LAPORAN TUGAS STRUKTUR DATA

“LINKED LIST”

Nama Kelompok 1:

Achmad Diky Setiawan (23091397178)

Ika Amelia Zalianti (23091397190)

Achmad Harris Abdillah (23091397204)

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2023/2024

Linked list

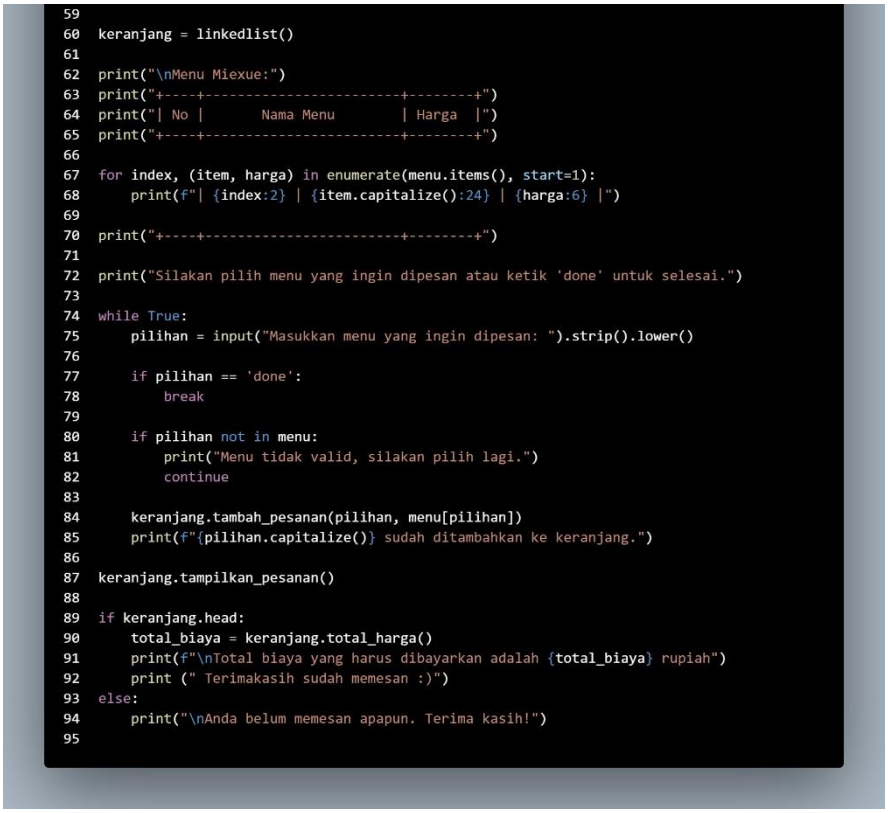
- **Linked list**

  Pengertian Linked list

  Linked list adalah salah satu struktur data yang terdiri dari serangkaian elemen yang disusun secara berurutan. Setiap elemen dalam linked list disebut "node" dan memiliki dua bagian utama: data dan pointer yang menunjuk ke node berikutnya dalam urutan. Node terakhir dalam linked list menunjuk ke nilai null, menandakan akhir dari linked list.

