

**RESPONSI**  
**SISTEM OPERASI PRAKTIK KELOMPOK V**  
**MANAJEMEN MEMORI DAN PENJADWALAN ROUND ROBIN**



Disusun Oleh:

Nama : Yodhi Anugrah Damar Saputra  
NIM : 5200411168  
Kelas/Kelompok : Kelompok XII  
Hari, Tanggal : Rabu, 22 Desember 2021  
Dosen Pengampu : Iwan Hartadi Tri Untoro, S.T., M.Kom.  
Asisten Dosen : Galang Aidil Akbar

**LABORATORIUM KOMPUTER 7**  
**SARJANA INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2021**

## Daftar Isi

KODE PROGRAM.....	2
A. Manajemen RAM.....	2
B. Penjadwalan Round Robin .....	6
SCREENSHOT RUNNING PROGRAM .....	11
A. Manajemen RAM.....	11
1. Berhasil .....	11
2. Gagal.....	12
B. Penjadwalan Round Robin .....	13
1. Berhasil .....	13
2. Gagal.....	14

## KODE PROGRAM

### (DALAM BAHASA PYTHON 3.10)

#### A. Manajemen RAM

```
# Responsi - Sistem Operasi Praktik Kelompok V - Manajemen RAM
# 5200411168 - Yodhi Anugrah Damar Saputra

# tampilkan judul program
print("")
print("=" * 45)
print("\t\tMANAJEMEN RAM")
print("=" * 45)

# instruksi untuk melakukan proses
print(" * untuk memproses, masukkan angka 0 (nol) \n  pada input nama program")
print("-" * 45)

# sistem menerima input yang sesuai
RAMGB = int(input("Masukkan kapasitas total RAM! \n GB => "))

print("")

PetaBit = int(input("Masukkan kapasitas peta bit! \n MB => "))

print("")

OSGB = int(input("Masukkan kapasitas RAM OS! \n GB => "))

# sistem memeriksa validitas input
if RAMGB < 0 or PetaBit < 0 or OSGB < 0:
    print("")
    print("=" * 24)
    print("|" + ("!" * 5) + " PERINGATAN " + ("!" * 5) + "|")
    print("|" + ("-" * 22) + "|")
    print("|" + " Input harus berupa " + "|")
```

```

print("|" + "  bilangan positif!  " + "|")
print("=" * 24)
print("")

# apabila input tidak valid, keluar dari sistem
exit()

# sistem melakukan perulangan untuk menghitung proses utama
print("-" * 45)
ProgramTotal = 0
Urutan = 1
while 1:

    # sistem memperoleh input program tereksekusi
    ProgramTereksekusi = int(input("Masukkan kapasitas RAM program ke-{}\n MB =>
.format(Urutan)))
    print("")

    # jumlah kapasitas semua program tereksekusi
    ProgramTotal += ProgramTereksekusi

    # sistem melakukan pengondisian
    # jika kapasitas program tereksekusi bernilai 0 maka akan masuk proses utama
    if ProgramTereksekusi == 0:

        # konversi data yang diperlukan
        RAMMB = RAMGB * 1024
        OSMB = OSGB * 1024
        BlokPerUnit = RAMMB / PetaBit

        # opearsi data sisa
        RAMMBTerpakai = OSMB + ProgramTotal
        BlokTerpakai = round(RAMMBTerpakai / PetaBit)
        RAMGBTerpakai = RAMMBTerpakai / 1024

        # opearsi data terpakai
        RAMMBSisa = RAMMB - RAMMBTerpakai
        BlokSisa = round(BlokPerUnit - BlokTerpakai)

