

LAPORAN AKHIR PERTEMUAN 1

Mata Praktikum : Algoritma dan Pemograman 2B
Kelas : 1IA13
Praktikum ke- : 1
Tanggal : 31 Maret 2023
Materi : Multithread (Python)
NPM : 51422161
Nama : Muhammad Tarmidzi Bariq
Ketua Asisten : Muhammad Wildan Firdausi
Paraf Asisten :
Nama Asisten : Muhammad Wildan Firdausi
Jumlah Lembar : 3 Lembar

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

1. Jelaskan Apa Yang Dimaksud dengan Thread Dan Multithreading!

Thread adalah sebuah bagian program yang dapat berjalan mandiri, sehingga dua atau lebih thread dapat berjalan bersamaan, tanpa yang satu harus menunggu selesainya yang lain.

Multithreading teknik pemrograman yang memungkinkan beberapa sub proses dalam program dapat berjalan secara paralel

2. buatlah program dengan Multithread seperti pada condoh, kemudian ubah rentang data yang digunakan. Screenshot program yang telah kalian buat beserta logika dan output nya!

```
1  from threading import Thread
2  import time
3
4  class olahData:
5      def __init__(self, rentang):
6          self.rentang = rentang
7          # read data
8      def readData(self):
9          print(f'[1] Read data ke : {self.rentang}')
10     def sortData(self):
11         # menyortir data
12         print(f'[2] Sort data ke : {self.rentang}')
13     def exportData(self):
14         # menarik data
15         print(f'[3] Export data ke : {self.rentang}')
16         time.sleep(1)
17     # untuk menjalankan 3 function diatas
18     def run(self):
19         self.readData()
20         self.sortData()
21         self.exportData()
22
23 if __name__ == "__main__":
24     # untuk menyimpan waktu
25     start = time.perf_counter()
26     # kita set
27     rentang = [
28         '1-1000',
29         '1001-2000',
30         '2001-3000',
31         '3001-4000',
32         '4001-5000',
33         '5001-6000',
34         '6001-7000',
35         '7001-8000',
36         '8001-9000',
37         '9001-10000',
38     ]
39     thread_list = []
40     for rentang in rentang:
41         t = Thread(target=olahData(rentang).run)
42         t.start()
43         time.sleep(0.1)
44         thread_list.append(t)
45     for thread in thread_list:
46         thread.join()
47     # mencetak rentan waktu proses
48     finish = time.perf_counter()
49     # cetak waktu yang dibutuhkan dari waktu pengolahan
50     print('Waktu yang butuh untuk eksekusi program', finish-start)
```

```

PS C:\Users\ASUS\Desktop\latihan coding\ALGO 1A> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ASUS/Desktop/latihan coding/ALGO 1A/sdad.py"
[1] Read data ke : 1-1000
[2] Sort data ke : 1-1000
[3] Export data ke : 1-1000
[1] Read data ke : 1001-2000
[2] Sort data ke : 1001-2000
[3] Export data ke : 1001-2000
[1] Read data ke : 2001-3000
[2] Sort data ke : 2001-3000
[3] Export data ke : 2001-3000
[1] Read data ke : 3001-4000
[2] Sort data ke : 3001-4000
[3] Export data ke : 3001-4000
[1] Read data ke : 4001-5000
[2] Sort data ke : 4001-5000
[3] Export data ke : 4001-5000
[1] Read data ke : 5001-6000
[2] Sort data ke : 5001-6000
[3] Export data ke : 5001-6000
[1] Read data ke : 6001-7000
[2] Sort data ke : 6001-7000
[3] Export data ke : 6001-7000
[1] Read data ke : 7001-8000
[2] Sort data ke : 7001-8000
[3] Export data ke : 7001-8000
[1] Read data ke : 8001-9000
[2] Sort data ke : 8001-9000
[3] Export data ke : 8001-9000
[1] Read data ke : 9001-10000
[2] Sort data ke : 9001-10000
[3] Export data ke : 9001-10000
Waktu yang dibutuhkan untuk eksekusi program 1.9867689000093378
PS C:\Users\ASUS\Desktop\latihan coding\ALGO 1A>

```

3. Jelaskan Kegunaan Dari Multithreading pada program yang kalian buat !

Program yang Anda sertakan merupakan contoh implementasi multithreading pada Python. Dalam program tersebut, beberapa thread dibuat untuk memproses data yang terletak pada rentang tertentu.

Beberapa kegunaan dari multithreading pada program ini adalah:

1. Meningkatkan kinerja: Dengan memproses data secara paralel pada beberapa thread, program ini dapat menyelesaikan tugas lebih cepat dibandingkan jika hanya menggunakan satu thread.
2. Mengoptimalkan penggunaan sumber daya: Dalam program ini, setiap thread akan memproses data pada rentang tertentu secara independen. Hal ini memungkinkan penggunaan sumber daya seperti CPU dan memori menjadi lebih optimal karena setiap thread hanya memproses sebagian kecil dari data.
3. Menjaga responsifitas aplikasi: Dalam program ini, beberapa thread digunakan untuk memproses data pada rentang tertentu secara bersamaan. Hal ini memungkinkan aplikasi tetap responsif dan dapat merespons input pengguna, sementara beberapa thread sedang bekerja untuk memproses data.

Dengan mengimplementasikan multithreading pada program, Anda dapat mempercepat pemrosesan data, meningkatkan efisiensi penggunaan sumber daya, dan menjaga responsifitas aplikasi.