

SISTEM OPERASI PRAKTIK V

Responsi



Disusun Oleh:

Nama : Muhamad Syarifudin

Nim : 5200411347

Program Studi Informatika

Fakultas Sains & Teknologi

Universitas Teknologi Yogyakarta

2021/2022

1.

Code:

```
#MuhamadSyarifudin
#5200411347

#Menghitung Kapasitas Total Ram & Total Petabit
TotalRam = int(input("Kapasitas total RAM (MBps): "))
TotalBlok = int(input("Blok/unit: "))

TotalPetabit = TotalRam/TotalBlok

print("Total Petabit:",TotalPetabit)
print("Kapasitas Per Petabit :",TotalPetabit)
print("Kapasitas Ram yang digunakan OS: ",TotalRam,"MBps")
#Total Ram Dipakai & Tidak Terpakai
PS = int(input("Ram yang digunakan oleh Program 1: "))
PD = int(input("Ram yang digunakan oleh Program 2: "))

TotalRT = PS + PD
TotalTD = TotalRam - (PS + PD)
print("Totak Ram yang terpakai: ",TotalRT,"MBps")
print("Total Ram yang tidak terpakai: ",TotalTD,"MBps")

#Jumlah Blok Bernilai 1 dan 0

JumlahBS = PS-TotalPetabit
JumlahBN = TotalBlok+JumlahBS

print("Jumlah Blok Bernilai 1: ",JumlahBS)
print("Jumlah Blok Bernilai 0: ",JumlahBN)
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\2021\Kuliah semester 3\Sistem Operasi Praktik> & C:/Users
py"
Kapasitas total RAM (MBps): 8192
Blok/unit: 32
Total Petabit: 256.0
Kapasitas Per Petabit : 256.0
Kapasitas Ram yang digunakan OS: 8192 MBps
Ram yang digunakan oleh Program 1: 2048
Ram yang digunakan oleh Program 2: 2048
Totak Ram yang terpakai: 4096 MBps
Total Ram yang tidak terpakai: 4096 MBps
Jumlah Blok Bernilai 1: 8.0
Jumlah Blok Bernilai 0: 24.0
PS D:\2021\Kuliah semester 3\Sistem Operasi Praktik> █
```

2.

Code:

```
#Muhamad Syarifudin
#5200411347

class RoundRobin:

    def ProsesData(self, no_of_processes):
        proses = []
        for i in range(no_of_processes):
            tenggat = []
            prosesKe = str(input("Memasukan Aplikasi : "))
            burstTime = int(input(f"Memasukan Burst Time Untuk Memproses {prosesKe}: "))
            tenggat.extend([prosesKe, 0, burstTime, 0, burstTime])

            proses.append(tenggat)
        quantumTime = int(input("Memasukan Quantum Time : "))
        RoundRobin.schedulingProcess(self, proses, quantumTime)

    def schedulingProcess(self, proses, quantumTime):
        Mulai = []
        selesai = []
        eksekusiProses = []
        urutan = []
        s_time = 0
        while 1:
            temp = []
            for i in range(len(proses)):
                if proses[i][1] <= s_time and proses[i][3] == 0:
                    present = 0
                    if len(urutan) != 0:
                        for k in range(len(urutan)):
                            if proses[i][0] == urutan[k][0]:
                                present = 1

                if present == 0:
                    temp.extend([proses[i][0], proses[i][1], proses[i][2], proses[i][4]])
                    urutan.append(temp)
                    temp = []

                if len(urutan) != 0 and len(eksekusiProses) != 0:
                    for k in range(len(urutan)):
                        if urutan[k][0] == eksekusiProses[len(eksekusiProses) - 1]:
                            urutan.insert((len(urutan) - 1), urutan.pop(k))

            if len(urutan) == 0:
                break
            if len(urutan) != 0:
                if urutan[0][2] > quantumTime:

                    Mulai.append(s_time)
                    s_time = s_time + quantumTime
                    e_time = s_time
                    selesai.append(e_time)
                    eksekusiProses.append(urutan[0][0])
                    for j in range(len(proses)):
                        if proses[j][0] == urutan[0][0]:
                            break
                    proses[j][2] = proses[j][2] - quantumTime
                    urutan.pop(0)
                elif urutan[0][2] <= quantumTime:

                    Mulai.append(s_time)
                    s_time = s_time + urutan[0][2]
                    e_time = s_time
                    selesai.append(e_time)
                    eksekusiProses.append(urutan[0][0])
                    for j in range(len(proses)):
                        if proses[j][0] == urutan[0][0]:
                            break
                    proses[j][2] = 0
                    proses[j][3] = 1
```

```

        proses[j].append(e_time)
        urutan.pop(0)
    t_time = RoundRobin.calculateTurnaroundTime(self, proses)
    w_time = RoundRobin.calculateWaitingTime(self, proses)
    RoundRobin.printData(self, proses, t_time, w_time, eksekusiProses)

def calculateTurnaroundTime(self, proses):
    total_putaran_waktu = 0
    for i in range(len(proses)):
        putaran_waktu = proses[i][5] - proses[i][1]

        total_putaran_waktu = total_putaran_waktu + putaran_waktu
        proses[i].append(putaran_waktu)
    rataTurnTime = total_putaran_waktu / len(proses)

    return rataTurnTime

def calculateWaitingTime(self, proses):
    total_waktu_tunggu = 0
    for i in range(len(proses)):
        waktu_tunggu = proses[i][6] - proses[i][4]

        total_waktu_tunggu = total_waktu_tunggu + waktu_tunggu
        proses[i].append(waktu_tunggu)
    rataWaitTime = total_waktu_tunggu / len(proses)

    return rataWaitTime

def printData(self, proses, rataTurnTime, rataWaitTime, eksekusiProses):
    proses.sort(key=lambda x: x[0])

    print(
        "prosesKe  Arrival_Time  Rem_burstTime  Completed  Original_burstTime  Completion_Time  Waktu_
Berputar  Waktu_Tunggu")
    for i in range(len(proses)):
        for j in range(len(proses[i])):
            print(proses[i][j], end="\t\t\t\t")
        print()
    print(f'Rata-Rata Waktu Berputar: {rataTurnTime}')
    print(f'Rata-Rata Waktu Tunggu: {rataWaitTime}')
    print(f'Urutan Proses: {eksekusiProses}')

if __name__ == "__main__":
    no_of_processes = int(input("Memasukan Jumlah Proses : "))
    rr = RoundRobin()
    rr.ProsesData(no_of_processes)

