

Program Studi Teknik Informatika Institut Teknologi Sumatera

2024

**Pemrograman Berorientasi Objek**

**Laporan Praktikum**

|  |  |
| --- | --- |
| **Modul :** | **Class & Object** |
| **Nama :** | **Akhdan Arif P.** |
| **NIM :** | **122140099** |
| **Kelas (Kelas Asal) :** | **RA** |

Instruksi sederhana :

* Disarankan kepada **Praktikan Pemrograman Berorientasi Objek** untuk mengeditnya menggunakan Google Docs agar tidak berantakan dan rapi,
* Silahkan mengganti **Nama Modul** baik yang ada pada **Cover** dan **Header** sesuai dengan materi praktikum,
* Gunakan text styling seperti **Heading 1**, **Normal Text** yang telah terformat / Text Style lainnya yang digunakan untuk menjaga estetika laporan,
* Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan kode yang sudah Praktikan buat ke dalam Laporan Praktikum.

# Materi Praktikum

Dalam Python, sebuah class digunakan sebagai cetak biru untuk membuat objek, yang berfungsi untuk mendefinisikan karakteristik dan perilaku dari objek yang akan dibuat. Objek sendiri merupakan contoh konkret dari sebuah class, yang memiliki atribut (variabel) dan metode (fungsi) sesuai dengan definisi dalam class tersebut. Sebagai analogi, bayangkan class sebagai resep untuk membuat kue, dimana resep tersebut menentukan jenis dan cara membuat kue, sedangkan objek adalah kue nyata yang dibuat berdasarkan resep tersebut. Sebagai contoh lain, kita bisa memiliki class "Hewan" dengan atribut seperti jenis dan makanan, serta metode seperti "makan" dan "berjalan". Dengan membuat objek-objek berdasarkan class "Hewan", kita bisa memiliki hewan-hewan dengan berbagai jenis dan perilaku sesuai dengan definisi class tersebut. Dengan konsep class dan objek, kita dapat mengatur dan mengelola data dalam program Python secara terstruktur dan efisien.

# Link Source Code

<https://onlinegdb.com/f8p7_U2-I>

# Source Code

Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan Source Code yang dipindahkan dari text editor anda ke dokumen ini.

class Buku:

    def \_\_init\_\_(*self*, *Nama*, *Genre*, *NamaPengarang*, *TahunTerbit*):

*self*.nama = *Nama*

*self*.genre = *Genre*

*self*.NamaPengarang = *NamaPengarang*

*self*.TahunTerbit = *TahunTerbit*

class DataBuku(Buku):

    def \_\_init\_\_(*self*, *Nama*, *Genre*, *NamaPengarang*, *TahunTerbit*):

        super().\_\_init\_\_(*Nama*, *Genre*, *NamaPengarang*, *TahunTerbit*)

*self*.daftar\_buku = []

    def TambahDataBuku(*self*):

        nama = input("Masukkan Nama Buku: ")

        genre = input("Masukkan Genre Buku: ")

        pengarang = input("Masukkan Nama Pengarang: ")

        tahun = int(input("Masukkan Tahun Terbit: "))

        buku\_baru = Buku(nama, genre, pengarang, tahun)

*self*.daftar\_buku.append(buku\_baru)

    def LihatListBuku(*self*):

        print("Daftar Buku:")

        for buku in *self*.daftar\_buku:

            print(f"Nama: {buku.nama}, Genre: {buku.genre}, Pengarang: {buku.NamaPengarang}, Tahun Terbit: {buku.TahunTerbit}")

    def Modifikasi\_buku\_lama(*self*):

        nama\_buku = input("Masukkan nama buku yang ingin dimodifikasi: ")

        for buku in *self*.daftar\_buku:

            if buku.nama == nama\_buku:

                buku.nama = input("Masukkan nama baru: ")

                buku.genre = input("Masukkan genre baru: ")

                buku.NamaPengarang = input("Masukkan nama pengarang baru: ")

                buku.TahunTerbit = int(input("Masukkan tahun terbit baru: "))

                print("Buku berhasil dimodifikasi.")

                return

        print("Buku tidak ditemukan.")

    def Hapus\_buku\_lama(*self*):

        nama\_buku = input("Masukkan nama buku yang ingin dihapus: ")

        for buku in *self*.daftar\_buku:

            if buku.nama == nama\_buku:

*self*.daftar\_buku.remove(buku)

                print("Buku berhasil dihapus.")

                return

        print("Buku tidak ditemukan.")

    def KeluarProgram(*self*):

        print("Keluar dari program.")

buku\_mula\_mula = int(input("Masukkan jumlah buku mula-mula: "))

data\_buku = DataBuku("", "", "", "")

for i in range(buku\_mula\_mula):

    print(f"Masukkan data buku ke-{i+1}:")

    data\_buku.TambahDataBuku()

while True:

    print("\nToko Buku Kelontong Yoimiya:")

    print("1. Tambah Data Buku")

    print("2. Lihat List Buku")

    print("3. Modifikasi Buku Lama")

    print("4. Hapus Buku Lama")

    print("5. Keluar Program")

    pilihan = input("Masukkan nomor inputan (1/2/3/4/5): ")

    if pilihan == '1':

        data\_buku.TambahDataBuku()

    elif pilihan == '2':

        data\_buku.LihatListBuku()

    elif pilihan == '3':

        data\_buku.Modifikasi\_buku\_lama()

    elif pilihan == '4':

        data\_buku.Hapus\_buku\_lama()

    elif pilihan == '5':

        data\_buku.KeluarProgram()

        break

    else:

        print("Pilihan tidak valid. Silakan pilih kembali.")

# Dokumentasi Hasil Running

|  |
| --- |
|  |
|  |
|  |
| **Gambar 1. Outbut Program Class Buku** |

Code di atas merupakan implementasi sederhana dari sistem manajemen buku dalam bahasa Python. Program ini memungkinkan pengguna untuk menambah, melihat, memodifikasi, dan menghapus data buku melalui antarmuka teks. Dengan menggunakan class `Buku` sebagai cetak biru untuk membuat objek buku, program dapat menyimpan informasi tentang setiap buku, termasuk nama, genre, nama pengarang, dan tahun terbitnya. Class `DataBuku` berperan sebagai pengelola data yang memungkinkan operasi-operasi seperti menambah, melihat, memodifikasi, dan menghapus buku dari daftar. Pengguna dapat memilih berbagai operasi melalui menu yang ditampilkan di layar, dan program akan menjalankan fungsi yang sesuai sesuai dengan input pengguna. Dengan menggunakan struktur class dan method, program ini memberikan pengguna kontrol yang mudah dan efisien dalam mengelola data buku.