```

```

RAMGBSisa = RAMMBSisa / 1024

# tampilkan info output yang diperlukan
print("-" * 45)
print(" => Kapasitas total RAM \t: ")
print("    - {} GB \n    - {} MB \n".format(RAMGB, RAMMB))

print(" => Kapasitas total peta bit \t: ")
print("    - {} MB \n".format(PetaBit))

print(" => Kapasitas per peta bit \t: ")
print("    - {} MB \n".format(BlokPerUnit))

print(" => Kapasitas RAM terpakai \t: ")
print("    - {} GB \n    - {} MB \n".format(RAMGBTerpakai,
RAMMBTerpakai))

print(" => Kapasitas RAM tersisa \t: ")
print("    - {} GB \n    - {} MB \n".format(RAMGBSisa, RAMMBSisa))

print(" => Jumlah blok bernilai 1 \t: ")
print("    - {} \n".format(BlokTerpakai))
print(" => Jumlah blok bernilai 0 \t: ")
print("    - {} \n".format(BlokSisa))

# tampilkan footer
print("=" * 45)
print("5200411168 - Sistem Operasi Praktik - Kelas V")
print("=" * 45)
print("")

# sistem memeriksa validitas sisa ram apabila bernilai negatif
if RAMGBSisa < 0 or RAMMBSisa < 0:

    # tampilkan peringatan
    print("=" * 24)
    print("|" + ("!" * 5) + " PERINGATAN " + ("!" * 5) + "|")
    print("|" + ("-" * 22) + "|")

```

```

        print("|" + " Kapasitas RAM tidak " + "|")
        print("|" + " memungkinkan untuk " + "|")
        print("|" + " menjalankan program!" + "|")
        print("=" * 24)
        print("")
        break
    break

# sistem memeriksa validitas kapasitas program tereksekusi
elif ProgramTereksekusi < 0:

    # tampilkan peringatan
    print("")
    print("=" * 24)
    print("|" + ("!" * 5) + " PERINGATAN " + ("!" * 5) + "|")
    print("|" + ("-" * 22) + "|")
    print("|" + " Input harus berupa " + "|")
    print("|" + " bilangan positif! " + "|")
    print("=" * 24)
    print("")
    continue

# sistem memeriksa validitas kapasitas program tereksekusi bernilai positif
elif ProgramTereksekusi > 0:

    # tambahkan nilai urutan
    Urutan += 1
    Continue

```

## B. Penjadwalan Round Robin

```
# Responsi - Sistem Operasi Praktik Kelompok V - Penjadwalan Round Robin
# 5200411168 - Yodhi Anugrah Damar Saputra

# tampilkan judul program
print("")
print("=" * 45)
print("\t\t\t\t\tMANAJEMEN ROUND ROBIN")
print("=" * 45)

# list kosong sebagai tempat menyimpan nilai
RRNamaProgram = []
RRBurstTime = []

# instruksi untuk melakukan proses
print(" * untuk memroses, masukkan karakter '='\n\t\t\t\t\t(sama dengan) pada input nama\nprogram")
print("-" * 45)

# masuk perulangan untuk melakukan input nama program dan burst time
Urutan = 1
while 1:

    # sistem mendapatkan input nama program
    print("")
    NamaProgram = str(input("Masukkan nama program ke-{}\n => ".format(Urutan)))

    # sistem masuk pada kondisi yang memeriksa nama program
    # jika nama program bernama '=' maka proses perhitungan akan berjalan
    if NamaProgram == "=":

        # sistem mendapatkan input quantum time
        QuantumTime = int(input("\nMasukkan quantum time \n msec => "))

        # sistem memeriksa validitas quantum time
        if QuantumTime <= 0:
            print("")
```

```

        print("=" * 24)
        print("|" + ("!" * 5) + " PERINGATAN " + ("!" * 5) + "|")
        print("|" + ("-" * 22) + "|")
        print("|" + " Quantum time harus " + "|")
        print("|" + "  bilangan bulat      " + "|")
        print("|" + "  positif!                " + "|")
        print("=" * 24)
        print("")
        continue

# sistem menghitung banyaknya jumlah program yang berjalan
JumlahProgram = 0
for Index in range(len(RRNamaProgram)):
    JumlahProgram += 1
    Index += 1

# sistem menghitung jumlah total burst time
TotalBurstTime = 0
for Index in range(len(RRBurstTime)):
    TotalBurstTime += int(RRBurstTime[Index])

# informasi program
print("")
print("-" * 45)
print(" => Daftar program \t: ")

for Index in range(len(RRBurstTime)):
    print("      {}. {} \t: {} msec".format(Index + 1, RRNamaProgram[Index],
RRBurstTime[Index]))

print("\n => Total program \t: {}".format(JumlahProgram))
print(" => Total burst time \t: {}".format(TotalBurstTime))
print(" => Quantum time \t: {}".format(QuantumTime))
print("-" * 45)
print("")

# sistem masuk pada pemrosesan utama round robin
SisaBurstTime = TotalBurstTime

