

Nama : Annisa Nugraheni

NIM : L200180066

Kelas : C

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

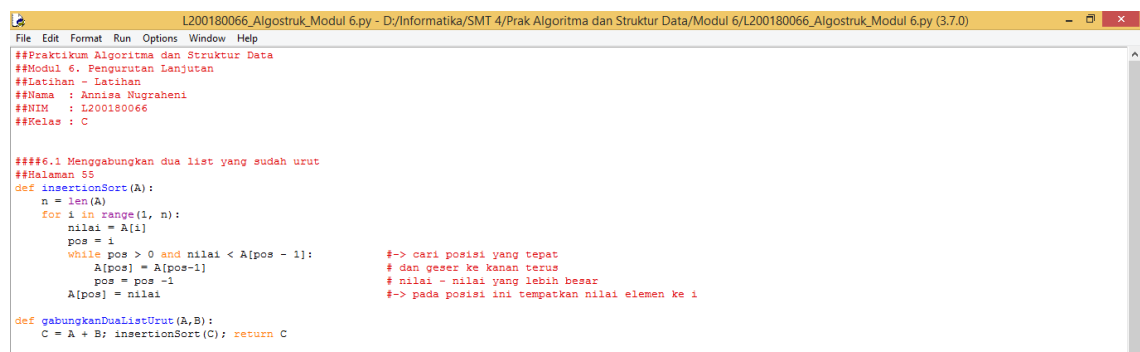
### MODUL 6

### PENGURUTAN LANJUTAN

#### ❖ 6.1 Menggabungkan dua list yang sudahurut

##### ➤ Halaman 55

Berikut adalah screenshot dari program:

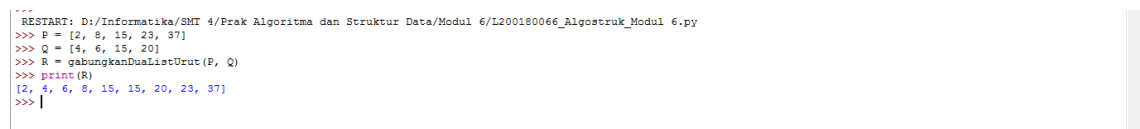


```
L200180066_Algostruk_Modul 6.py - D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 6/L200180066_Algostruk_Modul 6.py (3.7.0)
File Edit Format Run Options Window Help
##Praktikum Algoritma dan Struktur Data
##Modul 6, Pengurutan Lanjutan
##Latihan - Latihan
##Nama : Annisa Nugraheni
##NIM : L200180066
##Kelas : C

###6.1 Menggabungkan dua list yang sudahurut
##Halaman 55
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]:
            A[pos] = A[pos-1]
            pos = pos - 1
        A[pos] = nilai
        #-> cari posisi yang tepat
        # dan geser ke kanan terus
        # nilai - nilai yang lebih besar
        #-> pada posisi ini tempatkan nilai elemen ke i

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

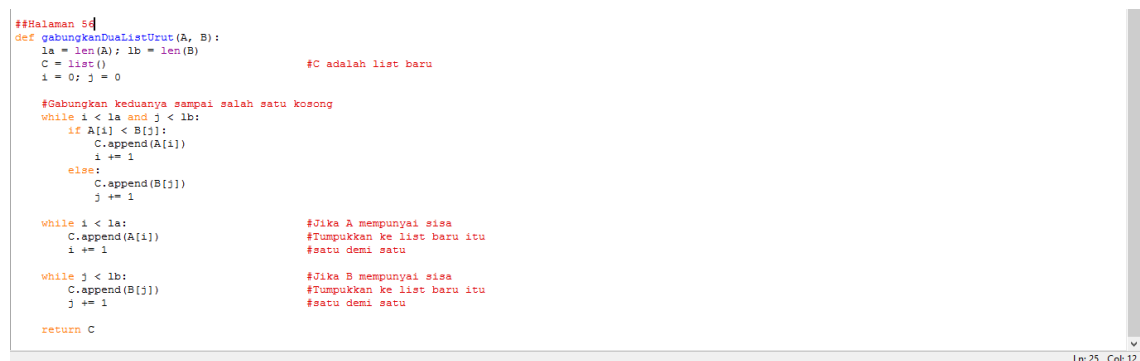
Berikut adalah screenshot dari hasil ketika program dijalankan:



```
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 6/L200180066_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]
>>> |
```

##### ➤ Halaman 56

Berikut adalah screenshot dari program:



```
##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
    #Jika A mempunyai sisa
    #Tumpukkan ke list baru itu
    #satu demi satu

    while j < lb:
        C.append(B[j])
        j += 1
    #Jika B mempunyai sisa
    #Tumpukkan ke list baru itu
    #satu demi satu

    return C
```

Berikut adalah screenshot dari ketika program dijalankan:

```
'''
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 6/L200180066_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]
>>> |
```

## ❖ 6.2 Merge Sort

### ➤ Halaman 57. Penyegaran: Fungsi Rekursif

Berikut adalah screenshot dari program:

```
####6.2 Merge Sort
##Halaman 57
##Penyegaran: Fungsi Rekursif
def faktorial(a):
    if a==1:
        return 1
    else:
        return a*faktorial(a-1)

Ln: 58 Col: 31
```

Berikut adalah screenshot dari hasil ketika program dijalankan:

```
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 6/L200180066_Algostruk_Modul 6.py
>>> faktorial(1)
1
>>> faktorial(2)
2
>>> faktorial(3)
6
>>> faktorial(4)
24
>>> faktorial(7)
5040
>>> |

Ln: 47 Col: 4
```

### ➤ Halaman 57 – 60

Berikut adalah screenshot dari program ketika perintah print(“Membelah ”, A) dan print(“Menggabungkan ”, A) belum diaktifkan :

```
####Halaman 57 – 59
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)

        #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)

Ln: 92 Col: 31
```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

```
'''
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 6/L200180066_Algostruk_Modul 6.py
>>> alist = [54, 26, 93, 17, 77, 31, 44, 55, 20]
>>> mergeSort(alist)
>>> print(alist)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>> |

Ln: 67 Col: 4
```

Berikut adalah screenshot dari program ketika perintah print(“Membelah ”, A) dan print(“Menggabungkan ”, A) sudah diaktifkan :

```
####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)

        #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)
```

Berikut adalah screenshot dari hasil ketika program diatas dijalankan:

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>>
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data/Modul 6/L200180066_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]
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]
>>>
```

## ❖ 6.3 Quick Sort

### ➤ Halaman 61 – 63

Berikut adalah screenshot dari program:

```
L200180066_Algostruk_Modul 6.py - D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data/Modul 6/L200180066_Algostruk_Modul 6.py (3.7.0)
File Edit Format Run Options Window Help
####6.3 Quick Sort
####Halaman 61 – 63
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

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

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

Berikut adalah screenshot dari hasil ketika program dijalankan:

```
RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 6\L200180066_Algostruk_Modul 6.py
>>> P = [54, 26, 93, 17, 77, 31, 44, 55, 20]
>>> partisi(P, 0, 8)
5
>>> quickSort(P)
>>> print(P)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>> |
```