Source Code

```python
class Node:
    def __init__(self, nama, harga):
        self.nama = nama
        self.harga = harga
        self.next = None

class linkedlist:
    def __init__(self):
        self.head = None

    def tambah_pesanan(self, nama, harga):
        new_node = Node(nama, harga)
        if not self.head:
            self.head = new_node
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_node

    print(" SELAMAT DATANG DI E-ORDER WARUNG D4 MI")
    print("MINUMAN DAN MIE PEDAS TERSEDIA DI SINI!!  ")

    def tampilkan_pesanan(self):
        if not self.head:
            print("Keranjang masih kosong.")
            return

        print("\nPesanan yang sudah ditambahkan:")
        print("+----+-----------------------+--------+")
        print("| No |       Nama Menu       | Harga  |")
        print("+----+-----------------------+--------+")

        current = self.head
        index = 1
        while current:
            print(f"| {index:2} | {current.nama.capitalize():24} | {current.harga:6} |")
            current = current.next
            index += 1

        print("+----+-----------------------+--------+")

    def total_harga(self):
        total = 0
        current = self.head
        while current:
            total += current.harga
            current = current.next
        return total

# Menu Miexue
menu = {
    'miexue ice cream': 5000,
    'boba shake': 16000,
    'mi sundae': 14000,
    'mi ganas': 11000,
    'creamy mango boba': 22000
}
```



```
59
60  keranjang = linkedlist()
61
62  print("\nMenu Miexue:")
63  print("+----+------------------------+--------+")
64  print("| No |       Nama Menu        | Harga  |")
65  print("+----+------------------------+--------+")
66
67  for index, (item, harga) in enumerate(menu.items(), start=1):
68      print(f"| {index:2} | {item.capitalize():24} | {harga:6} |")
69
70  print("+----+------------------------+--------+")
71
72  print("Silakan pilih menu yang ingin dipesan atau ketik 'done' untuk selesai.")
73
74  while True:
75      pilihan = input("Masukkan menu yang ingin dipesan: ").strip().lower()
76
77      if pilihan == 'done':
78          break
79
80      if pilihan not in menu:
81          print("Menu tidak valid, silakan pilih lagi.")
82          continue
83
84      keranjang.tambah_pesanan(pilihan, menu[pilihan])
85      print(f"{pilihan.capitalize()} sudah ditambahkan ke keranjang.")
86
87  keranjang.tampilkan_pesanan()
88
89  if keranjang.head:
90      total_biaya = keranjang.total_harga()
91      print(f"\nTotal biaya yang harus dibayarkan adalah {total_biaya} rupiah")
92      print (" Terimakasih sudah memesan :)")
93  else:
94      print("\nAnda belum memesan apapun. Terima kasih!")
95
```

Penjelasan step by step

- Pada baris 1 ini kita membuat deklarasi/definisi class node

```
1   class Node:
```

- Pada baris berikutnya kita mendefinisikan metode konstruktor ***__init__*** untuk **class node**. Ini digunakan untuk menginisialisasi objek node yang memiliki atribut ***nama*** untuk menyimpan nama menu dan ***harga*** untuk menyimpan harga menu. Atribut ***next*** adalah referensi ke node berikutnya dalam linked list.

```
2       def __init__(self, nama, harga):
3           self.nama = nama
4           self.harga = harga
5           self.next = None
```

- Setelah itu membuat definisi **class linked list** yang digunakan untuk membuat dan mengelola linked list untuk menyimpan pesanan. Serta mendefinisikan metode konstruktor ***__init__*** untuk kelas **linked list**. Ini digunakan untuk menginisialisasi objek **linked list** dengan atribut **head**.

```
7   class linkedlist:
8       def __init__(self):
9           self.head = None
10
```

- Membuat fungsi **tambah_pesanan** : Fungsi ini digunakan untuk menambahkan pesanan baru ke keranjang. Jika keranjang masih kosong, maka pesanan baru menjadi **head**. Jika tidak, pesanan baru ditambahkan di akhir keranjang.

```
11      def tambah_pesanan(self, nama, harga):
12          new_node = Node(nama, harga)
13          if not self.head:
14              self.head = new_node
15          else:
16              current = self.head
17              while current.next:
18                  current = current.next
19              current.next = new_node
```

- Mengeprint awalan pesan pembuka untuk pelanggan yang akan membeli

```
20
21      print(" SELAMAT DATANG DI E-ORDER WARUNG D4 MI")
22      print("MINUMAN DAN MIE PEDAS TERSEDIA DI SINI!!  ")
23
```

