RESPONSI

Sistem Operasi Praktik-V

Oleh :

520041123 – Marhani Wiji Ayu Kusumawati

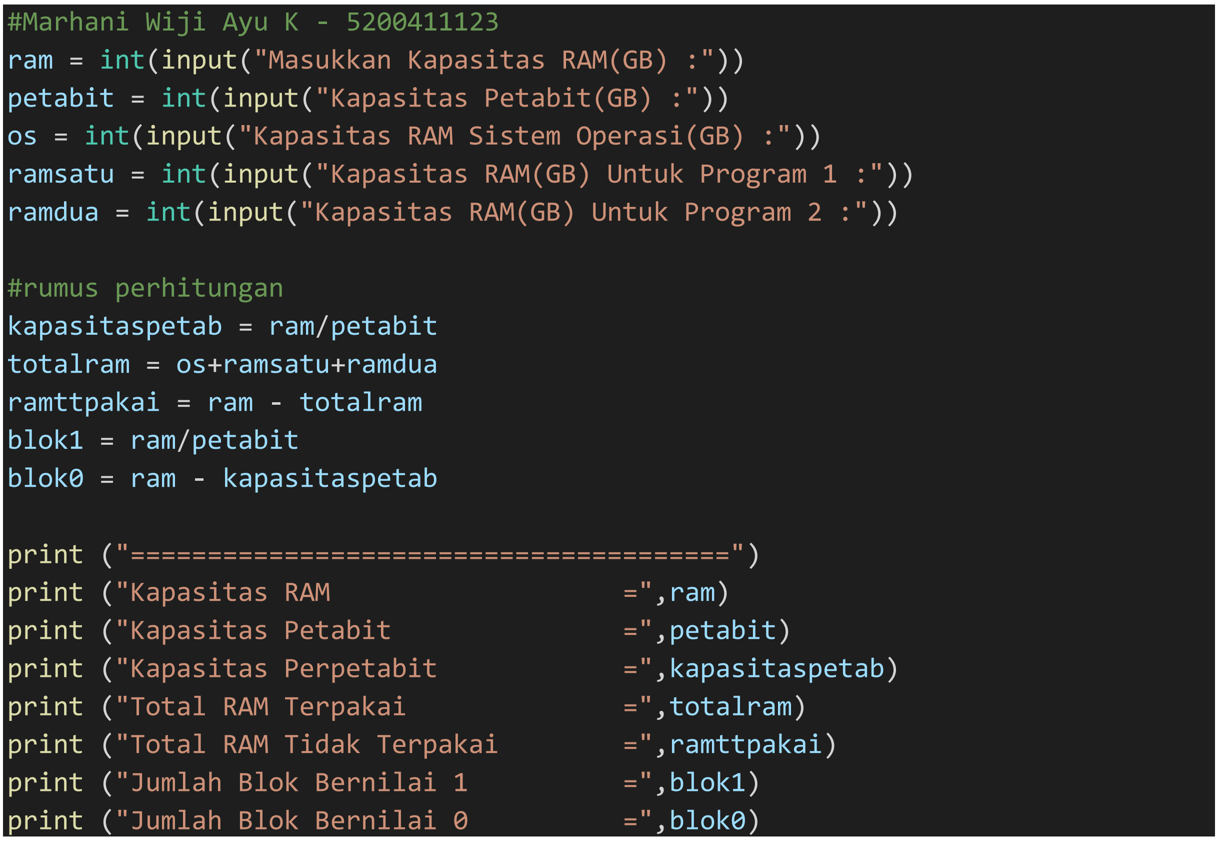
PRODI INFORMATIKA

FAKULTAS SAINS & TEKNOLOGI

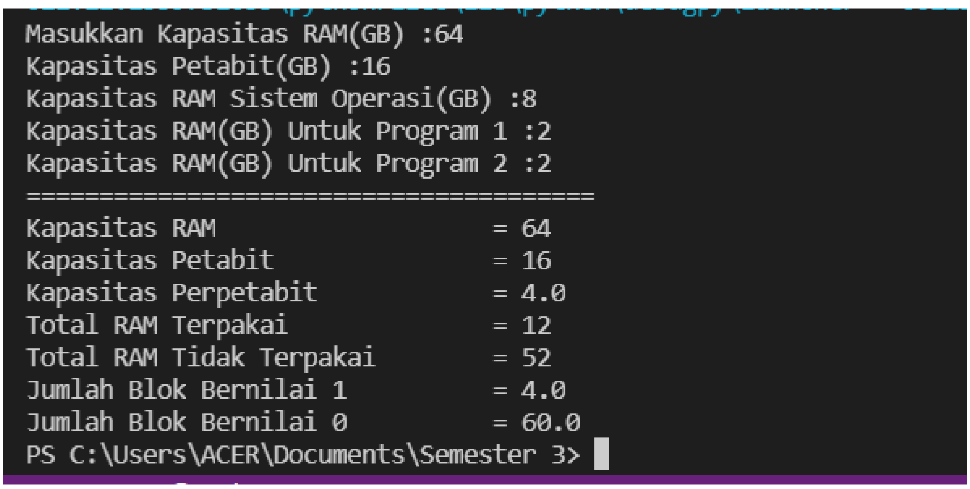
UNIVERSITAS TEKNOLOGI YOGYAKARTA

2021/2022

1.

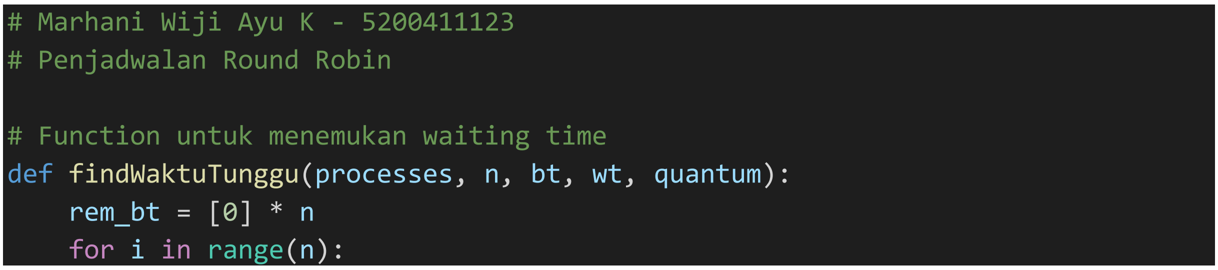


```
#Marhani Wiji Ayu K - 5200411123
ram = int(input("Masukkan Kapasitas RAM(GB) :"))
petabit = int(input("Kapasitas Petabit(GB) :"))
os = int(input("Kapasitas RAM Sistem Operasi(GB) :"))
ramsatu = int(input("Kapasitas RAM(GB) Untuk Program 1 :"))
ramdua = int(input("Kapasitas RAM(GB) Untuk Program 2 :"))

#rumus perhitungan
kapasitaspetab = ram/petabit
totalram = os+ramsatu+ramdua
ramttpakai = ram - totalram
blok1 = ram/petabit
blok0 = ram - kapasitaspetab

print ("=======================================")
print ("Kapasitas RAM                   =",ram)
print ("Kapasitas Petabit               =",petabit)
print ("Kapasitas Perpetabit            =",kapasitaspetab)
print ("Total RAM Terpakai              =",totalram)
print ("Total RAM Tidak Terpakai        =",ramttpakai)
print ("Jumlah Blok Bernilai 1          =",blok1)
print ("Jumlah Blok Bernilai 0          =",blok0)
```

