  
Name : Azaria Cindy Sahasika

Number Id : 2341760169 / 06

Class : 1G – Business Information System

Lesson : Algorithm and Data Structure

Material : Jobsheet 4

Github Link : <https://github.com/azariacindy/algorithm-ds>

JOBSHEET IV  
BRUTE FORCE DAN DIVIDE CONQUER

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

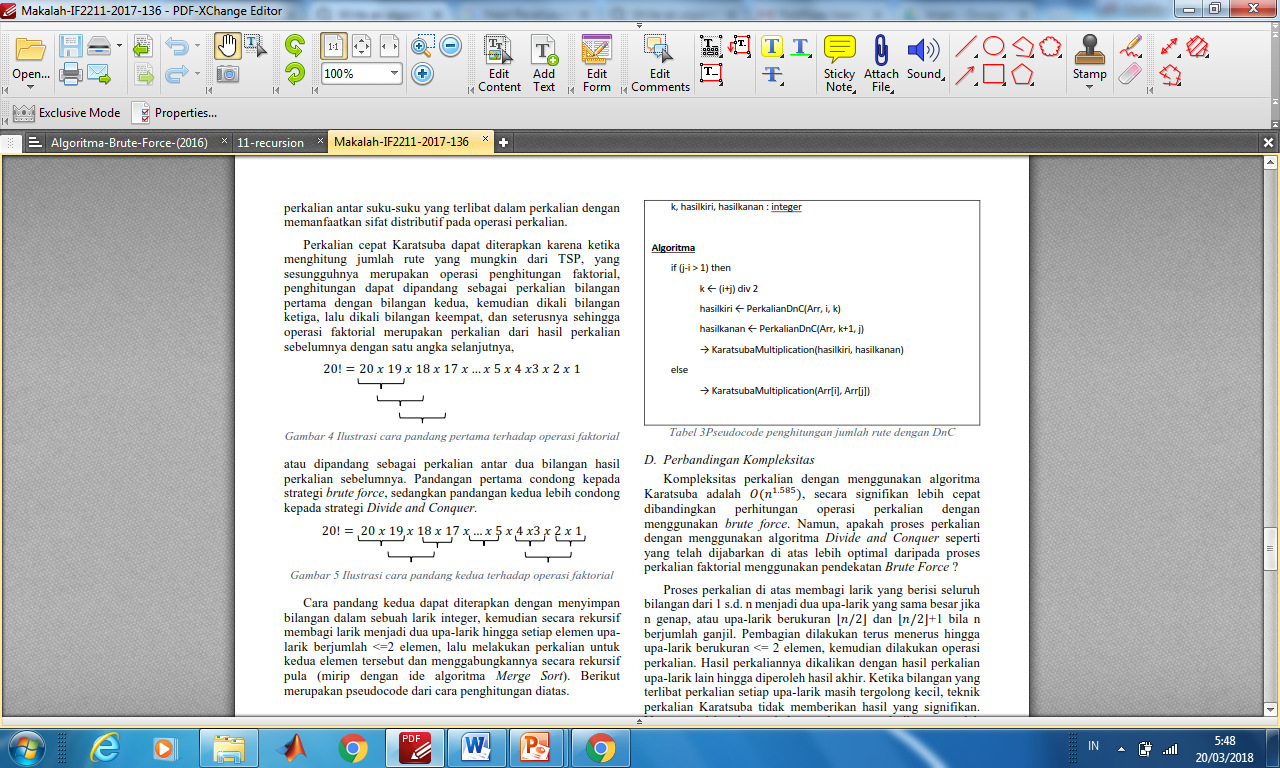
1. Mahasiswa mampu membuat algoritma bruteforce dan divide-conquer
2. Mahasiswa mampu menerapkan penggunaan algorima bruteforce dan divide-conquer
3. Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

Perhatikan Diagram Class berikut ini :

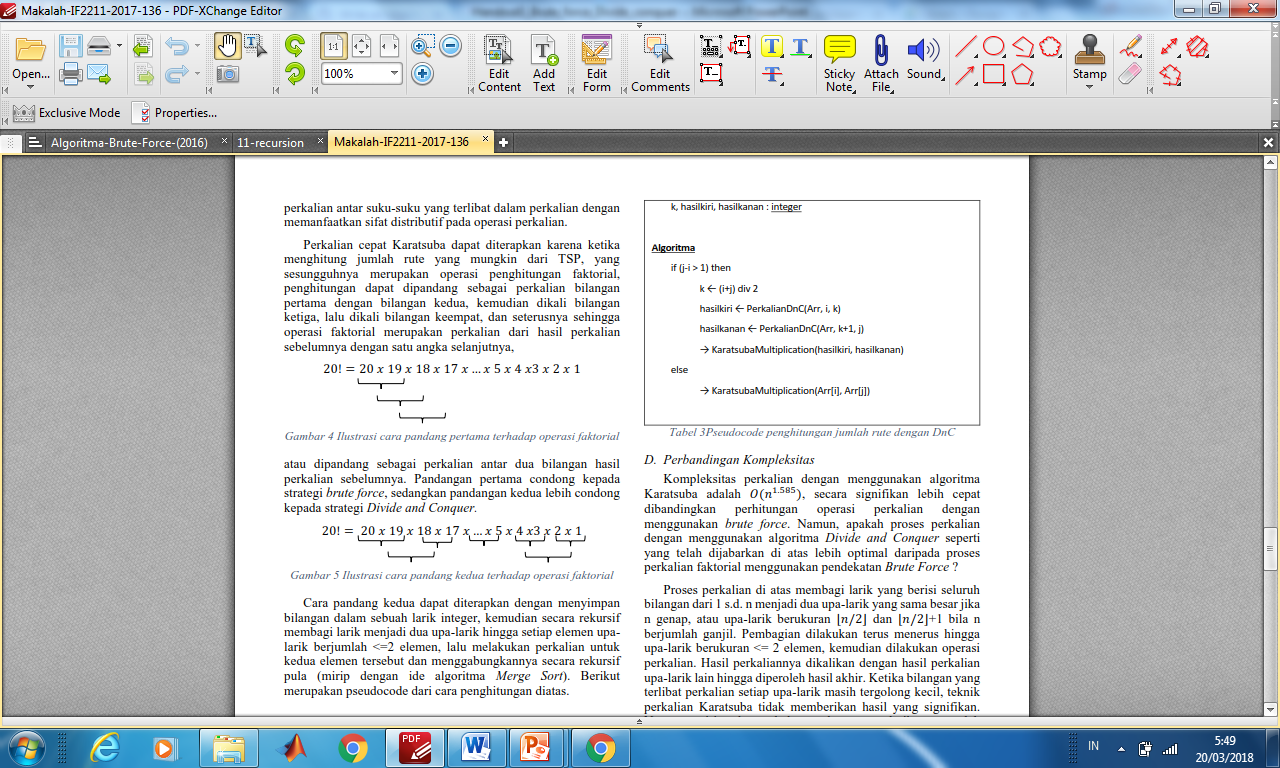
|  |
| --- |
| Faktorial |
| nilai: int |
| faktorialBF(): int  faktorialDC(): int |

Berdasarkan diagram class di atas, akan dibuat program class dalam Java. Untuk menghitung nilai faktorial suatu angka menggunakan 2 jenis algoritma, Brute Force dan Divide and Conquer. Jika digambarkan terdapat perbedaan proses perhitungan 2 jenis algoritma tersebut sebagai berikut :

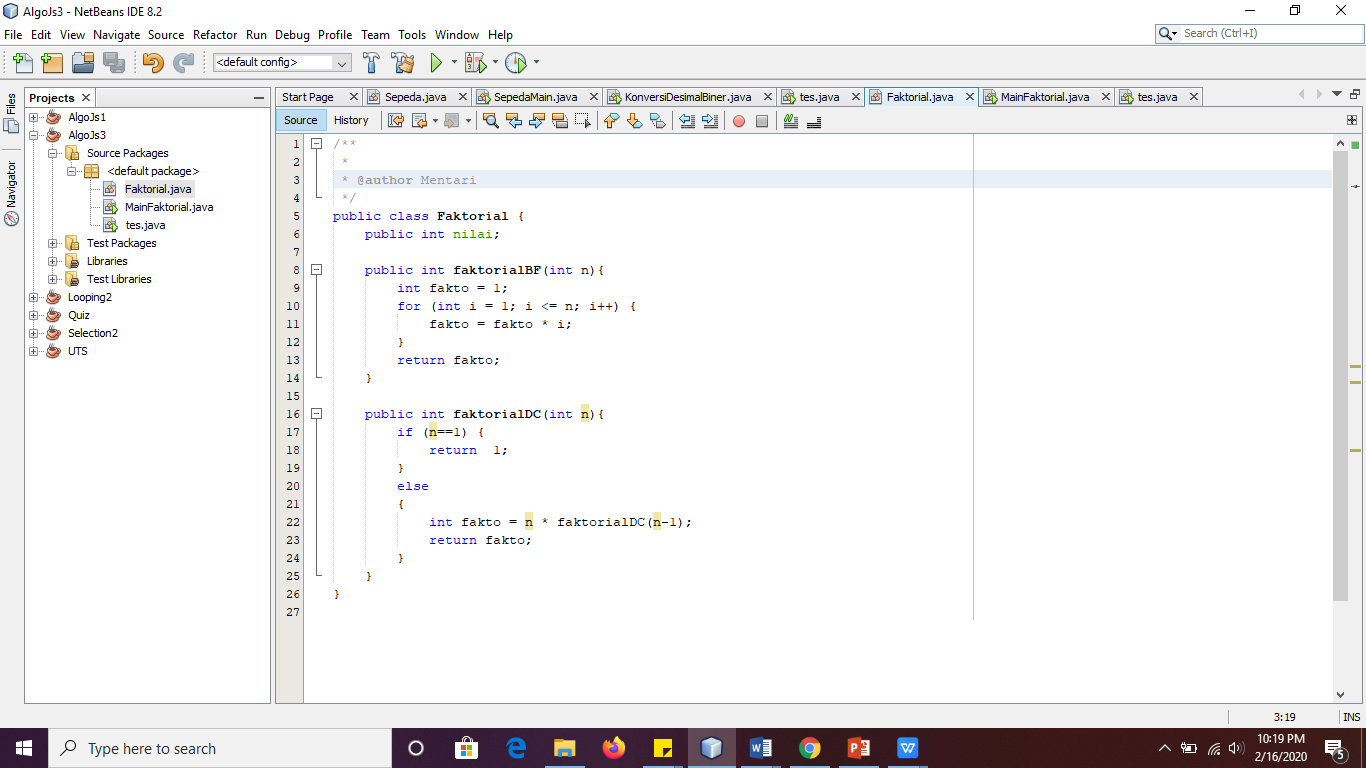
Tahapan pencarian nilai faktorial dengan algoritma Brute Force :



