

**LAPORAN C PRAKTIKUM ALGORITMA PEMROGRAMAN**  
**KE - 14**



**DISUSUN OLEH :**

Restu Wibisono (2340506061)

Sunny Alodia W (2320506057)

Muhamad Naveed (2320506040)

**JURUSAN TEKNOLOGI INORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS TIDAR**

**2023**

**LAPORAN**  
**ALGORITMA PEMROGRAMAN**



<b>Diisi Mahasiswa Praktikan</b>										
Nama Praktikan	Restu Wibisono, Sunny Alodia Widyadhana, Muhamad Naveed Sharaheel									
NPM	2340506061, 2320405057, 2320506040									
Rombel	03									
Judul Praktikum	Tugas Akhir Algoritma Pemrograman									
Tanggal Praktikum	06 Desember 2023									
<b>Diisi Asisten Praktikum</b>										
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 menggunakan library eksternal untuk memperluas fungsional program.
2. Mampu menggunakan konsep pemrograman berorientasi objek (OOP) dengan membuat kelas.
3. Untuk mengimplementasikan function dalam bentuk metode-metode di dalam kelas.
4. Menguasai pengendalian program menggunakan nested if dan nested loop.

## **II. Dasar Teori (minimal 3 halaman)**

Penggunaan library berfokus pada fungsionalitas yang diperluas, misalnya perpustakaan tabel digunakan untuk mengatur data dalam format tabel, sementara colorama digunakan untuk memberikan dimensi visual pada output konsol, yang dapat meningkatkan pengalaman pengguna.

Selanjutnya, pendekatan konsep class digunakan untuk membentuk struktur program yang terstruktur. Dalam beberapa kasus, kelas Kasir dibuat untuk mewakili objek kasir dengan atribut seperti cart dan menu, serta beragam metode seperti lihatMenu, masukkanKeranjang, lihatKeranjang, dan beli.

Penggunaan fungsi dicapai dengan mendefinisikan berbagai metode di dalam kelas, yang membantu memecah tugas suatu program menjadi komponen yang lebih terkelola dan modular. Fungsi-fungsi antara lain logika untuk melihat menu, menambahkan item ke dalam keranjang, menampilkan isi keranjang, dan melaksanakan proses checkout.

Prinsip yang ditetapkan di bawah ini digunakan dalam bentuk list untuk menyimpan data menu dan isi keranjang belanja. Pendekatan ini memberikan struktur data yang efisien untuk mengelola kumpulan informasi.

Terakhir, metode nested if dan nested loop digunakan dalam logika program untuk mengatasi pilihan menu dan memastikan validitas input dari

pengguna. Pernyataan kondisional bersarang (nested if) digunakan untuk mengevaluasi opsi menu yang dipilih, sedangkan nested loop digunakan secara tidak langsung dalam struktur pengulangan menu interaktif, memberikan kemungkinan pengguna untuk melakukan tindakan yang berbeda.

### **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. 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

```
from tabulate import tabulate
from colorama import init, Fore

class Kasir:
    def __init__(self):
        self.cart = []
        self.menu = [
            {"No": 1, "Item": "Kopi", "Price": 15000},
            {"No": 2, "Item": "Teh", "Price": 10000},
            {"No": 3, "Item": "Roti", "Price": 8000},
            {"No": 4, "Item": "Nasi Goreng", "Price": 25000},
            {"No": 5, "Item": "Kwetiaw", "Price": 15000},
            {"No": 6, "Item": "Es Teh Anget", "Price": 10000},
            {"No": 7, "Item": "Gado-gado", "Price": 8000},
            {"No": 8, "Item": "Nasi Putih", "Price": 25000},
        ]

    def lihatMenu(self):
        if not self.menu:
            print("Menu kosong.")
            return

        headers = self.menu[0].keys()
        print(tabulate(self.menu, headers="keys", tablefmt="fancy_grid", colalign=("center", "center", "center")))

    def masukkanKeranjang(self, itemId, quantity):
        for item in self.menu:
            if item["No"] == itemId:
                totalHarga = item["Price"] * quantity
                self.cart.append({"Item": item["Item"], "quantity": quantity, "totalPrice": totalHarga})
                print(f"{quantity} {item['Item']} ditambahkan ke keranjang.")
                return
        print("Item tidak ditemukan.")

    def lihatKeranjang(self):
        if not self.cart:
            print("Keranjang kosong.")
            return

        headers = ["Item", "Quantity", "Total Price"]
        cartData = [[item["Item"], item["quantity"], item["totalPrice"]] for item in self.cart]
        print(tabulate(cartData, headers=headers, tablefmt="fancy_grid", colalign=("center", "center", "center")))

    def beli(self):
        if not self.cart:
            print("Keranjang kosong. Tidak dapat checkout.")
            return

        total = sum(item["totalPrice"] for item in self.cart)
        print(f"Total pembayaran: {total}")
        print("Terima kasih telah berbelanja!")

if __name__ == "__main__":
    init(autoreset=True) # Inisialisasi colorama
    kasir = Kasir()

    while True:
        print("\n=== Program Kasir ===")
        print("1. Tampilkan Menu")
        print("2. Tambahkan Item ke Keranjang")
        print("3. Tampilkan Keranjang")
        print("4. Checkout")
        print("5. Keluar")

        try:
            pilih = int(input("Pilih menu (1-5): "))
        except ValueError:
            print("Masukkan harus berupa angka.")
            continue

        if pilih == 1:
            kasir.lihatMenu()
        elif pilih == 2:
            itemId = int(input("Masukkan ID item yang ingin ditambahkan: "))
            quantity = int(input("Masukkan jumlah item: "))
            kasir.masukkanKeranjang(itemId, quantity)
        elif pilih == 3:
            kasir.lihatKeranjang()
        elif pilih == 4:
            kasir.beli()
            break
        elif pilih == 5:
            print("Program selesai.")
            break
        else:
            print("Pilihan tidak valid. Silakan pilih menu 1-5.")
```

