

Nama : MuchFatan Rahmadan
NIM : L200180061
Kelas : C

Modul 6

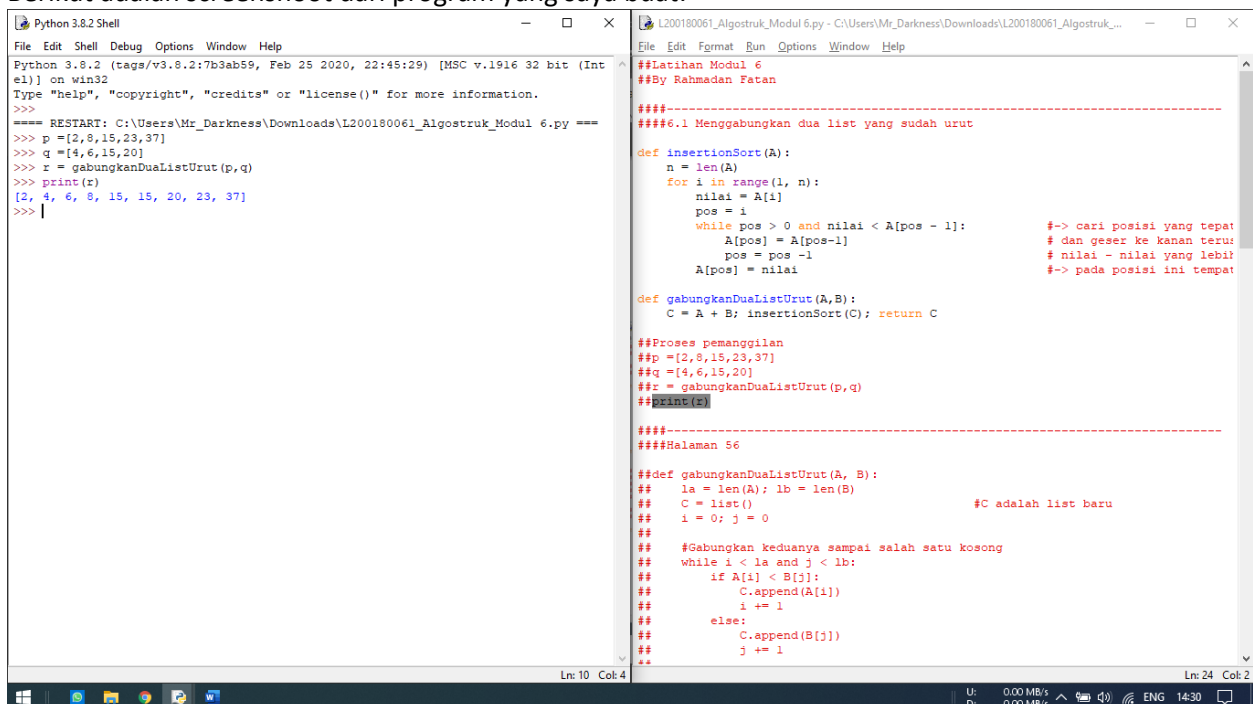
Pengaturan Lanjutan

6.1

Hal 55

Membuat suatu program untuk mengurutkan dua buah list menjadi sebuah list baru yg urut mulai dr yg terkecil-hingga terbesar

Berikut adalah screenshoot dari program yang saya buat:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_Modul 6.py ====
>>> p = [2,8,15,23,37]
>>> q = [4,6,15,20]
>>> r = gabungkanDuaListUrut(p,q)
>>> print(r)
[2, 4, 6, 8, 15, 15, 20, 23, 37]
>>>

L200180061_Algostruk_Modul 6.py - C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_...
File Edit Format Run Options Window Help
##Latihan Modul 6
##By Rahmadan Fatan

####-----
####6.1 Menggabungkan dua list yang sudah urut

def insertionSort(A):
    n = len(A)
    for i in range(1, n):
        nilai = A[i]
        pos = i
        while pos > 0 and nilai < A[pos - 1]:
            #-> cari posisi yang tepat
            A[pos] = A[pos-1]
            # dan geser ke kanan terus
            pos = pos - 1
            # nilai - nilai yang lebih
        A[pos] = nilai
        #-> pada posisi ini tempat

def gabungkanDuaListUrut(A,B):
    C = A + B; insertionSort(C); return C

##Proses pemanggilan
##p = [2,8,15,23,37]
##q = [4,6,15,20]
##r = gabungkanDuaListUrut(p,q)
##print(r)

