

**RESPONSI  
SISTEM OPERASI PRAKTIK V**



**DISUSUN OLEH :**

**RETNO PRIHATINI  
5200411450**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS & TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2021/2022**

## 1. Kode program

### Helper.py

```
def header(judul):
    print("=" * 70)
    print("\t\t", judul)
    print("=" * 70)

def konversiRAM(ttlRAM):
    return ttlRAM * 1024

def hitungPetaBit(ram, blok):
    return ram / blok

def ttlRamTerpakai(program1, program2):
    return program1 + program2

def ttlRamKosong(ramUntukProgram, ramTerpakai):
    return ramUntukProgram - ramTerpakai
```

### no1.py

```
import helper

helper.header("Responsi Sistem Operasi Praktik")

totalRam = int(input("Kapasitas RAM dalam Mb: "))
totalBlok = int(input("Total PetaBit: "))
petabit = helper.hitungPetaBit(helper.konversiRAM(totalRam), totalBlok)

print("\nProgram dijalankan")

ramUntukProgram = int(input("Kapasitas Sistem Operasi : "))
blokTerpakai = helper.hitungPetaBit(helper.konversiRAM(ramUntukProgram), petabit)
program1 = int(input("Kapasitas Program 1 : "))
program2 = int(input("Kapasitas Program 2 : "))
ramTerpakai = helper.ttlRamTerpakai(program1, program2)
ramKosong = helper.ttlRamKosong(ramUntukProgram, ramTerpakai)
```

```

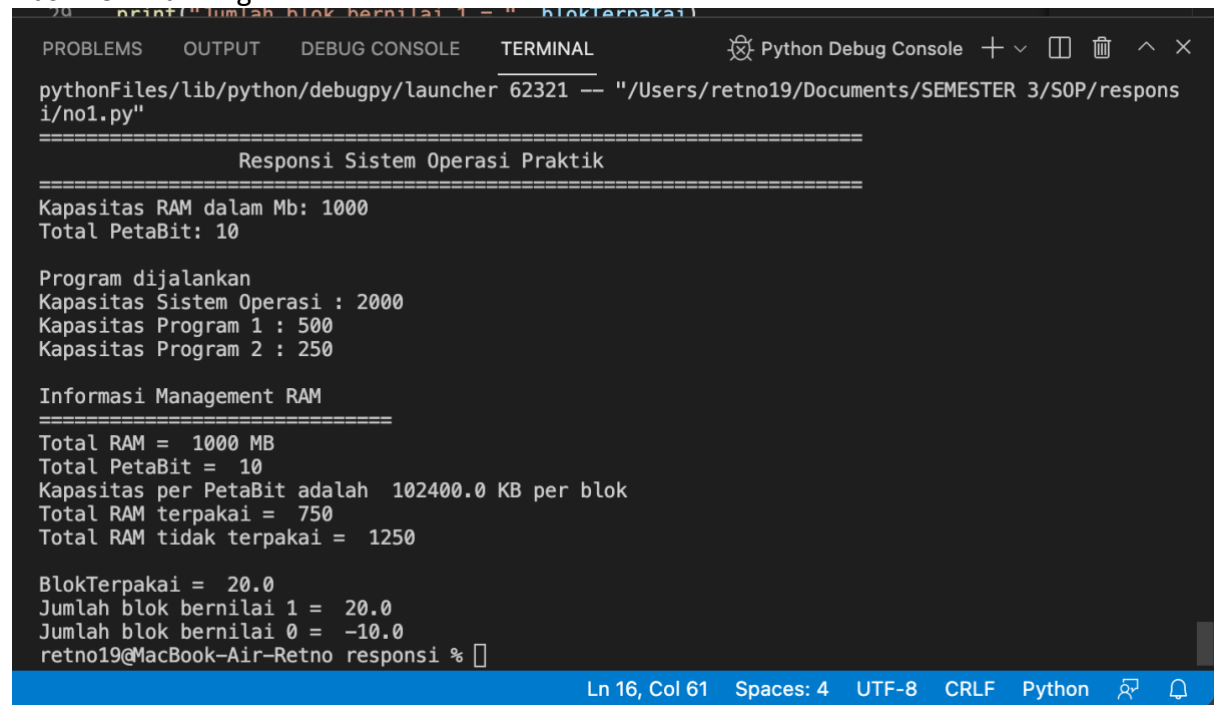
print("\nInformasi Management RAM")
print("="*30)

print("Total RAM = ", totalRam , "MB")
print("Total PetaBit = ", totalBlok)
print("Kapasitas per PetaBit adalah ", petabit , "KB per blok" )
print("Total RAM terpakai = ",ramTerpakai)
print("Total RAM tidak terpakai = ",ramKosong)

print("\nBlokTerpakai = ",blokTerpakai)
print("Jumlah blok bernilai 1 = ", blokTerpakai)
print("Jumlah blok bernilai 0 = ", totalBlok - blokTerpakai)

