LAPORAN TUGAS STRUKTUR DATA LINKED LIST



DISUSUN OLEH :

1. Dimas Wijil Pamungkas (23091397201)
2. M. Raka Phaedra Agus Putra (23091397210)
3. Vani Fransiska (23091397193)

DOSEN PENGAMPU :

I Gde Agung Sri Sidhimantra,S.Kom., M.Kom

**PRODI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2024**

**Source Code**

# Definisikan class Node untuk merepresentasikan setiap elemen dalam linked list

class Node:

    def \_\_init\_\_(self, nama\_menu, harga):

        self.nama\_menu = nama\_menu  # Simpan nama menu

        self.harga = harga  # Simpan harga menu

        self.next = None  # Inisialisasi pointer ke node berikutnya sebagai None

# Definisikan class LinkedList untuk merepresentasikan linked list dari pesanan

class LinkedList:

    def \_\_init\_\_(self):

        self.head = None  # Inisialisasi node pertama dalam linked list sebagai None

    # Method untuk menambahkan pesanan baru ke dalam linked list

    def tambah\_pesanan(self, nama\_menu, harga):

        new\_node = Node(nama\_menu, harga)  # Buat node baru dengan nama menu dan harga yang diberikan

        if not self.head:  # Jika linked list kosong

            self.head = new\_node  # Node baru menjadi head

        else:

            current = self.head

            while current.next:  # Iterasi hingga node terakhir

                current = current.next

            current.next = new\_node  # Tambahkan node baru setelah node terakhir

    # Method untuk menampilkan semua pesanan yang sudah ditambahkan ke dalam linked list

    def tampilkan\_pesanan(self):

        current = self.head

        if not current:  # Jika linked list kosong

            print("Belum ada pesanan.")

        else:

            print("Pesanan yang sudah ditambahkan:")

            count = 1

            while current:  # Iterasi dan tampilkan setiap node

                print(f"{count}. {current.nama\_menu} - Rp {current.harga}")

                current = current.next

                count += 1

    # Method untuk menghitung total harga dari semua pesanan dalam linked list

    def total\_harga(self):

        total = 0

        current = self.head

        while current:  # Iterasi dan jumlahkan harga setiap pesanan

            total += current.harga

            current = current.next

        return total

# Menu dan harga miexue disimpan dalam sebuah dictionary

menu = {

    "Miexue Ice Cream": 5000,

    "Boba Shake": 16000,

    "Mi Sundae": 14000,

    "Mi Ganas": 11000,

    "Creamy Mango Boba": 22000

}

# Membuat linked list baru untuk menyimpan pesanan miexue

miexue = LinkedList()

# Menampilkan pesan selamat datang kepada pengguna

print("========================================================")

print("========================================================")

print("Selamat datang di D4 MIE!")

# Program utama berjalan dalam loop tak terbatas hingga pengguna memilih untuk keluar

while True:

    print("========================================================")

    # Menampilkan menu miexue beserta harga

    print("\nMenu Miexue:")

    for item, harga in menu.items():

        print(f"{item} -> Rp {harga}")

    # Menampilkan pilihan menu untuk pengguna

    print("========================================================")

    print("\nPilihan Perintah:")

    print("1. Tambah pesanan ke keranjang")

    print("2. Tampilkan pesanan yang sudah ditambahkan")

    print("3. Jumlah harga yang dibayarkan")

    print("========================================================")

    # Meminta input pilihan dari pengguna

    pilihan = input("Masukkan pilihan: ")

    print("========================================================")

    # Memproses pilihan pengguna

    if pilihan == '1':

        pesanan = input("Masukkan pesanan: ")

        if pesanan in menu:

            miexue.tambah\_pesanan(pesanan, menu[pesanan])

            print(f"{pesanan} sudah ditambahkan ke keranjang")

            print("========================================================")

        else:

            print("Menu tidak tersedia.")

    elif pilihan == '2':

        miexue.tampilkan\_pesanan()

    elif pilihan == '3':

        total\_harga = miexue.total\_harga()

        print(f"Total biaya yang harus dibayarkan adalah Rp {total\_harga}. Terima kasih atas pesanannya!")

        print("========================================================")

        break

    else:

        print("Pilihan tidak valid, silakan coba lagi.")