- Membuat fungsi **tampilkan_pesanan** : Fungsi ini digunakan untuk menampilkan pesanan yang ada di dalam keranjang. Jika keranjang kosong, akan keluar output "Keranjang masih kosong" yang menunjukkan bahwa tidak ada pesanan dalam keranjang.

```
24      def tampilkan_pesanan(self):
25          if not self.head:
26              print("Keranjang masih kosong.")
27              return
```

- Mengeprint header dari tabel pesanan

```
28
29          print("\nPesanan yang sudah ditambahkan:")
30          print("+----+-----------------------+--------+")
31          print("| No |       Nama Menu       | Harga  |")
32          print("+----+-----------------------+--------+")
33
```

- menginisialisasi variabel **current** sebagai **head** dari linked list yang berarti kita mulai dari kepala linked list untuk mengakses setiap pesanan. Dan variabel **index** digunakan untuk melacak nomor urut setiap pesanan dalam tampilan. Dimulai dari 1 karena umumnya nomor urut dimulai dari 1 bukan 0. Selanjutnya, baris ini memulai loop while yang akan berjalan selama **current** tidak bernilai **None**. Dalam konteks ini, berarti loop akan berlanjut selama masih ada node/pesanan yang dapat diakses dalam linked list.

```
33
34          current = self.head
35          index = 1
36          while current:
```

- Selanjutnya mengeprint yang digunakan untuk mencetak informasi tentang setiap pesanan ke layar, berikut penjelasannya:

  - **{index:2}** digunakan untuk mencetak nomor urut dengan lebar 2 karakter.
  - **{current.nama.capitalize():24}** digunakan untuk mencetak nama pesanan dengan kapitalisasi huruf pertama dan lebar 24 karakter.
  - **{current.harga:6}** digunakan untuk mencetak harga pesanan dengan lebar 6 karakter.

  Sedangkan **current = current.next** digunakan untuk memindahkan current ke node berikutnya dalam linked list. Dengan ini, kita dapat melanjutkan iterasi melalui setiap pesanan dalam linked list dan **index += 1** ini digunakan agar setiap kali kita mencetak satu pesanan, nomor urutnya diinkrementasikan sehingga pesanan berikutnya akan memiliki nomor urut berikutnya.

```
37            print(f"| {index:2} | {current.nama.capitalize():24} | {current.harga:6} |")
38            current = current.next
39            index += 1
40
```

- Mengeprint garis pembatas untuk menandai akhir dari daftar pesanan yang ditampilkan.

```
40
41        print("+----+------------------------+--------+")
42
```

- Selanjutnya menginisialisasi fungsi **total harga** : fungsi ini digunakan untuk ntuk menghitung total biaya dari semua pesanan dalam keranjang belanja. Pada fungsi ini memuat variabel **total** dan **current.** Variabel **total** ini berfungsi menginisialisasi dengan nilai 0 dan variabel ini akan digunakan untuk menyimpan total biaya pesanan. Sedangkan variabel **current** berfungsi untuk menginisialisasi **current** dengan **self.head**, dengan ini kita akan mulai menghitung total biaya dari node pertama.

```
43    def total_harga(self):
44        total = 0
45        current = self.head
```

- Selanjutnya, dilakukan iterasi melalui setiap node dalam linked list dengan menggunakan loop while, dimana nilai harga dari setiap pesanan ditambahkan ke dalam variabel **total**. Kemudian, **current** digeser ke node berikutnya dalam linked list hingga mencapai akhir keranjang. Selanjutnya **return total** yang digunakan untuk mengembalikan nilai **total** setelah semua pesanan dalam keranjang telah dihitung yang artinya adalah total biaya dari semua pesanan dalam keranjang belanja.

```
46          while current:
47              total += current.harga
48              current = current.next
49          return total
50
```

- Codingan selanjutnya terdiri dari dua bagian:

  codingan pertama mendefinisikan sebuah kamus menu yang berisi beberapa jenis menu makanan/minuman beserta harganya.Membuat sebuah objek keranjang dari kelas Keranjang.untuk penjelasan lebih rinci nya sebagai beikut:

  > Kamus menu berisi beberapa pasangan kunci-nilai yang mewakili nama menu dan harganya. Misalnya, 'miexue ice cream' dengan harga 5000, 'boba shake' dengan harga 16000, dan seterusnya.

