

TUGAS PRAKTIKUM 5
RESUME MODUL 1-4



Oleh :

Antonius Munthe
121140032
PBO RB

TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2023

DAFTAR ISI

PERTEMUAN I	3
A. Pengenalan Bahasa Pemrograman Python dan Dasar Pemrograman Python	3
PERTEMUAN II.....	4
A. OBJEK DAN KELAS DALAM PYTHON (KONSTRUKTOR, SETTER, DAN GETTER)	
.....	4
A. 1. Kelas	4
A. 2. Objek.....	4
PERTEMUAN III	6
A. ABSTRAKSI DAN ENKAPSULASI (VISIBILITAS FUNGSI DAN VARIABEL, RELASI	
ANTAR KELAS)	6
A.1 Konsep Abstraksi pada Pemrograman Berorientasi Objek.....	6
A.2 Konsep Enkapsulasi pada Pemrograman Berorientasi Objek	6
PERTEMUAN IV	7
A. PEWARISAN DAN POLIMORFISME (OVERLOADING, OVERRIDING, DYNAMIC	
CAST)	7

PERTEMUAN I

A. Pengenalan Bahasa Pemrograman Python dan Dasar Pemrograman Python

Python merupakan bahasa pemrograman yang dibuat pada akhir tahun 1980-an oleh Guido Van Rossum di Centrum Wiskunde & Informatica Belanda. Bahasa ini mendukung banyak paradigma pemrograman seperti object-oriented, functional, dan structured. Bahasa Python sangat populer saat ini karena sintaksnya yang mudah dipahami, didukung oleh library/modul yang berlimpah, dan dapat digunakan untuk berbagai macam jenis pemrograman seperti desktop, mobile, CLI, GUI, web, otomatisasi, hacking, IoT, robotika, dan lain sebagainya.

Dalam filosofi inti dari bahasa pemrograman Python, yang disebut The Zen of Python (PEP 20), terdapat prinsip-prinsip seperti beautiful is better than ugly, explicit is better than implicit, simple is better than complex, complex is better than complicated, dan readability counts. Python sangat memperhatikan readability pada kode, sehingga untuk mengimplementasikan filosofi ini, Python tidak menggunakan kurung kurawal ({ }) atau keyword (ex. start, begin, end) sebagai gantinya menggunakan spasi (white space) untuk memisahkan blok-blok kode. Untuk melakukan instalasi Python, pada sistem operasi Linux/Unix secara default Python sudah terinstal. Sedangkan untuk Windows, pengguna dapat mengunduh Python pada link <https://www.python.org/downloads/release/python-3102/> dan melakukan instalasi dengan mengikuti instruksi yang ada.

Pada dasar pemrograman Python, semua perintah yang bisa dieksekusi Python disebut statement. Pada Python, akhir dari sebuah statement adalah baris baru (newline), namun dimungkinkan membuat statement yang terdiri dari beberapa baris menggunakan backslash (\). Python tidak menggunakan kurung kurawal sebagai grouping blok kode, melainkan menggunakan spasi ataupun tab (4 spasi). Kode yang berada di blok yang sama harus memiliki jumlah spasi yang sama di awal.

Variabel merupakan lokasi penyimpanan yang berguna untuk menyimpan suatu data atau suatu nilai. Dalam mendeklarasikan suatu variabel dalam pemrograman, perlu diketahui tipe-tipe data yang berhubungan dengan variabel yang akan dideklarasikan, di antaranya adalah bool, int, float, dan string.

Python memiliki sejumlah operator, seperti operator aritmatika dan operator perbandingan. Operator aritmatika digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya. Sementara itu, operator perbandingan digunakan untuk membandingkan 2 buah nilai, seperti lebih besar dari, lebih kecil dari, sama dengan, dan tidak sama dengan. Hasil perbandingannya adalah True atau False tergantung pada kondisinya.

PERTEMUAN II

A. OBJEK DAN KELAS DALAM PYTHON (KONSTRUKTOR, SETTER, DAN GETTER)

A. 1. Kelas

Kelas atau class pada Python bisa diartikan sebagai sebuah blueprint atau cetakan untuk membuat objek. Dalam kelas, terdapat atribut/property dan method untuk objek yang ingin dibuat. Kelas dapat membuat banyak objek dan harus diimplementasikan menjadi sebuah objek terlebih dahulu, yang disebut dengan instansiasi.

Atribut pada kelas merupakan variabel yang dideklarasikan di dalam kelas dan dimiliki oleh setiap objek yang dibuat menggunakan kelas tersebut. Terdapat dua jenis atribut, yaitu atribut kelas dan atribut objek. Atribut kelas merupakan sifat yang dimiliki oleh sebuah kelas dan juga akan dimiliki oleh setiap objek. Sedangkan atribut objek adalah sebuah atribut dari masing-masing objek. Untuk memanggil atribut pada objek, digunakan syntax objek.atribut.