**Penjelasan Step by Step**

* Class Node: digunakan untuk Mendefinisikan sebuah kelas bernama Node yang akan merepresentasikan setiap elemen dalam linked list. *Def \_\_init\_\_(self, nama\_menu, harga)*: digunakan untuk membuat objek dari kelas Node. Metode ini menerima dua parameter: nama\_menu dan harga, yang akan digunakan untuk inisialisasi atribut-atribut dari objek yang dibuat.*Self.nama\_menu = nama\_menu:* Atribut nama\_menu dari objek akan diinisialisasi dengannilai nama\_menu yang diterima sebagai parameter.*Self.harga = harga* merupakanAtribut harga dari objek akan diinisialisasi dengan nilai harga yang diterima sebagai parameter.*Self.next = None*: Atribut next dari objek akan diinisialisasi dengan None. Atribut ini akan digunakan untuk menunjukkan elemen berikutnya dalam linked list.

class Node:

    def \_\_init\_\_(self, nama\_menu, harga):

        self.nama\_menu = nama\_menu  # Simpan nama menu

        self.harga = harga  # Simpan harga menu

        self.next = None  # Inisialisasi pointer ke node berikutnya sebagai None

* *class LinkedList:`*Mendefinisikan sebuah kelas bernama `LinkedList` yang akan merepresentasikan linked list.*`def \_\_init\_\_(self):`*: Metode konstruktor yang akan dipanggil saat membuat objek dari kelas `LinkedList`. Metode ini tidak menerima parameter tambahan selain `self`. Metode ini bertujuan untuk inisialisasi atribut-atribut dari objek yang dibuat.*`self.head = None`*: Atribut `head` dari objek `LinkedList` akan diinisialisasi dengan `None`. Atribut `head` digunakan untuk menunjukkan elemen pertama atau awal dari linked list.

class LinkedList:

    def \_\_init\_\_(self):

        self.head = None  # Inisialisasi node pertama dalam linked list sebagai None

* *`def tambah\_pesanan(self, nama\_menu, harga):`*: Mendefinisikan sebuah metode `tambah\_pesanan` yang menerima dua parameter: `nama\_menu` (nama menu pesanan) dan `harga` (harga menu pesanan).*`new\_node = Node(nama\_menu, harga)*`: digunakan untuk Membuat objek `Node` baru dengan menggunakan konstruktor `Node` yang telah didefinisikan sebelumnya. Objek `new\_node` akan memiliki atribut `nama\_menu` dan `harga` sesuai dengan nilai yang diteruskan ke dalam metode `tambah\_pesanan`.*`if not self.head:*`digunakan untuk Memeriksa apakah linked list kosong dengan cara memeriksa apakah atribut `head` adalah `None`. Jika linked list kosong, maka `self.head` akan bernilai `None`.*`self.head = new\_node*`: Jika linked list kosong (nilai `head` adalah `None`), maka `new\_node` akan menjadi elemen pertama atau `head` dari linked list.*`else:`* Jika linked list tidak kosong, maka program akan melanjutkan ke blok kode berikutnya.`*current = self.head*`: digunakan untuk Membuat variabel `current` yang menunjuk ke elemen pertama atau `head` dari linked list.`*while current.next:`*: Melakukan iterasi dengan menggunakan loop `while` untuk mencari elemen terakhir dalam linked list. Loop ini akan terus berjalan selama `current.next` tidak `None`, yang menunjukkan bahwa `current` bukanlah elemen terakhir.`*current = current.next`*: Memindahkan `current` ke elemen berikutnya dalam linked list.`*current.next = new\_node`*: Setelah mencapai elemen terakhir (`current` merupakan elemen terakhir), maka `new\_node` akan ditambahkan setelah elemen terakhir tersebut dengan mengubah nilai `next` dari elemen terakhir menjadi `new\_node`.

