerja()
33     Worker.printTotalWorker(Worker)
```

### Output:

```
Name: Budi Setiawan , Salary: $ 1000
Name: Alan Suryajana , Salary: $ 1000
Total worker: 2
```

### 3. Method

#### Percobaan & Latihan 11.3:

```
# Removed totalWorkerCount 1, after call del function
def __del__(self):
    Worker.totalWorkerCount -= 1
```

#### Output:

```
Name: Budi Setiawan , Salary: $ 1000
Name: Alan Suryajana , Salary: $ 1000
Total worker: 2
Remove one object
Total worker: 1
```

### 4. Constructor

#### Percobaan & Latihan 11.4:

```
# Constuctor
# Inititalize Object
# @param name mendefisikan nama perkerja
# @param salaray mendefisikan gaji gaji
def __init__(self, name, salary):
    self.name = name
    self.salary = salary
    Worker.totalWorkerCount += 1
```

#### Penjelasan Constuctor

"\_\_init\_\_" adalah metode reserved dalam kelas python. Ia dikenal sebagai konstruktor dalam konsep berorientasi objek. Metode ini dipanggil ketika suatu objek dibuat dari kelas dan itu memungkinkan kelas untuk menginisialisasi atribut suatu kelas.

## **Laporan Resmi:**

### **a. Analisa & Summary**

- **Latihan 11.1**

Kelas adalah cetak biru atau prototipe dari objek dimana kita mendefinisikan atribut dari suatu objek. Atribut ini terdiri dari data member (variabel) dan fungsi (metode).

- **Latihan 11.2**

Objek adalah instansiasi atau perwujudan dari sebuah kelas. Bila kelas adalah prototipenya, dan objek adalah barang jadinya.

- **Latihan 11.3**

Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lain yang digunakan untuk 'mempersiapkan' objek.

- **Latihan 11.4**

Constructor menggunakan `__init__()` merupakan metode menginisialisasi sebuah instance dari class.

### **b. Kesimpulan**

OOP adalah paradigma pemrograman yang menyediakan sarana untuk menyusun program sehingga sifat dan perilaku digabungkan menjadi objek individual.

Repository:

[https://github.com/indrian16/PRAKTIKUM\\_DASAR\\_PEMROGRAMAN\\_PYTHON](https://github.com/indrian16/PRAKTIKUM_DASAR_PEMROGRAMAN_PYTHON)