**JOBSHEET 14**

**DASAR PEMROGRAMAN**

****

**ALVINO VALERIAN D.R**

**2341720027**

**D-IV TEKNIK INFORMATIKA**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023**

**Pertanyaan**1. Apa yang dimaksud dengan fungsi rekursif?

**Fungsi rekursif dalam Java merujuk pada kemampuan suatu fungsi untuk memanggil dirinya sendiri selama proses eksekusi. Dengan kata lain, fungsi tersebut dapat melakukan pemanggilan diri sendiri untuk menyelesaikan tugas yang lebih kompleks.**  
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

**Perhitungan Faktorial, Pencarian dalam Struktur Data, Pembagian dan Penaklukan, Fibonacci Sequence**  
3. Pada **Percobaan1**, apakah hasil yang diberikan fungsi **faktorialRekursif()** dan fungsi **faktorialIteratif()** sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

**Pada percobaan ini, hasil yang diberikan oleh fungsi factorialRekursif() dan factorialLiteratif() akan sama**

**Fungsi factorialRekursif():**

**-Program akan memanggil fungsi rekursif factorialRekursif(5).**

**-Fungsi ini akan memanggil dirinya sendiri dengan nilai yang berkurang hingga mencapai base case (ketika n == 0).**

**-Setiap pemanggilan rekursif akan mengalikan nilai n dengan hasil dari pemanggilan rekursif sebelumnya.**

**-Proses ini terus berlanjut hingga mencapai base case, dan hasilnya dikembalikan ke pemanggil sebelumnya.**

**Fungsi factorialLiteratif():**

**-Program menggunakan pendekatan iteratif dengan menggunakan loop for.**

**-Sebuah variabel faktor diinisialisasi dengan nilai 1.**

**-Loop akan berjalan dari n ke 1, mengalikan faktor dengan nilai iterasi saat ini.**

**-Proses ini terus berlanjut hingga loop selesai, dan hasil faktorial dikembalikan.**

**Perbedaan Utama:**

**Fungsi rekursif menggunakan pemanggilan diri sendiri dan memerlukan pemanggilan rekursif berulang sampai mencapai base case.Fungsi iteratif menggunakan loop untuk mengulangi operasi secara berurutan hingga mencapai kondisi berhenti.**

**Pertanyaan**1. Pada **Percobaan2**, terdapat pemanggilan fungsi rekursif **hitungPangkat(bilangan,pangkat)** pada fungsi main, kemudian dilakukan pemanggilan fungsi hitungPangkat() secara berulangkali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!

**Dalam contoh ini, proses pemanggilan fungsi hitungPangkat() akan terus berlangsung hingga mencapai base case ketika pangkat == 0. Pada titik ini, nilai 1 akan dikembalikan, dan proses rekursif akan berhenti. Jadi, proses pemanggilan fungsi tersebut akan dihentikan ketika pangkat mencapai nilai 0.**  
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh :  
**hitungPangkat(2,5)** dicetak 2x2x2x2x2x1 = 32



**Pertanyaan**1. Pada **Percobaan3**, sebutkan blok kode program manakah yang merupakan **“base case”**

dan **“recursion call”!**



2. Jabarkan trace fase ekspansi dan fase subtitusi algoritma perhitungan laba di atas jika diberikan nilai **hitungLaba(100000,3)**



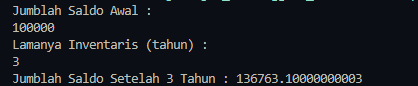
**Fase Ekspansi (Expanding Phase):**

* **Pemanggilan pertama: hitungLaba(100000, 3)**
* **Pada pemanggilan ini, fungsi akan melakukan pemanggilan dirinya sendiri dengan nilai tahun yang berkurang satu, yaitu hitungLaba(100000, 2).**
* **Pemanggilan kedua: hitungLaba(100000, 2)**
* **Pada pemanggilan ini, fungsi kembali memanggil dirinya sendiri dengan nilai tahun yang kembali berkurang satu, yaitu hitungLaba(100000, 1).**
* **Pemanggilan ketiga: hitungLaba(100000, 1)**
* **Karena nilai tahun sekarang adalah 0 (basis atau base case), fungsi langsung mengembalikan nilai saldo (yaitu 100000).**

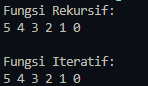
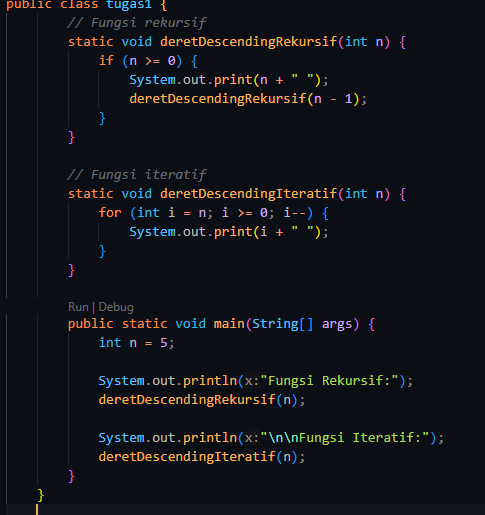
**Fase Substitusi (Substitution Phase):**

* **Pemanggilan ketiga mengembalikan nilai 100000.**
* **Pemanggilan kedua menggunakan nilai kembalian dari pemanggilan ketiga dan mengalikannya dengan 1.11, yaitu 1.11 \* 100000.**
* **Pemanggilan pertama menggunakan nilai kembalian dari pemanggilan kedua dan mengalikannya lagi dengan 1.11, yaitu 1.11 \* (1.11 \* 100000).**

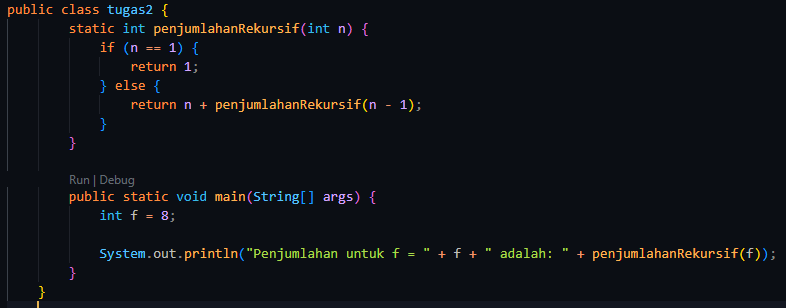
**Hasil =**



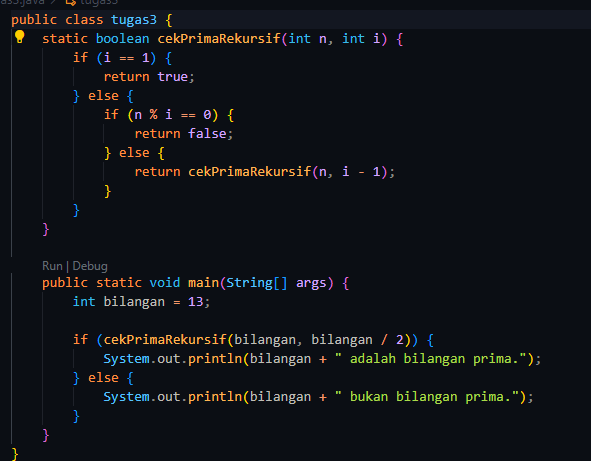
**TUGAS**



**2.**



**3.**



**4.**

