**Laporan Praktikum ASD**

**Jobsheet 05**

**Nama: Gegas Anugrah Derajat**

**Kelas: SIB-1F**

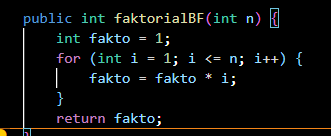
**Absen: 11**

Percobaan 4.2

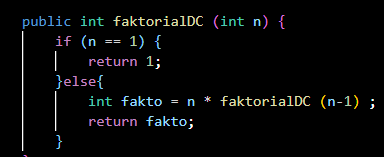
Membuat class dan atribut



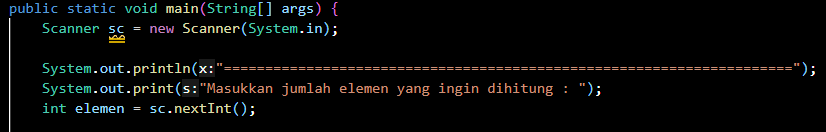
Menambahkan method



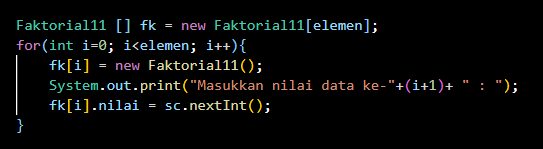
Menambahkan method



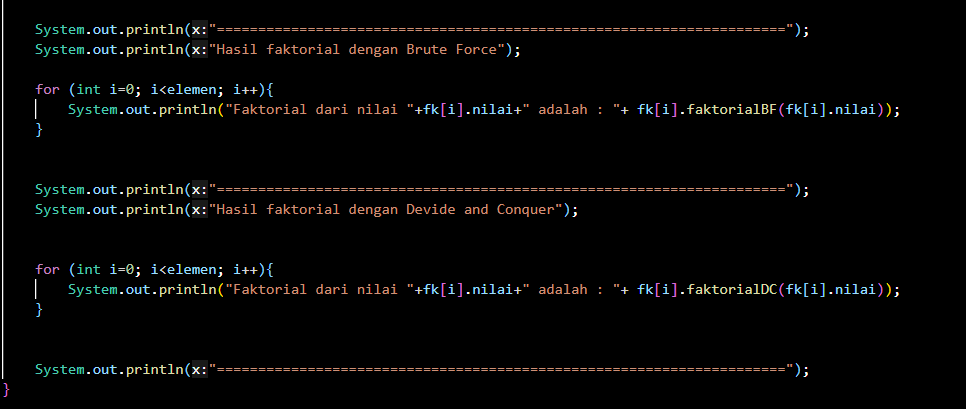
pada fungsi main ditambahkan input nilai



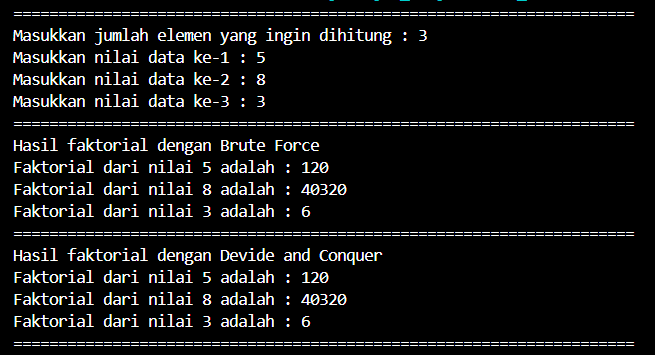
membuat array of objek

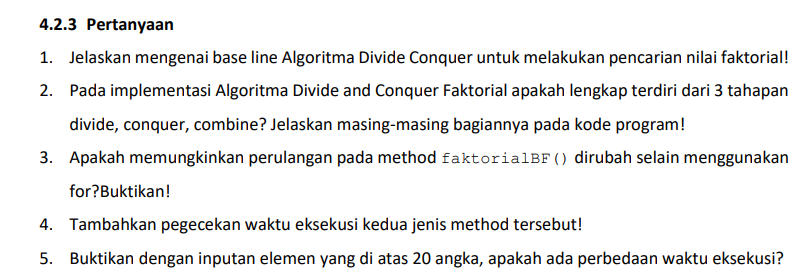


menampilkan hasil



hasil





**Jawaban:**

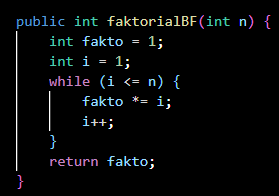
1. **Nilai pertama yang dikirimkan akan di cek apakah sama dengan 1 atau tidak, jika nilai sama dengan 1 maka nilai akan dikembalikan dengan value 1. Jika tidak sama dengan satu maka akan dikalikan dengan -1**
2. **divide**