    def tambah\_pesanan(self, nama\_menu, harga):

        new\_node = Node(nama\_menu, harga)  # Buat node baru dengan nama menu dan harga yang diberikan

        if not self.head:  # Jika linked list kosong

            self.head = new\_node  # Node baru menjadi head

        else:

            current = self.head

            while current.next:  # Iterasi hingga node terakhir

                current = current.next

            current.next = new\_node  # Tambahkan node baru setelah node terakhir

* *def tampilkan\_pesanan(self):`*: Mendefinisikan sebuah metode `tampilkan\_pesanan` yang tidak menerima parameter tambahan selain `*self`.current = self.head`*: Membuat variabel `current` yang menunjuk ke elemen pertama atau `head` dari linked list.`if not current:`: Memeriksa apakah linked list kosong dengan cara memeriksa apakah `current` bernilai `None`. Jika linked list kosong, maka tidak ada pesanan yang ditampilkan.`print*("Belum ada pesanan.")*`: Jika linked list kosong, maka program akan mencetak pesan "Belum ada pesanan."`*else*:`: Jika linked list tidak kosong, maka program akan melanjutkan ke blok kode berikutnya.`*print("Pesanan yang sudah ditambahkan:")`*: Mencetak pesan "Pesanan yang sudah ditambahkan:" untuk menandakan bahwa daftar pesanan akan ditampilkan.`count = 1`: Membuat variabel `count` yang digunakan untuk menghitung nomor urut dari setiap pesanan yang ditampilkan.`while current:`: Melakukan iterasi dengan menggunakan loop `*while*` untuk menampilkan setiap pesanan dalam linked list. Loop ini akan terus berjalan selama `*current*` tidak `None`, yang menunjukkan bahwa masih ada elemen dalam linked list yang harus ditampilkan.`*print(f"{count}. {current.nama\_menu} - Rp {current.harga}")`*: digunakan untuk qqx2xMencetak nama menu dan harga dari pesanan yang saat ini sedang ditampilkan, beserta nomor urutnya (`count`). Pesanan ini diambil dari atribut `nama\_menu` dan `harga` dari objek `current`, yang merupakan objek `Node` dalam linked list.`*current = current.next`*: Memindahkan `current` ke elemen berikutnya dalam linked list.`count += 1`: Menambahkan nilai `count` untuk mengupdate nomor urut pesanan berikutnya yang akan ditampilkan.

    def tampilkan\_pesanan(self):

        current = self.head

        if not current:  # Jika linked list kosong

            print("Belum ada pesanan.")

        else:

            print("Pesanan yang sudah ditambahkan:")

            count = 1

            while current:  # Iterasi dan tampilkan setiap node

                print(f"{count}. {current.nama\_menu} - Rp {current.harga}")

                current = current.next

                count += 1

* *`def total\_harga(self):*`: Mendefinisikan sebuah metode `total\_harga` yang tidak menerima parameter tambahan selain `self`.`total = 0`: Membuat variabel `total` yang akan digunakan untuk menyimpan total harga dari semua pesanan.`*current = self.head`*: Membuat variabel `current` yang menunjuk ke elemen pertama atau `head` dari linked list.`while current:`: Melakukan iterasi dengan menggunakan loop `while` untuk menjumlahkan harga dari setiap pesanan dalam linked list. Loop ini akan terus berjalan selama `current` tidak `None`, yang menunjukkan bahwa masih ada elemen dalam linked list yang harus dihitung harganya.`*total += current.harga`*: Menambahkan harga dari pesanan yang saat ini sedang dihitung ke dalam total harga. Harga pesanan diambil dari atribut `harga` dari objek `current`, yang merupakan objek `Node` dalam linked list.`*current = current.next`*: Memindahkan `current` ke elemen berikutnya dalam linked list.`*return total`:* Mengembalikan nilai `total` sebagai hasil dari perhitungan total harga dari semua pesanan dalam linked list.