```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\2021\Kuliah semester 3\Sistem Operasi Praktik> & C:/Users/asus/anaconda3/python.exe "d:/2021/Kuliah semester 3/Sistem Operasi Praktik/Responsi2.py"
Memasukan Jumlah Proses : 5
Memasukan Aplikasi : Google
Memasukan Burst Time Untuk Memproses Google: 5
Memasukan Aplikasi : Zoom
Memasukan Burst Time Untuk Memproses Zoom: 8
Memasukan Aplikasi : YT
Memasukan Burst Time Untuk Memproses YT: 6
Memasukan Aplikasi : VSCode
Memasukan Burst Time Untuk Memproses VSCode: 3
Memasukan Aplikasi : Discord
Memasukan Burst Time Untuk Memproses Discord: 2
Memasukan Quantum Time : Notion
Traceback (most recent call last):
  File "d:/2021/Kuliah semester 3/Sistem Operasi Praktik/Responsi2.py", line 117, in <module>
    rr.ProsesData(no_of_processes)
  File "d:/2021/Kuliah semester 3/Sistem Operasi Praktik/Responsi2.py", line 16, in ProsesData
    quantumTime = int(input("Memasukan Quantum Time : "))
ValueError: invalid literal for int() with base 10: 'Notion'
PS D:\2021\Kuliah semester 3\Sistem Operasi Praktik> & C:/Users/asus/anaconda3/python.exe "d:/2021/Kuliah semester 3/Sistem Operasi Praktik/Responsi2.py"
Memasukan Jumlah Proses : 5
Memasukan Quantum Time : 5
prosesKe  Arrival_Time  Rem_burstTime  Completed  Original_burstTime  Completion_Time  Waktu_Berputar  Waktu_Tunggu
7  DOTA 2  0  0  0  0  1  20  3
37  17
Discord  0  0  0  1  3  6
VSCode  0  0  0  1  4  1
3  13  9
Youtube  0  0  1  9  2
2  22  13
Zoom  0  0  1  1  5
Rata-Rata Waktu Berputar: 17.4
Rata-Rata Waktu Tunggu: 10.0
Urutan Proses: ['Youtube', 'Zoom ', 'Discord', 'VSCode', 'DOTA 2', 'Youtube', 'DOTA 2', 'DOTA 2', 'DOTA 2']
PS D:\2021\Kuliah semester 3\Sistem Operasi Praktik> |
```