```



```

UrutanHasil = 1
CompletionTime = 0
while 1:

    # sistem mendapatkan index dari tiap element dalam array burst time
    for Index in range(len(RRBurstTime)):

        # memperoleh nilai element array burst time dari index
        FetchData = RRBurstTime[Index]

        # sistem memeriksa perbandingan nilai element array
        # apabila nilai element array lebih besar dari quantum time, dan
total burst time tidak nol
        if FetchData >= QuantumTime and SisaBurstTime != 0:

            # kurangi total burst time dengan burst time yang telah
terpakai

            SisaBurstTime -= QuantumTime

            # kurangi nilai element array berdasarkan indeks sesuai
dengan burst time yang telah terpakai
            RRBurstTime[Index] = FetchData - QuantumTime

            # cetak informasi program sesuai urutan serta sisa burst time
            print(" => Proses ke-{}".format(UrutanHasil))
            print(" - {} dieksekusi selama {}
msec".format(RRNamaProgram[Index], QuantumTime))
            print(" - Sisa burst time {} adalah {}
msec".format(RRNamaProgram[Index], RRBurstTime[Index]))
            print(" - Total burst time tersisa {}
\n".format(SisaBurstTime))

            # menambahkan nomor urutan hasil
            UrutanHasil += 1

        # lainnya apabila nilai element array lebih kecil dari quantum
time namun bukan nol, dan total burst time tidak nol

```

```

elif FetchData < QuantumTime and FetchData != 0 and SisaBurstTime
!= 0:

    # kurangi total burst time dengan burst time yang telah
    terpakai

    SisaBurstTime -= FetchData

    # kurangi nilai element array berdasarkan indeks sesuai
    dengan burst time yang telah terpakai
    RRBurstTime[Index] = FetchData - FetchData

    # cetak informasi program sesuai urutan serta sisa burst time
    print(" => Proses ke-{}".format(UrutanHasil))
    print(" - {} dieksekusi selama {}
msec".format(RRNamaProgram[Index], FetchData))
    print(" - Sisa burst time {} adalah {}
msec".format(RRNamaProgram[Index], RRBurstTime[Index]))
    print(" - Total burst time tersisa {}
\n".format(SisaBurstTime))

    # menambahkan nomor urutan hasil
    UrutanHasil += 1

    # sistem melakukan pengkondisian apabila sisa total burst time
    sama dengan nol
    if SisaBurstTime == 0:

        # tampilkan footer
        print("=" * 45)
        print("5200411168 - Sistem Operasi Praktik - Kelas V")
        print("=" * 45)
        print("")

        # sistem akan menghentikan proses dan keluar untuk
        menghindari perulangan tak terbatas
        exit()

