**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET PERTEMUAN KE-5**

****

**IRSA CAHAYA WIDODO (15) | 2341720193  
TI-1A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

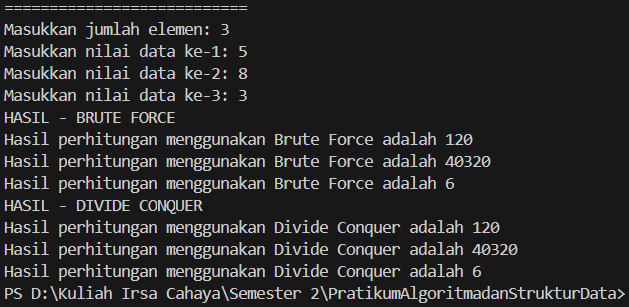
**4.2 Menghitung Faktorial dengan Algoritma Brute Force dan Divide Conquer**

package Pekan5;public class Faktorial15 { public int nilai; public int faktorialBF(int n) { int fakto = 1; for (int i = 1; i <= n; i++) { fakto = fakto \* i; } return fakto; } public int faktorialDC(int n) { if (n == 1) { return 1; } else { int fakto = n \* faktorialDC(n-1); return fakto; } }}

package Pekan5;import java.util.Scanner;public class MainFaktorial15 { public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.println("==========================="); System.out.print("Masukkan jumlah elemen: "); int iJml = sc.nextInt(); Faktorial15[] fk = new Faktorial15[10]; for (int i = 0; i < iJml; i++) { fk[i] = new Faktorial15(); System.out.print("Masukkan nilai data ke-" + (i+1) + ": "); fk[i].nilai = sc.nextInt(); } sc.close(); System.out.println("HASIL - BRUTE FORCE"); for (int i = 0; i < iJml; i++) { System.out.println ("Hasil perhitungan menggunakan Brute Force adalah "+ fk[i].faktorialBF(fk[i].nilai)); }

System.out.println("HASIL - DIVIDE CONQUER"); for (int i = 0; i < iJml; i++) { System.out.println ("Hasil perhitungan menggunakan Divide Conquer adalah "+ fk[i].faktorialDC(fk[i].nilai)); } }}

* Output



**4.2.3 Pertanyaan**

1. Jelaskan mengenai base line Algoritma Divide Conquer untuk melakukan pencarian nilai factorial, jelaskan perbedaan bagian kode pada penggunaan if else!

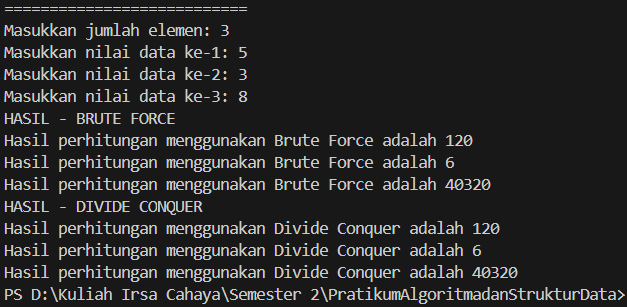
**Jawab:** Ketika n == 1, maka proses rekursi akan dihentikan. Selama n != 1, maka fungsi faktorialDC akan memanggil dirinya dengan n-1 sampai mencapai n==1. Ketika n==1, maka faktorialDC(n) akan mengembalikan nilai n saat itu. Kemudian nilai 1 akan mengembalikan nilai factorialDC(n) sebelumnya. Kemudian nilai faktorialDC(n) tersebut akan dikalikan dengan nilai sebelumnya.

1. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for?Buktikan!

**Jawab:** Memungkinkan. Menggunakan while loop

public int faktorialBP(int n) { int fakto = 1; int i = 1; while (i <= n) { fakto \*= i; i++; } return fakto; }

* **Output**



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

**Jawab:** Perbedaanya ialah, ketika fakto \*= i maka perkaliannya akan runtut dari awal. Sedangkan untuk int fakto = n\*faktorialDC(n-1) dibagi menjadi sub masalah kecil, selama n != 1, maka akan n-1. Ketika sudah mencapai 1, maka secara rekursif akan mengalikan n \* factorial(n-1) sampai ke n awal. Kemudian hasil perkalian tersebut akan dicombine.

