

TUGAS MENJELASKAN PROGRAM

Tugas Ini Dibuat Guna Memenuhi Mata Kuliah Struktur Data



Dosen pengampu:

Adam bachtiar, s.kom, M.MT

Disusun Oleh :

Nama : Ratih Purnamasari

NIM : 24241039

Kelas : PTI B

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
FAKULTAS SAINS, TEKNIK DAN TERAPAN
UNIVERSITAS PENDIDIKAN MANDALIKA MATARAM
2025**

❖ Kelas Node

```
1 # Kelas Node
2 class Node:
3     def __init__(self, data):
4         self.data = data          # Menyimpan nilai data dari node
5         self.prev = None          # Pointer ke node sebelumnya
6         self.next = None          # Pointer ke node selanjutnya
7
```

Penjelasan:

- Ini adalah class untuk setiap elemen (node) dari double linked list.
- Setiap node menyimpan data, serta prev (ke node sebelum) dan next (ke node sesudahnya).

❖ DoubleLinkedList

```
8 # Kelas DoubleLinkedList
9 class DoubleLinkedList:
10     def __init__(self):
11         self.head = None          # Pointer ke node pertama dari linked list
12
```

Penjelasan:

- Ini adalah class utama untuk mengelola linked list.
- self.head menyimpan node pertama (bisa None jika kosong).

❖ Menambahkan Node (append)

```
13 # Menambahkan Node (append)
14 def append(self, data):
15     new_node = Node(data)         # Buat node baru dengan data
16     if not self.head:             # Jika list kosong
17         self.head = new_node
18         return
19     curr = self.head
20     while curr.next:              # Iterasi sampai akhir list
21         curr = curr.next
22     curr.next = new_node          # Set next dari node terakhir ke new_node
23     new_node.prev = curr          # Set prev dari new_node ke node terakhir
24
```

Penjelasan:

- Fungsi ini menambahkan node di akhir.
- Jika list kosong, node baru jadi head.
- Kalau tidak, cari node terakhir dan sambungkan new_node di belakangnya.

❖ Menampilkan Isi Linked List

```
25 # Menampilkan Isi Linked List
26 def display(self):
27     curr = self.head
28     while curr:
29         print(curr.data, end=" <-> ")
30         curr = curr.next
31     print("None")
32
```

Penjelasan:

- Menampilkan isi linked list dari depan ke belakang.
- Format output seperti: 10 <-> 20 <-> 30 <-> None

❖ Menghapus Node Awal

```
33 # Menghapus Node Awal
34 def delete_awal(self):
35     if not self.head:          # List kosong
36         print("Linked list kosong!")
37         return
38     if not self.head.next:      # Hanya ada 1 node
39         self.head = None
40     else:
41         self.head = self.head.next # Geser head ke node berikutnya
42         self.head.prev = None      # Putuskan hubungan ke node lama
43
```

Penjelasan:

- Hapus node pertama (head).
- Tangani kasus: kosong dan hanya satu elemen.

❖ Menghapus Node Akhir

```
44 # Menghapus Node Akhir
45 def delete_akhir(self):
46     if not self.head:
47         print("Linked list kosong!")
48         return
49     curr = self.head
50     if not curr.next:          # Hanya ada satu node
51         self.head = None
52         return
53     while curr.next:          # Cari node terakhir
54         curr = curr.next
55     curr.prev.next = None     # Putuskan hubungan dengan node terakhir
56
```

Penjelasan:

- Menghapus node terakhir.
- Jika hanya satu node, langsung kosongkan list.

❖ Menghapus Node Berdasarkan Nilai

```
57 # Menghapus Node Berdasarkan Nilai
58 def delete_berdasarkan_nilai(self, target):
59     if not self.head:
60         print("Linked list kosong!")
61         return
62     curr = self.head
63     if curr.data == target:
64         self.delete_awal()      # Hapus jika target ada di head
65         return
66     while curr:
67         if curr.data == target:
68             if curr.next:
69                 curr.prev.next = curr.next
70                 curr.next.prev = curr.prev
71             else:
72                 curr.prev.next = None # Hapus jika node terakhir
73             return
74         curr = curr.next
75     print(f"Data {target} tidak ditemukan dalam linked list.")
76
```

Penjelasan:

- Mencari node dengan nilai target dan menghapusnya.
- Tangani tiga kasus:
 1. Node di awal.
 2. Node di tengah.
 3. Node di akhir.

❖ Contoh Penggunaan

```
80 # Contoh Penggunaan
81 dll = DoubleLinkedList()
82 dll.append(10)
83 dll.append(20)
84 dll.append(30)
85 dll.append(40)
86
```

Penjelasan:

- Membuat linked list: 10 <-> 20 <-> 30 <-> 40

❖ Menampilkan dan Menghapus Node

```
87 # Menampilkan dan Menghapus Node
88 print("Linked list awal:")
89 dll.display()
90
91 print("\nHapus node awal:")
92 dll.delete_awal()
93 dll.display()
94
95 print("\nHapus node akhir:")
96 dll.delete_akhir()
97 dll.display()
98
99 print("\nHapus node dengan nilai 20:")
100 dll.delete_berdasarkan_nilai(20)
101 dll.display()
102
103 print("\nCoba hapus data yang tidak ada (50):")
104 dll.delete_berdasarkan_nilai(50)
```

Penjelasan:

- Menampilkan isi awal.
- Menghapus node pertama (10), hasil: 20 <-> 30 <-> 40
- Menghapus node terakhir (40), hasil: 20 <-> 30
- Hapus node dengan data 20, hasil: 30
- Mencoba hapus node yang tidak ada → muncul pesan: Data 50 tidak ditemukan dalam linked list.

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\PENYIMPANAN DATA\Struktur Data\Modul 2 Linked list> & C:/Users/user/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:/Users/user/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:\PENYIMPANAN DATA\Struktur Data\Modul 2 Linked list\tugaslinked.py
Linked list awal:
10 <-> 20 <-> 30 <-> 40 <-> None

Hapus node awal:
20 <-> 30 <-> 40 <-> None

Hapus node akhir:
20 <-> 30 <-> None

Hapus node dengan nilai 20:
30 <-> None

Coba hapus data yang tidak ada (50):
Data 50 tidak ditemukan dalam linked list.
PS C:\PENYIMPANAN DATA\Struktur Data\Modul 2 Linked list>
```