**LAPORAN PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET 5**



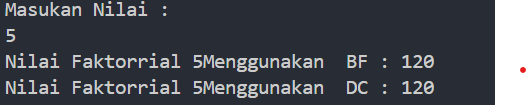
**NAMA = RAKAGALI RESDA KRISANDI PUTRA**

**KELAS = TI -1B / 19**

**NIM = 244107020136**

**PERCOBAAN 1**

* **Verifikasi Hasil Percobaan**

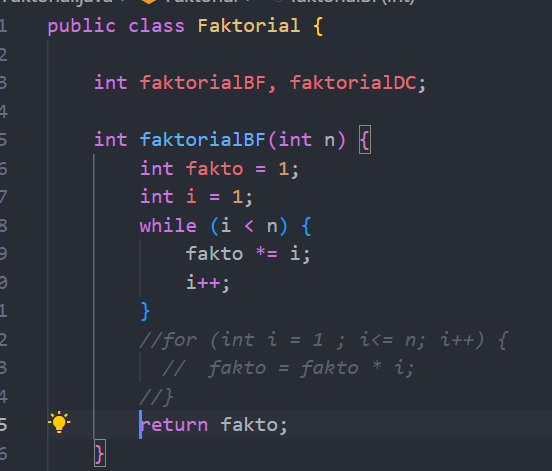
****

1. **Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!**

If digunakan sebagai base case atau batas nilai itu di cari, sedangkan untuk else digunakan untuk pemanggilan rekursif jika n lebih besar dari 1 maka proses rekursif ini akan terus berjalan dan akan berhenti Ketika sudah mencapai base case

1. **Apakah memungkinkan perulangan pada method faktorialBF() diubah selain menggunakan for? Buktikan**

**Bisa**

****

1. **Jelaskan perbedaan antara fakto \*= i; dan int fakto = n \* faktorialDC(n-1); !**

- fakto \*= I digunakan dalam iterasi Dimana nilai di perbarui setiap selesai iterasi tanpa memanggil suatu fungsi terlebih dahulu

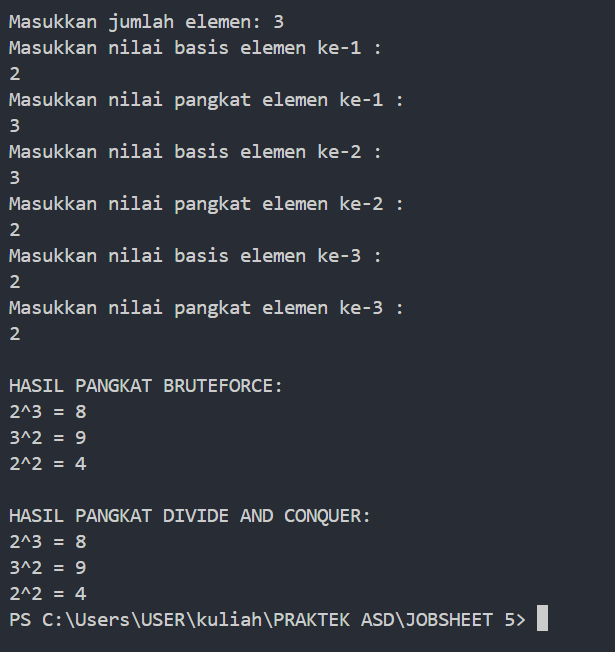
Fakto = n \* faktorialDc(n-1) ; digunakan untuk melakukan pemanggilan rekursif dimana fungsi dipanggil Ketika ingin melakukakan operasi dan akan berhenti Ketika mencapai base case

1. **Buat Kesimpulan tentang perbedaan cara kerja method faktorialBF() dan faktorialDC()!**

Kesimpulan factorialBf() digunakan ketika program anda membutuhkan efisiensi waktu dan penyimpanan, sedangkan faktorialDc() ketika anda ingin kode tersebut lebih mudah di pahami

**PERCOBAAN 2**

* **Verifikasi Hasil Percobaan**

****

* **Pertanyaan**

1. **Jelaskan mengenai perbedaan 2 method yang dibuat yaitu pangkatBF() dan pangkatDC()!**

Pada method pangkatBF() menggunakan perulangan for untuk mengalika angka secara berulang sebanyak nilai pangkat.sedangkan pangkatDC() melakukan perhitungan secara rekursif dengan membagi pangkat menjadi setengah untuk mengurangi jumlah perhitungan.

1. **Apakah tahap combine sudah termasuk dalam kode tersebut? Tunjukkan!**

Ya, terdapat pada method pangkatDC() Ketika rekursi dijalankan nilai pangkat dibagi menjadi 2(n/2), sehingga terdapat dua hasil perhitungan.

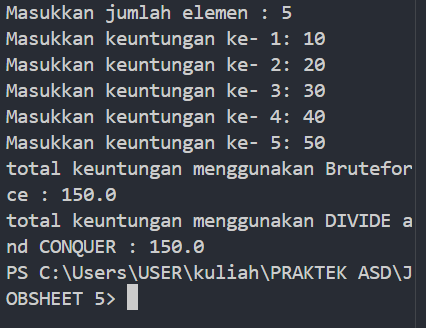
**3.** Bisa dibuat tanpaparameter karena bisa langsung menggunakan atribut nilai dan pangkat

1. **Tarik tentang cara kerja method pangkatBF() dan pangkatDC()!**.

Metode pangkatBF() bekerja secara iteratif dengan kompleksitas O(n) yang kurang optimal untuk pangkat besar, sedangkan pangkatDC() menggunakan pendekatan rekursif dengan kompleksitas O(log n) yang lebih efisien karena membagi masalah menjadi bagian-bagian lebih kecil.

**PERCOBAAN 3**

* **Verifikasi Hasil Percobaan**

****

* **Pertanyaan**

1. **Kenapa dibutuhkan variable mid pada method TotalDC()?**

Untuk membagi masalah menjadi dua bagian agar bisa diselesaikan dengan cepat

1. **Untuk apakah statement di bawah ini dilakukan dalam TotalDC()?**

Memebagi menjadi dua tempat lsum untuk menyelesaikan permasalahan dari l sampai mid dan untuk rsum menyelesaikan masalah dari mid+1 sampai r

1. **Kenapa diperlukan penjumlahan hasil lsum dan rsum seperti di bawah ini?**

Untuk menggabungkan / combine hasil dari pemecahan masalah.

1. **Apakah base case dari totalDC()?**

If (l == r){

Return arr[l];

}

1. **Tarik Kesimpulan tentang cara kerja totalDC()**

Fungsi totalDC() menggunakan metode **Divide and Conquer** untuk membagi array menjadi dua bagian hingga mencapai elemen tunggal, lalu menggabungkan hasilnya secara rekursif, sehingga menghasilkan perhitungan yang lebih efisien dengan kompleksitas **O(n log n)** dibandingkan metode iteratif