

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

## TUGAS PRA PERTEMUAN 12

### PEMROGRAMAN BERORIENTASI OBJEK

---

Buatlah program sederhana untuk perkalian matriks tanpa multithreading dan dengan menggunakan multithreading, kemudian jelaskan perbedaannya!

**Jawab:**

#### 1. Source Code

```
import java.util.Random;

public class Main {
    static final int MAX = 3;
    static final int MAX_THREAD = 3;
    static int[][] matA = new int[MAX][MAX];
    static int[][] matB = new int[MAX][MAX];
    static int[][] matC = new int[MAX][MAX];
    static int[][] matD = new int[MAX][MAX];
    static int step_i = 0;

    static class Worker implements Runnable {
        int i;

        Worker(int i) {
            this.i = i;
        }

        @Override
        public void run() {
            for (int j = 0; j < MAX; j++) {
                for (int k = 0; k < MAX; k++) {
                    matC[i][j] += matA[i][k] * matB[k][j];
                }
            }
        }
    }

    public static void main(String[] args) {
        Random rand = new Random();

        for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++) {
                matA[i][j] = rand.nextInt(10);
                matB[i][j] = rand.nextInt(10);
            }
        }

        System.out.println("Matrix A");
        for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++) {
                System.out.print(matA[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

```
System.out.println("Matrix B");
for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
        System.out.print(matB[i][j] + " ");
    }
    System.out.println();
}

Thread[] threads = new Thread[MAX_THREAD];

for (int i = 0; i < MAX_THREAD; i++) {
    threads[i] = new Thread(new Worker(step_i++));
    threads[i].start();
}

for (int i = 0; i < MAX_THREAD; i++) {
    try {
        threads[i].join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

System.out.println("Multiplication of A and B with
multithread");
for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
        System.out.print(matC[i][j] + " ");
    }
    System.out.println();
}

for(int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
        matD[i][j] = 0;
        for (int k = 0; k < MAX; k++) {
            matD[i][j] += matA[i][k] * matB[k][j];
        }
    }
}

System.out.println("Multiplication of A and B without
multithread");
for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
        System.out.print(matD[i][j] + " ");
    }
    System.out.println();
}

}
}
```

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

## 2. Output

```
Matrix A
1 1 9
5 9 8
1 3 2
Matrix B
1 2 9
8 0 9
9 5 3
Multiplication of A and B with multithread
90 47 45
149 50 150
43 12 42
Multiplication of A and B without multithread
90 47 45
149 50 150
43 12 42
```

## 3. Penjelasan

Program tersebut berisi perkalian matriks dengan 2 cara. Cara yang pertama yaitu memanfaatkan multithread. Thread pada CPU dimanfaatkan untuk menghitung perkalian pada tiap-tiap baris matriks A dan tiap-tiap kolom matriks B sekaligus lalu dilakukan join untuk menggabungkan hasilnya. Sehingga penghitungan dapat dilakukan dengan lebih cepat. Kompleksitas waktu jika menggunakan cara ini yaitu  $O(1)$ .

Sedangkan, pada cara kedua yaitu cara biasa, penghitungan dilakukan secara berurut dengan mengalikan baris pertama matriks A dengan kolom kedua matriks B sampai baris terakhir. Kompleksitas waktu jika menggunakan cara ini yaitu  $O(n^3)$ .