Selain atribut, elemen lain yang terdapat dalam suatu kelas adalah method. Method adalah fungsi yang terdapat di dalam kelas dan merupakan aktivitas/proses yang dapat dilakukan oleh sebuah objek. Setiap objek yang dibuat menggunakan kelas yang sama akan memiliki method yang sama pula. Untuk memanggil method, digunakan syntax objek.method().

A. 2. Objek

Objek adalah sesuatu yang mewakili kelas. Objek dapat mewakili sebuah kelas saja dan digunakan untuk mempermudah pemanggilan kelas, terutama jika nama kelas terlalu panjang atau kelas tersebut perlu berinteraksi dengan kelas lain.

Magic method adalah method yang diawali dan diakhiri dengan double underscore (dunder). Method ini tidak dipanggil secara langsung, tetapi dipanggil sistem secara internal ketika melakukan suatu operasi, seperti menggunakan operator tambah (add) atau membuat objek (init). Tujuan utama dari penggunaan magic method adalah untuk mengubah sifat bawaan dari suatu objek. Magic method dapat ditemukan di setiap jenis objek dan variabel.

Contoh penggunaan kelas, objek, dan magic method dapat dilihat pada kode program Python berikut:

```
# Contoh penggunaan kelas, objek, dan magic method pada Python
```

```
# Deklarasi kelas
```

```
class Mobil:
```

```
    # Atribut kelas
```

```
    jumlah_roda = 4
```

```
    # Konstruktor untuk membuat kelas Mobil
```

```
    def __init__(self, merk, warna):
```

```
        # Atribut objek
```

```
        self.merk = merk
```

```
        self.warna = warna

# Method
def maju(self):
    print("Mobil", self.merk, "warna", self.warna, "berjalan maju")

# Membuat objek
mobil1 = Mobil("Toyota", "putih")

# Memanggil atribut
print(mobil1.merk)
print(mobil1.warna)
print(mobil1.jumlah_roda)

# Memanggil method
mobil1.maju()

# Magic method
class Bilangan:
    def __init__(self, angka):
        self.angka = angka

    # Magic method untuk operator tambah
    def __add__(self, objek):
        return self.angka + objek.angka

# Membuat objek
bil1 = Bilangan(10)
bil2 = Bilangan(20)

# Menggunakan operator tambah
print(bil1)
```

PERTEMUAN III

A. ABSTRAKSI DAN ENKAPSULASI (VISIBILITAS FUNGSI DAN VARIABEL, RELASI ANTAR KELAS)

Modul ini membahas dua konsep utama dalam pemrograman berorientasi objek yaitu Abstraksi dan Enkapsulasi. Konsep abstraksi adalah cara untuk mengurangi kompleksitas objek dengan memperlihatkan atribut yang esensial dan menyembunyikan detail-detail yang tidak penting dari user. Sedangkan konsep enkapsulasi adalah sebuah metode yang digunakan untuk mengatur struktur kelas dengan cara menyembunyikan alur kerja dari kelas tersebut. Dalam enkapsulasi, kita dapat "menyembunyikan" property dan method dari suatu kelas agar hanya property dan method tertentu saja yang dapat diakses dari luar kelas.

A.1 Konsep Abstraksi pada Pemrograman Berorientasi Objek

Abstraksi adalah konsep yang sangat penting dalam pemrograman berorientasi objek. Tujuannya adalah untuk mengurangi kompleksitas dan menyembunyikan detail yang tidak penting dari pengguna atau user. User hanya perlu mengetahui apa yang objek lakukan, tetapi tidak perlu mengetahui mekanisme di belakang layar bagaimana objek tersebut bekerja. Misalnya, ketika kita mengendarai mobil, kita hanya perlu mengetahui bagaimana cara menyalakan mobil, menghentikan, dan sebagainya, tetapi kita tidak perlu mengetahui mekanisme apa yang terjadi pada mobil ketika mendapat perintah tersebut.

Saat kita mendefinisikan kelas, sebenarnya kita sedang membuat sebuah abstraksi dari suatu objek. Kelas adalah bentuk abstrak atau cetak biru (blueprint) dari suatu objek nyata. Wujud nyata dari suatu kelas dinamakan instance atau objek. Sebagai contoh, "Str" adalah kelas, sedangkan teks "Python" adalah objek.

