LAPORAN TUGAS
ALGORITMA DAN STRUKTUR DATA
PEKAN 4 DAN 5

Disusun oleh:

Mhd. Farhan Lubis

L200220277

C

TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2023/2024

DAFTAR ISI

TUGAS

1. **Finding the Longest Word**

```
def find_longest_word(sentence):
    """
    Find the longest word in the given sentence.
    """
    # Mengembalikan pesan berupa kata terpanjang yang
dapat dengan
    # Mencari kata terpanjang dalam kalimat yang
diberikan
    # Dengan cara membagi kalimat menjadi kata-kata
    # Dan kemudian mencari kata terpanjang berdasarkan
panjangnya
    return f"The longest word is: {max(sentence.split(),
key=len)}"

print(find_longest_word('I love to learn python'))
print("\nProgram Completed!\n\n--- By L200220277 ---")
```

```
010709-algoritma-dan-struktur-data on ᚠ main
❯ py .\240313-array\assignments\longest.py
The longest word is: python


Program Completed!


--- By L200220277 ---
```

2. **Finding missing Items**

```
def find_missing_number(numbers_list):
    """
    Find the missing number in the given array of
numbers between 1 and 10.
    """
    # Mendefinisikan variabel missing_number
    # untuk menyimpan angka yang hilang
    missing_number = 0
    # Melakukan perulangan dari angka dari 1 sampai 10
    for number in range(1, 11):
        # Memeriksa jika angka pada jangkauan diatas
        # tidak ada pada numbers_list
        # maka missing_number diisi dengan angka
tersebut
        if number not in numbers_list:
```

```
            missing_number = number
    # Mengembalikan pesan serta angka yang hilang
    return f"The Missing Number is:
{str(missing_number)}"

print(find_missing_number([1, 2, 3, 4, 5, 6, 8, 9, 10]))
print("\nProgram Completed!\n\n--- By L200220277 ---")
```

```
010709-algoritma-dan-struktur-data on ᚠ main
❯ py .\240313-array\assignments\missing.py
The Missing Number is: 7


Program Completed!


--- By L200220277 ---
```

3. **Previous Source Code**

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

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

    def print_linked_list(self):
        print_val = self.head
        while print_val is not None:
            print(print_val.data)
            print_val = print_val.pointer

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

    def at_end(self, new_data):
        new_node = Node(new_data)
        laste = self.head
        while (laste.pointer != None):
            laste = laste.pointer
        laste.pointer = new_node
```

```python
    def in_between(self, middle_node, new_data):
        new_node = Node(new_data)
        new_node.pointer = middle_node.pointer
        middle_node.pointer = new_node

    def remove_node(self, remove_key):
        head_val = self.head
        if (head_val is not None):
            if (head_val.data == remove_key):
                self.head = head_val.pointer
                head_val = None
                return
        while (head_val is not None):
            if head_val.data == remove_key:
                break
            prev = head_val
            head_val = head_val.pointer
        if (head_val == None):
            return
        prev.pointer = head_val.pointer
        head_val = None

# Membuat instance LinkedList
ll = LinkedList()

# Menambahkan node baru di awal
print('Menambah angka 2 di awal list')
ll.at_beginning(2)
ll.print_linked_list()

# Menambahkan node baru di akhir
print('\nmenambahkan angka 8 diakhir list')
ll.at_end(8)
ll.print_linked_list()

# Menambahkan node baru setelah node tertentu
print('\nMenambahkan angka 4 setelah 2')
ll.in_between(ll.head, 4)
ll.print_linked_list()

# Menghapus node dengan nilai tertentu
print('\nMenghapus angka 4')
ll.remove_node(4)
ll.print_linked_list()

print("\nProgram Completed!\n\n--- By L200220277 ---")
```

```
010709-algoritma-dan-struktur-data on  main
❯ py .\240313-linear_list\assignments\try_previous.py
Menambah angka 2 di awal list
2

menambahkan angka 8 diakhir list
2
8

Menambahkan angka 4 setelah 2
2
4
8

Menghapus angka 4
2
8

Program Completed!

--- By L200220277 ---
```

4. Sum Integer in Linked List

```
class Node:
    def __init__(self, data=None):
        # Inisialisasi node dengan data yang diberikan
        self.data = data
        # Pointer untuk menunjuk ke node berikutnya,
awalnya diatur None
        self.pointer = None

class LinkedList:
    def __init__(self):
        # Inisialisasi linked list dengan kepala (head)
yang awalnya None
```

```
        self.head = None

    def print_linked_list(self):
        # Mencetak semua data dalam linked list
        print_val = self.head
        while print_val is not None:
            print(print_val.data)
            print_val = print_val.pointer

    def at_beginning(self, new_data):
        # Menambahkan node baru di awal linked list
        new_node = Node(new_data)
        new_node.pointer = self.head
        self.head = new_node

    def at_end(self, new_data):
        # Menambahkan node baru di akhir linked list
        new_node = Node(new_data)
        # Jika linked list kosong, node baru akan
menjadi kepala
        if self.head is None:
            self.head = new_node
            return
        # Jika tidak, mencari node terakhir dan
menambahkan node baru di belakangnya
        laste = self.head
        while laste.pointer is not None:
            laste = laste.pointer
        laste.pointer = new_node

    def in_between(self, middle_node, new_data):
        # Menambahkan node baru setelah node tertentu
        if middle_node is None:
            print("Node referensi tidak ada")
            return
        new_node = Node(new_data)
        new_node.pointer = middle_node.pointer
        middle_node.pointer = new_node

    def remove_node(self, remove_key):
        # Menghapus node dengan nilai tertentu
        head_val = self.head
        # Jika node yang dihapus adalah kepala
        # maka head_val diubah menjadi pointer node
selanjutnya
        if head_val is not None and head_val.data ==
remove_key:
            self.head = head_val.pointer
            head_val = None
            return
        # Mencari node yang akan dihapus
```

```
        while head_val is not None:
            if head_val.data == remove_key:
                break
            prev = head_val
            head_val = head_val.pointer
        # Jika node tidak ditemukan
        if head_val is None:
            return
        # Menghapus node dan mengatur pointer node
sebelumnya
        prev.pointer = head_val.pointer
        head_val = None

    def sum_linked_list(self):
        # Menghitung jumlah dari semua data dalam linked
list
        current = self.head
        total_sum = 0
        while current is not None:
            total_sum += current.data
            current = current.pointer
        # Mengembalikan total jumlah dan cetak jumlahnya
        return total_sum

# Membuat instance linked list
ll = LinkedList()

# Menambahkan 6 node baru di awal
print('Menambah angka 3,5,2,6,9,7 di awal list')
for i in [3,5,2,6,9,7]:
    ll.at_beginning(i)

# Menghitung semua data dalam linked list
print('\nMenghitung semua data')
ll.print_linked_list()
print('Total =', ll.sum_linked_list())

print("\nProgram Completed!\n\n--- By L200220277 ---")
```

```
010709-algoritma-dan-struktur-data on  main
❯ py .\240313-linear_list\assignments\sum_linked_lis
Menambah angka 3,5,2,6,9,7 di awal list

Menghitung semua data
7
9
6
2
5
3
Total = 32

Program Completed!

--- By L200220277 ---
```