

Nama : Afiq Tri Nugraha

NIM : L200180080

Kelas : C

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

MODUL 6

6.1 Menggabungkan dua list yang sudahurut

- Halaman 55

```
L200180080_Algostruk_Modul 6.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_Algostruk_Modul 6.py (3.7.0)
File Edit Format Run Options Window Help

##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

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_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

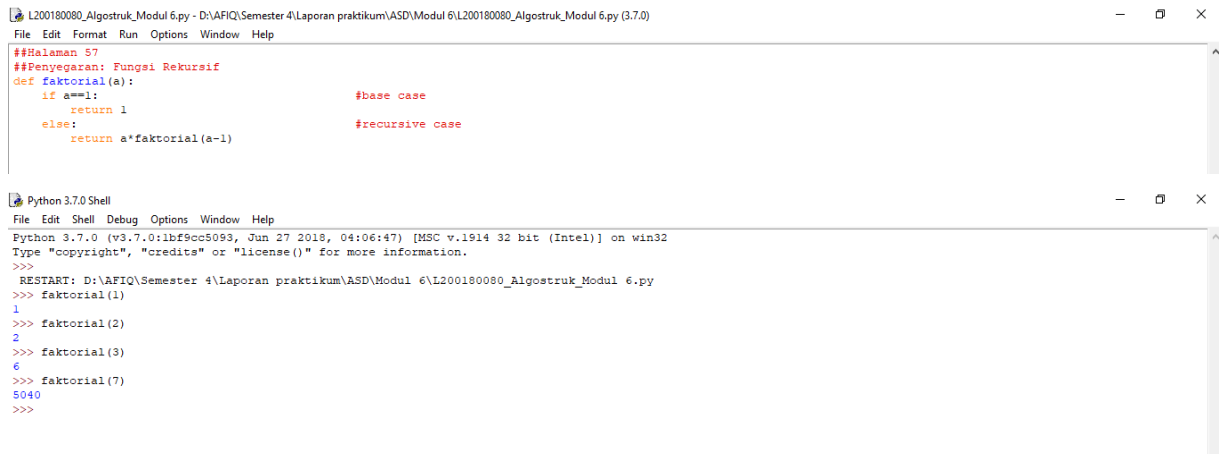
```
L200180080_Algostruk_Modul 6.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_Algostruk_Modul 6.py (3.7.0)
File Edit Format Run Options Window Help

##Halaman 56
def gabungkanDuaListUrut(A, B):
    la = len(A); lb = len(B)
    C = list()
    i = 0; j = 0
    #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
    #Jika A mempunyai sisa
    #Tumpukkan ke list baru itu
    #satu demi satu
    #Jika B mempunyai sisa
    #Tumpukkan ke list baru itu
    #satu demi satu

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_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



The screenshot shows a Python IDE with two windows. The top window, titled 'L200180080_Algostruk_Modul 6.py', contains a recursive function `faktorial` defined as follows:

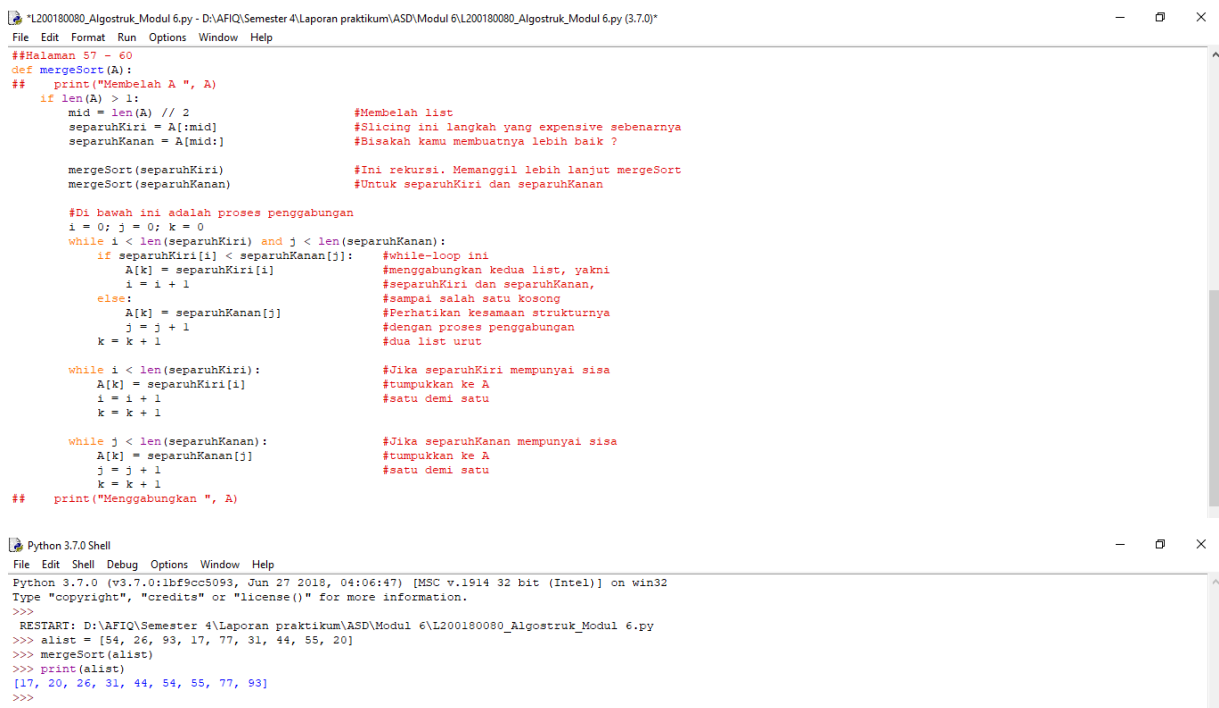
```
##Halaman 57
##Penyegaran: Fungsi Rekursif
def faktorial(a):
    if a==1:
        return 1
    else:
        return a*faktorial(a-1)
```