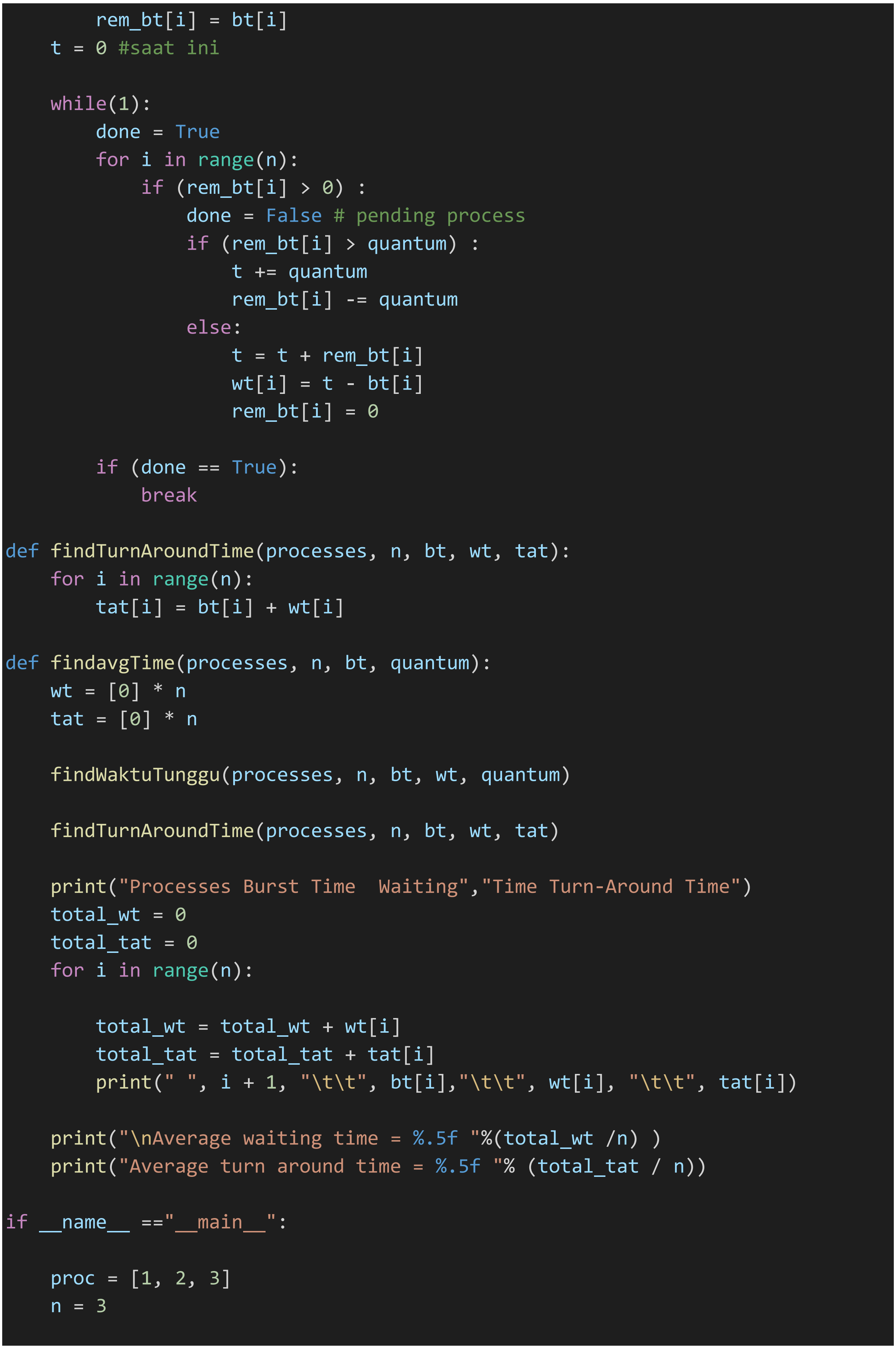


```
Masukkan Kapasitas RAM(GB) :64
Kapasitas Petabit(GB) :16
Kapasitas RAM Sistem Operasi(GB) :8
Kapasitas RAM(GB) Untuk Program 1 :2
Kapasitas RAM(GB) Untuk Program 2 :2
=======================================
Kapasitas RAM                   = 64
Kapasitas Petabit               = 16
Kapasitas Perpetabit            = 4.0
Total RAM Terpakai              = 12
Total RAM Tidak Terpakai        = 52
Jumlah Blok Bernilai 1          = 4.0
Jumlah Blok Bernilai 0          = 60.0
PS C:\Users\ACER\Documents\Semester 3>
```