    def total\_harga(self):

        total = 0

        current = self.head

        while current:  # Iterasi dan jumlahkan harga setiap pesanan

            total += current.harga

            current = current.next

        return total

* Membuat fungsi yang berisikan tentang menu beserta harganya agar bisa ditampilkan di program utama

menu = {

    "Miexue Ice Cream": 5000,

    "Boba Shake": 16000,

    "Mi Sundae": 14000,

    "Mi Ganas": 11000,

    "Creamy Mango Boba": 22000

}

* Membuat linkedlist baru untuk menyimpan pesanan mixue

miexue = LinkedList()

* Membuat fungsi untuk ditampilkan kepada pengguna sebagai ucapan selamat datang

print("========================================================")

print("========================================================")

print("Selamat datang di D4 MIE!")

* *while True:`*: Mendefinisikan sebuah loop tak terbatas, yang akan terus berjalan selama kondisinya adalah `True`.`*print("\nMenu Miexue:")*`: Mencetak pesan yang menandakan bahwa daftar menu miexue beserta harganya akan ditampilkan.`for item, harga in menu.items():`: Melakukan iterasi melalui setiap item dan harganya dalam dictionary `menu`. `item` akan mewakili nama menu, sedangkan `harga` akan mewakili harga dari setiap menu tersebut.`*print(f"{item} -> Rp {harga}")*`: Mencetak nama menu beserta harganya dalam format tertentu.*`print("\nPilihan Perintah:")*`: Mencetak pesan yang menandakan bahwa pilihan perintah untuk pengguna akan ditampilkan.`*print("1. Tambah pesanan ke keranjang")`*: Mencetak opsi untuk menambahkan pesanan ke dalam keranjang.`*print("2. Tampilkan pesanan yang sudah ditambahkan")`*: Mencetak opsi untuk menampilkan pesanan yang sudah ditambahkan ke dalam keranjang.`*print("3. Jumlah harga yang dibayarkan")`*: Mencetak opsi untuk menampilkan jumlah harga yang harus dibayarkan untuk seluruh pesanan yang sudah ditambahkan ke dalam keranjang.

while True:

    print("========================================================")

    # Menampilkan menu miexue beserta harga

    print("\nMenu Miexue:")

    for item, harga in menu.items():

        print(f"{item} -> Rp {harga}")

    # Menampilkan pilihan menu untuk pengguna

    print("========================================================")

    print("\nPilihan Perintah:")

    print("1. Tambah pesanan ke keranjang")

    print("2. Tampilkan pesanan yang sudah ditambahkan")

    print("3. Jumlah harga yang dibayarkan")

    print("========================================================")

* Membuat fungsi input agar pengguna bisa memasukkan pilihannya.

    pilihan = input("Masukkan pilihan: ")

print("========================================================")

