

Nama : W. Fasial Hari Dewanto  
NIM : L200180046  
Kelas : B

## Modul 4

1.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 04\1.py
Nomer 1
Mahasiswa yang tinggal di Kota Klaten [6, 8]
>>> |

1.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    #membuat list
    internalData = list([None])

    #mengambil data di list
    def __getitem__(self, item):
        return self.internalData[item]

    #mengatur posisi data dan index-nya pada list
    def __setitem__(self, key, value):
        self.internalData[key] = value

    #01 cariKota(self, data):
    d = []
    t = 0
    for i in self:
        if i.kotaTinggal == data:
            d.append(t)
            t += 1
    return d

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Sukoharjo', 2400000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 2300000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 2500000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 2350000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 2400000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 2500000)
c[6] = MhsTIF('Deni', 13, 'Klaten', 2450000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 2450000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 2450000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 2700000)
c[10] = MhsTIF('Khalid', 29, 'Purwodadi', 2650000)

print ("Nomer 1")
print ("Mahasiswa yang tinggal di Kota klaten", c.cariKota('Klaten'))
```

2.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 04\2.py
Nomer 2
Uang saku yang terkecil adalah 230000
>>> |

2.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    #membuat list
    internalData = list([None])

    #mengambil data di list
    def __getitem__(self, item):
        return self.internalData[item]

    #mengatur posisi data dan index-nya pada list
    def __setitem__(self, key, value):
        self.internalData[key] = value

    #02 cariuangSaku(self):
    terkecil = self[0].uangSaku
    for i in self:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
    return terkecil

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Sukoharjo', 2400000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 2300000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 2500000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 2350000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 2400000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 2500000)
c[6] = MhsTIF('Deni', 13, 'Klaten', 2450000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 2450000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 2450000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 2700000)
c[10] = MhsTIF('Khalid', 29, 'Purwodadi', 2650000)

print ("Nomer 2")
print ("Uang saku yang terkecil adalah", c.cariuangSaku())
```

3.

The screenshot shows two windows. The left window is a Python 3.8.1 Shell with the following code:

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 03\3.py
True
False
2 14 12 22 21 1 9
>>> |
```

The right window is a Python script editor showing the implementation of a Singly Linked List:

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

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

    def pushho(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node

    def pushho(self, data):
        if (self.head == None):
            self.head = Node(data)
        else:
            current = self.head
            while (current.next != None):
                current = current.next
            current.next = Node(data)

    def insert(self, data, pos):
        node = Node(data)
        if not self.head:
            self.head = node
        elif pos==0:
            node.next = self.head
            self.head = node
        else:
            prev = None
            current = self.head
            current_pos = 0
            while (current_pos < pos) and current.next:
                prev = current
                current = current.next
                current_pos +=1
            prev.next = node
            node.next = current
            return self.head

    def deleteNode(self, position):
        if self.head == None:
            return
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
            return
        for i in range(position-1):
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        next = temp.next.next
        temp.next = next
        temp.next = None

    def search(self, x):
        current = self.head
        while current != None:
            if current.data == x:
```

4.

The screenshot shows two windows. The left window is a Python 3.8.1 Shell with the following code:

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 03\4.py
menambah pada awal 7
menambah pada awal 1
menambah pada akhir 6
menambah pada akhir 4

Dari Depan :
1
7
6
4

Dari Belakang :
4
6
7
1
>>> |
```

The right window is a Python script editor showing the implementation of a Doubly Linked List:

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

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

    def awal(self, new_data):
        print("menambah pada awal", new_data)
        new_node = Node(new_data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node

    def akhir(self, new_data):
        print("menambah pada akhir", new_data)
        new_node = Node(new_data)
        new_node.next = None
        if self.head is None:
            new_node.prev = None
            self.head = new_node
            return
        last = self.head
        while (last.next is not None):
            last = last.next
        last.next = new_node
        new_node.prev = last
        return

    def printList(self, node):
        print("\nDari Depan :")
        while (node is not None):
            print(" % d" % (node.data))
            node = node.next
        print("\nDari Belakang :")
        while (last is not None):
            print(" % d" % (last.data))
            last = last.prev

l1 = DoublyLinkedList()
l1.awal(7)
l1.awal(1)
l1.akhir(6)
l1.akhir(4)
l1.printList(l1.head)
```

5.