  > Objek keranjang dibuat menggunakan kelas Keranjang. Ini menunjukkan bahwa ada kelas Keranjang yang digunakan untuk mengelola pesanan-pesanan yang dimasukkan oleh pengguna.

  Dengan demikian, bagian pertama mendefinisikan daftar menu beserta harganya, sedangkan bagian kedua membuat objek keranjang yang akan digunakan untuk mengelola pesanan-pesanan tersebut.

```
51  # Menu Miexue
52  menu = {
53      'miexue ice cream': 5000,
54      'boba shake': 16000,
55      'mi sundae': 14000,
56      'mi ganas': 11000,
57      'creamy mango boba': 22000
58  }
59
60  keranjang = linkedlist()
61
```

- Codingan selanjutnya di gunakan untuk mengprint header dari tabel pesanan

```
62  print("\nMenu Miexue:")
63  print("+----+-------------------------+--------+")
64  print("| No |        Nama Menu        | Harga  |")
65  print("+----+-------------------------+--------+")
```

- codingan selanjutnya merupakan sebuah loop for yang melakukan enumerasi terhadap pasangan kunci-nilai dalam kamus menu. Setiap pasangan kunci-nilai tersebut direpresentasikan oleh variabel item dan harga.

```
67  for index, (item, harga) in enumerate(menu.items(), start=1):
68      print(f"| {index:2} | {item.capitalize():24} | {harga:6} |")
```

- codingan selanjutnya yaitu untuk Mengeprint garis pembatas untuk menandai akhir dari daftar pesanan yang ditampilkan.

```
70  print("+----+------------------------+--------+")
```

- codingan ini mencetak pesan ke layar yang meminta pengguna untuk memilih menu yang ingin dipesan atau untuk mengetik 'done' jika pengguna telah selesai dengan pemesanan. Pesan ini bertujuan untuk memberikan petunjuk kepada pengguna tentang langkah selanjutnya dalam proses pemesanan menu. Dengan demikian, pengguna diarahkan untuk memilih menu atau mengakhiri proses pemesanan dengan mengetik 'done'.

```
72  print("Silakan pilih menu yang ingin dipesan atau ketik 'done' untuk selesai.")
```

- codingan selanjutnya ini merupakan sebuah loop while yang akan terus berjalan selama kondisi True. Di dalam loop, pengguna diminta untuk memasukkan menu yang ingin dipesan melalui fungsi input(). Selanjutnya, codingan tersebut akan melakukan pengecekan terhadap input yang dimasukkan pengguna.

  Jika pengguna memasukkan 'done', loop akan dihentikan dengan menggunakan pernyataan break.Jika menu yang dimasukkan pengguna tidak ada dalam kamus menu, maka akan dicetak pesan "Menu tidak valid, silakan pilih lagi." dan loop akan melanjutkan ke iterasi berikutnya dengan menggunakan pernyataan continue.

  Dengan demikian, kode ini memastikan bahwa pengguna hanya dapat memasukkan menu yang valid, dan memberikan opsi untuk menghentikan loop dengan memasukkan 'done'.

```
74  while True:
75      pilihan = input("Masukkan menu yang ingin dipesan: ").strip().lower()
76
77      if pilihan == 'done':
78          break
79
80      if pilihan not in menu:
81          print("Menu tidak valid, silakan pilih lagi.")
82          continue
```

- codingan ini bertujuan untuk menambahkan pesanan ke dalam keranjang dengan menggunakan metode tambah_pesanan dari objek keranjang. Metode ini menerima

dua argumen, yaitu pilihan yang merupakan nama menu yang dipilih oleh pengguna, dan menu[pilihan] yang mengambil harga dari menu yang dipilih dari kamus menu.

