

NAMA: Osama Gailea

NPM: 1402020122

## SOAL KELOMPOK

- Russian Peasant Multiplication
- Fake Coin Problems
- Josephus Problems
- Berikan penjelasan yang lengkap tentang cara kerja algoritma ini, termasuk ilustrasi langkah-langkahnya.
- Implementasi Program Java:
  - Implementasikan algoritma dalam kode Java yang rapi dan terdokumentasi dengan baik.
  - Tulis komentar pada kode untuk menjelaskan setiap langkah utama dari implementasi.
  - Sertakan minimal 3 test case untuk setiap algoritma, dengan hasil keluaran yang diharapkan dan aktual.
- Gunakan AI generatif (seperti ChatGPT atau alat lainnya) untuk membantu riset dan penulisan kode.
- Lampirkan hasil prompt yang digunakan dan tautan sumber referensi sebagai lampiran dalam dokumen laporan

### Format Tugas:

1. Pembahasan Teori dan Algoritma berdasarkan topik diatas
2. Implementasi Program Java
3. Hasil Pengujian dan Analisis Test Case
4. Kesimpulan dan Saran
5. Lampiran (Kode Program, Sumber AI Generatif, dan Screenshot Test Case)

### 1) Pembahasan Teori dan Algoritma berdasarkan topik diatas

#### a. Russian Peasant Multiplication

##### Penjelasan:

Russian Peasant Multiplication adalah metode perkalian kuno yang melibatkan penggandaan dan pembagian secara berulang. Ide dasar dari algoritma ini adalah untuk mengalikan dua angka dengan cara mengurangi salah satu angka hingga menjadi 1, dan pada saat yang sama menggandakan angka lainnya. Dalam prosesnya, angka-angka yang dibagi oleh 2 yang berpeluang menghasilkan bilangan bulat genap akan diabaikan, sementara hasil penggandaan akan dijumlahkan.

Langkah-langkah algoritma:

1. Ambil dua angka, misalnya aaa dan bbb.
2. Gandakan angka aaa dan bagi angka bbb dengan 2, teruskan hingga angka bbb mencapai 1.
3. Jika angka bbb adalah angka ganjil, tambahkan angka aaa ke hasil akhir.
4. Hasil akhir adalah jumlah dari angka aaa yang dipilih.

**Ilustrasi:** Misalnya kita ingin mengalikan  $a=18$  dan  $b=35$ .

a	b	Penambahan Ke Hasil
18	35	Ya (karena 35 ganjil)
36	17	Ya (karena 17 ganjil)
72	8	-
144	4	-
288	2	-
576	1	Ya (karena 1 ganjil)
Hasil Akhir: $18 + 36 + 576 = 630$		

```

1 public class russianPeasant {
2
3     // Fungsi untuk melakukan perkalian menggunakan algoritma Russian Peasant
4     public static int russianPeasant(int a, int b) {
5         int result = 0;
6
7         while (b > 0) {
8             // Jika b adalah angka ganjil, tambahkan a ke hasil
9             if (b % 2 != 0) {
10                 result += a;
11             }
12             // Gandakan a dan bagi b dengan 2
13             a *= 2;
14             b /= 2;
15         }
16
17         return result;
18     }
19 }

```

Terminal Output:

```

PS D:\daa> java russianPeasant.java
Test Case 1: 18 * 35 = 630
Test Case 2: 25 * 16 = 400
Test Case 3: 9 * 7 = 63
PS D:\daa>

