

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
KE - 13

Python Classes and Objects



DISUSUN OLEH :

Restu Wibisono

2340506061

JURUSAN TEKNOLOGI INORMASI

FAKULTAS TEKNIK

UNIVERSITAS TIDAR

2023

LAPORAN
SESUAI NAMA MATA KULIAH PRAKTIKUM



Diisi Mahasiswa Praktikan										
Nama Praktikan	Restu Wibisono									
NPM	2340506061									
Rombel	03									
Judul Praktikum	Python Classes and Objects									
Tanggal Praktikum	22 November 2023									
Diisi Asisten Praktikum										
Tanggal Pengumpulan	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>									
Catatan										

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
Ludfi Arfiani	Imam Adi Nata	

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR
2023

I. Tujuan Praktikum

Adapun tujuan praktikum ini sebagai berikut :

1. Dapat memahami konsep dasar classes dan objects, perbedaan antara class dan object, membuat class sederhana serta menginstansiasikannya menjadi object dan dapat mengakses atribut serta metode object.
2. Dapat menggunakan metode `__init__` untuk inisialisasi object dan memberikan nilai awal atribut object.
3. Dapat manipulasi atribut dan metode dengan mengganti nilai atribut objek dengan menggunakan metode setter dan dapat mengakses nilai atribut objek dengan menggunakan metode getter.

II. Dasar Teori

Dalam pembelajaran pemrograman berorientasi objek (OOP) menggunakan Python, konsep dasar kelas dan objek memegang peran sentral. Suatu kelas dalam Python bukan hanya sekadar kumpulan atribut dan metode; lebih dari itu, kelas berfungsi sebagai blueprint yang menentukan struktur dan perilaku objek. Pemahaman yang mendalam terhadap kelas dan objek memberikan kemampuan kepada pengembang untuk merancang dan membangun sistem yang terorganisir dan mudah dimengerti.

Pertama-tama, perlu dipahami bahwa suatu objek merupakan suatu instansi dari suatu kelas. Dengan kata lain, objek dibuat berdasarkan definisi yang telah diberikan oleh kelas. Proses pembuatan objek ini melibatkan instansiasi kelas menggunakan sintaks `objek = Class()`.

Dalam lingkup praktikum ini, konsep-konsep tersebut dapat diperdalam melalui implementasi dari class `AlatTulis`. Class ini berperan sebagai representasi dari entitas alat tulis, memiliki atribut-atribut seperti `nama`, `stok`, `harga satuan`, dan `harga total`. Metode khusus, yaitu `__init__` atau metode konstruktor, digunakan untuk menginisialisasi objek ketika objek tersebut pertama kali dibuat. Dengan demikian, nilai awal atribut-alat tulis dapat ditetapkan dengan jelas.

Selanjutnya, terdapat metode-metode seperti `setNama`, `setStok`, dan `setHargaSatuan`, yang dikenal sebagai setter methods. Fungsi utama setter

methods adalah mengubah nilai atribut objek dengan aman, memastikan integritas data, dan melakukan perhitungan otomatis jika diperlukan. Penggunaan setter methods ini menggambarkan kemampuan OOP dalam memberikan kontrol dan validasi terhadap perubahan nilai atribut.

Pada tingkat aplikasi yang lebih tinggi, praktikum ini mencakup class `PenjualAlatTulis`, yang menggunakan objek-objek dari class `AlatTulis` sebagai atribut. Melalui metode `totalHarga`, class ini dapat mengakses objek-objek tersebut untuk menghitung total harga dari semua alat tulis yang dijual. Dengan demikian, penerapan class sebagai komponen lainnya menunjukkan bagaimana konsep OOP memfasilitasi pembangunan sistem yang terorganisir dan modular.

Dengan pemahaman konsep dasar seperti class, objek, metode, metode konstruktor, dan setter methods, individu dapat mengaplikasikan pemrograman berorientasi objek dalam Python dengan lebih efektif. Pemahaman ini membantu dalam merancang solusi perangkat lunak yang lebih bersih, modular, dan mudah untuk dipelihara.