####-----
####Halaman 56

##def gabungkanDuaListUrut(A, B):
##    la = len(A); lb = len(B)
##    C = list()
##    i = 0; j = 0
##    #C adalah list baru
##    #Gabungkan keduanya sampai salah satu kosong
##    while i < la and j < lb:
##        if A[i] < B[j]:
##            C.append(A[i])
##            i += 1
##        else:
##            C.append(B[j])
##            j += 1
##    #Salah satu list sudah habis, tambahkan sisa dari list lainnya
##    while i < la:
##        C.append(A[i])
##        i += 1
##    while j < lb:
##        C.append(B[j])
##        j += 1
##    return C
```

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_Modul 6.py =====
>>> p = [2,8,15,23,37]
>>> q = [4,6,15,20]
>>> r = gabungkanDuaListUrut(p,q)
>>> print(r)
[2, 4, 6, 8, 15, 15, 20, 23, 37]
>>>
===== RESTART: C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_Modul 6.py =====
>>> p = [2,8,15,23,37]
>>> q = [4,6,15,20]
>>> r = gabungkanDuaListUrut(p,q)
>>> print(r)
[2, 4, 6, 8, 15, 15, 20, 23, 37]
>>>

L200180061_Algostruk_Modul 6.py - C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_...
File Edit Format Run Options Window Help
#####Halaman 56#####
def gabungkanDuaListUrut(A, B):
    la = len(A); lb = len(B)
    C = list()
    i = 0; j = 0
    #C adalah list baru
    #Gabungkan keduanya sampai salah satu kosong
    while i < la and j < lb:
        if A[i] < B[j]:
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1
    while i < la:
        C.append(A[i])
        i += 1
    while j < lb:
        C.append(B[j])
        j += 1
    return C
#####
####6.2 Merge Sort
####Penyelesaian: Fungsi Rekursif
def faktorial(a):
    if a==1:
        return 1
    else:
        return a*faktorial(a-1)
#####
Ln: 17 Col: 4
U: 0.00 MB/s
D: 0.00 MB/s
ENG 14:31
```

Hal 56-60

Merge Sort

Membuat suatu program untuk mengurutkan nilai angka menggunakan Teknik strategi divide and conquer / bagi dan taklukkan

Berikut adalah screenshoot dari program yang saya buat:

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_Modul 6.py =====
>>> alist = [54,26,93,17,77,31,44,55,20]
>>> mergeSort(alist)
Membelah A [54, 26, 93, 17, 77, 31, 44, 55, 20]
Membelah A [54, 26, 93, 17]
Membelah A [54, 26]
Membelah A [54]
Menggabungkan [54]
Membelah A [26]
Menggabungkan [26]
Menggabungkan [26, 54]
Membelah A [93, 17]
Membelah A [93]
Menggabungkan [93]
Membelah A [17]
Menggabungkan [17]
Menggabungkan [17, 93]
Menggabungkan [17, 26, 54, 93]
Membelah A [77, 31, 44, 55, 20]
Membelah A [77, 31]
Menggabungkan [77]
Menggabungkan [31]
Menggabungkan [31, 77]
Membelah A [44, 55, 20]
Membelah A [44]
Menggabungkan [44]
Membelah A [55, 20]
Membelah A [55]
Menggabungkan [55]
Membelah A [20]
Menggabungkan [20]
Menggabungkan [20, 55]
Menggabungkan [20, 44, 55]
Menggabungkan [20, 31, 44, 55, 77]
Menggabungkan [17, 20, 26, 31, 44, 54, 55, 77, 93]
>>> print(alist)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>