setelah pesanan ditambahkan ke keranjang, codingan tersebut akan mencetak pesan yang menyatakan bahwa menu yang dipilih (dengan huruf kapital) sudah ditambahkan ke dalam keranjang. Dengan demikian, pengguna akan mendapat konfirmasi bahwa pesanannya telah berhasil ditambahkan ke dalam keranjang.

```
84      keranjang.tambah_pesanan(pilihan, menu[pilihan])
85      print(f"{pilihan.capitalize()} sudah ditambahkan ke keranjang.")
```

- codingan yang selanjutnya ini memanggil metode tampilkan_pesanan() dari objek keranjang.
  metode ini bertujuan untuk menampilkan daftar pesanan yang telah ditambahkan ke dalam keranjang. dengan memanggil metode ini, pengguna dapat melihat daftar pesanan yang telah mereka tambahkan sejauh ini. Hal ini memungkinkan pengguna untuk memeriksa kembali pesanan mereka sebelum melanjutkan proses pemesanan atau pembayaran.

```
87  keranjang.tampilkan_pesanan()
```

- ini adalah codingan terakhir yang akan dijalankan, codingan ini akan melakukan pengecekan terhadap apakah keranjang memiliki pesanan atau tidak. Jika keranjang memiliki setidaknya satu pesanan, maka total biaya dari pesanan-pesanan tersebut dihitung dengan memanggil metode total_harga() dari objek keranjang. Selanjutnya, total biaya tersebut dicetak ke layar bersama dengan pesan "Terimakasih sudah memesan :)".

  Jika keranjang tidak memiliki pesanan (atau dalam konteks lain, jika keranjang.head adalah None atau False), maka pesan "Anda belum memesan apapun. Terima kasih!" akan dicetak ke layar. Hal ini bertujuan untuk memberi tahu pengguna bahwa mereka belum menambahkan pesanan apapun ke dalam keranjang.

```
89  if keranjang.head:
90      total_biaya = keranjang.total_harga()
91      print(f"\nTotal biaya yang harus dibayarkan adalah {total_biaya} rupiah")
92      print (" Terimakasih sudah memesan :)")
93  else:
94      print("\nAnda belum memesan apapun. Terima kasih!")
95
```

Output dari code diatas

```
 SELAMAT DATANG DI E-ORDER WARUNG D4 MI
MINUMAN DAN MIE PEDAS TERSEDIA DI SINI!!

Menu Miexue:
+----+-------------------------+--------+
| No |       Nama Menu         | Harga  |
+----+-------------------------+--------+
|  1 | Miexue ice cream        |   5000 |
|  2 | Boba shake              |  16000 |
|  3 | Mi sundae               |  14000 |
|  4 | Mi ganas                |  11000 |
|  5 | Creamy mango boba       |  22000 |
+----+-------------------------+--------+
Silakan pilih menu yang ingin dipesan atau ketik 'done' untuk selesai.
Masukkan menu yang ingin dipesan: mi ganas
Mi ganas sudah ditambahkan ke keranjang.
Masukkan menu yang ingin dipesan: boba shake
Boba shake sudah ditambahkan ke keranjang.
Masukkan menu yang ingin dipesan: mi sundae
Mi sundae sudah ditambahkan ke keranjang.
Masukkan menu yang ingin dipesan: done

Pesanan yang sudah ditambahkan:
+----+-------------------------+--------+
| No |       Nama Menu         | Harga  |
+----+-------------------------+--------+
|  1 | Mi ganas                |  11000 |
|  2 | Boba shake              |  16000 |
|  3 | Mi sundae               |  14000 |
+----+-------------------------+--------+

Total biaya yang harus dibayarkan adalah 41000 rupiah
 Terimakasih sudah memesan :)
```

Link GitHub:

https://github.com/dikysetiawan21/2023F_KELOMPOK_1