

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Algoritma dan Pemrograman 2B
Kelas : 1IA24
Praktikum Ke- : 1
Tanggal : Selasa, 28 Maret 2023
Materi : Thread Dan Multi-Thread
NPM : 50422818
Nama : Lutfi Robbani
Ketua asisten : Gede Oke
Nama asisten :
Paraf asisten :
Jumlah Lembar : 5 lembar

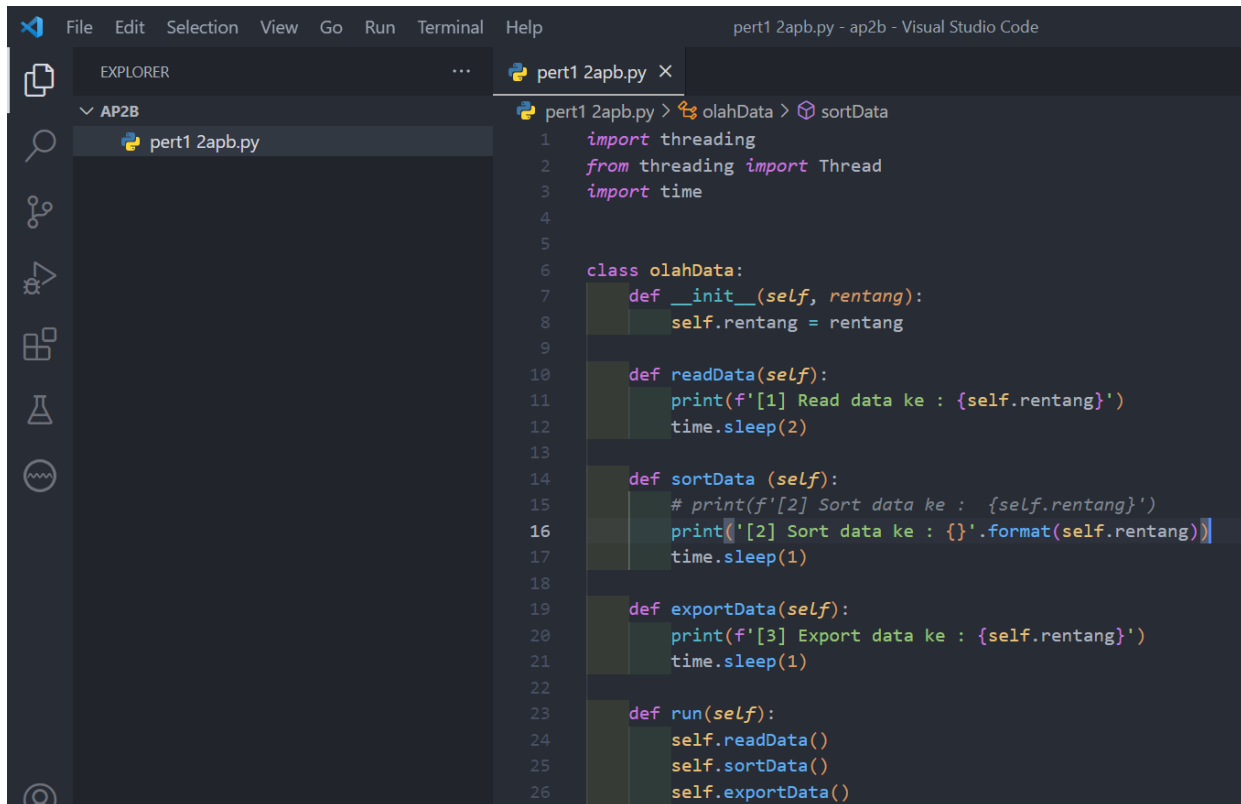


LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2022

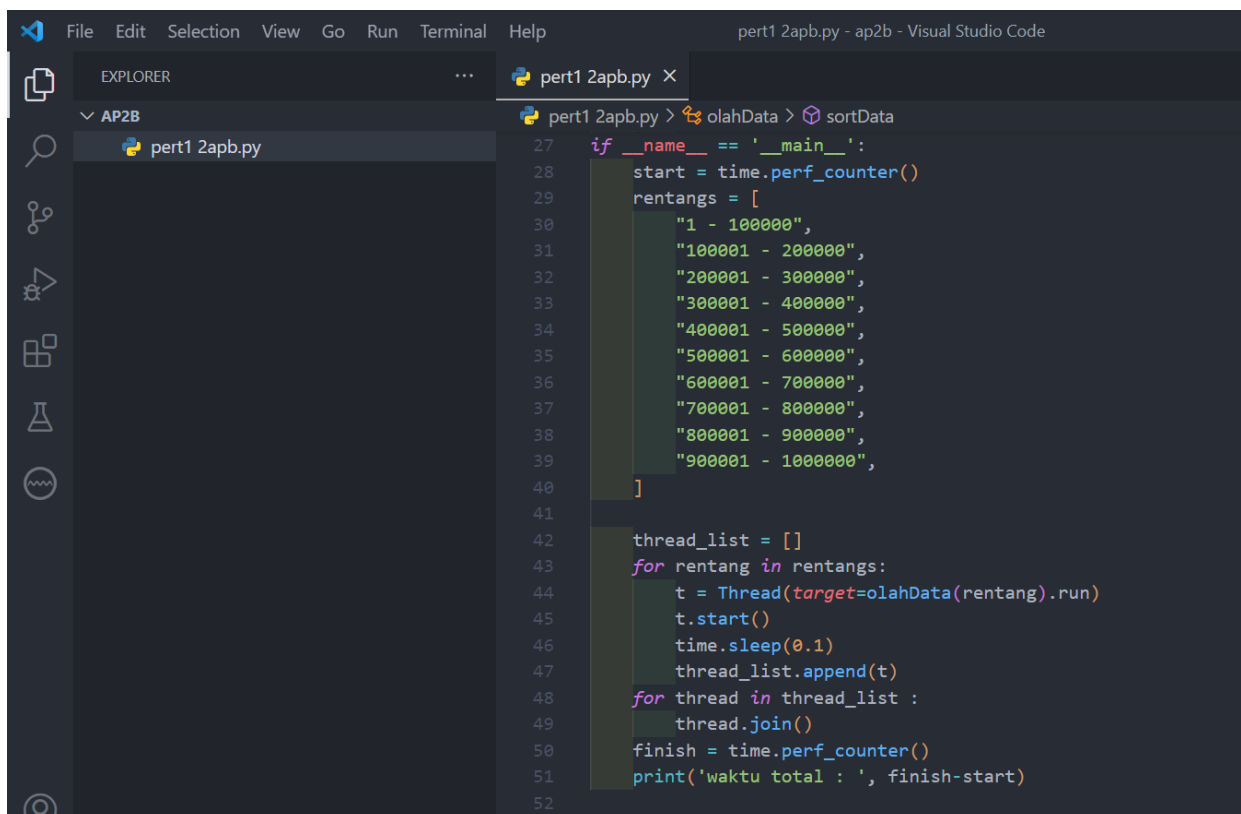
LISTENING



```
File Edit Selection View Go Run Terminal Help pert1_2apb.py - ap2b - Visual Studio Code

EXPLORER
AP2B
  pert1_2apb.py

pert1_2apb.py > olahData > sortData
1  import threading
2  from threading import Thread
3  import time
4
5
6  class olahData:
7      def __init__(self, rentang):
8          self.rentang = rentang
9
10     def readData(self):
11         print(f'[1] Read data ke : {self.rentang}')
12         time.sleep(2)
13
14     def sortData (self):
15         # print(f'[2] Sort data ke : {self.rentang}')
16         print(f'[2] Sort data ke : {}'.format(self.rentang))
17         time.sleep(1)
18
19     def exportData(self):
20         print(f'[3] Export data ke : {self.rentang}')
21         time.sleep(1)
22
23     def run(self):
24         self.readData()
25         self.sortData()
26         self.exportData()
```



```
File Edit Selection View Go Run Terminal Help pert1_2apb.py - ap2b - Visual Studio Code

EXPLORER
AP2B
  pert1_2apb.py

pert1_2apb.py > olahData > sortData
27  if __name__ == '__main__':
28      start = time.perf_counter()
29      rentangs = [
30          "1 - 100000",
31          "100001 - 200000",
32          "200001 - 300000",