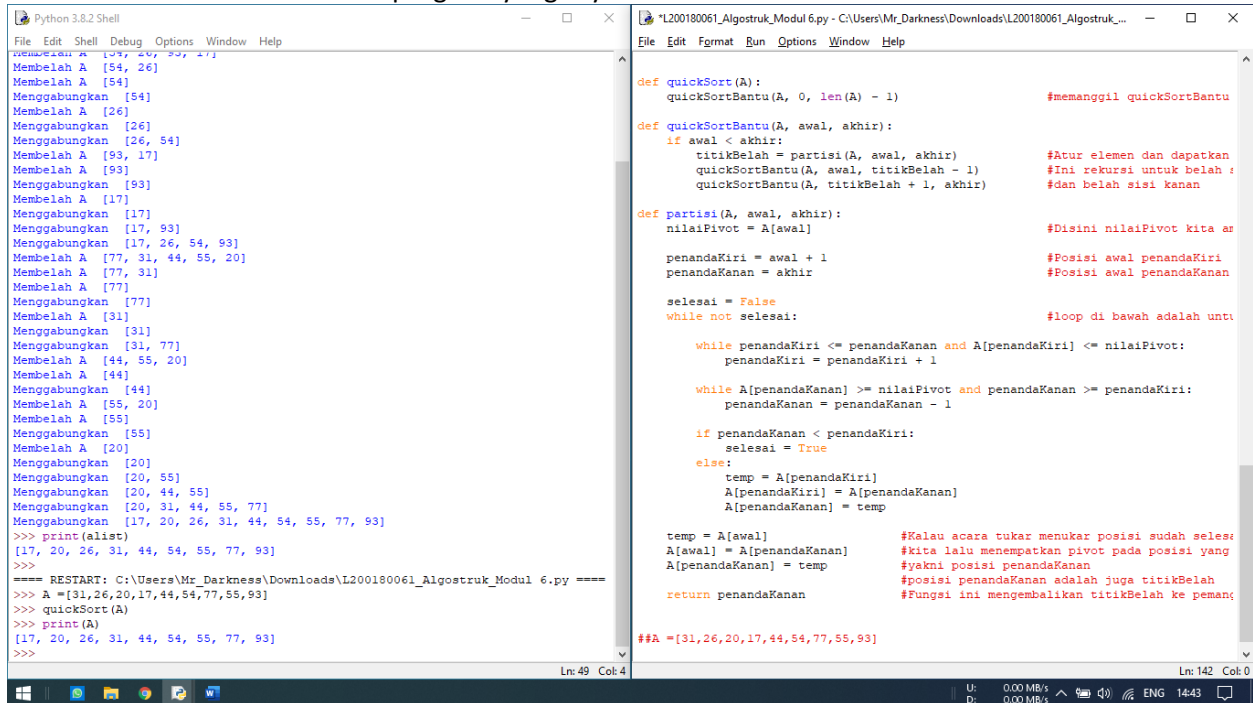
L200180061_Algostruk_Modul 6.py - C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_...
File Edit Format Run Options Window Help
#####Halaman 57 - 60#####
def mergeSort(A):
    print("Membelah A ", A)
    if len(A) > 1:
        mid = len(A) // 2
        separuhKiri = A[:mid]
        separuhKanan = A[mid:]
        mergeSort(separuhKiri)
        mergeSort(separuhKanan)
        #Ini rekursi. Memanggil lebih lan:
        #Untuk separuhKiri dan separuhKanan
        #Di bawah ini adalah proses penggabungan
        i = 0; j = 0; k = 0
        while i < len(separuhKiri) and j < len(separuhKanan):
            if separuhKiri[i] < separuhKanan[j]:
                A[k] = separuhKiri[i]
                i = i + 1
            else:
                A[k] = separuhKanan[j]
                j = j + 1
            k = k + 1
        while i < len(separuhKiri):
            A[k] = separuhKiri[i]
            i = i + 1
            k = k + 1
        while j < len(separuhKanan):
            A[k] = separuhKanan[j]
            j = j + 1
            k = k + 1
        print("Menggabungkan ", A)
#####
####6.3 Quick Sort
####Halaman 61 - 63
def partisi(A, awal, akhir):
    nilaiPivot = A[awal]
    #Disini nilaiPivot kita as
#####
Ln: 43 Col: 4
U: 0.00 MB/s
D: 0.00 MB/s
ENG 14:38
```

Hal 61-63

Quick Sort

Membuat suatu program untuk mengurutkan nilai angka menggunakan Teknik belah dan taklukan sama dgn merge sort namun tidak memerlukan penyimpanan tambahan

Berikut adalah screenshoot dari program yang saya buat:



The screenshot shows two windows. The left window is a Python 3.8.2 Shell with the following output:

```
Membelah A [17, 20, 26, 31, 44, 54, 55, 77, 93]
Membelah A [54, 26]
Membelah A [54]
Menggabungkan [54]
Membelah A [26]
Menggabungkan [26]
Menggabungkan [26, 54]
Membelah A [93, 17]
Membelah A [93]
Menggabungkan [93]
Membelah A [17]
Menggabungkan [17]
Menggabungkan [17, 93]
Menggabungkan [17, 26, 54, 93]
Membelah A [77, 31, 44, 55, 20]
Membelah A [77, 31]
Membelah A [77]
Menggabungkan [77]
Membelah A [31]
Menggabungkan [31]
Menggabungkan [31, 77]
Membelah A [44, 55, 20]
Membelah A [44]
Menggabungkan [44]
Membelah A [55, 20]
Membelah A [55]
Menggabungkan [55]
Membelah A [20]
Menggabungkan [20]
Menggabungkan [20, 55]
Menggabungkan [20, 44, 55]
Menggabungkan [20, 31, 44, 55, 77]
Menggabungkan [17, 20, 26, 31, 44, 54, 55, 77, 93]
>>> print(alist)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
==== RESTART: C:\Users\Mr_Darkness\Downloads\L200180061_Algostruk_Modul 6.py ====
>>> A = [31,26,20,17,44,54,77,55,93]
>>> quickSort(A)
>>> print(A)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

The right window is a Python script editor showing the following code:

```
def quickSort(A):
    quickSortBantu(A, 0, len(A) - 1)

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah - 1)
        quickSortBantu(A, titikBelah + 1, akhir)

def partisi(A, awal, akhir):
    nilaiPivot = A[awal]
    penandaKiri = awal + 1
    penandaKanan = akhir
    selesai = False
    while not selesai:
        while penandaKiri <= penandaKanan and A[penandaKiri] <= nilaiPivot:
            penandaKiri = penandaKiri + 1
        while A[penandaKanan] >= nilaiPivot and penandaKanan >= penandaKiri:
            penandaKanan = penandaKanan - 1
        if penandaKanan < penandaKiri:
            selesai = True
        else:
            temp = A[penandaKiri]
            A[penandaKiri] = A[penandaKanan]
            A[penandaKanan] = temp
    temp = A[awal]
    A[awal] = A[penandaKanan]
    A[penandaKanan] = temp
    return penandaKanan

##A = [31,26,20,17,44,54,77,55,93]
```