```

```

    # Jika nama program yang dimasukkan selain karakter '=', maka sistem akan
    meminta input burst time
    else:

        # sistem mendapatkan input burst time
        BurstTime = int(input("Masukkan burst time dari {} \n msec =>
{}".format>NamaProgram)))

        # sistem memeriksa apabila quantum time bernilai positif
        if BurstTime <= 0:
            print("")
            print("=" * 24)
            print("|" + ("!" * 5) + " PERINGATAN " + ("!" * 5) + "|")
            print("|" + ("-" * 22) + "|")
            print("|" + " Burst time harus " + "|")
            print("|" + " bilangan bulat " + "|")
            print("|" + " positif! " + "|")
            print("=" * 24)
            print("")
            continue

        # inputan nama program dan burst time dimasukan ke masing masing array
        RRNamaProgram.append>NamaProgram)
        RRBurstTime.append>BurstTime)

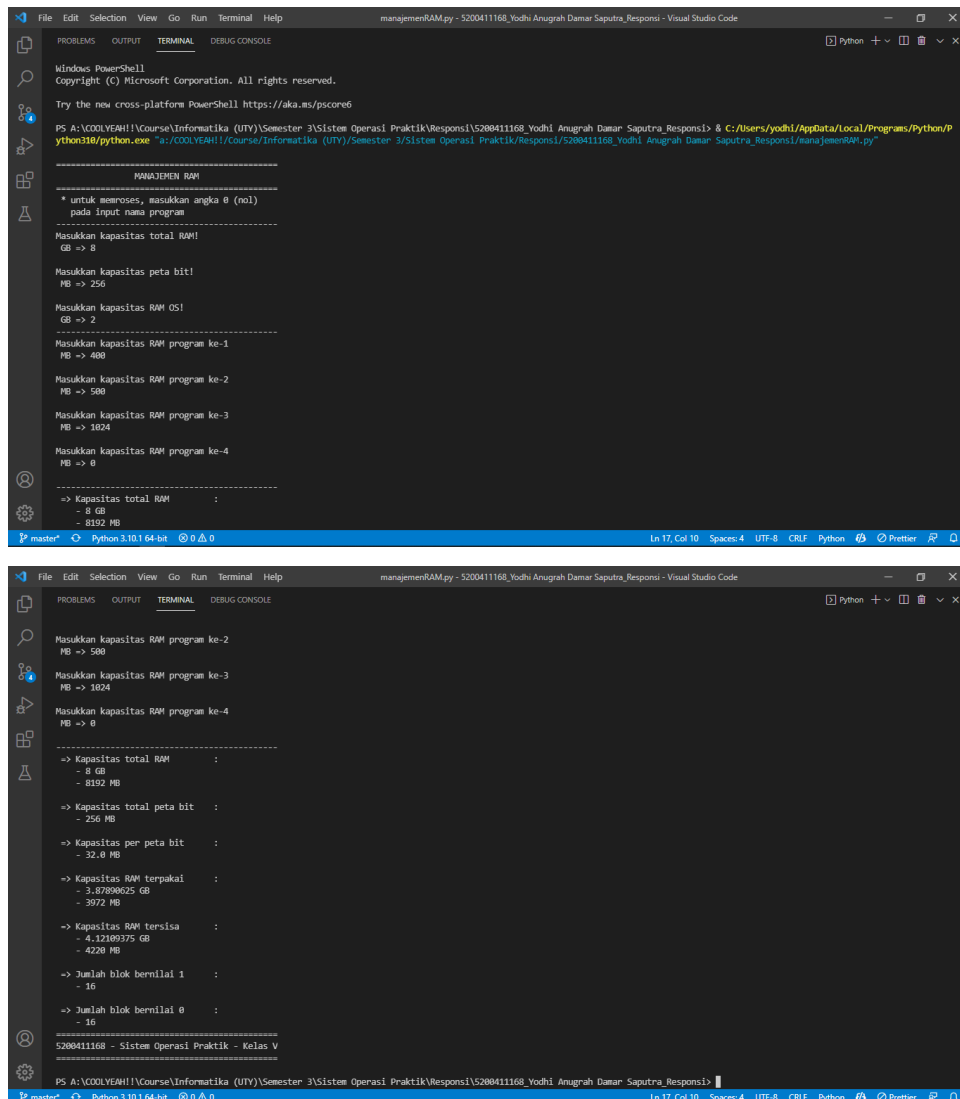
        # menambah nomor urutan
        Urutan += 1

