Nama : Alpian Roymundus Siringo-Ringo
NIM : 11211009
Tugas 1 Sistem Terdistribusi

ESSAI 1:

1. Jelaskan dengan singkat perbedaan antara proses dan thread!

   Proses dan **Thread** adalah unit eksekusi pada sistem operasi. **Proses** dapat diibaratkan sebagai sebuah program yang sedang berjalan. Setiap proses berjalan secara terpisah dan tidak saling mengganggu. Memulai sebuah proses lebih lambat dibandingkan ketika memulai sebuah **Thread**. **Thread** adalah unit eksekusi yang lebih kecil dari sebuah proses. Beberapa thread dapat berada di dalam satu proses.Thread-thread ini berbagi ruang alamat memori dan sumber daya yang sama dengan proses induknya.

   Jika sebuah proses adalah sebuah rumah, maka thread adalah penghuni rumah tersebut. Setiap penghuni memiliki aktivitasnya sendiri (thread), namun mereka berbagi ruang yang sama (ruang alamat) dan sumber daya yang sama (dapur, kamar mandi, dll).

2. Sebutkan dan jelaskan 3 keuntungan dari penggunaan multithreading.
   a. **Peningkatan Responsivitas**: Dengan multithreading, tugas yang berat dapat dilakukan di thread yang berbeda, sehingga UI tetap responsif dan pengguna dapat terus berinteraksi dengan aplikasi.
   b. **Peningkatan Efisiensi**: Pada sistem multi-core, setiap thread dapat dijalankan pada core yang berbeda. Ini memungkinkan CPU untuk bekerja secara paralel dan menyelesaikan tugas lebih cepat.
   c. **Struktur Program yang Lebih baik**: Multithreading memungkinkan Anda untuk memecah program menjadi bagian-bagian yang lebih kecil (thread) yang lebih mudah dikelola dan diuji.

3. Jelaskan skenario di mana multithreading dapat meningkatkan performa aplikasi (misalnya, server web).

   Multithreading adalah teknik dalam pemrograman komputer yang memungkinkan sebuah program menjalankan beberapa tugas secara bersamaan dalam satu proses, yang mana berguna untuk meningkatkan kinerja aplikasi, terutama pada server web yang seringkali harus menangani banyak permintaan sekaligus. Dengan multithreading, setiap permintaan dapat diproses oleh thread yang berbeda, sehingga waktu respon menjadi lebih cepat dan server dapat melayani lebih banyak pengguna. Ini juga memungkinkan pemisahan tugas yang kompleks menjadi bagian-bagian yang lebih kecil, membuat program lebih mudah dikelola dan dipelihara. Tapi yang perlu diingat adalah multithreading juga membawa tantangan seperti kondisi race dan deadlock yang perlu dikelola dengan hati-hati.

KASUS 1:

1. Buat review mengenai literatur yang membahas perbedaan antara model **many-to-one**, **one-to-one**, dan **many-to-many** dalam multithreading.

   Judul Buku : Distributed Systems
   Penulis : Maarten Van Steen dan Andrew S. Tanenbaum
   Edisi Ke-4

   Multithreading adalah teknik pemrograman yang memungkinkan beberapa bagian dari sebuah program (disebut thread) berjalan secara bersamaan. Ada tiga model utama untuk mengimplementasikan multithreading: many-to-one, one-to-one, dan many-to-many.

   Model Many-to-One:

   Konsep: Dalam model ini, banyak thread dipetakan ke satu thread sistem operasi.

   Kelebihan: Efisiensi dalam hal penggunaan sumber daya sistem dan juga implementasi relatif Sederhana.

   Kekurangan: Jika satu thread melakakn operasi *blocking* (mislanya, meunggu I/O), semua thread lainnya akan terblokir, serta tidak dapat memanfaatkan sepenuhnya sisetem *multi-core.*

   Model One-to-One:

   Konsep: Setiap thread aplikasi dipetakan ke satu thread sistem operasi.

   Kelebihan: Setiap thread dapat berjalan secara independen, sehingga jika satu thread terblokir, thread lainnya dapat terus berjalan dan juga dapat memanfaatkan sepenuhnya sistem *multi-core*.

   Model Many-to-Many:

   Konsep: Model ini merupakan kombinasi dari model many-to-one dan one-to-one. Beberapa thread aplikasi dapat dipetakan ke satu thread sistem operasi, dan beberapa thread sistem operasi dapat dipetakan ke satu thread aplikasi.

   Kelebihan: Fleksibilitas dalam mengelola thread, dapat menghindari masalah blocking yang terjadi pada odel many-to-one, serta dapat meanfaatkan sepenuhnya sistem *multi-core*.

   Kekurangan: Implementasi lebih kompleks dibandingkan model lainnya.

