

TUGAS MATA KULIAH
SISTEM OPERASI PRAKTIK



Dosen Pengampu : Iwan Hartadi Tri Untoro, S.T., M.Kom.

Asisten Dosen : Galang Aidil Akbar

Disusun oleh

Nama : Pandu Teja Sutrisno

NPM : 5200411343

Kelompok : V

UNIVERSITAS TEKNOLOGI YOGYAKARTA

Fakultas Sains dan Teknologi

INFORMATIKA

2021 / 2022

Nomor 1

```
helper.py ResponsiNo1.py X
C: > Users > asus > Documents > UTY > Semester 3 > Sistem Operasi Praktik > Responsi > ResponsiNo1.py > ...
1  import helper
2
3  helper.header("Responsi SOP NOMOR 1")
4
5  RAM = int(input("Kapasitas RAM dalam MB : "))
6  Blok = int(input("Masukan Blok : "))
7  petabit = helper.hitungPetaBit(helper.ubahRamKeMbps(RAM), Blok)
8  kapasitas = petabit - RAM
9
10 print("-" * 50)
11 print("Kapasitas Petabit : ", petabit)
12 print("Kapasitas /Petabit : ", kapasitas)
13
14 print("\nProgram dijalankan")
15
16 sisop = int(input("Kapasitas Sistem Operasi : "))
17 program1 = int(input("RAM yang digunakan oleh program 1: "))
18 program2 = int(input("RAM yang digunakan oleh program 2: "))
19
20
21 totalRAM = program1 + program2
22 totalRAMtdkterpakai = RAM - totalRAM
23
24 print("Total RAM Terpakai : ", totalRAM)
25 print("Total RAM Tak Terpakai : ", totalRAMtdkterpakai)
26
27 blok1 = RAM / petabit
28 blok0 = Blok - blok1
29
30
31 print("Kapasitas RAM", RAM)
32 print("Jumlah blok bernilai 1 = ", blok1)
33 print("Jumlah blok bernilai 0 = ", blok0)
```

```
-----
                        Responsi SOP NOMOR 1
-----
Kapasitas RAM dalam MB : 64
Masukan Blok : 8
-----
Kapasitas Petabit : 8192.0
Kapasitas /Petabit : 8128.0

Program dijalankan
Kapasitas Sistem Operasi : 6
RAM yang digunakan oleh program 1: 20
RAM yang digunakan oleh program 2: 17
Total RAM Terpakai : 37
Total RAM Tak Terpakai : 27
Kapasitas RAM 64
Jumlah blok bernilai 1 = 0.0078125
Jumlah blok bernilai 0 = 7.9921875
```