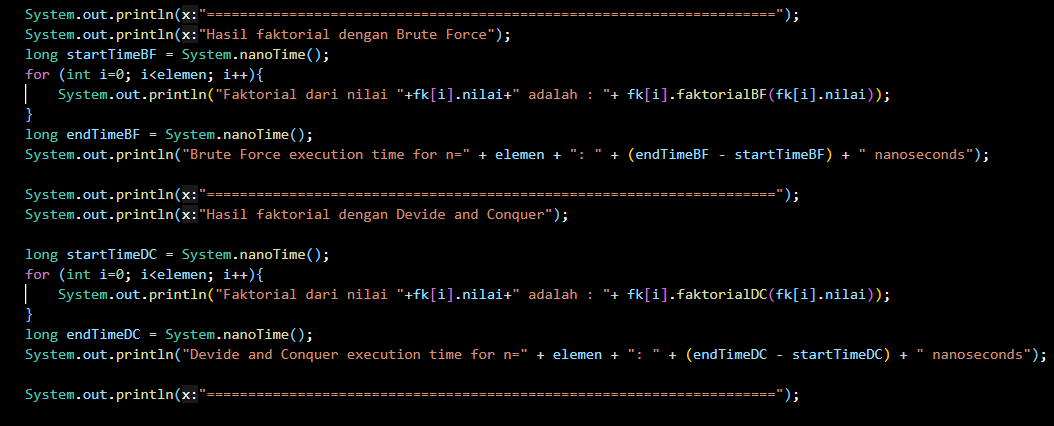


**conquer**

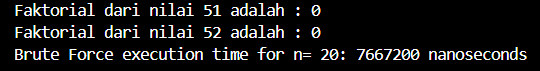


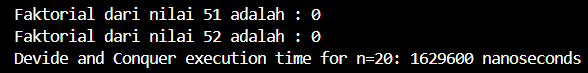
1. **menggunakan while**





1. **divide conquer lebih cepat dibandingkan dengan brute force**



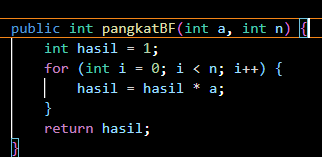


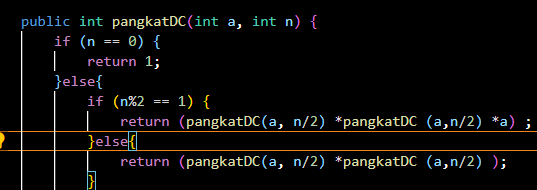
Percobaan 4.3

membuat class dan atribut

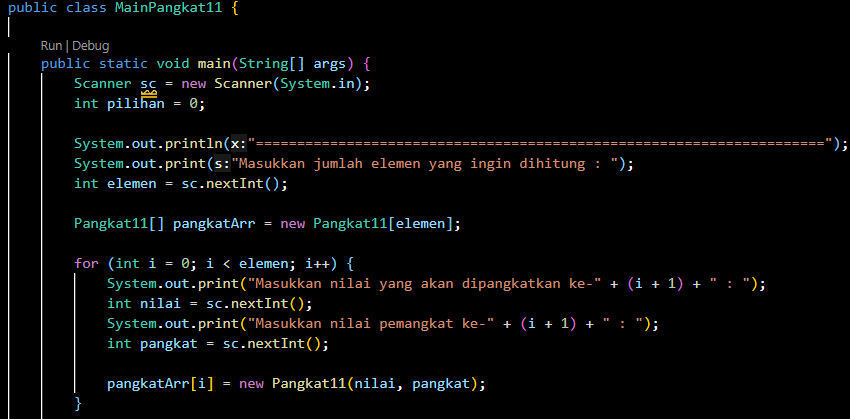


menambahkan method BF dan DC

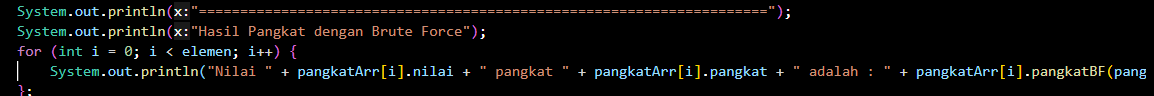


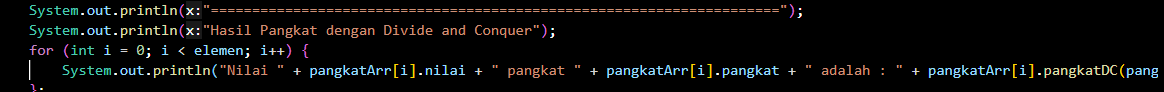


membuat class main

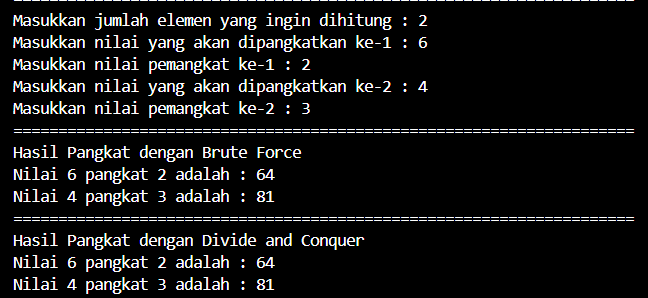


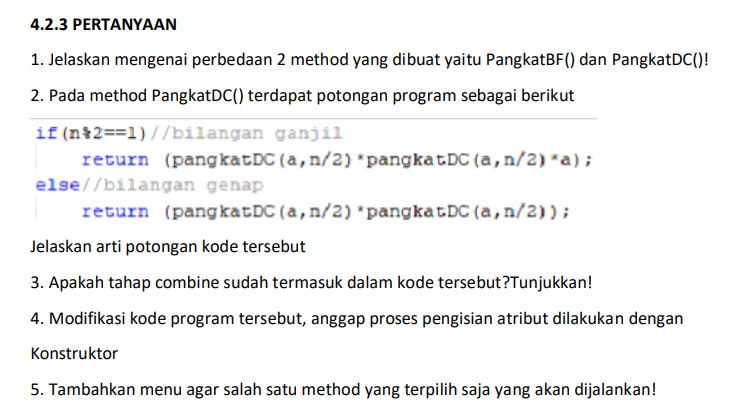
menambahkan pemanggilan method





hasil