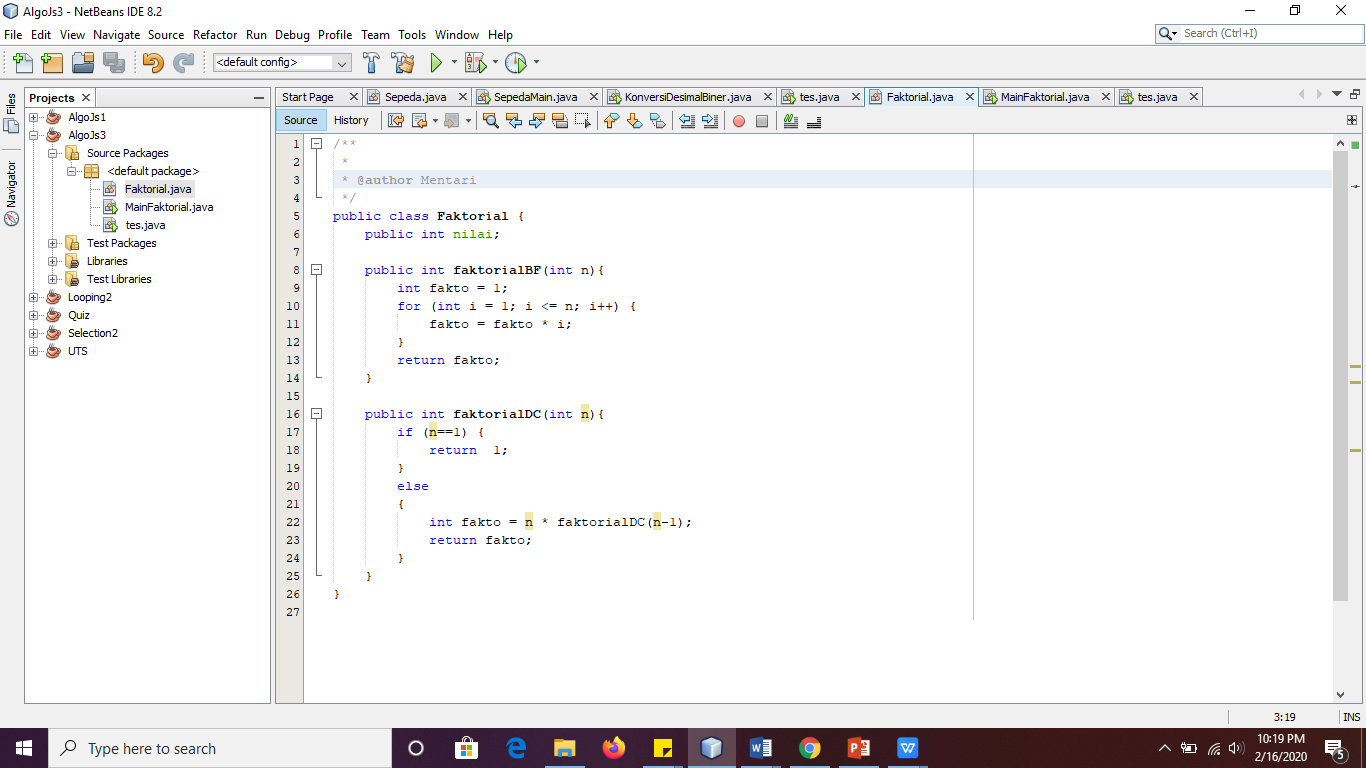
Tahapan pencarian nilai faktorial dengan algoritma Divide and Conquer :



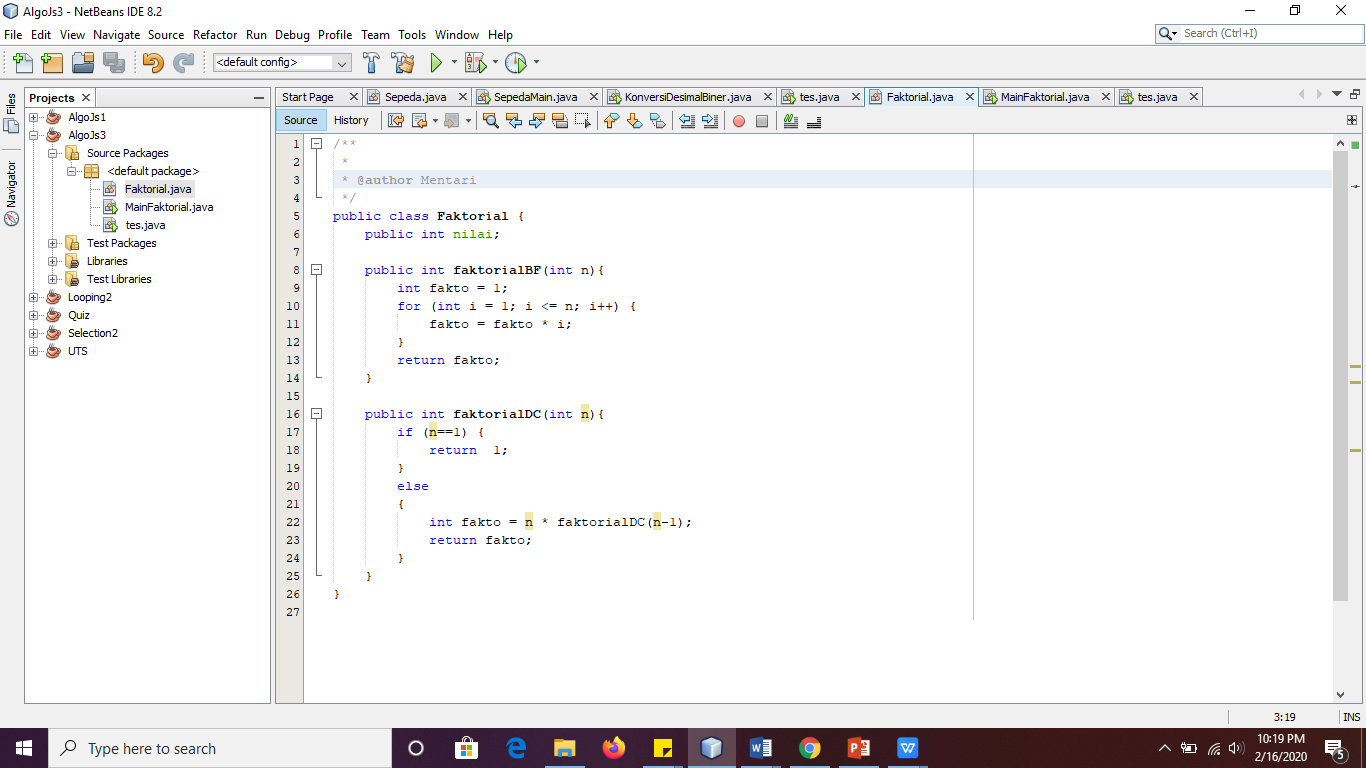
1. Langkah-langkah Percobaan
2. Buat Project baru, dengannama “**BruteForceDivideConquer**”. Buat package dengan nama minggu5.
3. Buatlah class baru dengan nama **Faktorial**
4. Lengkapi class **Faktorial** dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:
5. Tambahkan atribut nilai



