

# TUGAS PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 1

Pertemuan ke-7

BAB 9: Fungsi/Method

Deadline: 10 November 2024 Pukul 18:29

## Petunjuk

- Kerjakan semua soal di bawah ini dengan menggunakan bahasa Java.
- Implementasikan konsep Fungsi/Method sesuai materi Modul 9.
- Program harus dapat di-compile dan di-run tanpa error.
- Nama file source code (.java) harus sesuai dengan nama class.
- Kumpulkan file source code (.java) untuk setiap program dan laporan praktikum (.pdf).
- Source code di dalam laporan wajib dilampirkan menggunakan syntax highlighter.
- Format laporan praktikum dapat dilihat di myITS Classroom.
- Penamaan file laporan praktikum adalah LaporanPraktikum7\_Kelompok1\_Nama\_Lengkap.pdf.
- Hasil penggerjaan dikumpulkan di myITS Classroom dalam satu file (.zip) dengan nama LaporanPraktikum7\_Kelompok1\_Nama\_Lengkap.zip yang berisi file source code (.java) dan laporan praktikum (.pdf).
- Deadline pengumpulan: **10 November 2024 Pukul 18:29**

## Soal 1 Kalkulator Pola Bilangan Rekursif

Buat program kalkulator yang dapat menghitung berbagai pola bilangan menggunakan method rekursif. Program harus memiliki menu untuk memilih jenis pola bilangan yang ingin dihitung.

### Spesifikasi Program:

- Tampilkan menu pilihan pola bilangan:
  1. Fibonacci
  2. Faktorial
  3. Segitiga Pascal
  4. Keluar
- Gunakan method rekursif untuk setiap perhitungan
- Program berjalan sampai user memilih keluar

### Rumus Pola Bilangan:

- **Fibonacci:**  $F(n) = F(n - 1) + F(n - 2)$  dengan  $F(1) = 1, F(2) = 1$
- **Faktorial:**  $n! = n \times (n - 1)!$  dengan  $0! = 1$
- **Segitiga Pascal:**  $C(n, k) = C(n - 1, k - 1) + C(n - 1, k)$  dengan  $C(n, 0) = 1, C(n, n) = 1$

### Contoh Output:

```
==== KALKULATOR POLA BILANGAN ====
1. Fibonacci
2. Faktorial
3. Segitiga Pascal
4. Keluar

Pilihan Anda: 1
Masukkan nilai n: 7
F(7) = 13

Pilihan Anda: 2
Masukkan nilai n: 5
5! = 120

Pilihan Anda: 3
Masukkan nilai n: 4
Baris ke-4: 1 3 3 1

Pilihan Anda: 4
Terima kasih!
```

### Requirements:

- Buat method rekursif untuk setiap pola bilangan dengan base case yang tepat
- Gunakan method overloading jika diperlukan untuk handling input yang berbeda
- Program harus dapat menangani input yang tidak valid (misal: bilangan negatif) dan menampilkan pesan error yang jelas

## Soal 2 Petualangan Java Mencari Harta Karun

Java adalah seorang petualang pemberani yang gemar menjelajah dunia pemrograman. Suatu hari, ia mendengar kisah tentang sebuah harta karun legendaris yang tersembunyi jauh di Pulau A-1. Tanpa ragu, Java memulai perjalanan panjang untuk menemukannya.

Namun, perjalanan Java tidak mudah. Ia harus mengatur energi, melangkah sejauh 50 satuan, dan terkadang mencari petunjuk rahasia agar tidak tersesat. Tugas kamu adalah membantu Java menyelesaikan misinya dengan melengkapi method yang belum diimplementasikan.

