TASK REPORT
ALGORITHMS AND DATA STRUCTURE
WEEK 5

Created by:

Onic Agustino

L200234275

X

INFORMATICS ENGINEERING

FACULTY OF COMMUNICATION AND INFORMATICS

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2024/2025

TASK

1. **Try the previous source code to make a linkedlist structure and use traverse, insert at the beginning, insert at last, insert at the middle and remove nodes method.**

Week5 > 1.py > ...

```python
class Node:
    def __init__(self, data):
        self.data = data
        self.pointer = None

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

    def traverse(self):
        temp = self.head
        while temp:
            print(temp.data, end=" -> ")
            temp = temp.pointer
        print("None")

    def insert_at_beginning(self, new_data):
        new_node = Node(new_data)
        new_node.pointer = self.head
        self.head = new_node

    def insert_at_end(self, new_data):
        new_node = Node(new_data)
        if not self.head:
            self.head = new_node
            return
        temp = self.head
        while temp.pointer:
            temp = temp.pointer
        temp.pointer = new_node

    def insert_at_middle(self, prev_node, new_data):
        if not prev_node:
            print("Previous node must be in the LinkedList.")
            return
        new_node = Node(new_data)
        new_node.pointer = prev_node.pointer
        prev_node.pointer = new_node

    def remove_node(self, key):
        temp = self.head
        if temp and temp.data == key:
```

```
            self.head = temp.pointer
            temp = None
            return
        prev = None
        while temp and temp.data != key:
            prev = temp
            temp = temp.pointer
        if not temp:
            return
        prev.pointer = temp.pointer
        temp = None

# Inisialisasi linked list
llist = LinkedList()
llist.insert_at_end(3)
llist.insert_at_end(5)
llist.insert_at_end(2)
llist.insert_at_end(6)
llist.insert_at_end(9)
llist.insert_at_end(7)

print("Linked List awal:")
llist.traverse()

# Insert di awal
llist.insert_at_beginning(1)
print("Setelah insert di awal:")
llist.traverse()

# Insert di akhir
llist.insert_at_end(10)
print("Setelah insert di akhir:")
llist.traverse()

# Insert di tengah setelah node dengan nilai 5
temp = llist.head
while temp and temp.data != 5:
    temp = temp.pointer
llist.insert_at_middle(temp, 4)
print("Setelah insert di tengah:")
llist.traverse()
```

```
81    # Hapus node dengan nilai 6
82    llist.remove_node(6)
83    print("Setelah menghapus node 6:")
84    llist.traverse()
```

Picture 1.1 the code.

```
PS D:\Semester 4\Algorithm_and_structuredata> & C:/Users/Acer/AppData/Local/Programs/Python/Python311/python.exe "d:/Semester 4/Algorithm_and_structuredata/Week5/1.py"
Linked List awal:
3 -> 5 -> 2 -> 6 -> 9 -> 7 -> None
Setelah insert di awal:
1 -> 3 -> 5 -> 2 -> 6 -> 9 -> 7 -> None
Setelah insert di akhir:
1 -> 3 -> 5 -> 2 -> 6 -> 9 -> 7 -> 10 -> None
Setelah insert di tengah:
1 -> 3 -> 5 -> 4 -> 2 -> 6 -> 9 -> 7 -> 10 -> None
Setelah menghapus node 6:
1 -> 3 -> 5 -> 4 -> 2 -> 9 -> 7 -> 10 -> None
```

Picture 1.2 the output.

2. **Build a single linked list which contains 6 nodes, where each node has an integer value. The value of the integers are 3,5,2,6,9,7. Then, sum all of these integers by traversing the linked list from the head node until the tail node.**

```
87    class Node:
88        def __init__(self, data):
89            self.data = data
90            self.pointer = None
91
92    class LinkedList:
93        def __init__(self):
94            self.head = None
95
96        def traverse(self):
97            temp = self.head
98            while temp:
99                print(temp.data, end=" -> ")
100               temp = temp.pointer
101           print("None")
102
103       def insert_at_end(self, new_data):
104           new_node = Node(new_data)
105           if not self.head:
106               self.head = new_node
107               return
108           temp = self.head
109           while temp.pointer:
110               temp = temp.pointer
111           temp.pointer = new_node
112
113       def sum_linked_list(self):
114           temp = self.head
115           total = 0
116           while temp:
117               total += temp.data
118               temp = temp.pointer
119           return total
120
```

```
121   # Inisialisasi linked list
122   llist = LinkedList()
123   llist.insert_at_end(3)
124   llist.insert_at_end(5)
125   llist.insert_at_end(2)
126   llist.insert_at_end(6)
127   llist.insert_at_end(9)
128   llist.insert_at_end(7)
129
130   print("Linked List:")
131   llist.traverse()
132
133   # Menjumlahkan nilai dalam linked list
134   total_sum = llist.sum_linked_list()
135   print("Total sum of linked list:", total_sum)
136
```

Picture 2.1 the code.

```
Linked List:
3 -> 5 -> 2 -> 6 -> 9 -> 7 -> None
Total sum of linked list: 32
```

Picture 2.2 the output.