

Nama: Marvelino Husca

NIM: 254107020184

Kelas: TI – 1H

## Jobsheet 12

### Percobaan 1:

1. Apa yang dimaksud dengan fungsi rekursif?
  - Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan tugas.
2. Bagaimana contoh kasus penggunaan fungsi rekursif?
  - Fungsi rekursif sangat cocok digunakan untuk masalah yang bisa dipecah menjadi sub-masalah yang lebih kecil namun mirip dengan masalah aslinya. Contohnya faktorial, deret fibonacci atau pangkat, struktur data pohon (tree), algoritma sorting, game/puzzle
3. Perbedaan hasil dan alur jalan program (rekursif vs iteratif)
  - Hasilnya sama tapi prosesnya akan berbeda
  - Berikut adalah perbedaan cara kerjanya ibarat mengerjakan tugas:

#### **1. Alur Rekursif (Gerakan Turun lalu Naik)**

Fungsi ini ibarat menyelam ke dasar laut untuk mengambil mutiara, lalu berenang kembali ke permukaan.

- **Fase Turun (Memanggil Diri Sendiri)** Pertama, Fungsi 5 memanggil Fungsi 4. Kemudian, Fungsi 4 memanggil Fungsi 3. Lalu, Fungsi 3 memanggil Fungsi 2. Terakhir, Fungsi 2 memanggil Fungsi 1. Di sini proses berhenti turun karena dasar sudah dicapai.
- **Fase Naik (Mengembalikan Nilai)** Fungsi 1 melapor bahwa hasilnya adalah 1. Fungsi 2 menerima laporan itu, lalu menghitung 2 dikali 1 hasilnya 2. Fungsi 3 menerima laporan itu, lalu menghitung 3 dikali 2 hasilnya 6. Fungsi 4 menerima laporan itu, lalu menghitung 4 dikali 6 hasilnya 24. Fungsi 5 menerima laporan itu, lalu menghitung 5 dikali 24 hasilnya 120. Selesai.

#### **2. Alur Iteratif (Gerakan Maju Lurus)**

Fungsi ini ibarat menggelindingkan bola salju ke depan. Bola mulai dari kecil, digulung terus ke depan sampai menjadi besar.

- **Langkah Awal:** Kita mulai dengan nilai 1.
  - **Perulangan 1:** Nilai 1 dikali 5 menjadi 5. Angka disimpan.
  - **Perulangan 2:** Nilai 5 dikali 4 menjadi 20. Angka disimpan.
  - **Perulangan 3:** Nilai 20 dikali 3 menjadi 60. Angka disimpan.

- **Perulangan 4:** Nilai 60 dikali 2 menjadi 120. Angka disimpan.
- **Hasil Akhir:** Nilai 120 diambil sebagai jawaban final.

**Kesimpulan Perbedaan Alur:**

- **Rekursif:** Bergerak dua arah. Pergi dengan tangan kosong, pulang membawa hasil perhitungan.
- **Iteratif:** Bergerak satu arah. Maju terus ke depan sambil menghitung hasil saat itu juga.

**Percobaan 2:**

1. Sampai kapan proses pemanggilan fungsi tersebut akan dijalankan?
  - Proses pemanggilan fungsi `hitungPangkat()` akan terus dilakukan berulang-ulang sampai nilai `y` (pangkat) berkurang menjadi 0. Setiap fungsi dipanggil, nilai `y` akan dikurangi 1. Jika `y` masih `> 0` maka fungsi memanggil diri nya sendiri lagi, tapi jika `y = 0` maka fungsi berhenti memanggil dan mengembalikan nilai.
2. Modifikasi

**Percobaan 3:**

1. Analisis base case dan recursion call:
  - Base case Adalah kondisi dimana fungsi berhenti memanggil dirinya sendiri. Contohnya Adalah `if (tahun == 0){ return saldo;}`
  - Recursion call adalah baris Dimana fungsi memanggil dirinya sendiri dengan parameter yang berubah. Contohnya adalah `return (1.11 * hitungLaba(saldo, tahun - 1));`
2. Trace Algoritma `hitungLaba(100000, 3)`:
 

A. Fase Ekspansi (Penundaan Perhitungan):

  - Panggilan 1: `hitungLaba(100000, 3)`
    - Cek: Apakah `tahun == 0`? Tidak
    - Aksi: `1.11 * hitungLaba(100000, 2)`
  - Panggilan 2: `hitungLaba(100000, 2)`:
    - Cek: Apakah `tahun == 0`? Tidak
    - Aksi: `1.11 * hitungLaba(100000, 1)`
  - Panggilan 3: `hitungLaba(100000, 1)`
    - Cek: Apakah `tahun == 0`? Tidak
    - Aksi: `1.11 * hitungLaba(100000, 0)`
  - Panggilan 4: `hitungLaba(100000, 0)`
    - Cek: Apakah `tahun == 0`? Ya
    - Aksi: `return 100000`

## B. Fase Substitusi (Pengembalian Nilai)

- Kembali ke Panggilan 3:
  - Hitung:  $1.11 * 100000$
  - Hasil: 111.000
- Kembali ke Panggilan 2:
  - Hitung:  $1.11 * 111000$
  - Hasil: 123.210
- Kembali ke Panggilan 1 (Awal):
  - Hitung:  $1.11 * 123210$
  - Hasil Akhir: 136.763,1

Tugas