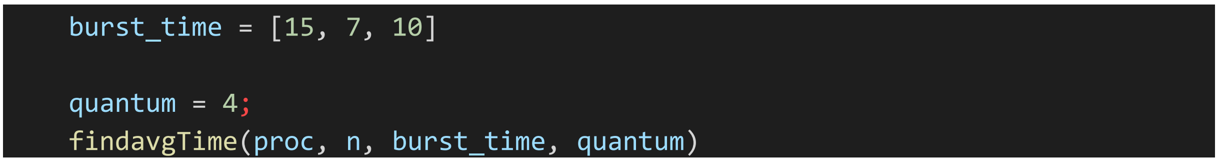
2.



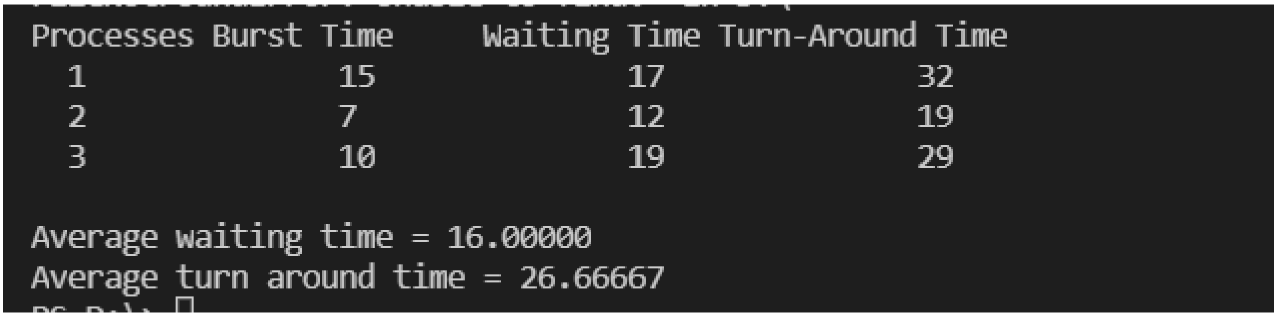
```
# Marhani Wiji Ayu K - 5200411123
# Penjadwalan Round Robin

# Function untuk menemukan waiting time
def findWaktuTunggu(processes, n, bt, wt, quantum):
    rem_bt = [0] * n
    for i in range(n):
```



```
        rem_bt[i] = bt[i]
    t = 0 #saat ini

    while(1):
        done = True
        for i in range(n):
            if (rem_bt[i] > 0) :
                done = False # pending process
                if (rem_bt[i] > quantum) :
                    t += quantum
                    rem_bt[i] -= quantum
                else:
                    t = t + rem_bt[i]
                    wt[i] = t - bt[i]
                    rem_bt[i] = 0

        if (done == True):
            break

def findTurnAroundTime(processes, n, bt, wt, tat):
    for i in range(n):
        tat[i] = bt[i] + wt[i]

def findavgTime(processes, n, bt, quantum):
    wt = [0] * n
    tat = [0] * n

    findWaktuTunggu(processes, n, bt, wt, quantum)

    findTurnAroundTime(processes, n, bt, wt, tat)

    print("Processes Burst Time  Waiting","Time Turn-Around Time")
    total_wt = 0
    total_tat = 0
    for i in range(n):

        total_wt = total_wt + wt[i]
        total_tat = total_tat + tat[i]
        print(" ", i + 1, "\t\t", bt[i],"\t\t", wt[i], "\t\t", tat[i])

    print("\nAverage waiting time = %.5f "%(total_wt /n) )
    print("Average turn around time = %.5f "% (total_tat / n))

if __name__ =="__main__":

    proc = [1, 2, 3]
    n = 3
```



```
    burst_time = [15, 7, 10]

    quantum = 4;
    findavgTime(proc, n, burst_time, quantum)
```



```
Processes Burst Time     Waiting Time Turn-Around Time
  1              15              17              32
  2              7               12              19
  3              10              19              29

Average waiting time = 16.00000
Average turn around time = 26.66667
```