

**NAMA : WULAN AULIA**

**NIM : 24241029**

**MATKUL : STRUKTUR DATA**

## **MODUL 2**

class Node:

```
def __init__(self, data): # Perbaikan di sini
    self.data = data
    self.prev = None
    self.next = None
```

class DoubleLinkedList:

```
def __init__(self): # Perbaikan di sini
    self.head = None
```

# Tambah node di akhir

```
def append(self, data):
    new_node = Node(data)
    if self.head is None:
        self.head = new_node
        return
    curr = self.head
    while curr.next:
        curr = curr.next
    curr.next = new_node
    new_node.prev = curr
```

```
# Hapus node awal
def delete_front(self):
    if self.head is None:
        print("List kosong.")
        return
    print(f"Menghapus node awal: {self.head.data}")
    self.head = self.head.next
    if self.head:
        self.head.prev = None
```

```
# Hapus node akhir
def delete_end(self):
    if self.head is None:
        print("List kosong.")
        return
    curr = self.head
    while curr.next:
        curr = curr.next
    print(f"Menghapus node akhir: {curr.data}")
    if curr.prev:
        curr.prev.next = None
    else:
        self.head = None
```

```
# Hapus node berdasarkan nilai
def delete_by_value(self, value):
    curr = self.head
```

```
while curr:
    if curr.data == value:
        print(f"Menghapus node dengan nilai: {value}")
        if curr.prev:
            curr.prev.next = curr.next
        else:
            self.head = curr.next
        if curr.next:
            curr.next.prev = curr.prev
        return
    curr = curr.next
print(f"Data {value} tidak ditemukan.")
```

# Cetak semua data

```
def display(self):
    curr = self.head
    while curr:
        print(curr.data, end=" <-> ")
        curr = curr.next
    print("None")
```

# Contoh penggunaan

```
dll = DoubleLinkedList()
dll.append(10)
dll.append(20)
dll.append(30)
dll.append(40)
```

```
print("Linked List Awal:")
```

```
dll.display()
```

```
dll.delete_front()
```

```
dll.display()
```

```
dll.delete_end()
```

```
dll.display()
```

```
dll.delete_by_value(20)
```

```
dll.display()
```

### **outputnya :**

Linked List Awal:

10 <-> 20 <-> 30 <-> 40 <-> None

Menghapus node awal: 10

20 <-> 30 <-> 40 <-> None

Menghapus node akhir: 40

20 <-> 30 <-> None

Menghapus node dengan nilai: 20

30 <-> None

### **Penjelasannya :**

#### **1. Class Node**

- Baris ini membuat class Node, yaitu struktur dasar dari list.
- Fungsi `_init_` dijalankan saat objek Node dibuat.

- Self.data menyimpan nilai yang ingin ditaruh dalam node.
- Self.prev adalah penghubung ke node sebelumnya (default-nya kosong)

Self.next adalah penghubung ke node berikutnya (default-nya kosong).

## 2. Class DoubleLinkedList

Ini adalah class utama untuk daftar berantai ganda (double linked list).

Fungsi `_init_` membuat list kosong, dengan head menunjuk ke None.

## 3. Menambahkan Node di Akhir

- Membuat node baru dengan data yang diberikan.
- Jika list masih kosong, node baru langsung jadi kepala (head).
- Jika tidak kosong, kita cari node terakhir dengan perulangan.
- Setelah ketemu node terakhir, kita sambungkan node baru ke belakangnya, dan sebaliknya.

## 4. Menghapus Node Pertama (Depan)

- Cek apakah list kosong. Jika ya, tampilkan pesan.
- Tampilkan nilai yang dihapus, lalu pindahkan head ke node berikutnya.
- Jika masih ada node setelahnya, putuskan hubungan ke node yang dihapus tadi.

## 5. Menghapus Node Terakhir (Belakang)

- Jika list kosong, tampilkan pesan.
- Temukan node terakhir dengan perulangan.

Tampilkan nilai yang dihapus.

Jika node punya sebelumnya, putuskan koneksinya.

Jika tidak (hanya 1 node), kosongkan list.

## 6. Menghapus Node Berdasarkan Nilai

- Mulai dari head dan cari node yang cocok.
- Jika ditemukan, tampilkan bahwa node akan dihapus.
- Jika bukan node pertama, sambungkan node sebelumnya ke sesudahnya.

- Jika itu adalah node pertama, geser head.
- Jika ada node setelahnya, sambungkan ke node sebelumnya.

Keluar dari fungsi setelah menghapus

- Jika data tidak ditemukan setelah pencarian selesai, tampilkan pesan.

## 7. Menampilkan Isi List

- fungsi ini mencetak semua data dari awal sampai akhir.

Menambahkan panah "<->" sebagai penghubung antar node.

Di akhir, mencetak "None" untuk menandai akhir list.

## 8. Contoh Penggunaan

- Membuat objek list kosong

Menambahkan 4 data ke dalam list.

- Menampilkan isi list: 10 <-> 20 <-> 30 <-> 40 <-> None
- Menghapus data paling depan (10), lalu tampilkan hasilnya.
- Menghapus data paling belakang (40), lalu tampilkan hasilnya.
- Menghapus node dengan nilai 20, lalu tampilkan list yang tersisa.