Lengkapilah program berikut dengan mengisi bagian yang ditandai. Ikuti setiap instruksi pada komentar agar petualangan Java berjalan dengan sukses!

```
1 import java.util.Scanner;
2
3 public class PetualanganJava {
4     private static int energi = 100;
5     private static int jarak = 0;
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9
10        System.out.println("== PETUALANGAN JAVA ==");
11        System.out.println("Energi awal: " + energi);
12
13        // LENKAPI 1: Panggil method tampilkanPeta()
14
15        while (energi > 0 && jarak < 50) {
16            System.out.println("\nEnergi: " + energi + " | Jarak: " +
17                jarak + "/50");
18            System.out.print("Aksi (1=Jalan, 2=Istirahat,
19                3=Petunjuk): ");
20            int aksi = input.nextInt();
21
22            // LENKAPI 2: Gunakan switch-case untuk aksi
23            // case 1 -> jalan() lalu tambahkan langkah ke jarak
24            // case 2 -> istirahat()
25            // case 3 -> tampilkan petunjuk dari cariPetunjuk()
26            // default -> aksi tidak valid
27
28            System.out.println("Energi sekarang: " + energi);
29
30            if (energi <= 20 && energi > 0) {
31                System.out.println("Energi menipis! Disarankan
32                    istirahat.");
33            }
34
35            if (energi <= 0) {
36                System.out.println("\nGame Over! Energi habis.");
37                break;
38            }
39
40            if (cekHarta()) {
41                System.out.println("\nSELAMAT! Harta ditemukan!");
42            }
43
44        }
45    }
46}
```

```
39         break;
40     }
41 }
42
43     input.close();
44 }
45
46 public static void tampilkanPeta() {
47     // LENGKAPI 3: Cetak peta perjalanan Java
48 }
49
50 public static int jalan() {
51     // LENGKAPI 4: Generate langkah 1-10
52     // LENGKAPI 5: Energi berkurang 5-15, energi tidak boleh
53     // negatif
54     // LENGKAPI 6: Tampilkan berapa langkah dan energi berkurang
55     return 0; // ubah dengan langkah sebenarnya
56 }
57
58 public static void istirahat() {
59     // LENGKAPI 7: Tambah energi 20-40, maksimal 100
60     // LENGKAPI 8: Cetak pesan energi bertambah
61 }
62
63 public static String cariPetunjuk() {
64     // LENGKAPI 9: Energi berkurang 10
65     // LENGKAPI 10: Return petunjuk acak dari array string
66     return ""; // ubah dengan petunjuk sebenarnya
67 }
68
69 public static boolean cekHarta() {
70     // LENGKAPI 11: return true jika jarak >= 50
71     return false;
72 }
```

**Contoh Output:**

```
==== PETUALANGAN JAVA ====
Energi awal: 100

PETA:
Start [ ]-[ ]-[ ]-[ ]-[ ]-[HARTA]
    0   10  20  30  40  50

Energi: 100 | Jarak: 0/50
Aksi (1=Jalan, 2=Istirahat, 3=Petunjuk): 1
Anda berjalan 7 langkah! (-8 energi)
Energi sekarang: 92

Energi: 92 | Jarak: 7/50
Aksi (1=Jalan, 2=Istirahat, 3=Petunjuk): 3
Petunjuk: Terus maju hingga jarak 50!
Energi sekarang: 82

SELAMAT! Harta ditemukan!
```

**Petunjuk:**

- Gunakan `Math.random()` untuk angka acak
- Energi tidak boleh negatif
- Jika energi habis, permainan langsung berakhir
- Untuk method `cariPetunjuk()` gunakan array string yang berisi beberapa petunjuk. Konsep array akan kita bahas pada modul 10, tapi untuk sementara ini sebagai pengenalan:

```
1 String[] petunjuk = {
2     "Terus maju hingga jarak 50!",
3     "Jangan lupa istirahat saat energi menipis!",
4     "Harta karun berada di jarak 50 satuan!",
5     "Setiap langkah membutuhkan energi!",
6     "Gunakan aksi 'Istirahat' untuk mengisi energi!"
7 };
8 int index = (int)(Math.random() * petunjuk.length);
```