

LATIHAN LINKED LIST PERTEMUAN 3

Nama: Haikal Faqih Aufa

NIM: J0403251021

Kelas: B

Kelas Praktikum: P1

1. LATIHAN 1

A. SS

```
#!/usr/bin/python3
# Tugas Latihan 1 TGT ATTAWI
# Untuk menulis implementasi Singly linked list yang
# mampu menambahkan dan menghapus node
# Author : Haikal Faqih Aufa
# NIM : J0403251021
# Value : PGD-DI

# Class untuk menyimpan node(s) dan memiliki
# pointer untuk node selanjutnya
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None
    def __str__(self):
        return str(self.data)

# Class untuk menyimpan fungsi dan yang akan
# kembali sebuah list
class LinkedList:
    def __init__(self):
        self.head = None
    # Fungsi untuk menambahkan node baru, dan memiliki p
    # titik node yang di tambah
    def append(self,value):
        new_node = Node(value)
        if self.head == None:
            self.head = new_node
            return
        last = self.head
        while last.next:
            last = last.next
        last.next = new_node
    # Fungsi untuk menambah node baru
    def delete(self,value):
        temp = self.head
    # Memerlukan list diindeks
    if temp is None:
        print("List tidak ada")
        return
    # Mencari node yang ingin dihapus
    if temp.data == value:
        self.head = temp.next
        temp = None
        print("Data berhasil dihapus")
        return
    prev = None
    # Mencari node yang ingin dihapus
    while temp and temp.data != value:
        prev = temp
        temp = temp.next
    if temp is None:
        print("Data tidak ditemukan")
        return
    # Menghapus node
    prev.next = temp.next
    temp = None
    print("Data berhasil dihapus")
    # Menampilkan isi dari linked list
    def print_list(self):
        temp = self.head
        while temp:
            print(temp.data, end=" > ")
            temp = temp.next
        print("None")
    # Pengujian hasil
    if __name__ == "__main__":
        ll = LinkedList()
        ll.append(10)
        ll.append(20)
        ll.append(30)
        ll.append(40)
        print("Sesudah dimasukkan:")
        ll.print_list()
        ll.delete(10)
        print("Setelah dihapus:")
        ll.print_list()
```

B. OUTPUT

```
PS E:\ARSIP AKADEMIK IPB\SEMESTER 2 IPB\Algoritma dan Struktur Data\Praktik\Praktikk_J0403251021_HaikalFaqihAufa\Pertemu
an 3> py latihan1.py
Sebelum dihapus:
10 -> 20 -> 30 -> 40 -> None
Masukan data yang ingin dihapus: 10
Data berhasil dihapus.
Sesudah dihapus:
20 -> 30 -> 40 -> None
```