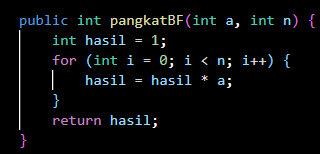


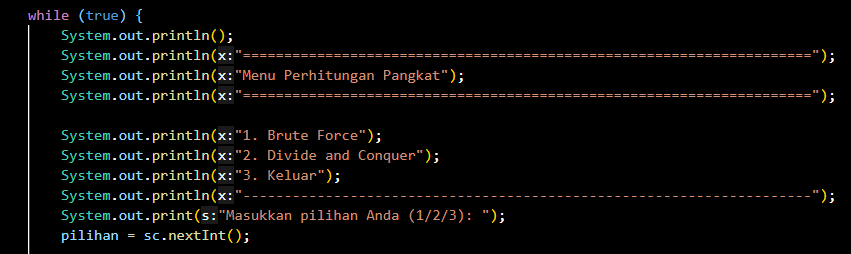


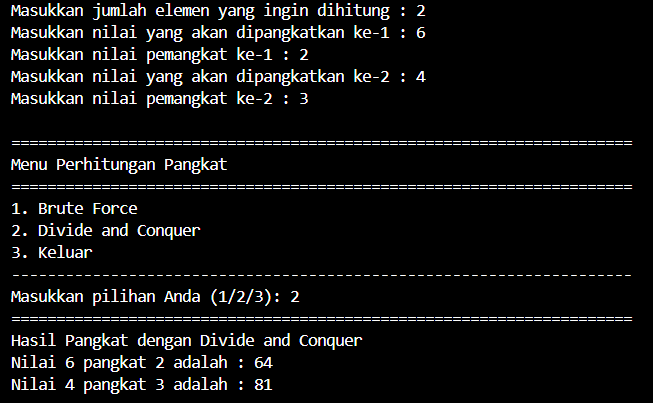
**Jawaban:**

1. **Method brute force melakukan perulangan secara runtut dan setiap iterasinya akan mengalikan hasil bilangan dengan pangkat. Method divide conquer menggunakan metode rekursif, dengan melakukan cek apakah pangkat bernilai 0 atau tidak dan mengecek juga apakah bilangan ganjil atau tidak**
2. **Pada tahap awal akan di cek apakah bilangan ganjil atau tidak, jika ganjil maka fungsi akan menghitung pangkat dan dikalikan imbuhan diakhir akan dilakikan dengan nilai. jika genap maka akan fungsi menghitung pangkat dikalikan dengan fungsi penghitung pangkat**