```

# SCREENSHOT RUNNING PROGRAM

## A. Manajemen RAM

### 1. Berhasil



```
manajemenRAM.py - 5200411168_Yodhi Anugrah Damar Saputra_Responi - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS A:\COOLYEAH\Course\Informatika (UTY)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responi> & C:\Users\yodhi\AppData\Local\Programs\Python\Python310\python.exe "a:\COOLYEAH\Course\Informatika (UTY)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responi\manajemenRAM.py"

=====
MANAJEMEN RAM
=====
* untuk memproses, masukkan angka 0 (nol)
pada input nama program

Masukkan kapasitas total RAM!
GB => 8

Masukkan kapasitas peta bit!
MB => 256

Masukkan kapasitas RAM OS!
GB => 2

=====
Masukkan kapasitas RAM program ke-1
MB => 400

Masukkan kapasitas RAM program ke-2
MB => 500

Masukkan kapasitas RAM program ke-3
MB => 1024

Masukkan kapasitas RAM program ke-4
MB => 0

=====
=> Kapasitas total RAM      :
    - 8 GB
    - 8192 MB

=====

Masukkan kapasitas RAM program ke-2
MB => 500

Masukkan kapasitas RAM program ke-3
MB => 1024

Masukkan kapasitas RAM program ke-4
MB => 0

=====
=> Kapasitas total RAM      :
    - 8 GB
    - 8192 MB

=> Kapasitas total peta bit  :
    - 256 MB

=> Kapasitas per peta bit   :
    - 32.0 MB

=> Kapasitas RAM terpakai   :
    - 3.87090625 GB
    - 3972 MB

=> Kapasitas RAM tersisa    :
    - 4.12109375 GB
    - 4220 MB

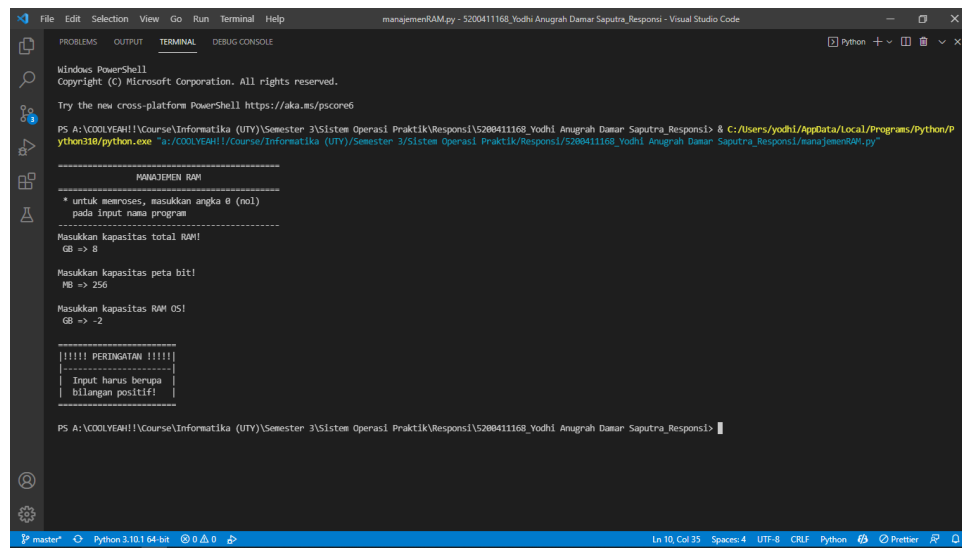
=> Jumlah blok bernilai 1   :
    - 16

=> Jumlah blok bernilai 0   :
    - 16

=====
5200411168 - Sistem Operasi Praktik - Kelas V
=====

PS A:\COOLYEAH\Course\Informatika (UTY)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responi>
```

## 2. Gagal



```
manajemenRAM.py - 5200411168_Yodhi Anugrah Damar Saputra_Responsi - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS A:\CDOL\VEAH!!!\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responsi> & C:\Users\yodhi\AppData\Local\Programs\Python\Python310\python.exe "a:\CDOL\VEAH!!!\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responsi\manajemenRAM.py"

=====
MANAJEMEN RAM
=====
* untuk memroses, masukkan angka 0 (no.)
pada input nama program
=====
Masukkan kapasitas total RAM!
GB => 8

Masukkan kapasitas peta bit!
MB => 256

Masukkan kapasitas RAM OS!
GB => 2

!!!!!! PERINGATAN !!!!!!!
|-----|
| Input harus berupa |
| bilangan positif! |
|-----|

PS A:\CDOL\VEAH!!!\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktik\Responsi\5200411168_Yodhi Anugrah Damar Saputra_Responsi>
Python 3.10.1 64-bit | In 10, Col 35 | Spaces: 4 | UTF-8 | CRLF | Python | Prettier
```