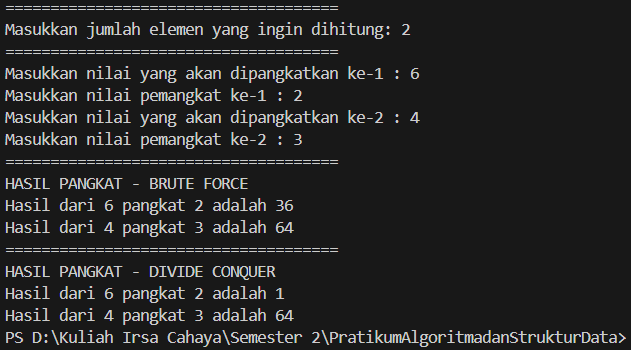
**4.3 Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide Conquer**

* Kode

package Pekan5;public class Pangkat15 { public int nilai, pangkat; public int pangkatBF(int a, int n) { int hasil = 1; for (int i = 0; i < n; i++) { hasil \*= a; } return hasil; } public int pangkatDC(int a, int n) { if (n==0) { return 1; } else { if (n % 2 == 1) { return (pangkatDC(a, n/2)\*pangkatDC(a, n/2)\*a); } else { return (pangkatDC(a, n/2)\*(pangkatDC(a, n/2))); } } }}

package Pekan5;import java.util.Scanner;public class PangkatMain15 { public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.println("====================================="); System.out.print("Masukkan jumlah elemen yang ingin dihitung: "); int elemen = sc.nextInt(); Pangkat15[] png = new Pangkat15[elemen]; System.out.println("====================================="); for (int i = 0; i < elemen; i++) { png[i] = new Pangkat15(); System.out.print("Masukkan nilai yang akan dipangkatkan ke-"+(i+1)+" : "); png[i].nilai = sc.nextInt(); System.out.print("Masukkan nilai pemangkat ke-"+(i+1)+" : "); png[i].pangkat = sc.nextInt(); } sc.close(); System.out.println("====================================="); System.out.println("HASIL PANGKAT - BRUTE FORCE"); for (int i = 0; i < elemen; i++) { System.out.println ("Hasil dari " + png[i].nilai+" pangkat " +png[i].pangkat+" adalah " +png[i].pangkatBF(png[i].nilai, png[i].pangkat)); } System.out.println("====================================="); System.out.println("HASIL PANGKAT - DIVIDE CONQUER"); for (int i = 0; i < elemen; i++) { System.out.println ("Hasil dari " + png[i].nilai+" pangkat " +png[i].pangkat+" adalah " +png[i].pangkatDC(png[i].nilai, png[i].pangkat)); } }}

* Output



**4.3.3 Pertanyaan**

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu PangkatBF() dan PangkatDC()!

**Jawab:**

* + Pada method PangkatBF() menggunakan metode iteratif dengan melakukan perulangan sebanyak n. Setiap perulangan, hasil akan dikalikan a.
  + Pada method PangkatDC() menggunakan metode rekrusif. Ketika n == 0 maka rekursi akan dihentikan. Terdapat 2 kemungkinan ketika n > 0, yaitu n bilangan ganjil atau n bilangan genap. Ketika n ganjil, n akan dibagi menjadi 2 bagian berupa n/2 dan n/2. Kemudian menghitung pangkat a dengan n/2 secara rekrusif dua kali. Kemudian hasil pangkatnya dikalikan dengan a.

Ketika n genap, sama seperti n ganjil. Tetapi, mengalikan hasil pangkat tersebut dengan a diubah menjadi mengkuadratkan hasil pangkat tersebut.

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

**Jawab:** Sudah. Berikut buktinya:

if (n % 2 == 1) { System.out.println((pangkatDC(a, n/2)\*pangkatDC(a, n/2)\*a)); return (pangkatDC(a, n/2)\*pangkatDC(a, n/2)\*a); } else { return (pangkatDC(a, n/2)\*(pangkatDC(a, n/2))); }