33          "300001 - 400000",
34          "400001 - 500000",
35          "500001 - 600000",
36          "600001 - 700000",
37          "700001 - 800000",
38          "800001 - 900000",
39          "900001 - 1000000",
40      ]
41
42      thread_list = []
43      for rentang in rentangs:
44          t = Thread(target=olahData(rentang).run)
45          t.start()
46          time.sleep(0.1)
47          thread_list.append(t)
48      for thread in thread_list :
49          thread.join()
50      finish = time.perf_counter()
51      print('waktu total : ', finish-start)
52
```

LOGIKA

```
pert1 2apb.py > olahData > sortData
1  import threading
2  from threading import Thread
3  import time
```

Pertama import library threading dan library time pada python.

```
6  class olahData:
7      def __init__(self, rentang):
8          self.rentang = rentang
9
10     def readData(self):
11         print(f'[1] Read data ke : {self.rentang}')
12         time.sleep(2)
13
14     def sortData (self):
15         # print(f'[2] Sort data ke : {self.rentang}')
16         print(f'[2] Sort data ke : {self.rentang}')
17         time.sleep(1)
18
19     def exportData(self):
20         print(f'[3] Export data ke : {self.rentang}')
21         time.sleep(1)
```

Setelah itu buat class Bernama olahdata, lalu di dalamnya masukan beberapa method, pertama buat method def __init__() method ini digunakan untuk konstruktor pada kelas olah data, Saat objek olahData dibuat, method ini yang dijalankan nantinya. Dan didalam method ini kita tambahkan parameter self, dan rentang. Kita deklarasikan nilai self.rentang pada parameter __init__() bernilai rentang.