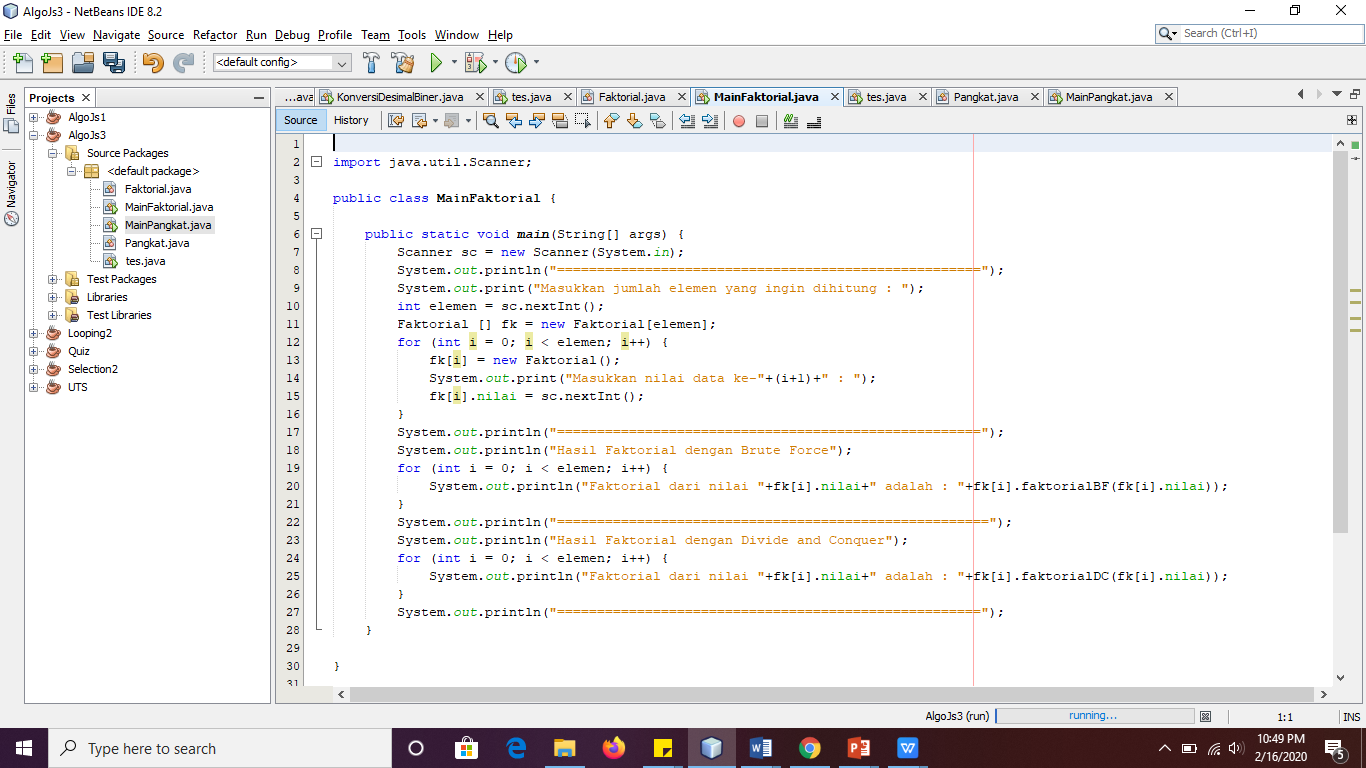
1. Tambahkan method faktorialBF() nilai



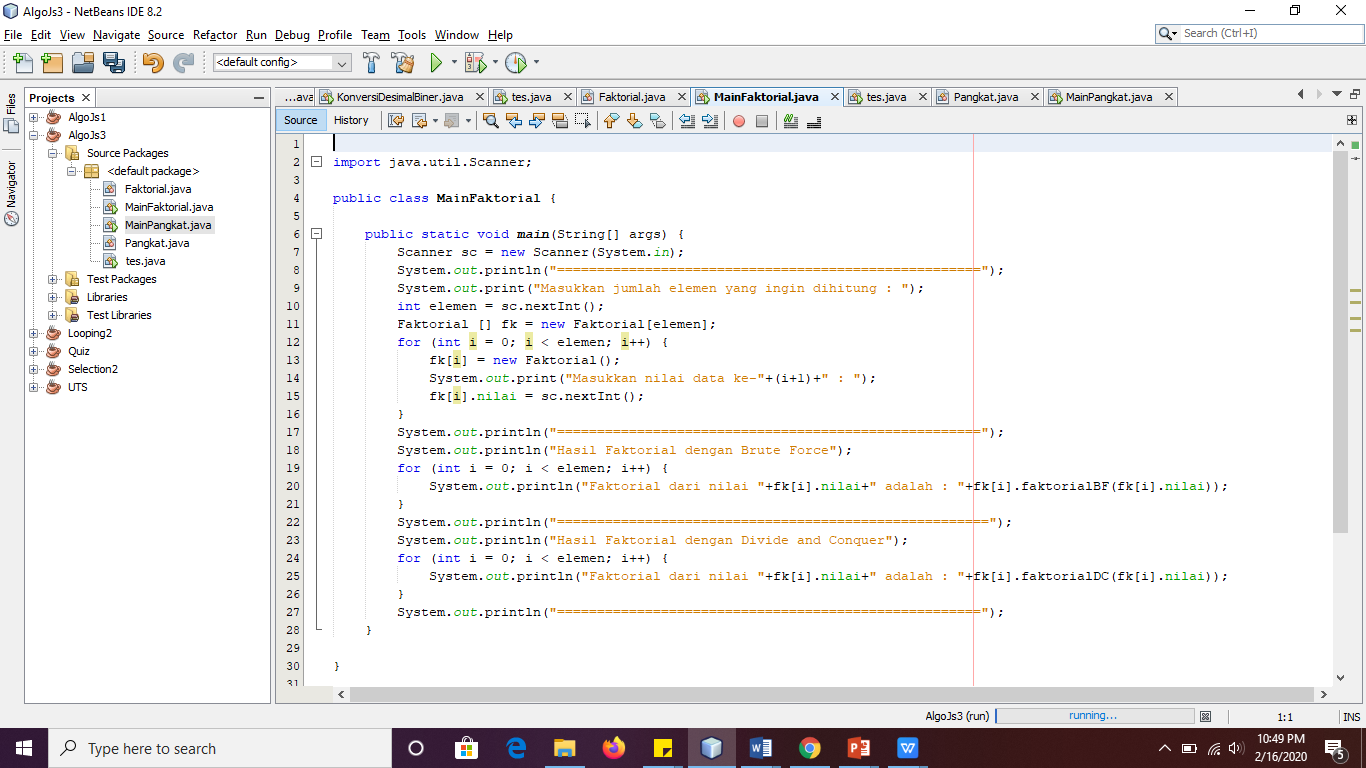
1. Tambahkan method faktorialDC() nilai



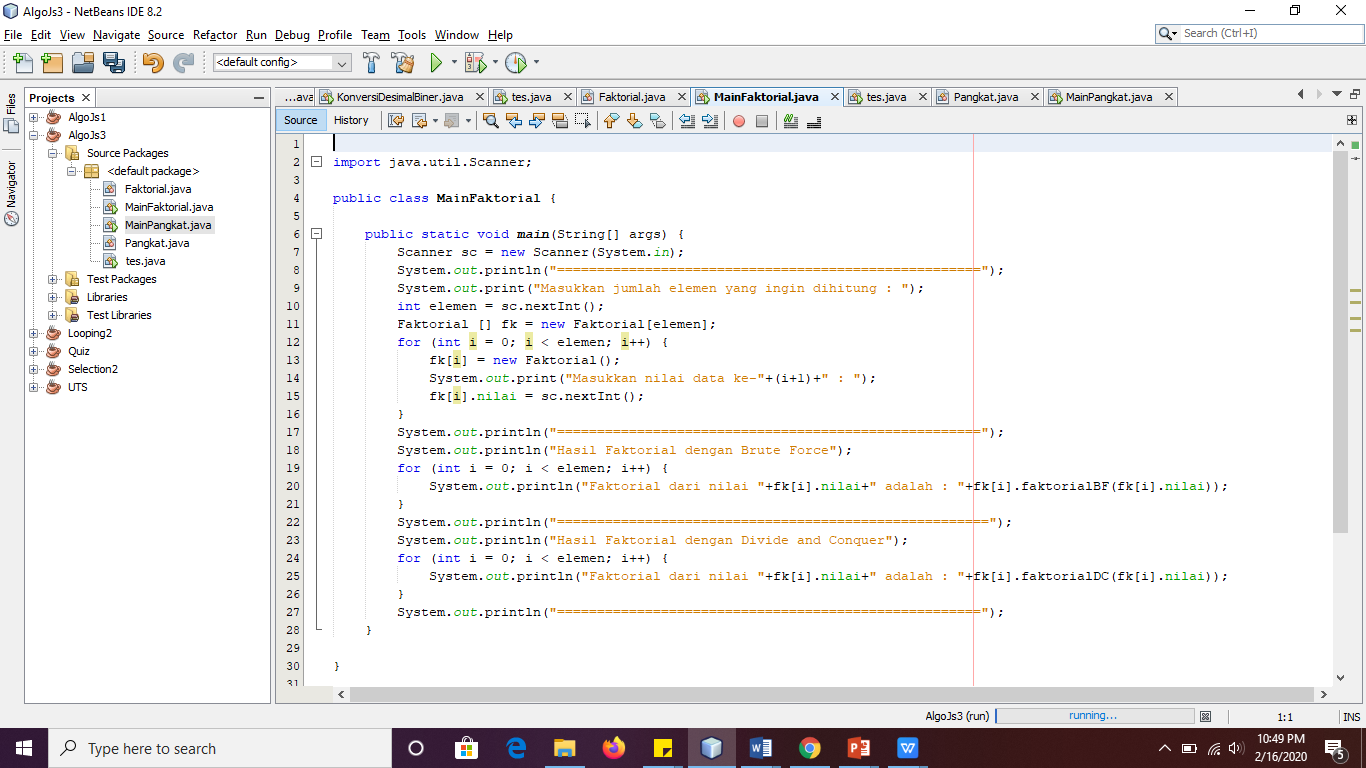
1. Coba jalankan (Run) class Faktorial dengan membuat class baru MainFaktorial.
2. Di dalam fungsi main sediakan komunikasi dengan user untuk menginputkan jumlah angka yang akan dicari nilai faktorialnya



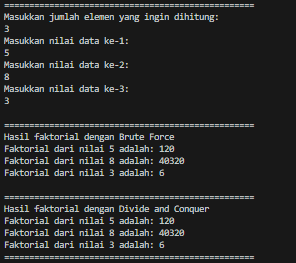
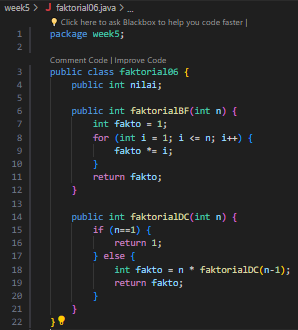
* 1. Buat Array of Objek pada fungsi main, kemudian inputkan beberapa nilai yang akan dihitung faktorialnya