* Program Ini menggunakan struktur kontrol if-elif-else dalam Python untuk menentukan tindakan berdasarkan input pengguna.Jika input pengguna sama dengan ‘1’, program akan meminta pengguna untuk memasukkan pesanan melalui perintah `input()`. Kemudian, program akan memeriksa apakah pesanan tersebut ada dalam daftar menu yang telah ditentukan sebelumnya. Jika pesanan ada dalam menu, maka pesanan akan ditambahkan ke keranjang menggunakan metode `tambah\_pesanan()` dari objek `miexue`, dengan menambahkan nama pesanan dan harganya. Setelah itu, program akan mencetak bahwa pesanan telah ditambahkan ke keranjang dan menampilkan garis pemisah.Jika input pengguna sama dengan ‘2’, program akan memanggil metode `tampilkan\_pesanan()` dari objek `miexue`, yang kemungkinan akan menampilkan semua pesanan yang sudah dimasukkan ke dalam keranjang.Jika input pengguna sama dengan ‘3’, program akan menghitung total harga dari semua pesanan yang ada di keranjang menggunakan metode `total\_harga()` dari objek `miexue`. Total harga akan dicetak dengan format “Total biaya yang harus dibayarkan adalah Rp {total\_harga}. Terima kasih atas pesanannya!”, dan program akan keluar dari loop menggunakan pernyataan `break`.Jika input pengguna tidak sama dengan ‘1’, ‘2’, atau ‘3’, maka program akan mencetak “Pilihan tidak valid, silakan coba lagi.”Kode ini diasumsikan bekerja di dalam suatu loop, sehingga pengguna dapat terus memasukkan pilihan hingga mereka memilih untuk keluar dari program.

    if pilihan == '1':

        pesanan = input("Masukkan pesanan: ")

        if pesanan in menu:

            miexue.tambah\_pesanan(pesanan, menu[pesanan])

            print(f"{pesanan} sudah ditambahkan ke keranjang")

            print("========================================================")

        else:

            print("Menu tidak tersedia.")

    elif pilihan == '2':

        miexue.tampilkan\_pesanan()

    elif pilihan == '3':

        total\_harga = miexue.total\_harga()