Nomor 2

```
helper.py ResponsiNo2.py X
C:\Users> asus > Documents > UTY > Semester 3 > Sistem Operasi Praktikum > Responsi > ResponsiNo2.py > RoundRobin > ProsesData
1 import helper
2
3 helper.header("Responsi SOP NOMOR 2")
4
5 class RoundRobin:
6
7     def ProsesData(self, no_of_processes):
8         proses = []
9         for i in range(no_of_processes):
10             tenggat = []
11             prosesKe = str(input("Masukan Nama Aplikasi : "))
12             burstTime = int(input(f"Masukan Burst Time Untuk Memproses {prosesKe}: "))
13             tenggat.extend([prosesKe, 0, burstTime, 0, burstTime])
14
15         proses.append(tenggat)
16         quantumTime = int(input("Masukan Quantum Time : "))
17         RoundRobin.schedulingProcess(self, proses, quantumTime)
18
19     def schedulingProcess(self, proses, quantumTime):
20         Mulai = []
21         selesai = []
22         eksekusiProses = []
23         urutan = []
24         s_time = 0
25         while 1:
26             temp = []
27             for i in range(len(proses)):
28                 if proses[i][1] <= s_time and proses[i][3] == 0:
29                     present = 0
30                     if len(urutan) != 0:
31                         for k in range(len(urutan)):
32                             if proses[i][0] == urutan[k][0]:
33                                 present = 1
34
35                     if present == 0:
36                         temp.extend([proses[i][0], proses[i][1], proses[i][2], proses[i][4]])
37                         urutan.append(temp)
38                         temp = []
39
40                     if len(urutan) != 0 and len(eksekusiProses) != 0:
41                         for k in range(len(urutan)):
42                             if urutan[k][0] == eksekusiProses[len(eksekusiProses) - 1]:
43                                 urutan.insert((len(urutan) - 1), urutan.pop(k))
44
45                     if len(urutan) == 0:
46                         break
47                     if len(urutan) != 0:
48                         if urutan[0][2] > quantumTime:
49
50                             Mulai.append(s_time)
51                             s_time = s_time + quantumTime
52                             e_time = s_time
53                             selesai.append(e_time)
54                             eksekusiProses.append(urutan[0][0])
55                             for j in range(len(proses)):
56                                 if proses[j][0] == urutan[0][0]:
57                                     break
58                             proses[j][2] = proses[j][2] - quantumTime
59                             urutan.pop(0)
60                         elif urutan[0][2] <= quantumTime:
61
62                             Mulai.append(s_time)
63                             s_time = s_time + urutan[0][2]
64                             e_time = s_time
65                             selesai.append(e_time)
66                             eksekusiProses.append(urutan[0][0])
67                             for j in range(len(proses)):
68                                 if proses[j][0] == urutan[0][0]:
69                                     break
70                             proses[j][2] = 0
71                             proses[j][3] = 1
72                             proses[j].append(e_time)
73                             urutan.pop(0)
74
75         t_time = RoundRobin.calculateTurnaroundTime(self, proses)
76         w_time = RoundRobin.calculateWaitingTime(self, proses)
77         RoundRobin.printData(self, proses, t_time, w_time, eksekusiProses)
```

```

78     def calculateTurnaroundTime(self, proses):
79         total_turnaround_time = 0
80         for i in range(len(proses)):
81             turnaround_time = proses[i][5] - proses[i][1]
82
83             total_turnaround_time = total_turnaround_time + turnaround_time
84             proses[i].append(turnaround_time)
85         rataaanTurnTime = total_turnaround_time / len(proses)
86
87         return rataaanTurnTime
88
89     def calculateWaitingTime(self, proses):
90         total_waiting_time = 0
91         for i in range(len(proses)):
92             waiting_time = proses[i][6] - proses[i][4]
93
94             total_waiting_time = total_waiting_time + waiting_time
95             proses[i].append(waiting_time)
96         rataanWaitTime = total_waiting_time / len(proses)
97
98         return rataanWaitTime
99
100    def printData(self, proses, rataaanTurnTime, rataanWaitTime, eksekusiProses):
101        proses.sort(key=lambda x: x[0])
102
103        print(
104            "prosesKe  Arrival_Time  Rem_burstTime  Completed  Original_burstTime  Completion_Time  Turnaround_Time  Waiting_Time")
105        for i in range(len(proses)):
106            for j in range(len(proses[i])):
107                print(proses[i][j], end="\t\t\t\t\t")
108            print()
109        print(f'Rataan Turnaround Time: {rataaanTurnTime}')
110        print(f'Rataan Waiting Time: {rataanWaitTime}')
111        print(f'Urutan Proses: {eksekusiProses}')
112
113
114    if __name__ == "__main__":
115        no_of_processes = int(input("Masukan Jumlah Proses : "))
116        rr = RoundRobin()
117        rr.ProsesData(no_of_processes)

```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\asus> & C:/Users/asus/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/asus/Documents/UTY/Semester 3/Sistem Operasi Praktik/Responsi/ResponsiNo2.py"

Responsi SOP NOMOR 2

Masukan Jumlah Proses : 4
Masukan Nama Aplikasi : Zoom
Masukan Burst Time Untuk Memproses Zoom: 12
Masukan Nama Aplikasi : Android Studio
Masukan Burst Time Untuk Memproses Android Studio: 6
Masukan Nama Aplikasi : Lens Studio
Masukan Burst Time Untuk Memproses Lens Studio: 4
Masukan Nama Aplikasi : Visual Studio
Masukan Burst Time Untuk Memproses Visual Studio: 7
Masukan Quantum Time : 3

prosesKe	Arrival_Time	Rem_burstTime	Completed	Original_burstTime	Completion_Time	Turnaround_Time	Waiting_Time
Android Studio			0		0	1	
8		18		12			6
Lens Studio			0		0	1	
9		19		15			4
Visual Studio			0		0	1	
6		26		19			7
Zoom		0		0		1	
9		17					12

Rataan Turnaround Time: 23.0

Rataan Waiting Time: 15.75

Urutan Proses: ['Zoom', 'Android Studio', 'Lens Studio', 'Visual Studio', 'Zoom', 'Android Studio', 'Lens Studio', 'Visual Studio', 'Zoom', 'Visual Studio', 'Zoom']

PS C:\Users\asus> █