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 04\5.py
Data 10 ada dalam linked list
Data 110 tidak ada dalam linked list
>>>

5.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 0...
File Edit Format Run Options Window Help
class node(object):
def \_\_init\_\_(self, data, next = None):
self.data = data
self.next = next

def cariLinkedList(self, dicari):
curNode = self
while curNode is not None:
if curNode.next != None:
if curNode.data != dicari:
curNode = curNode.next
else:
print ("Data ", dicari, "ada dalam linked list")
break
elif curNode.next == None:
print ("Data ", dicari, "tidak ada dalam linked list")
break

a = node(12)
menu = a
a.next = node(34)
a = a.next
a.next = node(10)
a = a.next
a.next = node(45)

menu.cariLinkedList(10)
menu.cariLinkedList(110)

6.

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 04\6.py
listnya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 56
False

listnya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 62
target berada di index 4
>>>

6.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 0...
File Edit Format Run Options Window Help
def binary (kumpulan,target):
low = 0
high = len(kumpulan) -1

while low <= high:
mid = (high + low) // 2
if kumpulan[mid] == target:
return "target berada di index " + str(mid)
break

elif target < kumpulan[mid]:
high = mid - 1

else:
low = mid + 1

return False

listnya = [23, 51, 78, 10, 62, 145, 410]
target1 = 56
target2 = 62

print("listnya adalah ",listnya)
print("nilai target adalah ", target1)
print(binary(listnya, target1))

print("\nlistnya adalah ",listnya)
print("nilai target adalah ", target2)
print(binary(listnya, target2))

7.

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 04\7.py
Posisi data 6 pada list [2, 6, 5, 6, 4, 6, 7, 8, 6, 10, 14, 15] adalah [1, 3, 5, 8]
>>>

7.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak\_ASD\MODUL - 0...
File Edit Format Run Options Window Help
def binary (kumpulan,target):
low = 0
high = len(kumpulan) -1
listku = []

while low <= high:
if kumpulan[low] == target:
listku.append(low)
low += 1
else:
low += 1

return listku

s = [2,6,5,6,4,6,7,8,6,10,14,15]
dicari = 6
print("Posisi data ", dicari, " pada list ", s, "adalah ",binary(s, dicari))

8.

```

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL
- 04\8.py

Soal :
    Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
    menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
    maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya?

Jawab :
    Ada dua kemungkinan pola yang bisa digunakan.
    Misalkan, angka yang akan ditebak adalah 70.
    -POLA PERTAMA-
        a = nilai tebakan pertama // 2
        tebakan selanjutnya = nilai tebakan "lebih dari" + a
        *jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
        tetap nilai lebih dari sebelumnya
        a = a // 2
    SIMULASI
        tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
        "
        tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
        tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
        tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
        tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
        tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
        tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70
    -POLA KEDUA-
        menggunakan barisan geometri  $S_n = 2^n$ 
        barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
        Misal angka yang akan ditebak adalah 68
        Tebakan ke-1 : 64 dijawab lebih dari itu
        Tebakan ke-2 : 96 (dari 64 + 32) dijawab "Kurang dari itu"
        Tebakan ke-3 : 80 (dari 64 + 16) dijawab "Kurang dari itu"
        Tebakan ke-4 : 72 (dari 64 + 8) dijawab "Kurang dari itu"
        Tebakan ke-5 : 68 (dari 64 + 4) dijawab "Lebih dari itu"
        Tebakan ke-6 : 70 (dari 68 + 2) dijawab "TEPAT"

8.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...
File Edit Format Run Options Window Help
print("""
Soal :
    Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
    menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
    maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya?""")

print("""
Jawab :
    Ada dua kemungkinan pola yang bisa digunakan.
    Misalkan, angka yang akan ditebak adalah 70.
    -POLA PERTAMA-
        a = nilai tebakan pertama // 2
        tebakan selanjutnya = nilai tebakan "lebih dari" + a
        *jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
        tetap nilai lebih dari sebelumnya
        a = a // 2
    SIMULASI
        tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
        tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
        tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
        tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
        tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
        tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
        tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70
    -POLA KEDUA-
        menggunakan barisan geometri  $S_n = 2^n$ 
        barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
        Misal angka yang akan ditebak adalah 68
        Tebakan ke-1 : 64 dijawab lebih dari itu
        Tebakan ke-2 : 96 (dari 64 + 32) dijawab "Kurang dari itu"
        Tebakan ke-3 : 80 (dari 64 + 16) dijawab "Kurang dari itu"
        Tebakan ke-4 : 72 (dari 64 + 8) dijawab "Kurang dari itu"
        Tebakan ke-5 : 68 (dari 64 + 4) dijawab "Lebih dari itu"
        Tebakan ke-6 : 70 (dari 68 + 2) dijawab "TEPAT"
""")

```