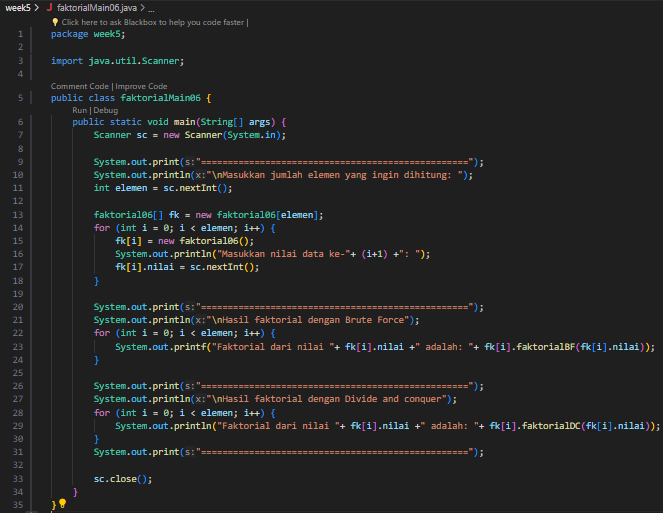


1. Tampilkan hasil pemanggilan method faktorialDC() dan faktorialBF()



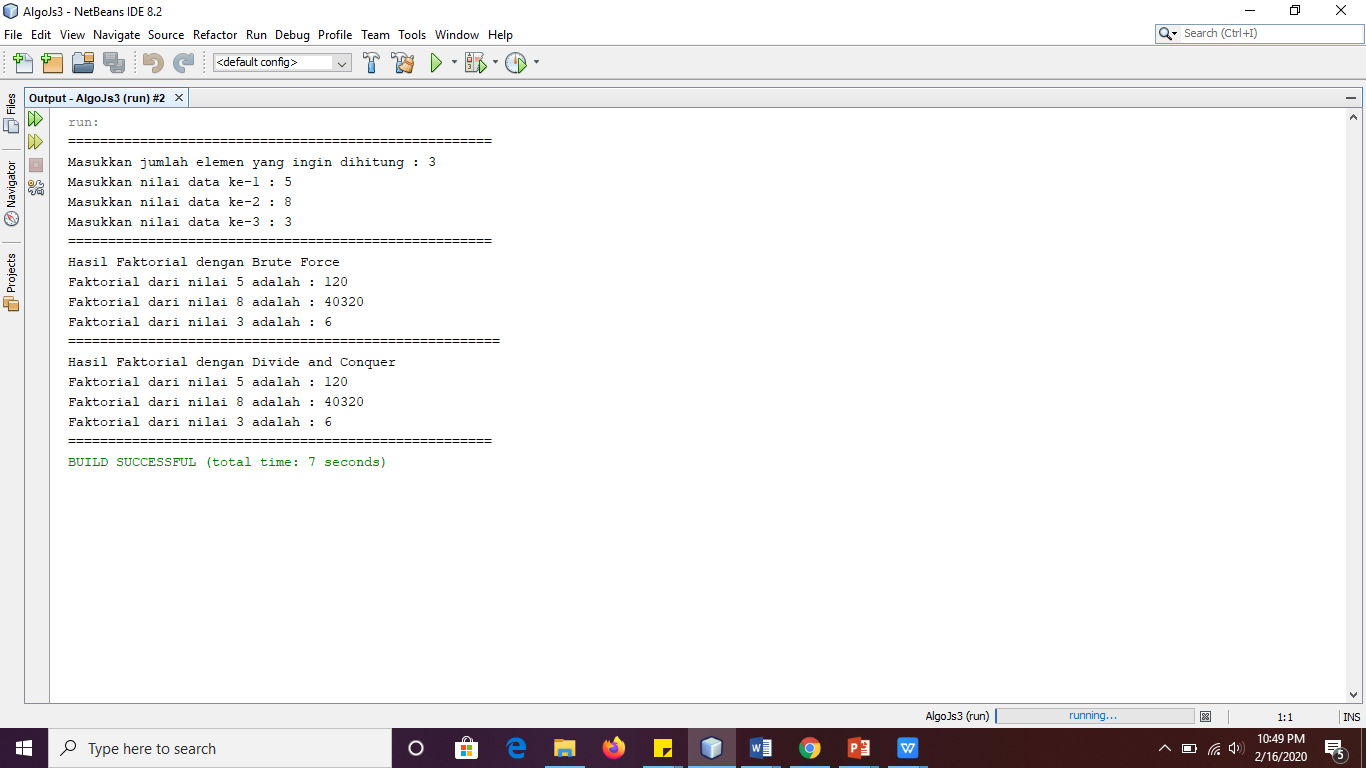
* 1. Pastikan program sudah berjalan dengan baik!





1. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



1. Pertanyaa
2. Jelaskan mengenai base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial!

* The baseline in the Divide and Conquer algorithm for calculating factorials is the condition where the problem becomes simple enough to solve directly. Here, the baseline is reached when the value of n (input) is equal to 1. Then, the value of the factorial is 1.

1. Pada implementasi Algoritma Divide and Conquer Faktorial apakah lengkap terdiri dari 3 tahapan divide, conquer, combine? Jelaskan masing-masing bagiannya pada kode program!

* Yes, the implementation of the Factorial Divide and Conquer algorithm in the complete program code consists of 3 stages:

1. Divide (pembagian)

Division is performed on the initial condition (n > 1). The program divides the problem into two smaller subproblems, which is to calculate the factorial n-1 and multiply it by n.

1. Conquer (penaklukan)

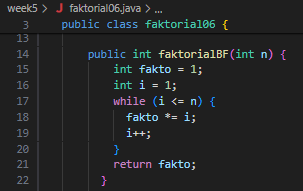
The conquest is done in a recursive manner. The program calls back the factorial functionDC(n-1) to calculate the value of factorial n-1.

1. Combine (penggabungan)

The combine is done by multiplying the result of subproblem n \* factorialDC(n-1). The result of this multiplication is the value of factorial n.

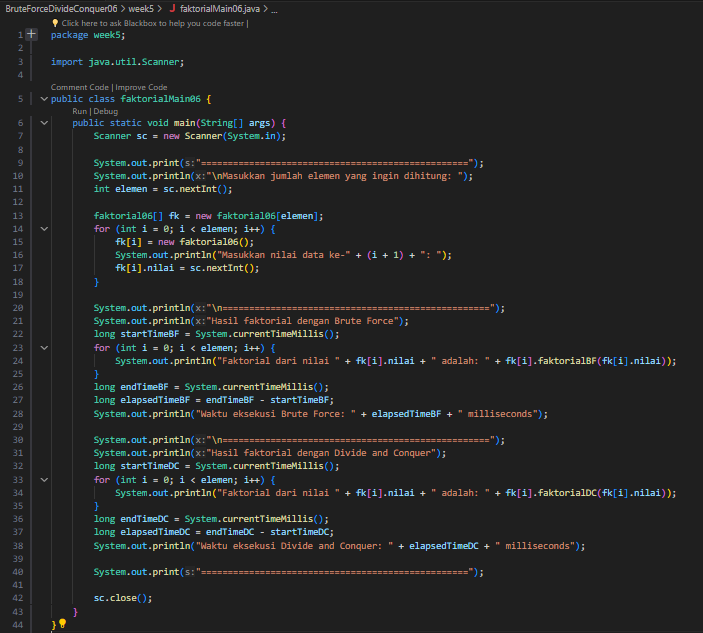
1. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for?Buktikan!

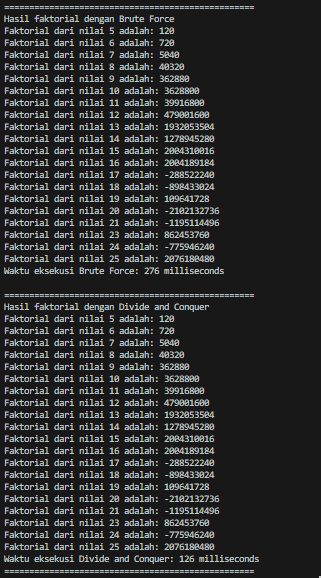
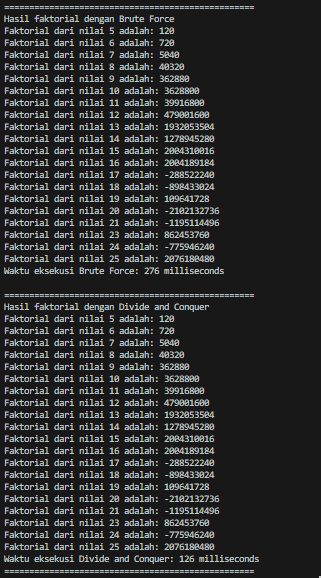
* use while loop:



1. Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!

* Execution time of the Divide and Conquer method is much faster than the Brute Force method.

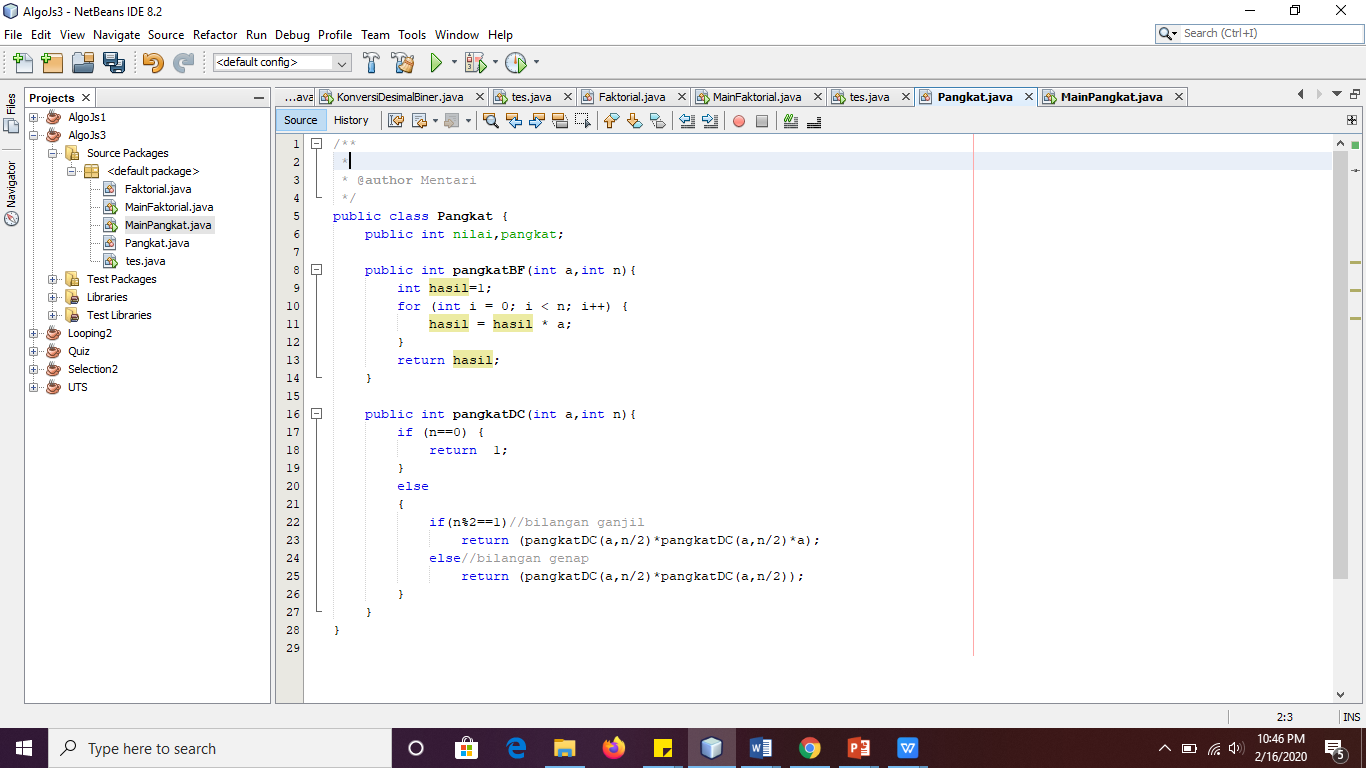


1. Buktikan dengan inputan elemen yang di atas 20 angka, apakah ada perbedaan waktu eksekusi?
2. Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