III. Metode Praktikum

A. Alat dan bahan

1. Komputer: Pastikan memiliki komputer yang dapat menjalankan Python. Komputer harus memiliki sistem operasi yang kompatibel dengan Python dan spesifikasi yang memadai untuk menjalankan aplikasi Python dengan lancar.
2. Python: Pastikan Python terinstal di komputer. Bisa didapatkan dengan mengunduh Python versi terbaru dari situs resmi Python dan mengikuti petunjuk instalasi terlampir.
3. Editor teks atau IDE: Pilih editor teks atau IDE (lingkungan pengembangan terintegrasi) yang sesuai dengan preferensi. Beberapa opsi populer termasuk PyCharm, Visual Studio Code, Atom, Sublime Text, atau IDLE (bawaan Python). Pastikan memiliki editor teks atau IDE yang mendukung Python dan diinstal di komputer.
4. Modul Latihan: Jika ada modul yang berkaitan dengan praktik pemrograman Python, baca dan pahami materi tersebut sebelum memulai praktik.
5. Koneksi internet (opsional): Jika ingin mencari lebih banyak materi referensi, mengunduh lebih banyak paket atau modul Python, atau mengakses dokumentasi online, pastikan memiliki koneksi Internet yang stabil.

B. Langkah kerja

1. Persiapan:
 - Pastikan telah menginstal Jupyter Notebook di komputer. Notebook Jupyter dapat diinstal menggunakan Anaconda atau pip.
 - Pastikan juga Python diinstal di komputer.
 - Buka Jupyter Notebook dengan menjalankan "jupyter notebook" di terminal atau baris perintah.

2. Buat buku catatan baru:
 - Saat buku catatan Jupyter terbuka di browser, klik tombol "Baru" untuk membuat buku catatan baru.
 - Pilih kernel yang sesuai yaitu. Python 2 atau Python 3.
3. Tulis kodenya:
 - Di buku catatan baru, akan melihat sel kosong yang bisa diisi dengan kode Python.
 - Tulis kodenya sesuai petunjuk pada modul latihan.
4. Jalankan kode:
 - Untuk menjalankan kode di dalam sel, klik tombol "Run" atau tekan "Shift+Enter" pada keyboard.
 - Hasil kode ditampilkan di bawah sel.
5. Periksa hasilnya dan perbaiki:
 - Perhatikan hasil yang ditampilkan setelah menjalankan kode.
 - Jika ada error atau kekeliruan, perbaiki kode yang salah tersebut.
6. Penyusunan Laporan:
 - Di akhir latihan, buat laporan yang memuat tujuan latihan, langkah-langkah yang dilakukan, hasil observasi, analisis dan kesimpulan yang diambil

IV. Hasil dan Analisis

A. Weekly Exercise 1

```
1 # Write your code below and press Shift+Enter to execute
2 class AlatTulis:
3     def __init__(self, nama, stok, hargaSatuan):
4         self.nama = nama
5         self.stok = stok
6         self.hargaSatuan = hargaSatuan
7         self.harga = stok * hargaSatuan
8
9     def setName(self, nama):
10        self.nama = nama
11
12    def setStok(self, stok):
13        self.stok = stok
14        self.harga = self.stok * self.hargaSatuan
15
16    def setHargaSatuan(self, hargaSatuan):
17        self.hargaSatuan = hargaSatuan
18        self.harga = self.stok * self.hargaSatuan
19
20    def getName(self):
21        return self.nama
22
23    def getStok(self):
24        return self.stok
25
26    def getHargaSatuan(self):
27        return self.hargaSatuan
28
29    def getHarga(self):
30        return self.harga
31
32
33 class PenjualAlatTulis:
34     def __init__(self):
35         self.bolpoint = AlatTulis("Bolpoint", 10, 2000)
36         self.pensil = AlatTulis("Pensil", 10, 1000)
37         self.penghapus = AlatTulis("Penghapus", 10, 500)
38
39     def totalHarga(self):
40        return self.bolpoint.getHarga() + self.pensil.getHarga() + self.penghapus.getHarga()
41
42
43 penjual = PenjualAlatTulis()
44
45 print("Nama\t\t\t\t:", penjual.bolpoint.getName())
46 print("Stok\t\t\t\t:", penjual.bolpoint.getStok())
47 print("Harga Satuan\t\t\t:", penjual.bolpoint.getHargaSatuan())
48 print("Harga\t\t\t\t:", penjual.bolpoint.getHarga())
49 print("R", "E", "S", "T", "O", "K")
50 penjual.bolpoint.setStok(20)
51 penjual.bolpoint.setHargaSatuan(2000)
52
53 print("Nama\t\t\t\t:", penjual.bolpoint.getName())
54 print("Stok\t\t\t\t:", penjual.bolpoint.getStok())
55 print("Harga Satuan\t\t\t:", penjual.bolpoint.getHargaSatuan())
56 print("Harga\t\t\t\t:", penjual.bolpoint.getHarga())
57
58 print("Total Harga Semua Barang\t:", penjual.totalHarga())
```