A.2 Konsep Enkapsulasi pada Pemrograman Berorientasi Objek

Enkapsulasi adalah sebuah metode yang digunakan untuk mengatur struktur kelas dengan cara menyembunyikan alur kerja dari kelas tersebut. Yang dimaksud dengan struktur kelas tersebut adalah property dan method. Dengan konsep ini, kita dapat "menyembunyikan" property dan method dari suatu kelas agar hanya property dan method tertentu saja yang dapat diakses dari luar kelas. Dengan menghalangi akses dari luar kelas, elemen penting yang terdapat dalam kelas dapat lebih terjaga, dan menghindari kesalahan jika elemen tersebut diubah secara tidak sengaja.

Abstraksi adalah cara untuk menyembunyikan informasi yang tidak dibutuhkan, sedangkan enkapsulasi adalah cara menyembunyikan atribut suatu entitas serta metode untuk melindungi informasi dari luar. Untuk membatasi hak akses terhadap property dan method dalam suatu kelas, terdapat 3 jenis access modifier yang terdapat dalam python, yaitu public access, protected access, dan private access.

>> Public Access Modifier: Pada umumnya, ketika kita mendeklarasikan suatu variabel atau method, maka itulah public access modifier. Setiap class, variabel, dan method yang dibuat secara default merupakan public.

>> Protected Access Modifier: Jika suatu variabel dan method dideklarasikan secara protected, maka variabel dan method tersebut

PERTEMUAN IV

A. PEWARISAN DAN POLIMORFISME (OVERLOADING, OVERRIDING, DYNAMIC CAST)

Pewarisan dan polimorfisme merupakan salah satu konsep dasar dalam Pemrograman Berbasis Objek (OOP). Konsep ini memungkinkan kita untuk mengembangkan program dengan lebih efisien, terstruktur, dan mudah dipelihara. Pewarisan atau inheritance memungkinkan kita untuk menurunkan kelas dari kelas lain untuk hirarki kelas yang saling berbagi atribut dan metode. Sementara itu, polimorfisme memungkinkan suatu method untuk bekerja dengan lebih dari satu tipe argumen. Dalam modul ini juga dibahas mengenai overriding, overloading, dan dynamic cast.

Inheritance adalah salah satu konsep dasar dari OOP. Pada inheritance, kita dapat menurunkan kelas dari kelas lain untuk hirarki kelas yang saling berbagi atribut dan metode. Contohnya adalah ketika kita memiliki base class "Animal" dan class "Horse" yang merupakan turunan dari class "Animal". Ini berarti class "Horse" memiliki atribut dan method yang sama dengan class "Animal". Dan objek "Horse" dapat menggantikan objek "Animal" dalam aplikasi.

Selain itu, terdapat juga inheritance identik yang merupakan pewarisan yang menambahkan constructor pada class child sehingga class child memiliki constructornya sendiri tanpa menghilangkan constructor pada class parentnya. Metode ini ditandai dengan adanya class child yang menggunakan constructor dan menggunakan kata kunci super(). Selain itu, pada child class dapat ditambahkan beberapa fitur tambahan baik atribut maupun method sehingga child class tidak identik dengan parent class.

Polymorphism berarti banyak (poly) dan bentuk (morphism), dalam Pemrograman Berbasis Objek konsep ini memungkinkan digunakannya suatu interface yang sama untuk memerintah objek agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda. Polymorphism merupakan kemampuan suatu method untuk bekerja dengan lebih dari satu tipe argumen. Pada bahasa lain (khususnya C++), konsep ini sering disebut dengan method overloading. Pada dasarnya, Python tidak menangani hal ini secara khusus. Hal ini disebabkan karena Python merupakan suatu bahasa pemrograman yang bersifat duck typing (dynamic typing).

Overriding adalah konsep di mana kita dapat menimpa suatu metode yang ada pada parent class dengan mendefinisikan kembali method dengan nama yang sama pada child class. Dengan begitu, method yang ada parent class tidak berlaku dan yang akan dijalankan adalah method yang terdapat di child class.

Metode overloading mengizinkan sebuah class untuk memiliki sekumpulan fungsi dengan nama yang sama dan argumen yang berbeda. Akan tetapi, Python tidak mengizinkan pendeklarasian fungsi (baik pada class ataupun tidak) dengan nama yang sama. Hal itu menyebabkan implementasi overloading pada python menjadi "tricky". Secara umum overloading memiliki beberapa signature, yaitu jumlah argumen, tipe argumen, tipe keluaran dan urutan argumen. Akan tetapi, seperti yang telah dijelaskan python tidak menangani overloading secara khusus. Karena python adalah bahasa pemrograman yang bersifat duck typing (dynamic typing), overloading secara tidak langsung dapat diterapkan