2. LATIHAN 3

A. SS



```
#-----#
# NAMA : LINDI LAJUHAN J
# STDIK : Pendidikan dan Kependidikan Pengaruan pada node tertentu
# Double linked list
#
# Nama : Haikal Faqih Aufa
# NIM : 30403251021
# Kelas : TPI-B1
#-----#
# Class untuk menyimpan data(isi nilai) dan next(Pontunjuk untuk node selanjutnya)
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None
#-----#
# Class untuk menyimpan fungsi aksi yang akan dilakukan pada linkedlist
class doublyLinkedList:
    def __init__(self):
        self.head = None
#-----#
# Fungsi untuk menambah data ke belakang (akhir list)
def append(self, data):
    new_node = Node(data)
    #-----
    # Kalau list masih kosong
    #-----
    if self.head is None:
        self.head = new_node
        return
    #-----
    # Kalau sudah ada isinya, cari node terakhir dulu
    #-----
    temp = self.head
    while temp.next:
        temp = temp.next
    #-----
    # Simpan pun node terakhir ke node baru
    #-----
    temp.next = new_node
    new_node.prev = temp
#-----#
# Fungsi untuk mencari data tertentu
def search(self, key):
    temp = self.head
    #-----
    # Cek satu-satu sampai habis
    #-----
    while temp:
        if temp.data == key:
            return True
        temp = temp.next
    return False
#-----#
# Fungsi untuk menampilkan semua isi list
def display(self):
    temp = self.head
    while temp:
        print(temp.data, end=" ")
        temp = temp.next
    print()
#-----#
# Membuat object dari Doubly linked list
#-----#
dll = doublyLinkedList()
#-----#
# Input data dari user (dipisahkan oleh koma)
#-----#
data_input = input("Masukkan elemen ke dalam Doubly linked List (pisahkan dengan koma): ")
data_list = data_input.split(",")
#-----#
# Memecah input menjadi list
#-----#
for item in data_list:
    dll.append(int(item.strip()))
#-----#
# Input data yang ingin dicari
#-----#
cari = int(input("Masukkan elemen yang ingin dicari: "))
#-----#
# Cek apakah datanya ada atau tidak
#-----#
if dll.search(cari):
    print(f"Elemen {cari} ditemukan dalam Doubly linked list.")
else:
    print(f"Elemen {cari} tidak ditemukan dalam Doubly linked list.")
```

B. OUTPUT

```
PS E:\ARSIP AKADEMIK IPB\SEMESTER 2 IPB\Algoritma dan Struktur Data\Praktik\Praktikk_J0403251021_HaikalFaqihAuFa\Pertemu  
an 3> py latihan3.py  
Masukkan elemen ke dalam Doubly Linked List (pisahkan dengan koma): 9,8,7,6,5  
Masukkan elemen yang ingin dicari: 9  
Elemen 9 ditemukan dalam Doubly Linked List.
```

3. LATIHAN 5

A. SS



```
=====
# TUGAS LIST LATIHAN 3
# Studi Kasus: Implementasikan Pencarian pada node tertentu
# Double linked list.

# Nama : Haikal Faqih AuFa
# NIM : J0403251021
# Kelas : TPL-B1
=====

=====
# Class untuk menyimpan data(sis nilai) dan next(Pemunjuk untuk node selanjutnya)
=====
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

=====
# Class untuk menyimpan fungsi aksi yang akan dilakukan pada LinkedList
t
=====
class LinkedList:
    def __init__(self):
        self.head = None

    # Fungsi untuk menambah data ke belakang (akhir list)
    def append(self, data):
        new_node = Node(data)

        if not self.head:
            self.head = new_node
            return

        temp = self.head
        while temp.next:
            temp = temp.next

        temp.next = new_node

    # Fungsi untuk menampilkan semua isi list
    def display(self):
        temp = self.head
        while temp:
            print(temp.data, end=" -> ")
            temp = temp.next
        print("null")

    # Fungsi untuk membalik isi list
    def reverse(self):
        prev = None
        current = self.head

        while current:
            next_node = current.next
            current.next = prev
            prev = current
            current = next_node

        self.head = prev

    # Membuat object LinkedList
=====
ll = LinkedList()

# Input dari user (pisahkan pakai koma)
=====
data_input = input("Masukkan elemen untuk Linked List: ")

# Ubah input jadi list
=====
data_list = data_input.split(",")

# Masukkan satu per satu ke Linked List
=====
for item in data_list:
    ll.append(int(item.strip()))

print("Linked list sebelum dibalik: ", end="")
ll.display()

ll.reverse()

print("Linked list setelah dibalik: ", end="")
ll.display()
```

B. OUTPUT

```
PS E:\ARSIP AKADEMIK IPB\SEMESTER 2 IPB\Algoritma dan Struktur Data\Praktik\Praktikk_J0403251021_HaikalFaqihAufa\Pertemu  
an 3> py latihan5.py  
Masukkan elemen untuk Linked List: 1,2,3,4,5,6,7  
Linked List sebelum dibalik: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> null  
Linked List setelah dibalik: 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1 -> null
```