

Aji Mustaqim
L200180141/E

TUGAS ALGOSTRUK

No 1

No 2

No 3

```
class MhsTIF(object) :
    def __init__(self,nama,um,kota,us) :
        self.nama = nama self.umur = um self.kotaTinggal = kota
        self.uangSaku = us
class Array(object) :
    internal_data = 11 * [None]
    def __getitem__(self, item) :
        return self.internal_data[item]
    def __setitem__(self, key, value) :
        self.internal_data[key] = value
##no 1
    def indexKota(self, data) :
        d = []
        t = 0
        for i in self :
            if i.kotaTinggal == data :
                d.append(t)
                t += 1
        return d
##no 2
    def uangTerkecil(self) :
        terkecil = self[0].uangSaku
        for i in self :
            if i.uangSaku < terkecil :
                terkecil = i.uangSaku
        return terkecil
##no 3
    def uangTerkecil2(self) :
        terkecil = self[0].uangSaku
        d = []
        for i in self :
            if i.uangSaku < terkecil :
                d.append((i.nama, i.umur, i.kotaTinggal, i.uangSaku))
        return d
```

```
>>> c.indexKota('Klaten')
[6, 8]
>>> c.uangTerkecil()
230000
>>> c.uangTerkecil2()
[('Budi', 51, 'Sragen', 230000), ('Chandra', 18, 'Surakarta', 23000)]
>>> |
```

No 4

```
##no 4
def uangTerkecil3(self) :
    terkecil = 250000
    d = []
    for i in self :
        if i.uangSaku < 250000 :
            d.append((i.nama, i.umur, i.kotaTinggal, i.uangSaku))
    for i in d :
        print (i)
```

```
c = Array()
c[0] = MhsTIF('Ika',10,'Sukoharjo', 240000)
c[1] = MhsTIF('Budi',51,'Sragen', 230000)
c[2] = MhsTIF('Ahmad',2,'Surakarta', 250000)
c[3] = MhsTIF('Chandra',18,'Surakarta', 235000)
c[4] = MhsTIF('Eka',4,'Boyolali', 240000)
c[5] = MhsTIF('Fandi',31,'Salatiga', 250000)
c[6] = MhsTIF('Deni',13,'Klaten', 245000)
c[7] = MhsTIF('Galuh',5,'Wonogiri', 245000)
c[8] = MhsTIF('Janto',23,'Klaten', 245000)
c[9] = MhsTIF('Hasan',64,'Karanganyar', 270000)
c[10] = MhsTIF('Khalid',29,'Purwodadi', 265000)|
```

===== RESTART: C:/Users/AJI TOSKA/Music/algo/Modul4/modul4.py =====

```
>>> c.uangTerkecil3()
('Ika', 10, 'Sukoharjo', 240000)
('Budi', 51, 'Sragen', 230000)
('Chandra', 18, 'Surakarta', 235000)
('Eka', 4, 'Boyolali', 240000)
('Deni', 13, 'Klaten', 245000)
('Galuh', 5, 'Wonogiri', 245000)
('Janto', 23, 'Klaten', 245000)
>>> |
```

No 5

```
## 5
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushAw(self, data_baru):
        node_baru = Node(data_baru)
        node_baru.next = self.head
        self.head = node_baru
    def pushAk(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)
        return self.head
    def insert(self, data, pos):
        node = Node(data)
        if not self.head:
            self.head = node
        elif posisi == 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 search(self, v):
    current = self.head
    while current != None:
        if current.data == v:
            return "True"
        current = current.next
    return "False"
def display(self):
    current = self.head
    while current != None:
        print(current.data)
        current = current.next

```

```

>>>
===== RESTART: C:/Users/AJI TOSKA/Music/algo/Modul4/modul4
.py =====
>>> A = LinkedList()
>>> A.pushAk(23)
<__main__.Node object at 0x02D4EED0>
>>> A.pushAk(24)
<__main__.Node object at 0x02D4EED0>
>>> A.display()
23
24
>>> A.pushAw(12)
>>> A.search(23)
'True'
>>> A.search(24)
'True'
>>> A.search(15)
'False'
>>> |

```

```

##no 6
def binSe(Daftar, Target) :
    low = 0
    high = len(Daftar) - 1

    while low <= high :
        mid = (high + low) // 2
        if Daftar[mid] == Target :
            return "Target berada pada index" + str(mid)
            break
        elif Target < Daftar[mid] :
            high = mid - 1
        else :
            low = mid + 1
    return False

listA = [12, 25, 34, 36, 57, 85, 90, 91]
Target1 = 29
Target2 = 57
print("\n6. List nya adalah", listA, "Nilai Target adalah ", Target1, "Hasil nya" , binSe(listA, Target1))
print("List nya adalah", listA, "Nilai Target adalah ", Target2, "Hasil nya" , binSe(listA, Target2))

>>>
===== RESTART: C:/Users/AJI TOSKA/Music/algo/Modul4/modul4.py =====
=====

6. List nya adalah [12, 25, 34, 36, 57, 85, 90, 91] Nilai Target adalah
29 Hasil nya False
List nya adalah [12, 25, 34, 36, 57, 85, 90, 91] Nilai Target adalah 57
Hasil nya Target berada pada index4
>>> |

```

No 7

```

##no 7
def binSe2(Daftar, Target) :
    loww = 0
    highh = len(Daftar) - 1
    listx = []

    while loww <= highh :
        if Daftar[loww] == Target :
            listx.append(loww)
            loww += 1
        else :
            loww += 1
    return listx

A = [2,3,5,6,6,6,8,9,9,10,11,12,13,13,14]
cari = 6

print ("\n7. Posisi data", cari, "pada list", A, "adalah", binSe2(A, cari))

--
--
>>>
===== RESTART: C:/Users/AJI TOSKA/Music/algo/Modul4/modul4.py =====

7. Posisi data 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] a
dalam [3, 4, 5]
>>> |

```

No 8

```
##no 8
def binSearch(kumpulan, target):
    """Mulai dari seluruh runtutan elemen"""
    low = 0
    high = len(kumpulan) - 1
    """Secara berulang belah runtutan itu menjadi separuhnya"""
    """sampai targetnya ditemukan"""
    while low <= high:
        """Temukan pertengahan runtut itu"""
        mid = (high + low) // 2
        """Apakah pertengahannya memuat target?"""
        if kumpulan[mid] == target:
            return mid
        elif kumpulan[mid] < target:
            high = mid + 1
        else :
            low = mid - 1

    return -1

b = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]

"""
untuk mencari berapa jumlah tebakan yang digunakan oleh Binary Search
yaitu dengan menggunakan Logaritma basis 2 (log2(n))
misalkan :
    // apabila terdapat elemen array berjumlah 100 maka memiliki maksimal 7 kali tebakan
    itu dikarenakan log2(100) = 6.643856189774725 sehingga diperoleh angka 7
    dapat juga diperoleh dari log2(128) = 7 karena yang mendekati dari 100 adalah 128
    // apabila terdapat elemen array berjumlah 1000 maka memiliki maksimal 10 kali tebakan
    itu dikarenakan log2(1000) = 9.965784284662087 sehingga diperoleh angka 10
    dapat juga diperoleh dari log2(1024) = 10 karena yang mendekati dari 1000 adalah 128
"""
```