**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

[DAFTAR ISI 2](#_Toc161782916)

[TUGAS 3](#_Toc161782917)

[1. Finding the Longest Word 3](#_Toc161782918)

[2. Finding missing Items 3](#_Toc161782919)

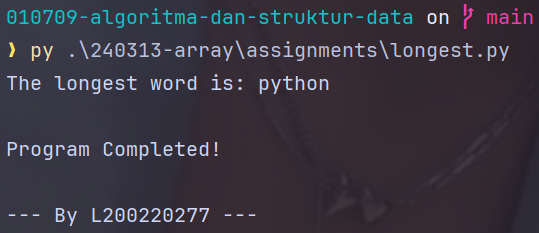
[3. Previous Source Code 4](#_Toc161782920)

[4. Sum Integer in Linked List 6](#_Toc161782921)

# TUGAS

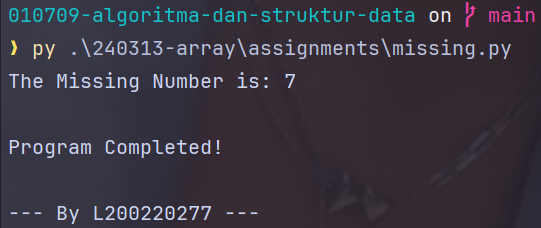
## Finding the Longest Word

|  |
| --- |
| 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 ---") |



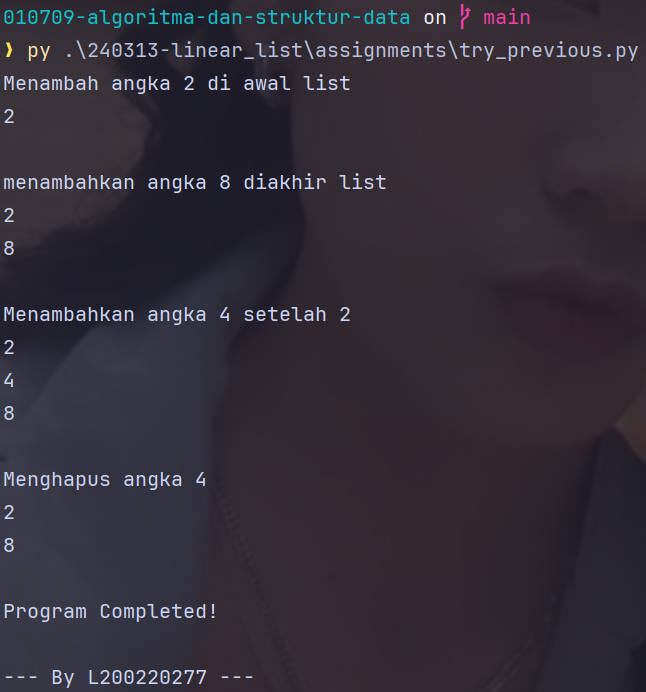
## Finding missing Items

|  |
| --- |
| 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 ---") |



## 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   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 ---") |



## 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 ---") |