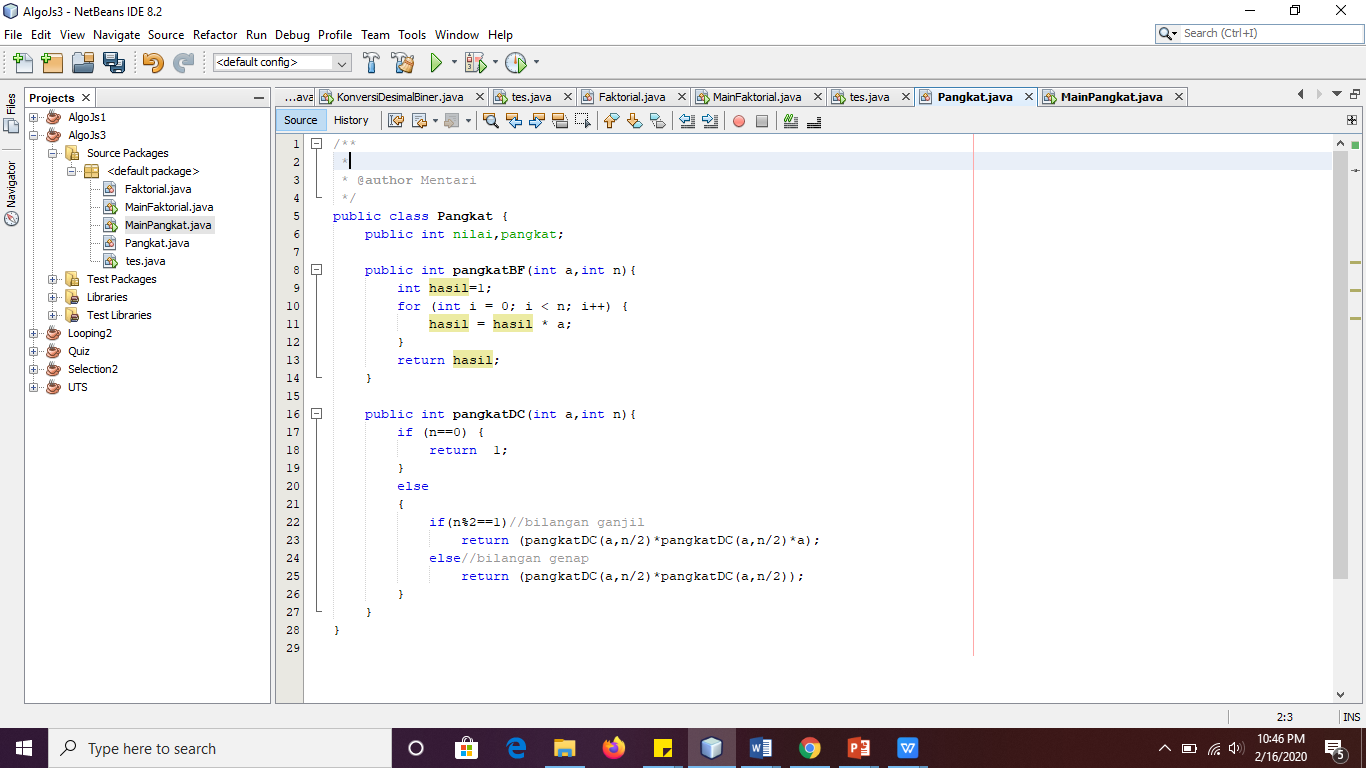
Pada praktikum ini kita akan membuat program class dalam Java. Untuk menghitung nilai pangkat suatu angka menggunakan 2 jenis algoritma, Brute Force dan Divide and Conquer.

* + 1. Langkah-langkah Percobaan

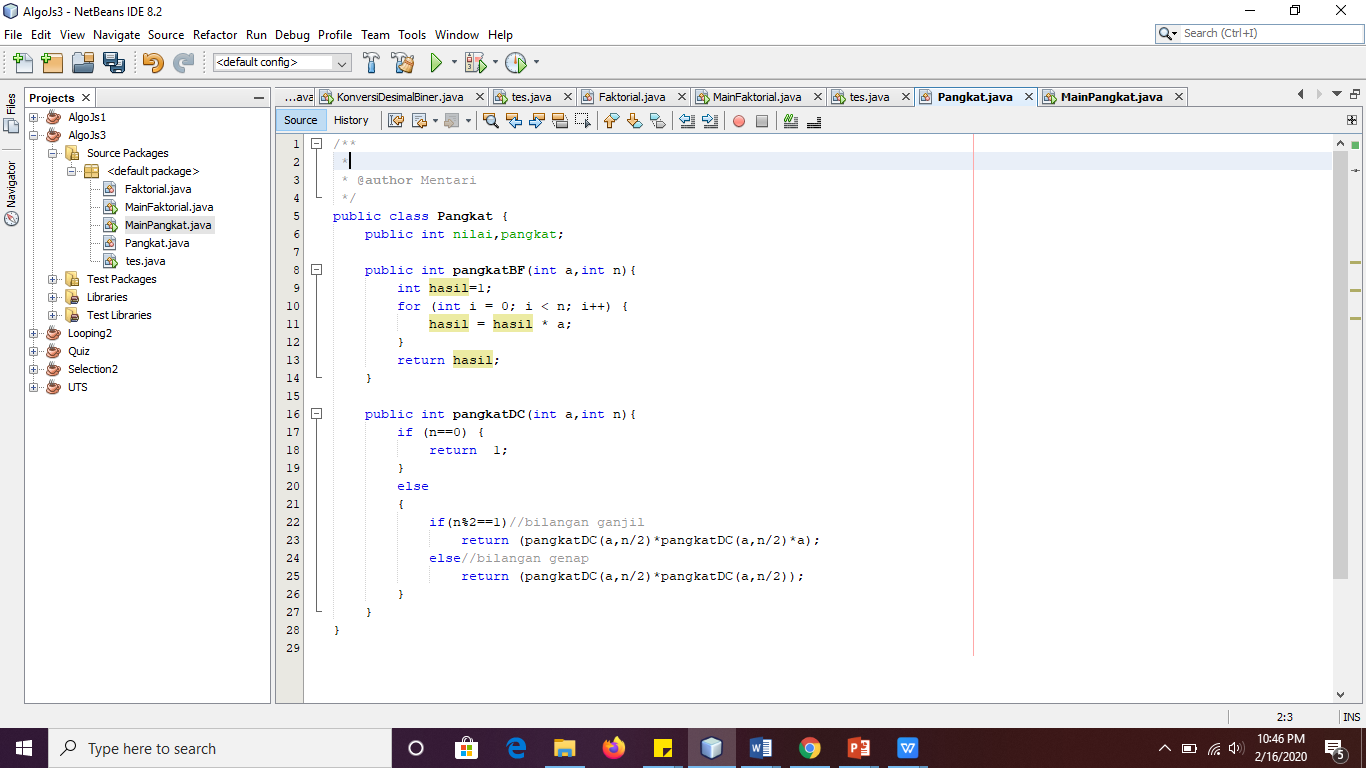
1. Di dalam paket **minggu5**, buatlah class baru dengan nama **Pangkat**. Dan di dalam class **Pangkat** tersebut, buat atribut angka yang akan dipangkatkan sekaligus dengan angka pemangkatnya



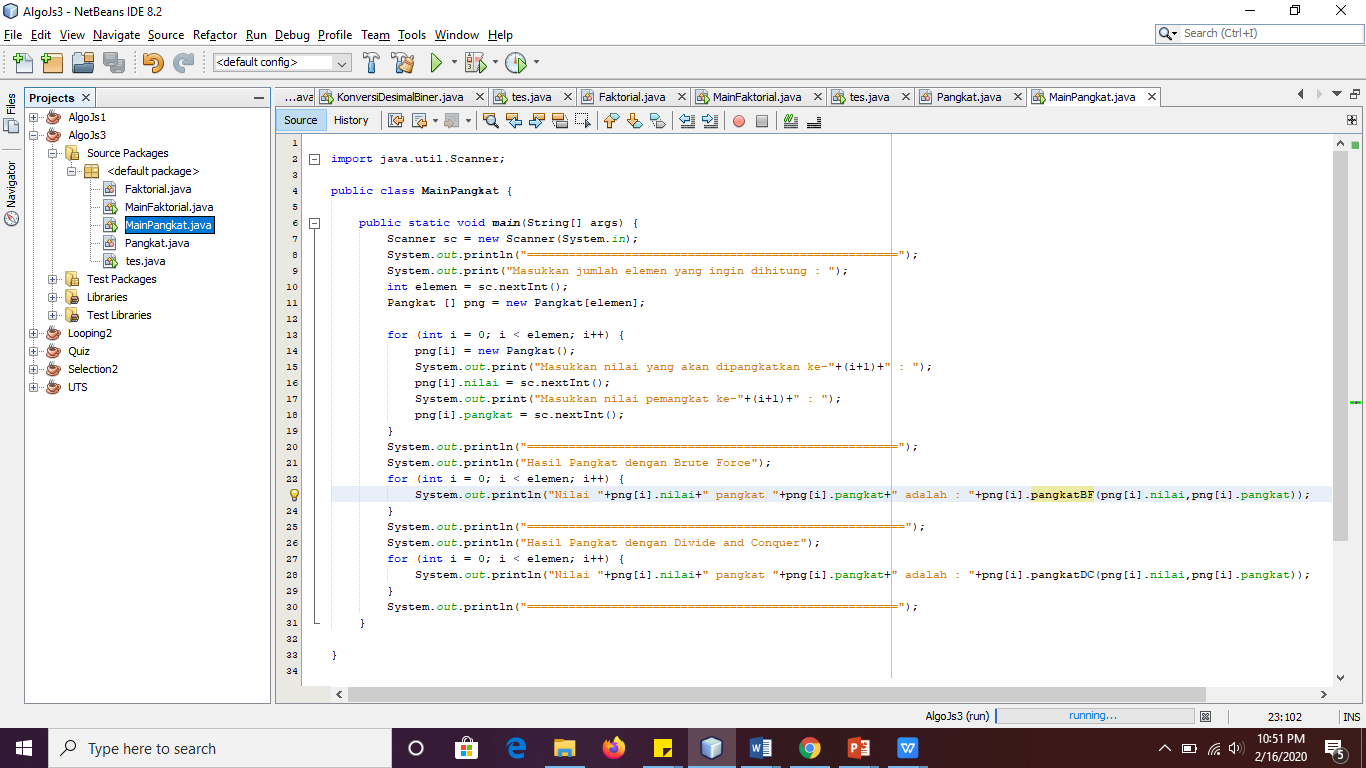
1. Pada class Pangkat tersebut, tambahkan method PangkatBF()