```

## Hasil FULL running



```

pythonFiles/lib/python/debugpy/launcher 62321 -- "/Users/retno19/Documents/SEMESTER 3/SOP/respons
i/no1.py"
=====
                Responsi Sistem Operasi Praktik
=====
Kapasitas RAM dalam Mb: 1000
Total PetaBit: 10

Program dijalankan
Kapasitas Sistem Operasi : 2000
Kapasitas Program 1 : 500
Kapasitas Program 2 : 250

Informasi Management RAM
=====
Total RAM = 1000 MB
Total PetaBit = 10
Kapasitas per PetaBit adalah 102400.0 KB per blok
Total RAM terpakai = 750
Total RAM tidak terpakai = 1250

BlokTerpakai = 20.0
Jumlah blok bernilai 1 = 20.0
Jumlah blok bernilai 0 = -10.0
retno19@MacBook-Air-Retno responsi % 

```

## Hasil inputan

```

=====
                Responsi Sistem Operasi Praktik
=====
Kapasitas RAM dalam Mb: 1000
Total PetaBit: 10

Program dijalankan
Kapasitas Sistem Operasi : 2000
Kapasitas Program 1 : 500
Kapasitas Program 2 : 250

```

## Hasil Output

```

Informasi Management RAM
=====
Total RAM = 1000 MB
Total PetaBit = 10
Kapasitas per PetaBit adalah 102400.0 KB per blok
Total RAM terpakai = 750
Total RAM tidak terpakai = 1250

BlokTerpakai = 20.0
Jumlah blok bernilai 1 = 20.0
Jumlah blok bernilai 0 = -10.0
retno19@MacBook-Air-Retno responsi % 

```

## 2. Kode Program

### No2.py

```

#bt/ burst time = waktu yang dibutuhkan
#wt / waiting time = waktu tunggu
#tat / turn around time = waktu penyelesaian
#task = tugas
programs=[]
def Waktu_tunggu(Proses ,tugas,bt,wt,Quantum):
    sisaWaktu=[0]*tugas
    for totalWaktuSelesai in range(tugas):
        sisaWaktu[totalWaktuSelesai]=bt[totalWaktuSelesai]
    t=0
    while(1):
        done=True
        for totalWaktuSelesai in range(tugas):
            if(sisaWaktu[totalWaktuSelesai]>0):
                done=False
                if(sisaWaktu[totalWaktuSelesai]>Quantum):
                    t+=Quantum
                    programs.append(("P"+str(totalWaktuSelesai))*(Quantum))
                    sisaWaktu[totalWaktuSelesai]-=Quantum
                else:
                    t+=sisaWaktu[totalWaktuSelesai]
                    wt[totalWaktuSelesai]=t-bt[totalWaktuSelesai]
                    programs.append(("P"+str(totalWaktuSelesai))*(sisaWaktu[totalWaktuSelesai]))
                    sisaWaktu[totalWaktuSelesai]=0
            if(done==True):
                break
def TurnAroundTime(Proses,tugas,bt,wt,tat):

```

```

for totalWaktuSelesai in range(tugas):
    tat[totalWaktuSelesai]=bt[totalWaktuSelesai]+wt[totalWaktuSelesai]
def FindAvgTime(Proses,tugas,bt,Quantum):
    wt=[0]*tugas
    tat=[0]*tugas
    Waktu_tunggu(Proses,tugas,bt,wt,Quantum)
    TurnAroundTime(Proses,tugas,bt,wt,tat)
    print("Program Waktu yang dibutuhkan Waktu tunggu", "Waktu penyelesaian")
    total_wt = 0
    total_tat = 0
    for totalWaktuSelesai in range(tugas):
        total_wt+=wt[totalWaktuSelesai]
        total_tat+=tat[totalWaktuSelesai]
        print(" ",totalWaktuSelesai+1, "\t\t", bt[totalWaktuSelesai],"\t\t", wt[totalWaktuSelesai], "\t\t",
tat[totalWaktuSelesai])
    print("\nRata-rata Waktu Tunggu = %.2f"%(total_wt /tugas) )
    print("Rata-rata Waktu Penyelesaian = %.2f"%(total_tat /tugas))

JumlahProses=int(input("Masukkan Jumlah Proses-->"))
JatahWaktu=int(input("Masukkan Jatah Waktu-->"))
Burst_time=[]
Process_id=[0,1,2,3]
for totalWaktuSelesai in range(JumlahProses):
    Burst_time.append(int(input("Masukkan waktu Proses "+str(input("Nama Program : "))
+str(totalWaktuSelesai)+ " ")))

def sort():
    return sorted(programs.items(), key=lambda item: item[1], reverse=False)
FindAvgTime(Process_id,JumlahProses,Burst_time,JatahWaktu)
Index=[int(totalWaktuSelesai) for totalWaktuSelesai in range(sum(Burst_time)+1)]
print("Banyak Program----> ",*programs,"\nTotal Waktu Penyelesaian----> ",*Index)

```

Hasil FULL running