The bottom window, titled 'Python 3.7.0 Shell', shows the execution of the function:

```
>>> faktorial(1)
1
>>> faktorial(2)
2
>>> faktorial(3)
6
>>> faktorial(7)
5040
>>>
```

- Halaman 57 – 60

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



The screenshot shows a Python IDE with two windows. The top window, titled 'L200180080_Algostruk_Modul 6.py', contains a `mergeSort` function defined as follows:

```
##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

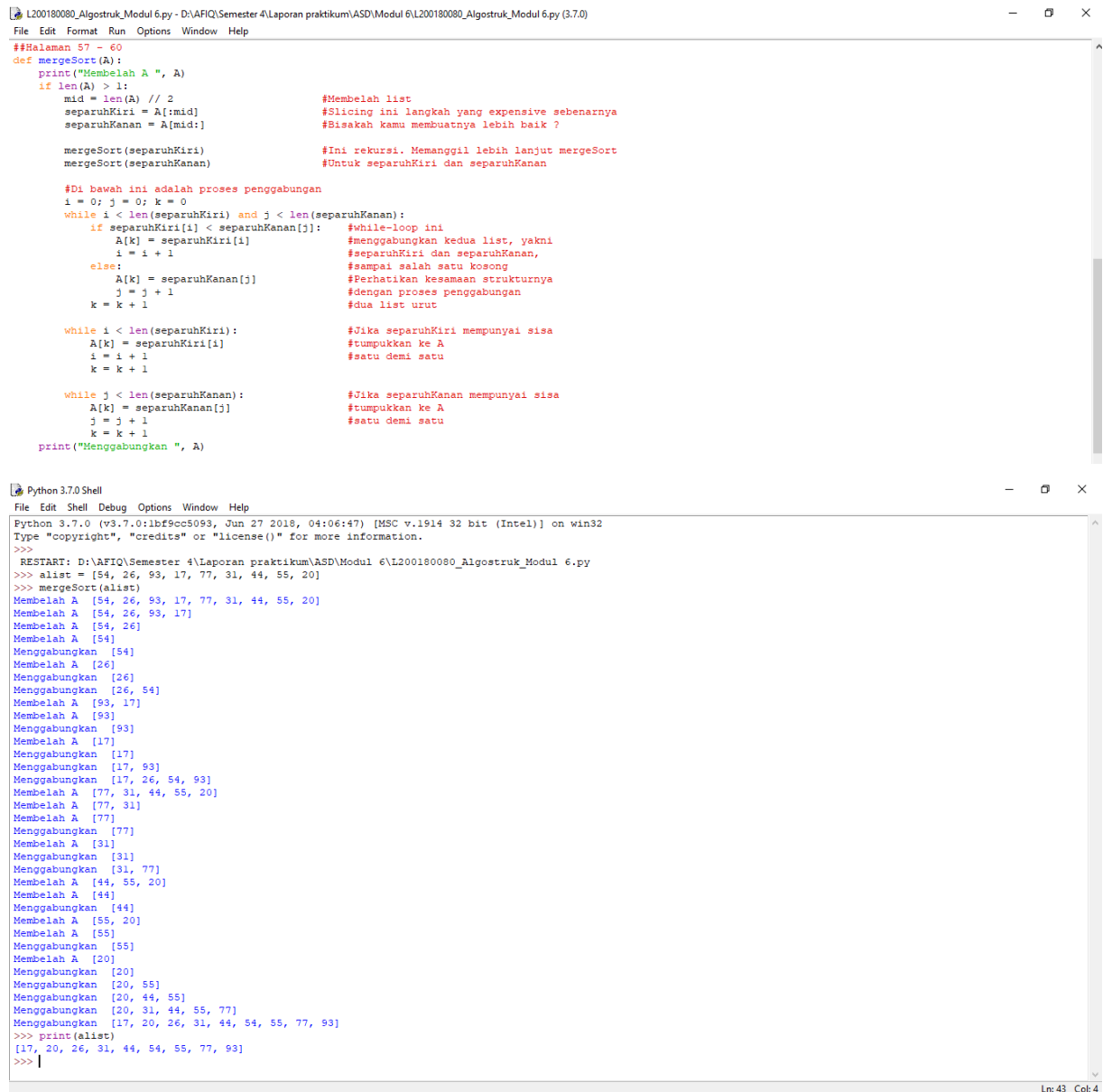
        while j < len(separuhKanan):
            A[k] = separuhKanan[j]
            j = j + 1
            k = k + 1

    ## print("Menggabungkan ", A)
```