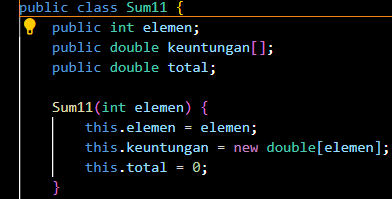




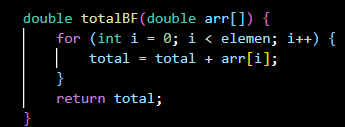


Percobaan 4.4

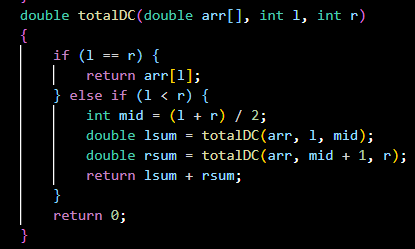
membuat class, atribut, dan konstruktor



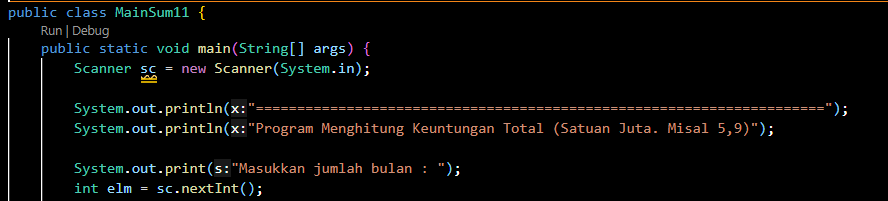
menambahkan method



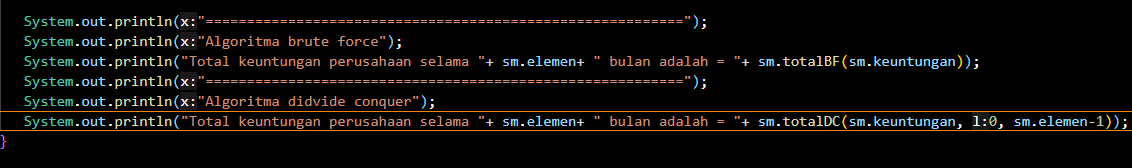
menambahkan method



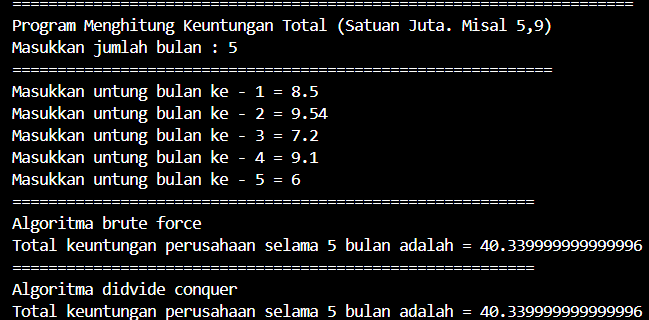
membuat class main

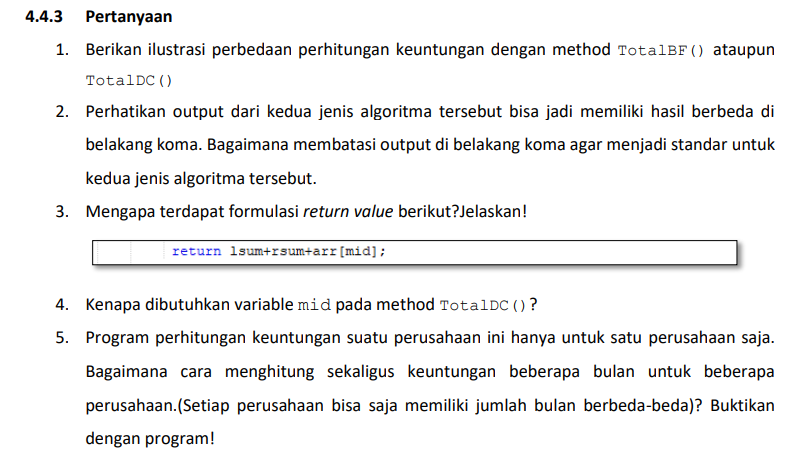


menampilkan hasil



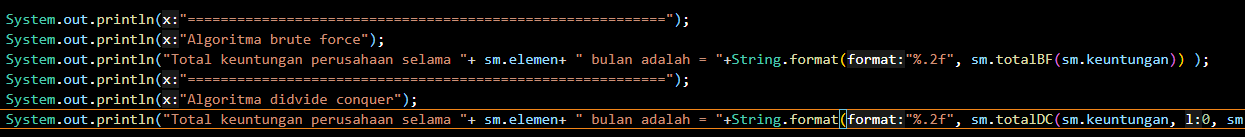
hasil



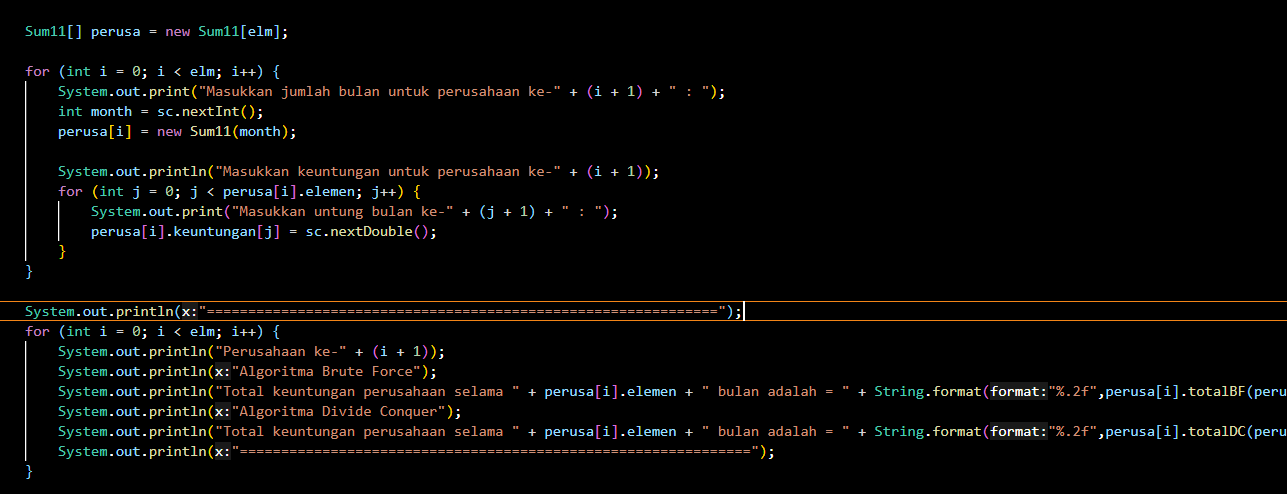


**Jawaban:**

1. **totalBF menggunakan perulangan dan di setiap literasi akan menambahkan total dari tiap array key ke var total, sehingga setelah perulangan var total berisi total dari value array. totalDC memanfaatkan fungsi rekursif, dengan terlebih dahulu melakukan cek apakah var l sama dengan var r, jika sama maka akan mengembalikan value array key yang sama dengan var l. Kemudian jika tidak maka akan melakukan pengecekan lagi apakah l < r, jika tidak maka akan mengembalikan 0, jika akan melakukan kalkulasi rekursif hingga selesai, setelah itu akan mengembalikan hasil dari kalkulasi.**



1. **Menghitung jumlah dari total elemen yang dihitung**
2. **mid digunakan untuk menampung nilai tengah dan menentukan nilai batas rekursif**



Latihan 4.5

Buatlah kode program untuk menghitung nilai akar dari suatu bilangan dengan algoritma Brute Force dan Divide Conquer! Jika bilangan tersebut bukan merupakan kuadrat sempurna, bulatkan angka ke bawah