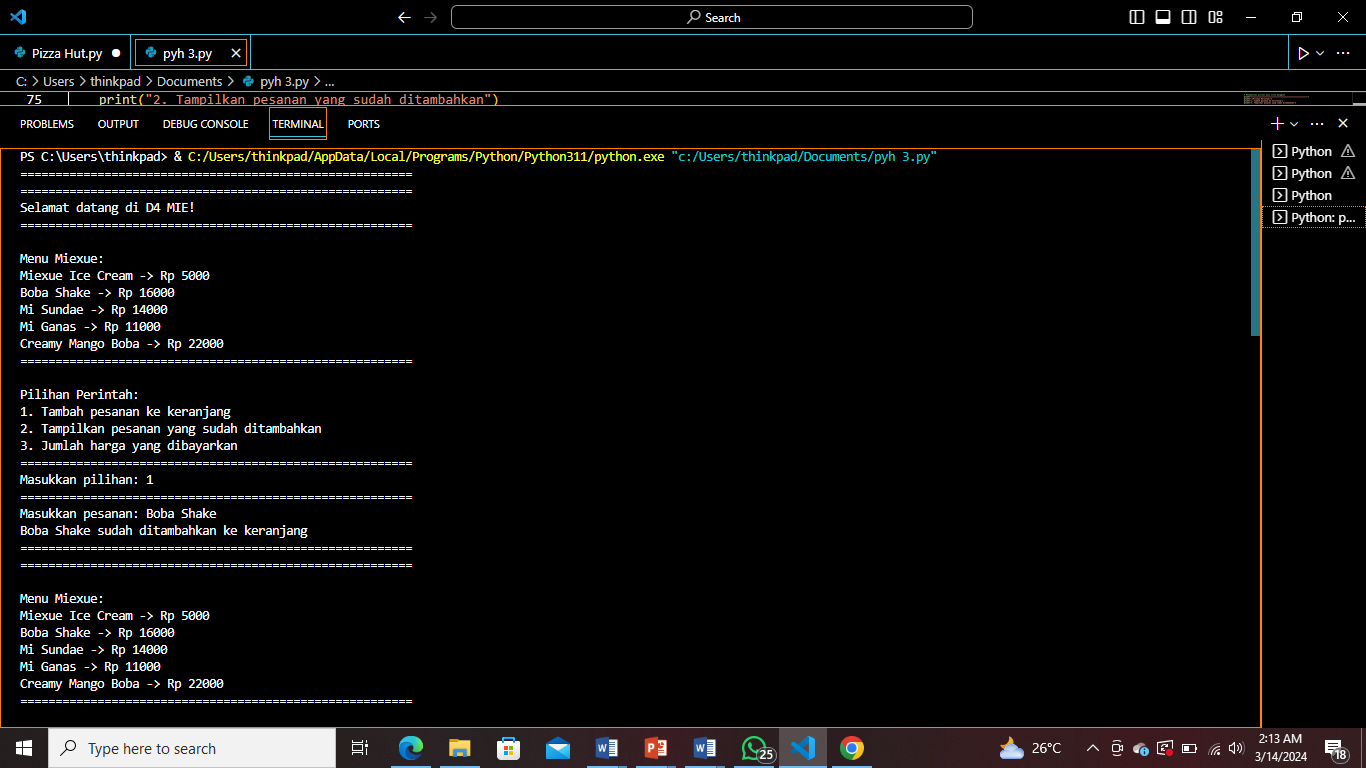
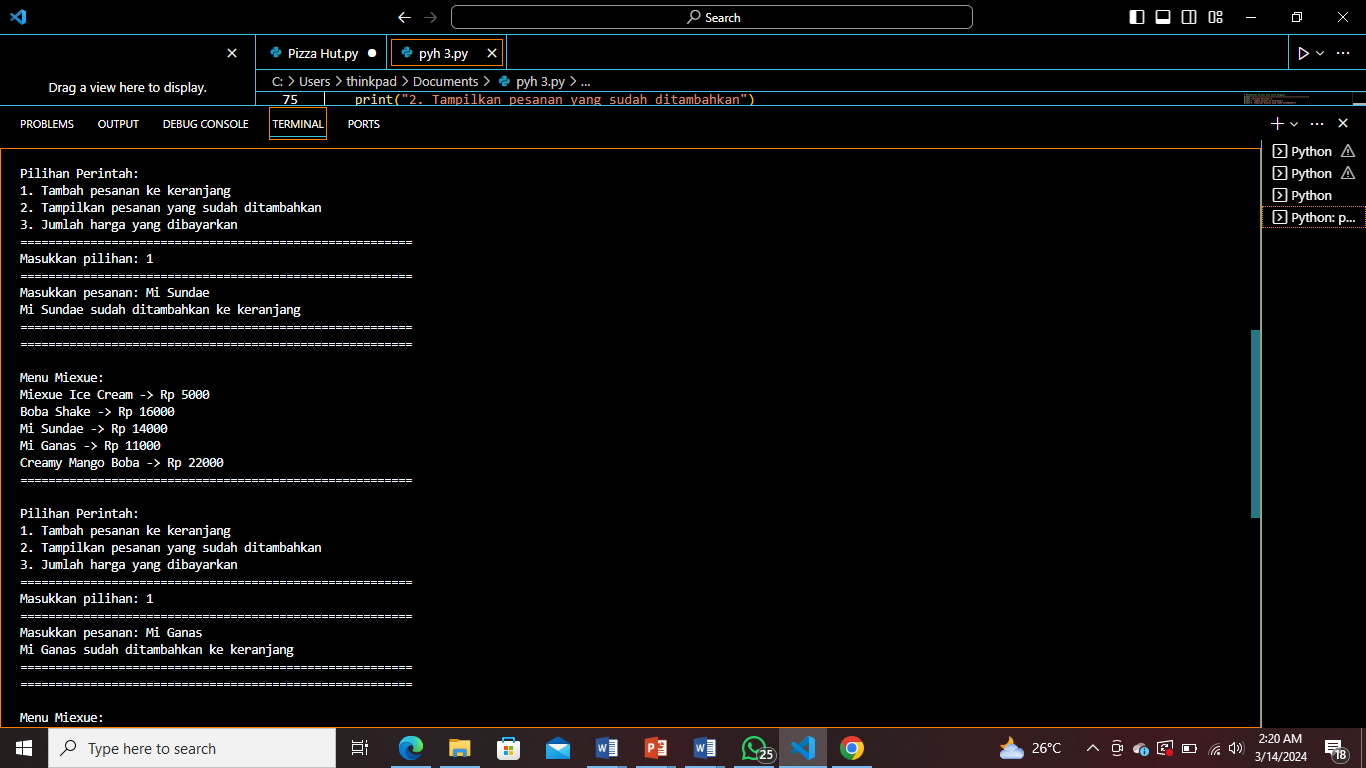
        print(f"Total biaya yang harus dibayarkan adalah Rp {total\_harga}. Terima kasih atas pesanannya!")

