```python
# Kelas Node untuk menyimpan data dan pointer ke node sebelumnya dan berikutnya
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
        self.next = None

# Kelas DoubleLinkedList untuk operasi dasar linked list
class DoubleLinkedList:
    def __init__(self):
        self.head = None

    # Tambah node di akhir
    def tambah_node(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_node
            new_node.prev = current

    # Tampilkan isi linked list
    def tampil(self):
        current = self.head
        if not current:
            print("Linked list kosong.")
            return
        print("Isi Linked List:")
        while current:
            print(current.data, end=" <-> " if current.next else "\n")
            current = current.next

    # Hapus node pertama
    def hapus_awal(self):
        if self.head is None:
            print("Linked list kosong, tidak ada yang bisa dihapus.")
            return
        print(f"Node dengan data '{self.head.data}' dihapus dari awal.")
        if self.head.next is None:
            self.head = None
        else:
            self.head = self.head.next
            self.head.prev = None

    # Hapus node terakhir
    def hapus_akhir(self):
        if self.head is None:
            print("Linked list kosong, tidak ada yang bisa dihapus.")
            return
        current = self.head
        if current.next is None:
            print(f"Node dengan data '{current.data}' dihapus dari akhir.")
            self.head = None
        else:
            while current.next:
                current = current.next
            print(f"Node dengan data '{current.data}' dihapus dari akhir.")
            current.prev.next = None

    # Hapus node berdasarkan nilai data
    def hapus_berdasarkan_nilai(self, nilai):
        if self.head is None:
            print("Linked list kosong.")
            return
        current = self.head
        while current:
            if current.data == nilai:
                print(f"Node dengan nilai '{nilai}' berhasil dihapus.")
                if current.prev:
                    current.prev.next = current.next
                else:
                    self.head = current.next
                if current.next:
                    current.next.prev = current.prev
                return
            current = current.next
        print(f"Node dengan nilai '{nilai}' tidak ditemukan.")

# Program utama
dll = DoubleLinkedList()
dll.tambah_node(10)
dll.tambah_node(20)
dll.tambah_node(30)
dll.tambah_node(40)

dll.tampil()                    # 10 <-> 20 <-> 30 <-> 40

dll.hapus_awal()              # Hapus node 10
dll.tampil()                  # 20 <-> 30 <-> 40

dll.hapus_akhir()             # Hapus node 40
dll.tampil()                  # 20 <-> 30

dll.hapus_berdasarkan_nilai(20)  # Hapus node 20
dll.tampil()                     # 30

dll.hapus_berdasarkan_nilai(99)  # Nilai tidak ditemukan
dll.tampil()                     # 30
```

Penjelasan Tugas Modul 2.md

```
# Kelas Node untuk menyimpan data dan pointer ke node sebelumnya dan berikutnya
class Node:
    def __init__(self, data):
        self.data = data         # Menyimpan data pada node
        self.prev = None         # Menunjuk ke node sebelumnya (double linked list)
        self.next = None         # Menunjuk ke node berikutnya
# Kelas DoubleLinkedList untuk operasi dasar linked list
class DoubleLinkedList:
    def __init__(self):
        self.head = None         # Inisialisasi head (kepala) linked list
    # Tambah node di akhir
    def tambah_node(self, data):
        new_node = Node(data)    # Buat node baru dengan data
        if self.head is None:
            self.head = new_node # Jika list kosong, node ini jadi head
        else:
            current = self.head
            while current.next:  # Loop ke node terakhir
                current = current.next
            current.next = new_node  # Sambungkan node baru di akhir
            new_node.prev = current  # Hubungkan balik ke node sebelumnya
    # Tampilkan isi linked list
    def tampil(self):
        current = self.head
        if not current:
            print("Linked list kosong.")
            return
        print("Isi Linked List:")
        while current:
```

```
        print(current.data, end=" <-> " if current.next else "\n")
        current = current.next
    # Hapus node pertama
    def hapus_awal(self):
        if self.head is None:
            print("Linked list kosong, tidak ada yang bisa dihapus.")
            return
        print(f"Node dengan data '{self.head.data}' dihapus dari awal.")
        if self.head.next is None:
            self.head = None
        else:
            self.head = self.head.next
            self.head.prev = None
    # Hapus node terakhir
    def hapus_akhir(self):
        if self.head is None:
            print("Linked list kosong, tidak ada yang bisa dihapus.")
            return
        current = self.head
        if current.next is None:
    print(f"Node dengan data '{current.data}' dihapus dari akhir.")
            self.head = None
        else:
            while current.next:
                current = current.next
            print(f"Node dengan data '{current.data}' dihapus dari akhir.")
            current.prev.next = None
    # Hapus node berdasarkan nilai data
    def hapus_berdasarkan_nilai(self, nilai):
        if self.head is None:
            print("Linked list kosong.")
```

```
            return
        current = self.head
        while current:
            if current.data == nilai:
                print(f"Node dengan nilai '{nilai}' berhasil dihapus.")
                if current.prev:
                    current.prev.next = current.next
                else:
                    self.head = current.next
                if current.next:
                    current.next.prev = current.prev
                return
            current = current.next
        print(f"Node dengan nilai '{nilai}' tidak ditemukan.")
# Program utama
dll = DoubleLinkedList()
dll.tambah_node(10)
dll.tambah_node(20)
dll.tambah_node(30)
dll.tambah_node(40)
dll.tampil()                # Tampilkan semua node
dll.hapus_awal()            # Hapus node pertama (10)
dll.tampil()                # Tampilkan hasil
dll.hapus_akhir()           # Hapus node terakhir (40)
dll.tampil()                # Tampilkan hasil
dll.hapus_berdasarkan_nilai(20)  # Hapus node dengan data 20
dll.tampil()                # Tampilkan hasil
dll.hapus_berdasarkan_nilai(99)  # Data 99 tidak ada
dll.tampil()                # Tampilkan hasil akhir
```