(Gambar 4.1)

1. `from tabulate import tabulate`: Mengimpor fungsi `tabulate` dari pustaka `tabulate` untuk memudahkan tampilan data dalam bentuk tabel.
2. `from colorama import init, Fore`: Mengimpor fungsi `init` dan kelas `Fore` dari pustaka `colorama` untuk memberikan warna pada output konsol.
3. `class Kasir::` Mendefinisikan kelas `Kasir` untuk merepresentasikan objek kasir.
4. `def __init__(self)::` Metode konstruktor kelas `Kasir` yang dijalankan ketika objek dibuat. Inisialisasi beberapa atribut seperti `cart` (keranjang belanja) dan `menu` (daftar menu).
5. `def lihatMenu(self)::` Metode untuk menampilkan menu. Menggunakan fungsi `tabulate` untuk memformat dan mencetak menu dalam bentuk tabel.
6. `def masukkanKeranjang(self, itemId, quantity)::` Metode untuk menambahkan item ke dalam keranjang belanja berdasarkan ID dan jumlah yang dimasukkan pengguna.
7. `def lihatKeranjang(self)::` Metode untuk menampilkan isi keranjang belanja dalam bentuk tabel.
8. `def beli(self)::` Metode untuk melakukan checkout dengan menghitung total harga belanjaan dalam keranjang.
9. `if __name__ == "__main__":`: Mengeksekusi blok program hanya jika skrip dijalankan sebagai program utama, bukan diimpor sebagai modul.
10. `init(autoreset=True)`: Inisialisasi `colorama` dengan opsi `autoreset=True` untuk mereset warna setelah setiap teks yang diwarnai.
11. `kasir = Kasir()`: Membuat objek dari kelas `Kasir`.
12. `while True::` Memulai loop tak terbatas untuk menjalankan program secara interaktif.
13. Menu interaktif:
  - Menampilkan opsi menu yang dapat dipilih oleh pengguna.



```

=== Program Kasir ===
1. Tampilkan Menu
2. Tambahkan Item ke Keranjang
3. Tampilkan Keranjang
4. Checkout
5. Keluar
Pilih menu (1-5):

```

- Menggunakan try-except untuk menangani kesalahan jika pengguna memasukkan input yang bukan angka.

#### 14. Logika pemilihan menu:

- Jika pengguna memilih 1, tampilkan menu.

Pilih menu (1-5): 1

No	Item	Price
1	Kopi	15000
2	Teh	10000
3	Roti	8000
4	Nasi Goreng	25000
5	Kwetiaw	15000
6	Es Teh Anget	10000
7	Gado-gado	8000
8	Nasi Putih	25000

- Jika pengguna memilih 2, minta ID item dan jumlah, lalu tambahkan ke keranjang.

```

Pilih menu (1-5): 2
Masukkan ID item yang ingin ditambahkan: 6
Masukkan jumlah item: 2
2 Es Teh Anget ditambahkan ke keranjang.

```

- Jika pengguna memilih 3, tampilkan isi keranjang.

Pilih menu (1-5): 3

Item	Quantity	Total Price
Es Teh Anget	2	20000

- Jika pengguna memilih 4, lakukan checkout dan keluar dari loop.

```
Pilih menu (1-5): 4  
Total pembayaran: 20000  
Terima kasih telah berbelanja!
```

- Jika pengguna memilih 5, keluar dari loop dan selesaikan program.

```
Pilih menu (1-5): 5  
Program selesai.
```

15. Akhirnya, cetak pesan sesuai dengan pilihan pengguna, dan program selesai.

## **V. Kesimpulan**

Latihan membangun aplikasi pembayaran ini akan memberi pemahaman yang lebih baik tentang mengembangkan aplikasi sederhana menggunakan bahasa pemrograman Python. Dalam proses ini, Anda akan diminta untuk merancang dan mengimplementasikan struktur dasar program pembayaran, termasuk membuat kategori, menangani data menu dan keranjang belanja, serta menerapkan logika untuk berinteraksi dengan pengguna melalui antarmuka teks. Selain itu, integrasi perpustakaan seperti "tabulasi" untuk memformat tabel dan "colorama" untuk mewarnai keluaran konsol menambah elemen visual yang meningkatkan pengalaman pengguna. Magang juga memberikan pengalaman dalam menangani kesalahan input, memastikan pengoperasian program yang benar, dan memberikan laporan informatif jika terjadi kesalahan. Secara keseluruhan, magang ini memberikan landasan yang kuat untuk pemahaman mendasar tentang pengembangan perangkat lunak, eksplorasi konsep pemrograman berorientasi objek, dan penggunaan perpustakaan eksternal untuk meningkatkan fungsionalitas program.

## **VI. Referensi**

Huda, A., Kom, S., Kom, M., *et-al*, 2020. *Dasar-Dasar Pemrograman Berbasis Python*. UNP PRESS.

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

Syahrudin, A. N., *et-al*, 2018. Input dan output pada bahasa pemrograman python. *Jurnal Dasar Pemograman Python STMIK*, 20, 1-7.