Nama : Riska Meilina Puspa

NIM : L200180192

Kelas : G

Modul 4

Praktikum Algoritma dan Struktur Data

1.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def cariKotaTinggal(list, target):
    a = []
    for i in list :
        if i.kotaTinggal == target:
            a.append(list.index(i))
    return a

a = cariKotaTinggal(Daftar, "Klaten")
print(a)
```

```
[6, 8]
>>>
```

2.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def cariUangSakuTerkecil(list):
    temp = list[0].uangSaku
    for i in list[1:]:
        if i.uangSaku < temp:
            temp = i.uangSaku
    return temp

a = cariUangSakuTerkecil(Daftar)
print(a)


230000
>>>
```

3.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def cariUangSakuTerkecilObject(list):
    temp = [list[0]]
    for i in list[1:]:
        if i.uangSaku < temp[0].uangSaku:
            temp = [i]
        elif i.uangSaku == temp[0].uangSaku:
            temp.append(i)
    return temp

a = cariUangSakuTerkecilObject(Daftar)
print(a)
```

```
[<__main__.Mahasiswa object at 0x03510310>]
>>>
```

4.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def cariUangSakuKurang250k(list):
    temp = []
    for i in list:
        if i.uangSaku < 250000:
            temp.append(i)
    return temp

a = cariUangSakuKurang250k(Daftar)
for i in a:
    print(i.nama)
```

```
Wosoek
Minhae
Hangyul
Seungyoun
Seungwoo
Junho
Eunsang
>>>
```

5.

```
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(45)
menu = a
a.next = node (9)
a = a.next
a.next = node (17)
a = a.next
a.next = node (23)

menu.cariLinkedList(9)
menu.cariLinkedList(22)
```

```
Data 9 ada dalam Linked List
Data 22 tidak ada dalam Linked List
>>>
```

6.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan)-1
    while low <= high:
        mid = (high+low)//2
        if kumpulan[mid] == target:
            return mid
        elif target < kumpulan[mid]:
            high = mid-1
        else:
            low = mid+1
    return False

kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
print(binSe(kumpulan, 5))
```

```
2
>>>
```

7.

```
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

h0 = Mahasiswa("Wosoek", 100, "Sukoharjo", 240000)
h1 = Mahasiswa("Minhae", 133, "Sragen", 230000)
h2 = Mahasiswa("Riska", 192, "Surakarta", 250000)
h3 = Mahasiswa("Hangyul", 180, "Surakarta", 235000)
h4 = Mahasiswa("Seungyoun", 155, "Boyolali", 240000)
h5 = Mahasiswa("Yohan", 189, "Salatiga", 250000)
h6 = Mahasiswa("Seungwoo", 177, "Klaten", 245000)
h7 = Mahasiswa("Junho", 143, "Wonogiri", 245000)
h8 = Mahasiswa("Eunsang", 211, "Klaten", 245000)
h9 = Mahasiswa("Dohyun", 130, "Karanganyar", 270000)
h10 = Mahasiswa("Hyeongjun", 199, "Purwodadi", 265000)

Daftar = [h0, h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

def binSeMass(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan)-1
    while low <= high :
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midKiri = mid-1
            while kumpulan[midKiri] == target:
                temp.append(midKiri)
                midKiri = midKiri-1
            temp.append(mid)
            midKanan = mid+1
            while kumpulan[midKanan] == target:
                temp.append(midKanan)
                midKanan = midKanan+1
            return temp
        elif target < kumpulan[mid]:
            high = mid-1

        else:
            low = mid+1
    return False

kumpulan = [2, 4, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
print(binSeMass(kumpulan, 6))
```

```
[3, 4, 5]
>>>
```

8.

```
print
"""Karena menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus O(log n) dengan rincian 1 = 1, 2 = 2, 4 = 3, 10 = 4, 100 = 7, 1000=10.
Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui jumlah
maksimal tebakan.
Untuk pola sendiri:
        apabila ingin menebak angka 70

        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!!!
"""
```