The bottom window, titled 'Python 3.7.0 Shell', shows the execution of the function:

```
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_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]
>>>
```

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



The image shows two windows from a Python IDE. The top window displays a Python script for a merge sort algorithm. The script includes comments in Indonesian explaining the steps: splitting the list, recursive calls, and merging. The bottom window shows the execution output, where the list [54, 26, 93, 17, 77, 31, 44, 55, 20] is repeatedly split into halves until single elements are reached, and then merged back in sorted order.

```
L200180080_Algostruk_Modul 6.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_Algostruk_Modul 6.py (3.7.0)
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)

        #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)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

Python 3.7.0 (vs3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_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]
>>>
```

Ln: 43 Cok 4

6.3 Quick Sort

- Halaman 61 – 63

```
L200180080_Algostruk_Modul 6.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_Algostruk_Modul 6.py (3.7.0)
File Edit Format Run Options Window Help

##Halaman 61 – 63
def partisi(A, awal, akhir):
    nilaiPivot = A[awal]                                #Disini nilaiPivot kita ambil dari elemen yang paling kiri

    penandaKiri = awal + 1                               #Posisi awal penandaKiri
    penandaKanan = akhir                                 #Posisi awal penandaKanan

    selesai = False                                     #loop di bawah adalah untuk mengatur ulang posisi semua elemen
    while not selesai:
        while penandaKiri <= penandaKanan and A[penandaKiri] <= nilaiPivot:    #penandaKiri bergerak ke kanan sampai ketemu suatu nilai yang
            penandaKiri = penandaKiri + 1                                     #lebih besar dari nilaiPivot

        while A[penandaKanan] >= nilaiPivot and penandaKanan >= penandaKiri:    #penandaKanan bergerak ke kiri sampai ketemu suatu nilai yang
            penandaKanan = penandaKanan - 1                                     #lebih kecil dari nilaiPivot

        if penandaKanan < penandaKiri:    #Kalau dua penanda sudah bersilangan
            selesai = True                #selesai & lanjut ke penempatan pivot
        else:
            temp = A[penandaKiri]          #tapi kalau belum bersilangan
            A[penandaKiri] = A[penandaKanan]    #tukarlah isi yang ditunjuk oleh
            A[penandaKanan] = temp             #penandaKiri dan penandaKanan

    temp = A[awal]                         #Kalau acara tukar menukar posisi sudah selesai
    A[awal] = A[penandaKanan]              #kita lalu menempatkan pivot pada posisi yang tepat
    A[penandaKanan] = temp                  #yakni posisi penandaKanan
                                           #posisi penandaKanan adalah juga titikBelah
    return penandaKanan                    #Fungsi ini mengembalikan titikBelah ke pemanggil

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)    #Atur elemen dan dapatkan titikBelah
        quickSortBantu(A, awal, titikBelah - 1) #Ini rekursi untuk belah sisi kiri
        quickSortBantu(A, titikBelah + 1, akhir) #dan belah sisi kanan

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

|
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 6\L200180080_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]
>>> |
```