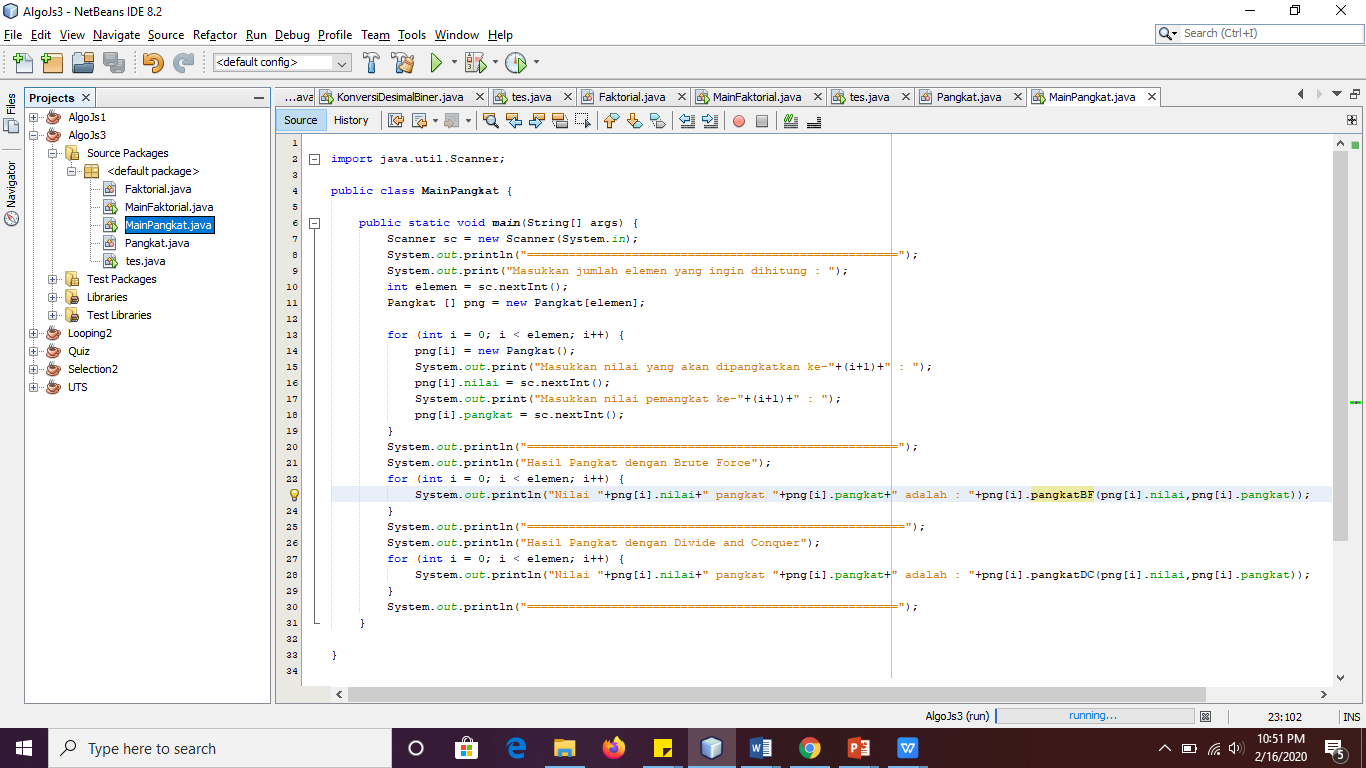
1. Pada class Pangkat juga tambahkan method PangkatDC()



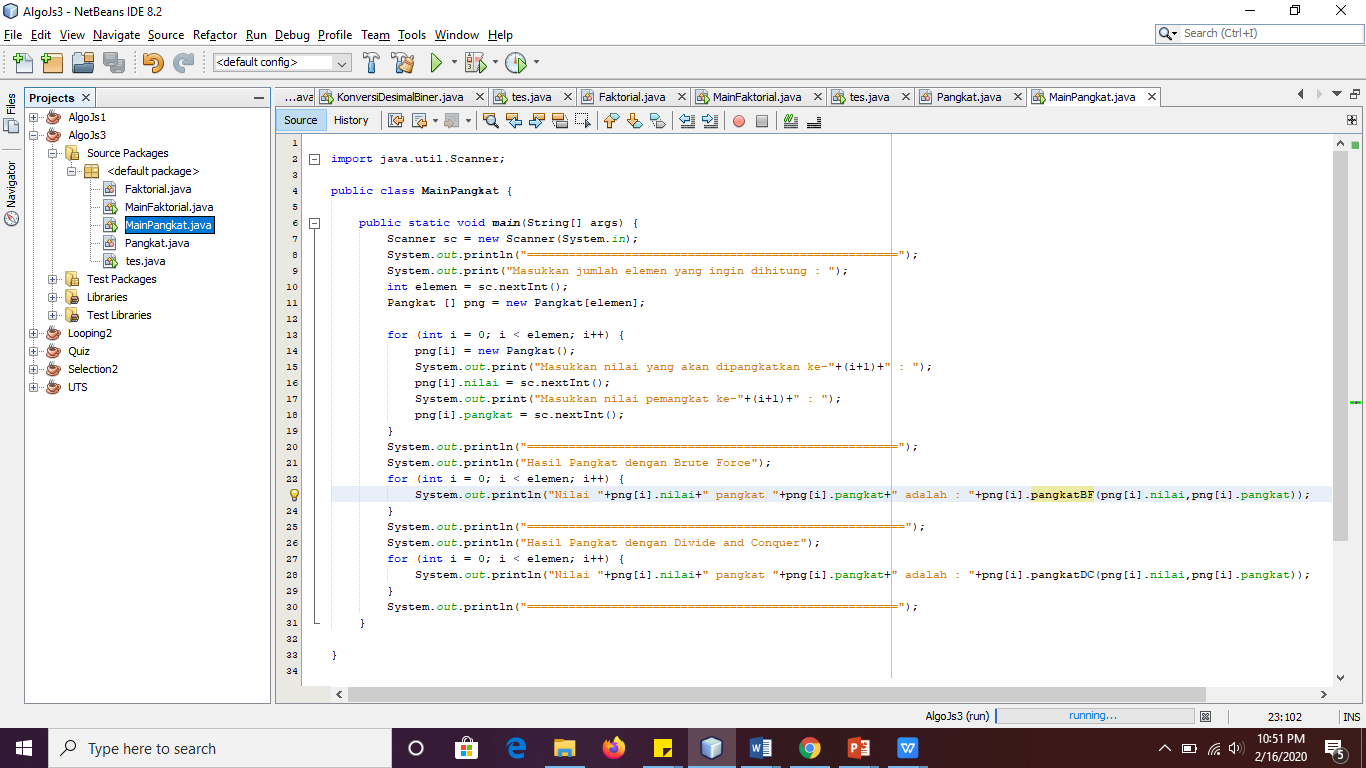
1. Perhatikan apakah sudah tidak ada kesalahan yang muncul dalam pembuatan class Pangkat
2. Selanjutnya buat class baru yang di dalamnya terdapat method main. Class tersebut dapat dinamakan MainPangkat. Tambahkan kode pada class main untuk menginputkan jumlah nilai yang akan dihitung pangkatnya.



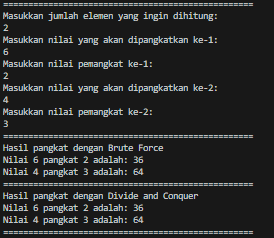
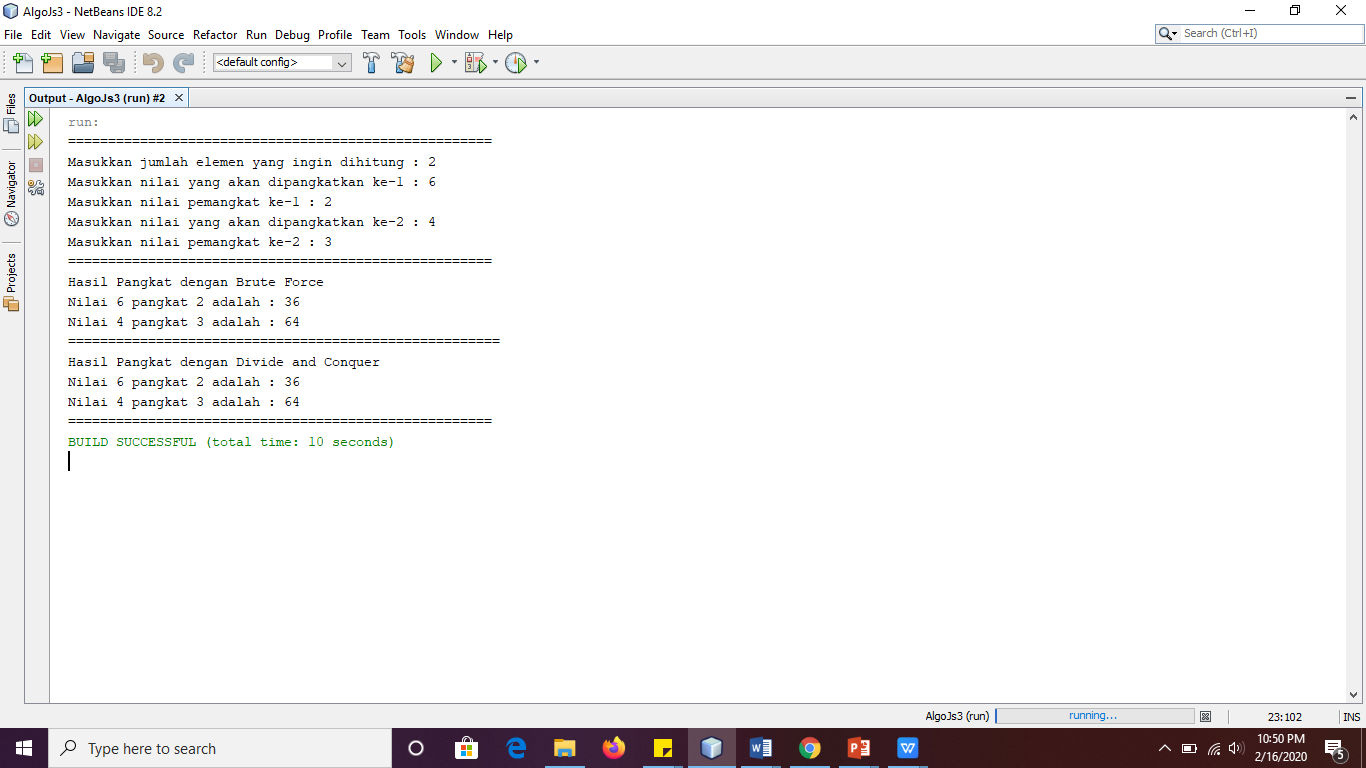
1. Nilai pada tahap 5 selanjutnya digunakan untuk instansiasi array of objek. Di dalam Kode berikut ditambahkan proses pengisian beberapa nilai yang akan dipangkatkan sekaligus dengan pemangkatnya.