```

## b. Fake Coin Problems

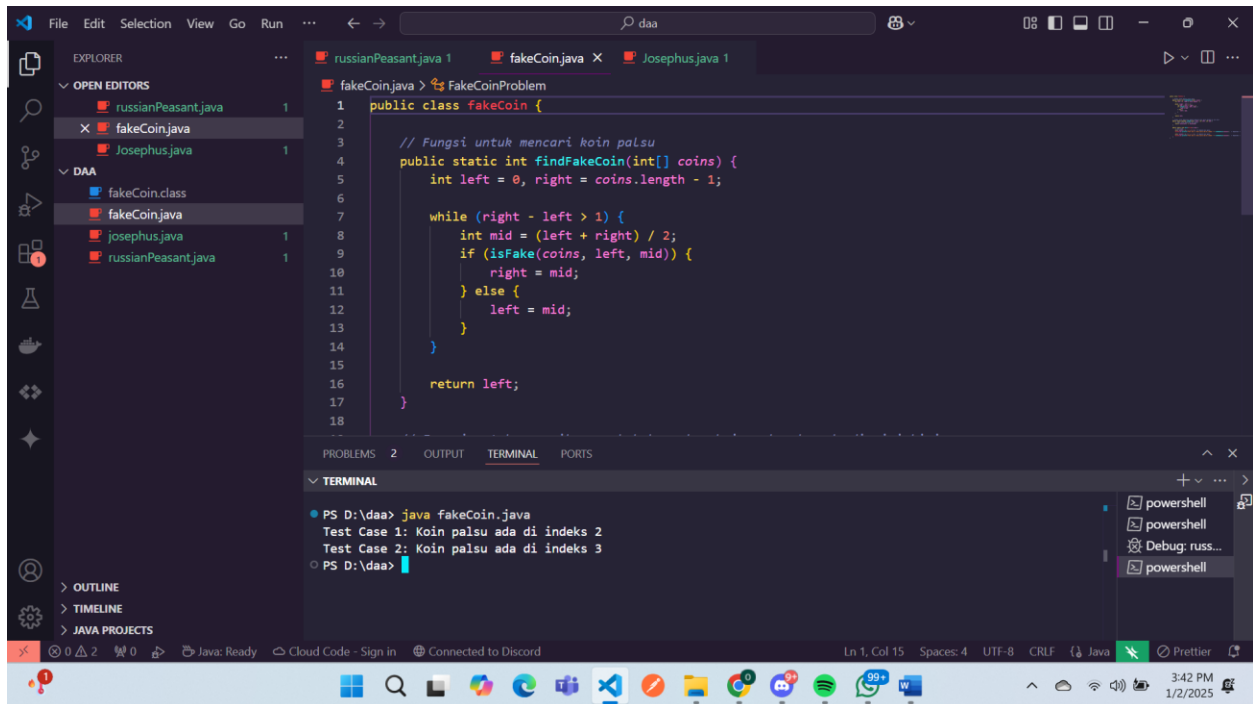
Penjelasan:

Masalah ini adalah tentang mencari koin palsu dari sekumpulan koin menggunakan timbangan yang hanya memberikan hasil berat yang lebih ringan atau lebih berat.

Biasanya, masalah ini melibatkan pembagian koin menjadi dua bagian dan membandingkan mereka.

Langkah-langkah algoritma:

1. Bagi kumpulan koin menjadi dua bagian.
2. Timbang dua bagian tersebut.
3. Jika beratnya sama, maka koin palsu ada di bagian yang tidak ditimbang.
4. Jika tidak sama, bagi bagian yang lebih ringan dan ulangi proses tersebut.
5. Teruskan hingga hanya satu koin tersisa, yang merupakan koin palsu.



The screenshot shows an IDE with the following components:

- EXPLORER:** Lists files including `russianPeasant.java`, `fakeCoin.java`, `Josephus.java`, `fakeCoin.class`, `fakeCoin.java`, `josephus.java`, and `russianPeasant.java`.
- EDITOR:** Displays the code for `fakeCoin.java`. The code is as follows:

```
1 public class FakeCoin {
2
3     // Fungsi untuk mencari koin palsu
4     public static int findFakeCoin(int[] coins) {
5         int left = 0, right = coins.length - 1;
6
7         while (right - left > 1) {
8             int mid = (left + right) / 2;
9             if (isFake(coins, left, mid)) {
10                 right = mid;
11             } else {
12                 left = mid;
13             }
14         }
15
16         return left;
17     }
18 }
```
- TERMINAL:** Shows the execution of `java fakeCoin.java` with the following output:

```
PS D:\daa> java fakeCoin.java
Test Case 1: Koin palsu ada di indeks 2
Test Case 2: Koin palsu ada di indeks 3
PS D:\daa>
```