(Gambar 1.1)

1. Buat class dengan nama AlatTulis untuk menampung berbagai method.

2. **__init__ method:** Konstruktor kelas ini digunakan untuk menginisialisasi objek AlatTulis dengan atribut seperti nama alat tulis, stok, dan harga satuan. Selain itu, ada atribut tambahan **harga** yang dihitung sebagai hasil perkalian stok dan harga satuan.
3. **nama, stok, dan hargaSatuan** adalah parameter yang digunakan untuk menginisialisasi atribut-atribut objek.
4. **nama, stok, dan hargaSatuan** adalah parameter yang digunakan untuk menginisialisasi atribut-atribut objek.
5. **setNama:** Mengubah nilai atribut **nama**.
6. **setStok:** Mengubah nilai atribut **stok** dan menghitung ulang nilai atribut **harga** berdasarkan perubahan stok.
7. **setHargaSatuan:** Mengubah nilai atribut **hargaSatuan** dan menghitung ulang nilai atribut **harga** berdasarkan perubahan harga satuan.
8. **getNama, getStok, getHargaSatuan, dan getHarga:** Mengembalikan nilai dari masing-masing atribut.
9. **Buat** class pembuatAlatTulis untuk menampung method dari **__init__** dan **totalHarga**.
10. Pada method **__init__** buat tiga objek AlatTulis (bolpoint, pensil, penghapus) sebagai atribut dari objek PenjualAlatTulis.
11. Pada **totalHarga** berfungsi untuk menghitung dan mengembalikan total harga semua alat tulis yang dijual oleh penjual.
12. Membuat objek penjualan menggunakan kelas PenjualanAlatTulis.

```
Nama : Bolpoint
Stok : 10
Harga Satuan : 2000
Harga : 20000

RESTOK

Nama : Bolpoint
Stok : 20
Harga Satuan : 2000
Harga : 40000
Total Harga Semua Barang : 55000
```


13. print akan membuat output dari program tersebut, yaitu dari `penjual.bolpoint.getNama` akan menampilkan nama dari barang yang akan dibutuhkan.
14. print dari `penjual.bolpoin.getStok` akan menampilkan sisa dari stok bolpoin saat ini, dan seterusnya.
15. Kemudian program akan memperbarui dari stok jika stok diganti menjadi `penjual.bolpoint.setStok(20)` dan mendapatkan harga dari `penjual.bolpoint.setHargaSatuan(2000)`.
16. Kemudian program akan membuat kembali output dan juga ditambah dengan total harga dari semua barang jual (bolpoin, pensil dan penghapus).

V. Kesimpulan

Praktikum ini memberikan pemahaman tentang konsep dasar pemrograman berorientasi objek menggunakan Python. Melalui contoh pembuatan class `AlatTulis` dan penggunaannya dalam class `PenjualAlatTulis`, dapat memahami cara membuat objek, menggunakan metode `__init__` untuk inisialisasi, dan penggunaan metode setter untuk memperbarui nilai atribut objek.

Dengan adanya contoh kasus alat tulis, dapat melihat bagaimana class dan objek dapat direpresentasikan dalam konteks nyata. Selain itu, perhitungan otomatis harga berdasarkan stok dan harga satuan menunjukkan fleksibilitas dan kemudahan dalam penggunaan class dalam pemrograman.

Penerapan metode setter juga memperlihatkan cara untuk mengubah nilai atribut objek dengan aman, sambil melakukan perhitungan tambahan jika diperlukan. Keseluruhan, praktikum ini memberikan dasar yang kuat untuk pemahaman konsep dasar dan penggunaan class dan object dalam bahasa pemrograman Python

VI. Referensi

Fachrie, M., *et-al.* Modul Praktikum: Pemrograman Berorientasi Objek.

Hidayah, A. *et-al.* 2023. *STRUKTUR DATA DENGAN PYTHON*. PT. Sonpedia Publishing Indonesia.

Id, I. D. 2021. *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python* (Vol. 1). Unri Press.

Rahman, S., *et-al.* 2023. *PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK*. Penerbit Tahta Media.