2. Buatlah review contoh kasus dari dunia nyata di mana sistem menggunakan multithreading untuk meningkatkan efisiensi.

   Penerapan Multithreading pada permainan video yang memiliki grafis yang kompleks dan melibatkan banyak tugas.

   Rendering Grafis: Proses rendering grafis (menghasilkan gambar di layar) dapat dibagi menjadi beberapa thread, sehingga setiap bagian gambar dapat dihitung secara paralel.

   Kecerdasan Buatan: Karakter non-pemain (NPC) dalam permainan sering kali memiliki perilaku yang kompleks. Setiap NPC dapat dikontrol oleh thread yang berbeda, sehingga memungkinkan banyak NPC bergerak dan berinteraksi secara simultan.

   Multithreading telah menjadi bagian integral dalam pengembangan permainan video modern. Dengan memanfaatkan kekuatan pemrosesan paralel, pengembang dapat menciptakan pengalaman bermain yang lebih kaya dan lebih imersif. Namun, penggunaan multithreading juga memerlukan pemahaman yang mendalam tentang konsep sinkronisasi dan manajemen thread untuk menghindari masalah yang tidak diinginkan.

KASUS 2:

1. Buatlah program (pilih salah satu: java, python, rust, golang). Buat program sederhana yang menggunakan dua thread. Satu thread bertugas membaca data dari file, dan thread lain bertugas menampilkan data ke layar.

```python
import threading
import time

def read_file(filename, shared_data):
    with open(filename, 'r') as file:

        data = file.read()
        shared_data.append(data)
    time.sleep(1)

def display_data(shared_data):
    while not shared_data:
        time.sleep(0.1)
    print("Data dari file:")
    print(shared_data[0])

filename = 'file.txt'

shared_data = []

thread1 = threading.Thread(target=read_file, args=(filename, shared_data))
thread2 = threading.Thread(target=display_data, args=(shared_data,))

thread1.start()
thread2.start()

thread1.join()
thread2.join()
print("Finish.")
```

```
[Running] python -u "c:\Users\alpia\Documents\Materi dan Tugas Kuliah\Semester 7\Sistem
Terdistribusi\Tugas\multitherading.py"
Data dari file:
Contoh teks untuk file
Nama = Alpian Roymundus Siringoringo
NIM  = 11211009
Finish.

[Done] exited with code=0 in 1.248 seconds
```

2. Buat modifikasi pada program sehingga proses pembacaan file dipecah menjadi beberapa thread yang bekerja secara bersamaan.

```
import threading
import time

def read_file_part(filename, start, end, shared_data, index):
    with open(filename, 'r') as file:
        file.seek(start)
        data = file.read(end - start)
        shared_data[index] = data
    time.sleep(1)


def display_data(shared_data):
    while not all(shared_data):
        time.sleep(0.1)
    print("Data dari file:")
    for part in shared_data:
        print(part)

filename = 'file.txt'
with open(filename, 'r') as file:
    file_size = len(file.read())

num_threads = 4
chunk_size = file_size // num_threads

shared_data = [None] * num_threads
threads = []
for i in range(num_threads):
    start = i * chunk_size
    end = (start + chunk_size) if i < num_threads - 1 else file_size
    thread = threading.Thread(target=read_file_part, args=(filename, start,
end, shared_data, i))
    threads.append(thread)
    thread.start()

display_thread = threading.Thread(target=display_data, args=(shared_data,))
display_thread.start()

for thread in threads:
    thread.join()

display_thread.join()
print("Finish")
```

```
Data dari file:
Contoh teks untuk
file
Nama = Alpian
n Roymundus Siring
oringo
NIM  = 1121100
Finish

[Done] exited with code=0 in 1.226 seconds
```

Analisis hasil eksekusi program multithreading. Tuangkan dalam tulisan bagaimana multithreading meningkatkan efisiensi aplikasi.

Multithreading merupakan teknik yang memungkinkan sebuah aplikasi untuk menjalankan beberapa proses secara bersamaan (concurrent), yang dalam konteks ini digunakan untuk membaca dan menampilkan isi file. Melalui penerapan multithreading, kita dapat meningkatkan efisiensi aplikasi dengan lebih baik dalam memanfaatkan sumber daya prosesor dan waktu eksekusi. Program modifikasi multith menggunakan lebih dari satu thread untuk membaca file secara paralel, di mana setiap thread bertanggung jawab membaca bagian tertentu dari file.
Pada contoh program sebelumnya, Concurrency meminimalkan waktu tunggu di antara proses membaca dan menampilkan file dan Parallelism mempercepat proses eksekusi dengan memecah tugas menjadi bagian-bagian yang lebih kecil dan mengerjakannya secara paralel.