## B. Penjadwalan Round Robin

1. Berhasil

```
File Edit Selection View Go Run Terminal Help perjadwalanRoundRobin.py - S200411168_Yodhi Anugrah Damar Saputra_Respons - Visual Studio Code  
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE Python + - [ ] [x] [v]  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/powershell  
  
PS A:\CODING\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktikum\Respons\5200411168_Yodhi Anugrah Damar Saputra_Respons> & C:\Users\yodhi\AppData\Local\Programs\Python\Python38\python.exe "a:\CODING\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktikum\Respons\5200411168_Yodhi Anugrah Damar Saputra_Respons\perjadwalanRoundRobin.py"  
-----  
MANAJEMEN ROUND ROBIN  
-----  
* untuk memroses, masukkan karakter '-'  
(sama dengan) pada input nama program  
-----  
  
Masukkan nama program ke-1  
-> Illustrator  
Masukkan burst time dari Illustrator  
msec -> 5800  
  
Masukkan nama program ke-2  
-> Google Chrome  
Masukkan burst time dari Google Chrome  
msec -> 3500  
  
Masukkan nama program ke-3  
-> Microsoft Word  
Masukkan burst time dari Microsoft Word  
msec -> 1200  
  
Masukkan nama program ke-4  
-> -  
  
Masukkan quantum time  
msec -> 2100  
  
-----  
-> Daftar program :  
1. Illustrator : 5800 msec  
-----  
  
Masukkan nama program ke-3  
-> Microsoft Word  
Masukkan burst time dari Microsoft Word  
msec -> 1200  
  
Masukkan nama program ke-4  
-> -  
  
Masukkan quantum time  
msec -> 2100  
  
-----  
-> Daftar program :  
1. Illustrator : 5800 msec  
2. Google Chrome : 3500 msec  
3. Microsoft Word : 1200 msec  
-----  
-> Total program : 3  
-> Total burst time : 9700  
-> Quantum time : 2100  
-----  
  
-> Proses ke-1  
- Illustrator dieksekusi selama 2100 msec  
- Sisa burst time Illustrator adalah 2900 msec  
- Total burst time tersisa 7600  
  
-> Proses ke-2  
- Google Chrome dieksekusi selama 2100 msec  
- Sisa burst time Google Chrome adalah 1400 msec  
- Total burst time tersisa 5500  
  
-> Proses ke-3  
- Microsoft Word dieksekusi selama 1200 msec  
- Sisa burst time Microsoft Word adalah 0 msec  
- Total burst time tersisa 4300  
  
-> Proses ke-4  
- Illustrator dieksekusi selama 2100 msec  
  
-----  
-> Total burst time : 9700  
-> Quantum time : 2100  
-----  
  
-> Proses ke-1  
- Illustrator dieksekusi selama 2100 msec  
- Sisa burst time Illustrator adalah 2900 msec  
- Total burst time tersisa 7600  
  
-> Proses ke-2  
- Google Chrome dieksekusi selama 2100 msec  
- Sisa burst time Google Chrome adalah 1400 msec  
- Total burst time tersisa 5500  
  
-> Proses ke-3  
- Microsoft Word dieksekusi selama 1200 msec  
- Sisa burst time Microsoft Word adalah 0 msec  
- Total burst time tersisa 4300  
  
-> Proses ke-4  
- Illustrator dieksekusi selama 2100 msec  
- Sisa burst time Illustrator adalah 800 msec  
- Total burst time tersisa 2200  
  
-> Proses ke-5  
- Google Chrome dieksekusi selama 1400 msec  
- Sisa burst time Google Chrome adalah 0 msec  
- Total burst time tersisa 800  
  
-> Proses ke-6  
- Illustrator dieksekusi selama 800 msec  
- Sisa burst time Illustrator adalah 0 msec  
- Total burst time tersisa 0  
  
-----  
5200411168 - Sistem Operasi Praktikum - Kelas V  
-----  
  
PS A:\CODING\Course\Informatika (UTV)\Semester 3\Sistem Operasi Praktikum\Respons\5200411168_Yodhi Anugrah Damar Saputra_Respons>
```

## 2. Gagal

[illegible]