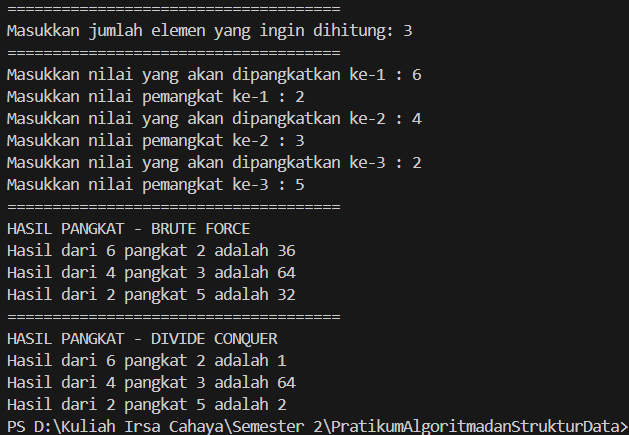
1. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.
   * Kode

package Pekan5;import java.util.Scanner;public class PangkatMain15 { public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.println("====================================="); System.out.print("Masukkan jumlah elemen yang ingin dihitung: "); int elemen = sc.nextInt(); Pangkat15[] png = new Pangkat15[elemen]; System.out.println("====================================="); for (int i = 0; i < elemen; i++) { System.out.print("Masukkan nilai yang akan dipangkatkan ke-"+(i+1)+" : "); int nilai = sc.nextInt(); System.out.print("Masukkan nilai pemangkat ke-"+(i+1)+" : "); int angkat = sc.nextInt(); png[i] = new Pangkat15(nilai, angkat); } sc.close();

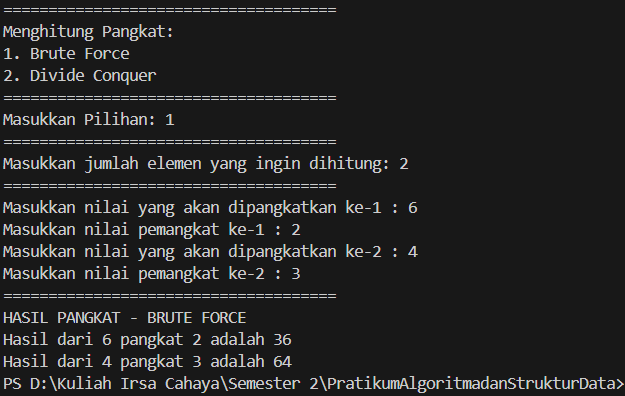
package Pekan5;public class Pangkat15 { public int nilai; public int pangkat; public Pangkat15(int nilai, int pangkat) { this.nilai = nilai; this.pangkat = pangkat; } public int pangkatBF(int a, int n) { int hasil = 1; for (int i = 0; i < n; i++) { hasil \*= nilai; } return hasil; } public int pangkatDC(int a, int n) { if (n==0) { return 1; } else { if (n % 2 == 1) { return (pangkatDC(a, n/2)\*pangkatDC(a, n/2)\*a); } else { return (pangkatDC(a, n/2)/pangkatDC(a, n/2)); } } }}

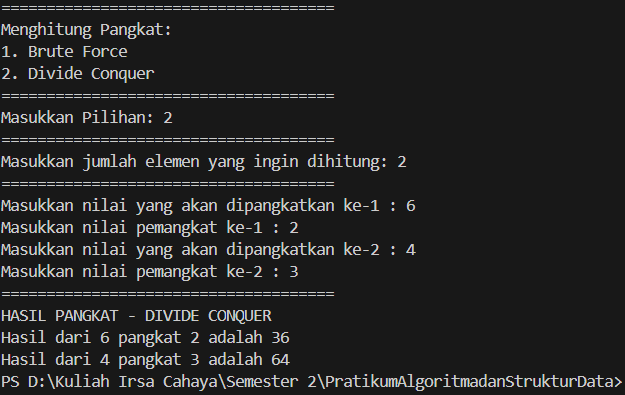
System.out.println("====================================="); System.out.println("HASIL PANGKAT - BRUTE FORCE"); for (int i = 0; i < elemen; i++) { System.out.println ("Hasil dari " + png[i].nilai+" pangkat " +png[i].pangkat+" adalah " +png[i].pangkatBF(png[i].nilai, png[i].pangkat)); } System.out.println("====================================="); System.out.println("HASIL PANGKAT - DIVIDE CONQUER"); for (int i = 0; i < elemen; i++) { System.out.println ("Hasil dari " + png[i].nilai+" pangkat " +png[i].pangkat+" adalah " +png[i].pangkatDC(png[i].nilai, png[i].pangkat)); } }}

* + Output



1. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan switch-case!





* 1. **Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer**
  + Kode