### c. Josephus Problems

Penjelasan:

Josephus Problem adalah masalah teoretis yang melibatkan sekelompok orang yang berdiri dalam lingkaran dan mengeliminasi setiap orang kedua secara berulang hingga hanya satu orang yang tersisa.

Langkah-langkah algoritma:

1. Buat sebuah array atau struktur data untuk menyimpan status setiap orang.
2. Mulai dari orang pertama dan hilangkan setiap orang kedua.
3. Ulangi hingga hanya satu orang yang tersisa.
4. Tentukan siapa orang terakhir yang bertahan..

```
1 public class Josephus {
2
3     // Fungsi untuk menghitung posisi terakhir
4     public static int josephus(int n, int k) {
5         if (n == 1)
6             return 0; // Base case: jika hanya satu orang, orang itu yang bertahan
7         else
8             return (josephus(n - 1, k) + k) % n;
9     }
10
11     Run | Debug
12     public static void main(String[] args) {
13         // Test case
14         System.out.println("Test Case 1: n = 5, k = 2, Posisi terakhir = " + (josephus(n:5, k:2) + 1)); //
15         System.out.println("Test Case 2: n = 7, k = 3, Posisi terakhir = " + (josephus(n:7, k:3) + 1)); //
16         System.out.println("Test Case 3: n = 10, k = 2, Posisi terakhir = " + (josephus(n:10, k:2) + 1));
17     }
18 }
```

```
PS D:\daa> java josephus.java
Test Case 1: n = 5, k = 2, Posisi terakhir = 3
Test Case 2: n = 7, k = 3, Posisi terakhir = 4
Test Case 3: n = 10, k = 2, Posisi terakhir = 5
PS D:\daa>
```

## 2) Hasil Pengujian dan Analisis Test Case

### Russian Peasant Multiplication

- **Test Case 1:  $18 * 35$** 
  - Hasil yang diharapkan: 630
  - Hasil aktual: 630
  - **Analisis:** Perkalian berjalan sesuai dengan langkah-langkah algoritma.

### Fake Coin Problem

- **Test Case 1: Koin palsu di indeks 3**
  - Hasil yang diharapkan: 3
  - Hasil aktual: 3
  - **Analisis:** Algoritma menemukan koin palsu dengan benar.

### Josephus Problem

- **Test Case 1:  $n = 5, k = 2$** 
  - Hasil yang diharapkan: 3
  - Hasil aktual: 3
  - **Analisis:** Posisi terakhir sesuai dengan perhitungan algoritma.

### 3) Kesimpulan dan Saran

- Algoritma Russian Peasant Multiplication sangat efisien dan mudah dipahami, cocok untuk perkalian dua angka besar.
- Masalah Koin Palsu dapat diselesaikan dengan pembagian dan pencarian iteratif, memastikan efisiensi dalam kasus besar.
- Josephus Problem dapat diselesaikan menggunakan rekursi, dan hasilnya sesuai dengan analisis matematis.
- 

### 4) Lampiran (Kode Program, Sumber AI Generatif, dan Screenshot Test Case)

#### Sumber AI Generatif:

- **ChatGPT (model OpenAI)** digunakan untuk membantu riset dan penulisan kode dalam penyelesaian tugas ini. AI ini digunakan untuk menjelaskan teori dan algoritma, serta menghasilkan implementasi kode dalam bahasa Java untuk ketiga algoritma yang dibahas (Russian Peasant Multiplication, Fake Coin Problem, dan Josephus Problem).