RESPONSI

Sistem Operasi Praktik-V

Oleh :
520041123 – Marhani Wiji Ayu Kusumawati

PRODI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2021/2022

1.

```python
#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.

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

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

```
        rem_burst[i] = burst[i]
    t = 0 #saat ini

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

        if (done == True):
            break

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

def findWaktuRata(processes, n, burst, kuantum):
    wt = [0] * n
    tat = [0] * n

    findWaktuTunggu(processes, n, burst, wt, kuantum)

    findWaktuTurnAround(processes, n, burst, wt, tat)

    print("Processes Burst Time  Waktu Tunggu","Waktu Turn-Around")
    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", burst[i],"\t\t", wt[i], "\t\t", tat[i])

    print("\nRata-rata Waktu Tunggu = %.5f "%(total_wt /n) )
    print("Rata-rata Turn Around Time = %.5f "% (total_tat / n))

if __name__ =="__main__":

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

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

  kuantum = 4;
  findWaktuRata(proc, n, burst_time, kuantum)
```

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

Rata-rata Waktu Tunggu = 16.00000
Rata-rata Turn Around Time = 26.66667
PS C:\Users\ACER\Documents>
```