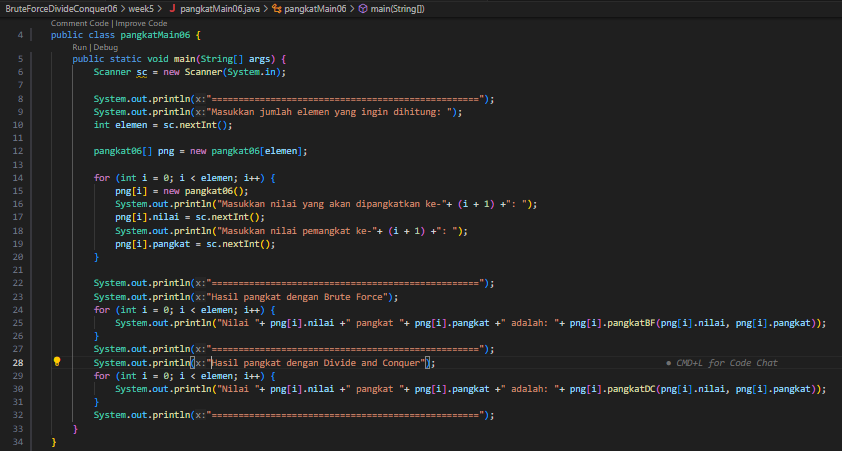


1. Kemudian, panggil hasil nya dengan mengeluarkan return value dari method PangkatBF() dan PangkatDC().



* + 1. Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini. 

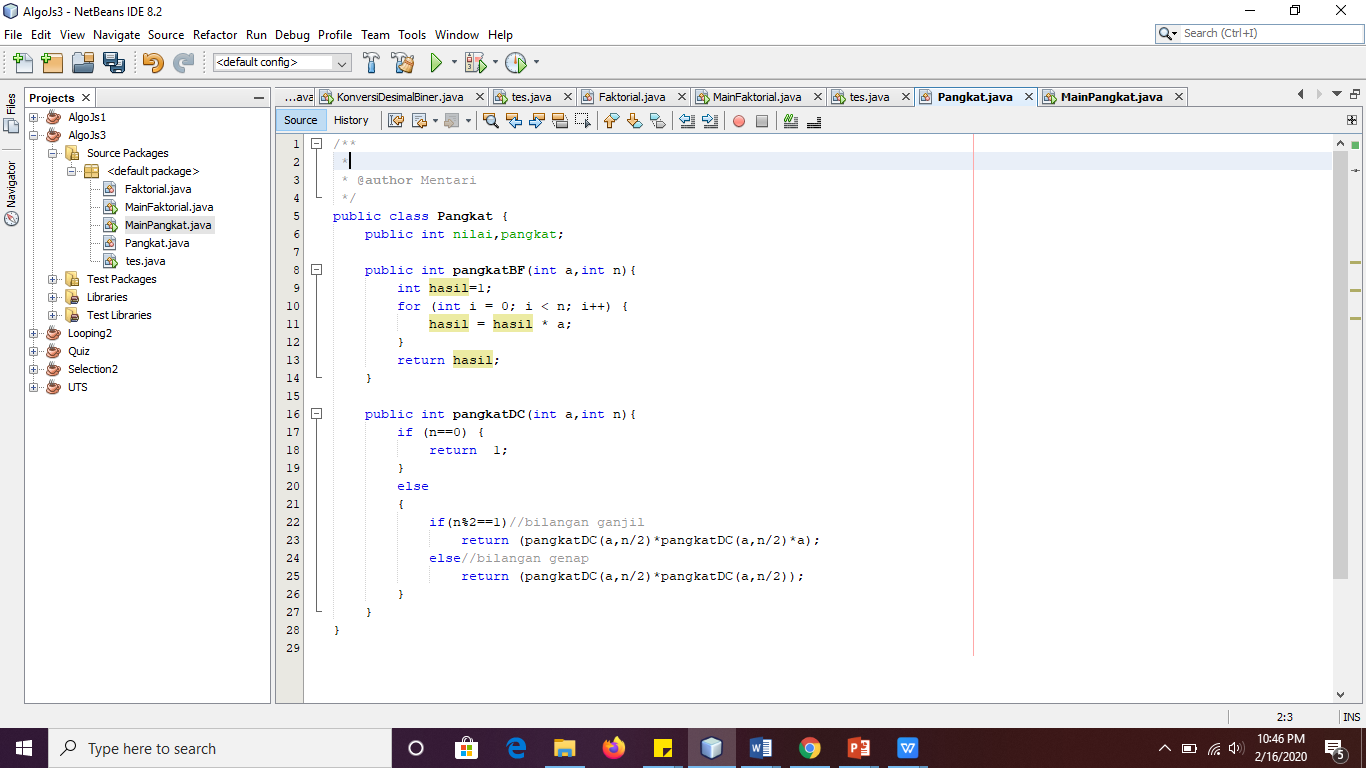


* + 1. Pertanyaan

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

* PangkatBF(): uses the Brute Force method to calculate the power result by multiplying the value to be multiplied by itself as many times as the power value.
* PangkatDC(): used the Divide and Conquer method to calculate the power result by dividing the problem into smaller subproblems, then calculating the power of half the power value, and then combining the results.

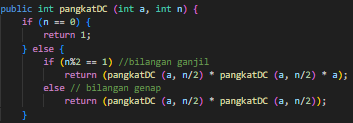
1. Pada method PangkatDC() terdapat potongan program sebagai berikut:



Jelaskan arti potongan kode tersebut

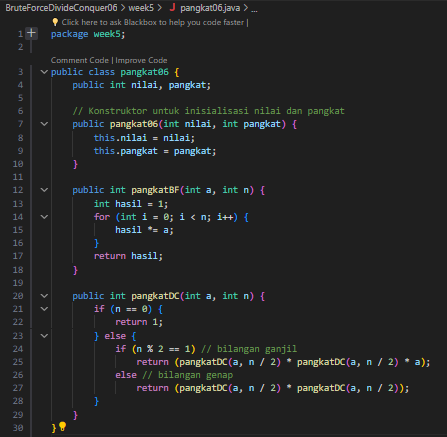
* ‘if (n == 0)’: if the power value is 0, then the result is 1 or true.
* ‘if (n%2 == 1)’: if odd, then calculate the result by dividing the power by two and multiply the result by itself and value(a). if odd, then calculate and divide the power by two then multiply the result by itself without value(a).

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

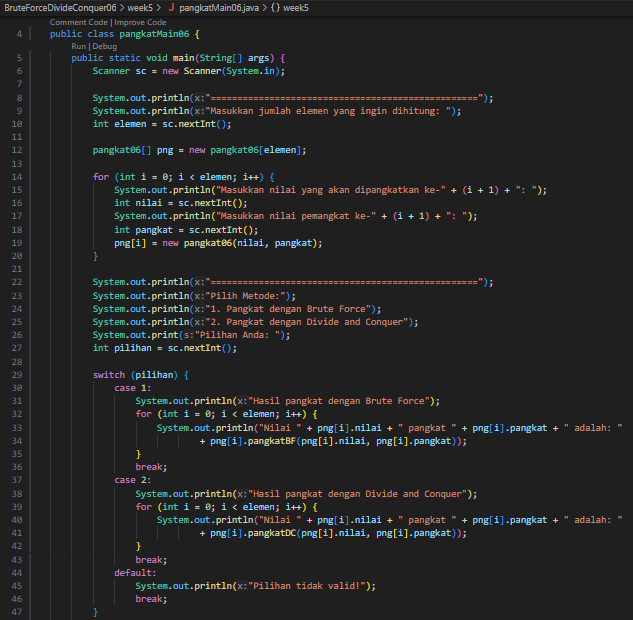


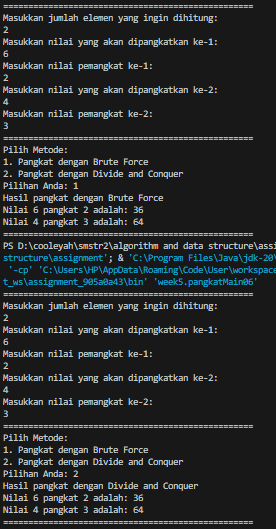
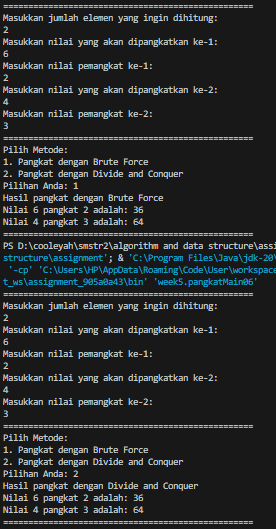
* The combine stage is usually not explicitly visible in the code, it usually occurs when combining results from smaller sub-problems to form the final result. The combine stage is implicit in the looping process of the 'pangkatDC()' method.

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



1. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan!