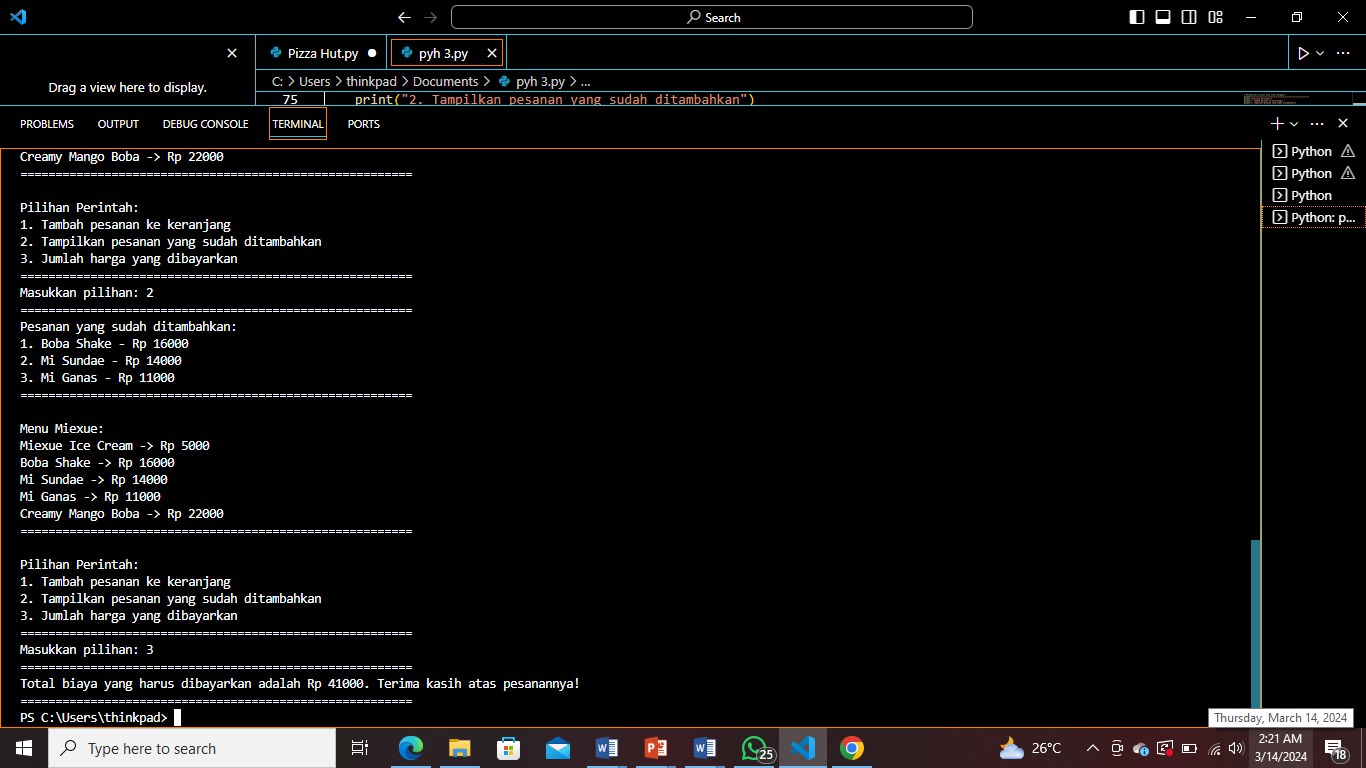
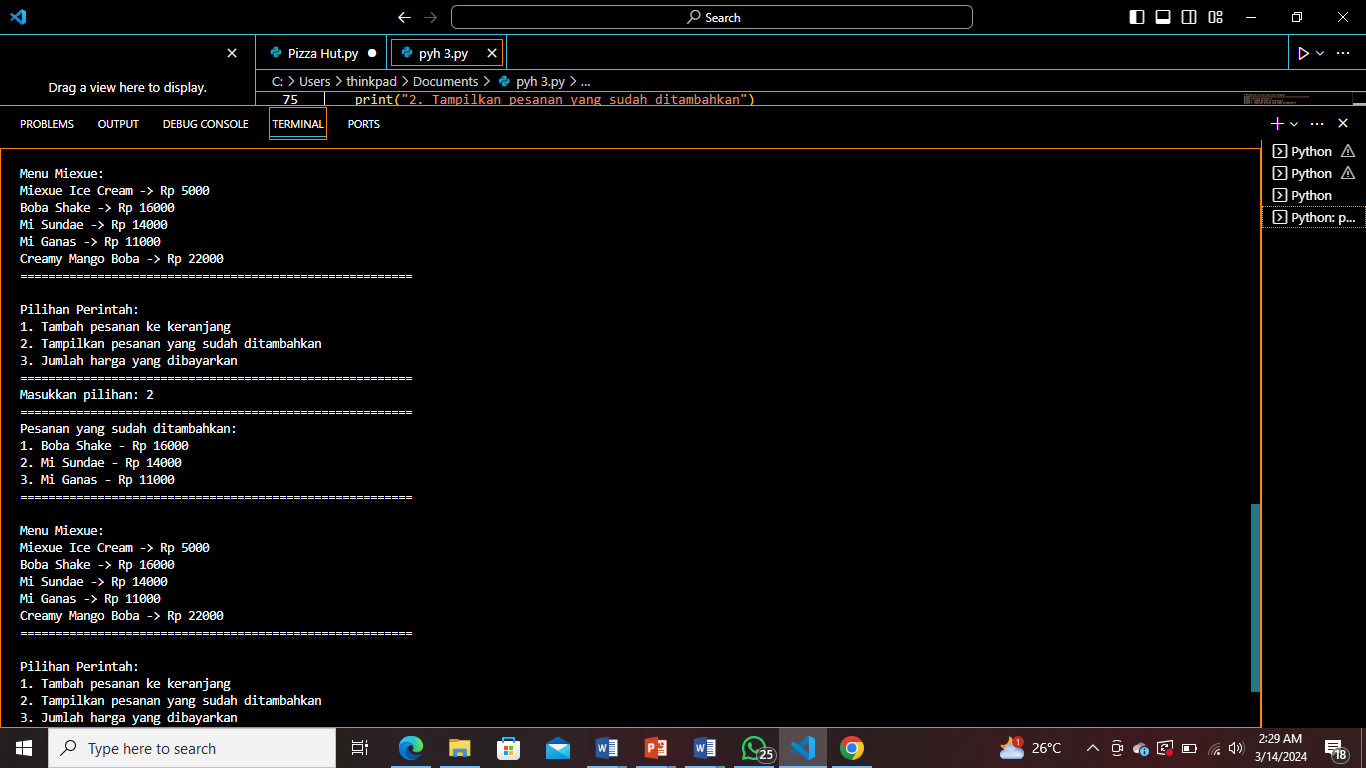
        print("========================================================")

        break

    else:

        print("Pilihan tidak valid, silakan coba lagi.")

**Hasil akhir program**

****