Selanjutnya kita buat method readData() didalamnya tambahkan parameter self, kemudian cetak tulisan “[1] read data ke : “ dan tambahkan self.rentang, untuk memanggil objek dari method __init__. Setelah mencetak output metode ini akan berjeda selama 2 detik menggunakan time.sleep(2).

Setelah itu tambahkan method lagi sortData() didalamnya tambahkan parameter self, lalu cetak “[2] sort data ke : {” dan tambahkan .format(self.rentang) untuk memanggil objek dari method __init__. Setelah mencetak output metode ini akan berjeda selama 1 detik menggunakan time.sleep(1).

Setelah itu tambahkan method lagi exportData() didalamnya tambahkan parameter self, lalu cetak “[3] export data ke : “ dan tambahkan self.rentang, untuk memanggil objek dari method __init__. Setelah mencetak output metode ini akan berjeda selama 1 detik menggunakan time.sleep(1).

```

22
23     def run(self):
24         self.readData()
25         self.sortData()
26         self.exportData()

```

Kemudian buat method run() dan parameter self, method ini berfungsi untuk menjalankan method yang telah kita buat sebelumnya yaitu method readData(), sortData(), exportData() secara berturut turut.

```

Python pert1 2apb.py > olahData > sortData
27     if __name__ == '__main__':
28         start = time.perf_counter()
29         rentangs = [
30             "1 - 100000",
31             "100001 - 200000",
32             "200001 - 300000",
33             "300001 - 400000",
34             "400001 - 500000",
35             "500001 - 600000",
36             "600001 - 700000",
37             "700001 - 800000",
38             "800001 - 900000",
39             "900001 - 1000000",
40         ]

```

Selanjutnya, buat statement kondisi if `__name__ == '__main__':` untuk menentukan apakah kode di dalam blok tersebut akan dijalankan atau tidak.

Lalu gunakan perintah `time.perf_counter()` digunakan untuk menghitung waktu mulai eksekusi program menggunakan fungsi `perf_counter()`, dan disimpan didalam variabel `start`.

kemudian mendefinisikan sebuah list `rentangs` yang berisi sepuluh item, yang masing-masing adalah string yang mewakili rentang data yang akan diolah.

```

42     thread_list = []
43     for rentang in rentangs:
44         t = Thread(target=olahData(rentang).run)
45         t.start()
46         time.sleep(0.1)
47         thread_list.append(t)

```

Selanjutnya buat perulangan for untuk memproses setiap elemen dari daftar rentangs. Setiap elemen rentang kemudian digunakan untuk membuat objek olahData yang dijalankan sebagai thread.

Selanjutnya program membuat thread-thread baru dan menjalankan fungsi run pada objek olahData yang diinisialisasi dengan parameter rentang yang diberikan. Kemudian setiap thread dimasukkan ke dalam thread_list dan dijalankan dengan memanggil method start(). Selanjutnya gunakan perintah time.sleep maksudnya di antara setiap pembuatan thread untuk memberikan jeda 0.1 detik.

```

48     for thread in thread_list :
49         thread.join()

```

Selanjutnya gunakan perulangan for untuk membuat perulangan pada list thread_list, program juga akan menunggu semua thread selesai dijalankan dengan menggunakan method join().

```

50     finish = time.perf_counter()
51     print('waktu total : ', finish-start)

```

Selanjutnya tambahkan variable finish dengan nilai time.perf_counter, ini berfungsi untuk membaca waktu program akan berakhir, lalu cetak “waktu total : “ dan tambahkan variable finish di kurang variable start untuk mengetahui waktu yang di butuhkan untuk menjalankan program.

OUTPUT

```
Help      pert1 2apb.py - ap2b - Visual Studio Code

PROBLEMS  4    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\sawal fitah\Documents\punya lupi\file2\buat belajar\ap2b> python documents\punya lupi\file2\buat belajar\ap2b\pert1 2apb.py"
[1] Read data ke : 1 - 100000
[1] Read data ke : 100001 - 200000
[1] Read data ke : 200001 - 300000
[1] Read data ke : 300001 - 400000
[1] Read data ke : 400001 - 500000
[1] Read data ke : 500001 - 600000
[1] Read data ke : 600001 - 700000
[1] Read data ke : 700001 - 800000
[1] Read data ke : 800001 - 900000
[1] Read data ke : 900001 - 1000000
[2] Sort data ke : 1 - 100000
[2] Sort data ke : 100001 - 200000
[2] Sort data ke : 200001 - 300000
[2] Sort data ke : 300001 - 400000
[2] Sort data ke : 400001 - 500000
[2] Sort data ke : 500001 - 600000
[2] Sort data ke : 600001 - 700000
[2] Sort data ke : 700001 - 800000
[2] Sort data ke : 800001 - 900000
[2] Sort data ke : 900001 - 1000000
[3] Export data ke : 1 - 100000
[3] Export data ke : 100001 - 200000
[3] Export data ke : 200001 - 300000
waktu total : 4.915264800000003
PS C:\Users\sawal fitah\Documents\punya lupi\file2\buat belajar\ap2b>
```