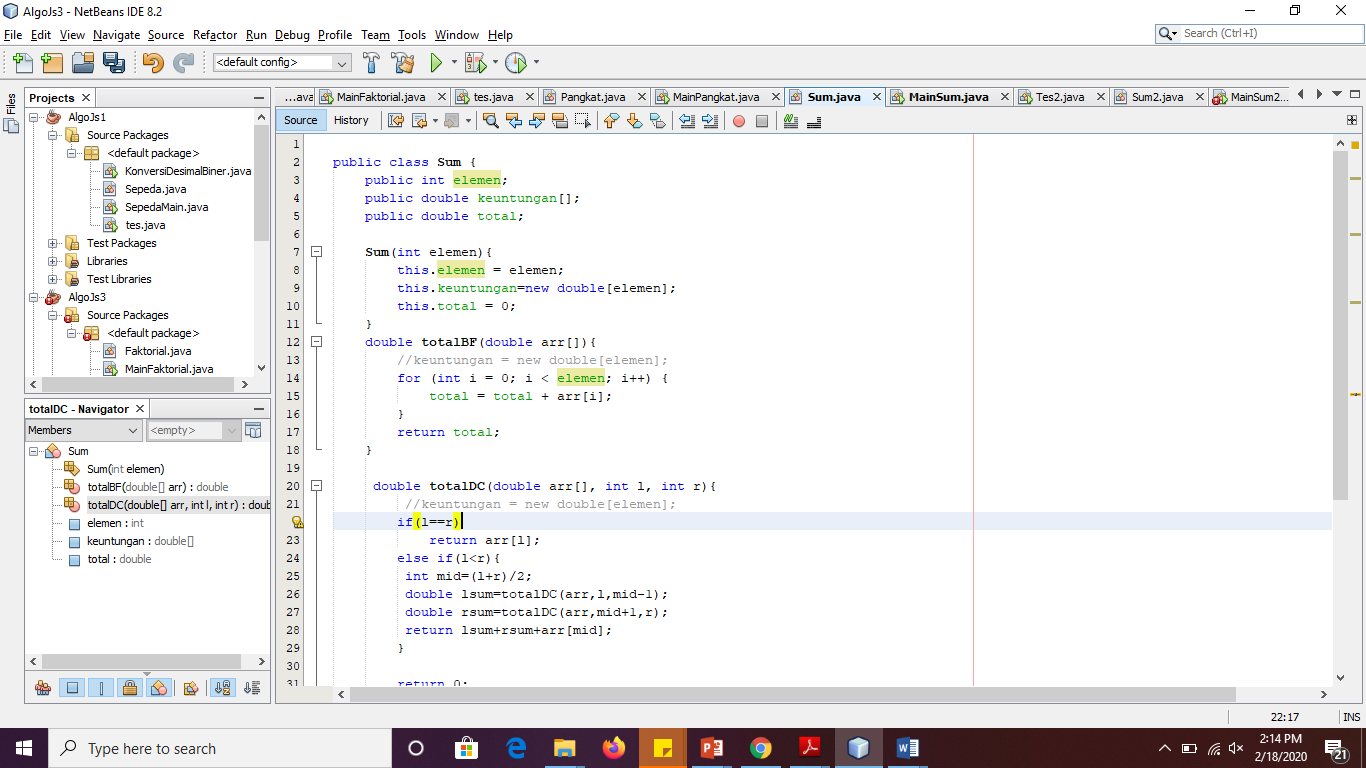
 

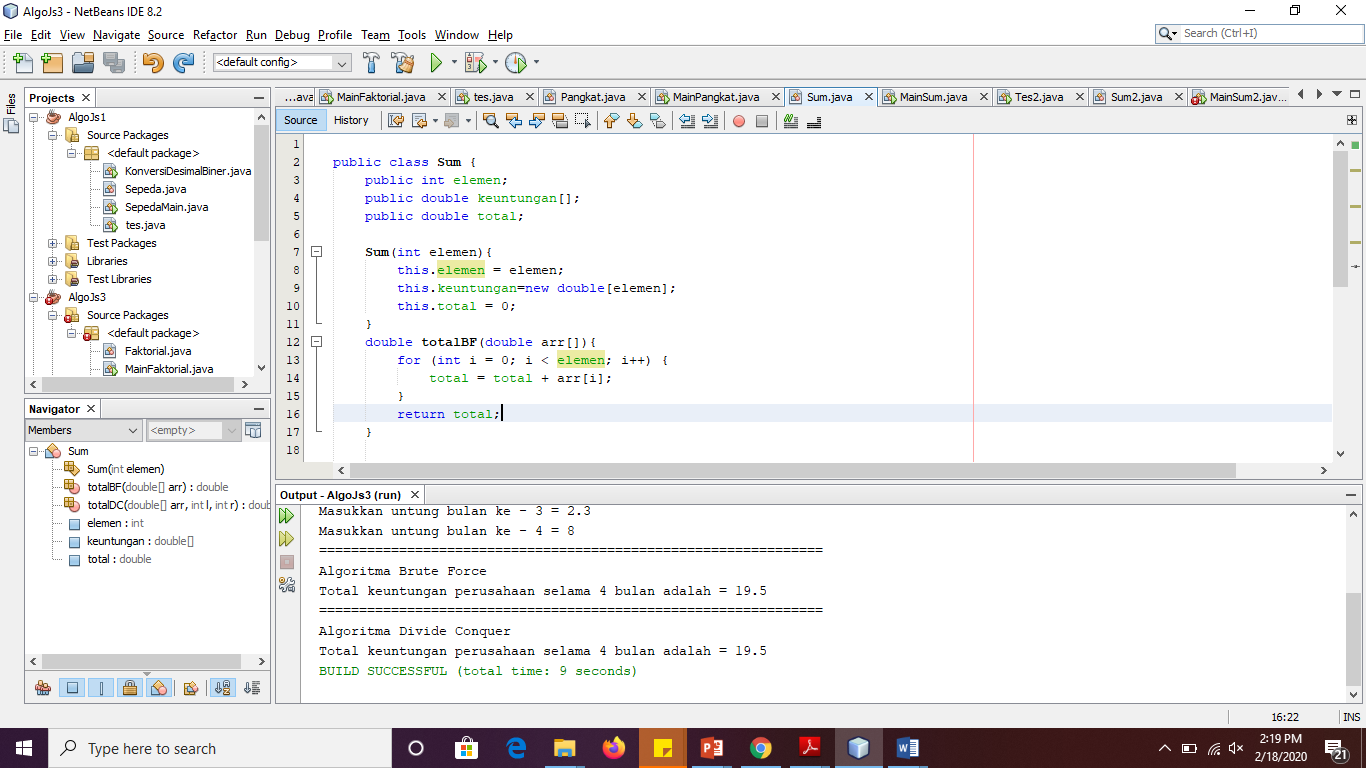
1. Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer

Di dalam percobaan ini, kita akan mempraktekkan bagaimana proses *divide, conquer*, dan *combine* diterapkan pada studi kasus penjumlahan keuntungan suatu perusahaan dalam beberapa bulan.

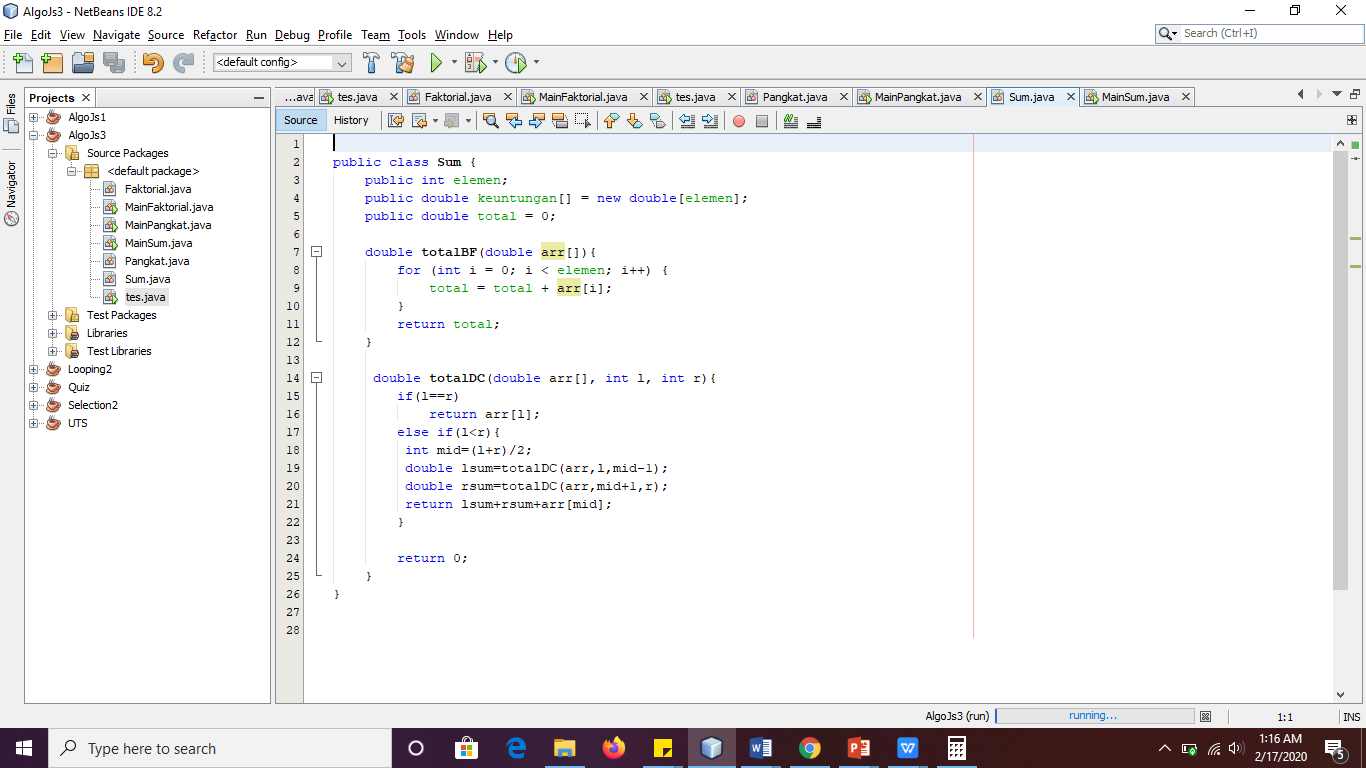
* + 1. Langkah-langkah Percobaan

1. Pada paket minggu5. Buat class baru yaitu class Sum. DI salam class tersebut terdapat beberapa atribut jumlah elemen array, array, dan juga total. Tambahkan pula konstruktor pada class Sum.

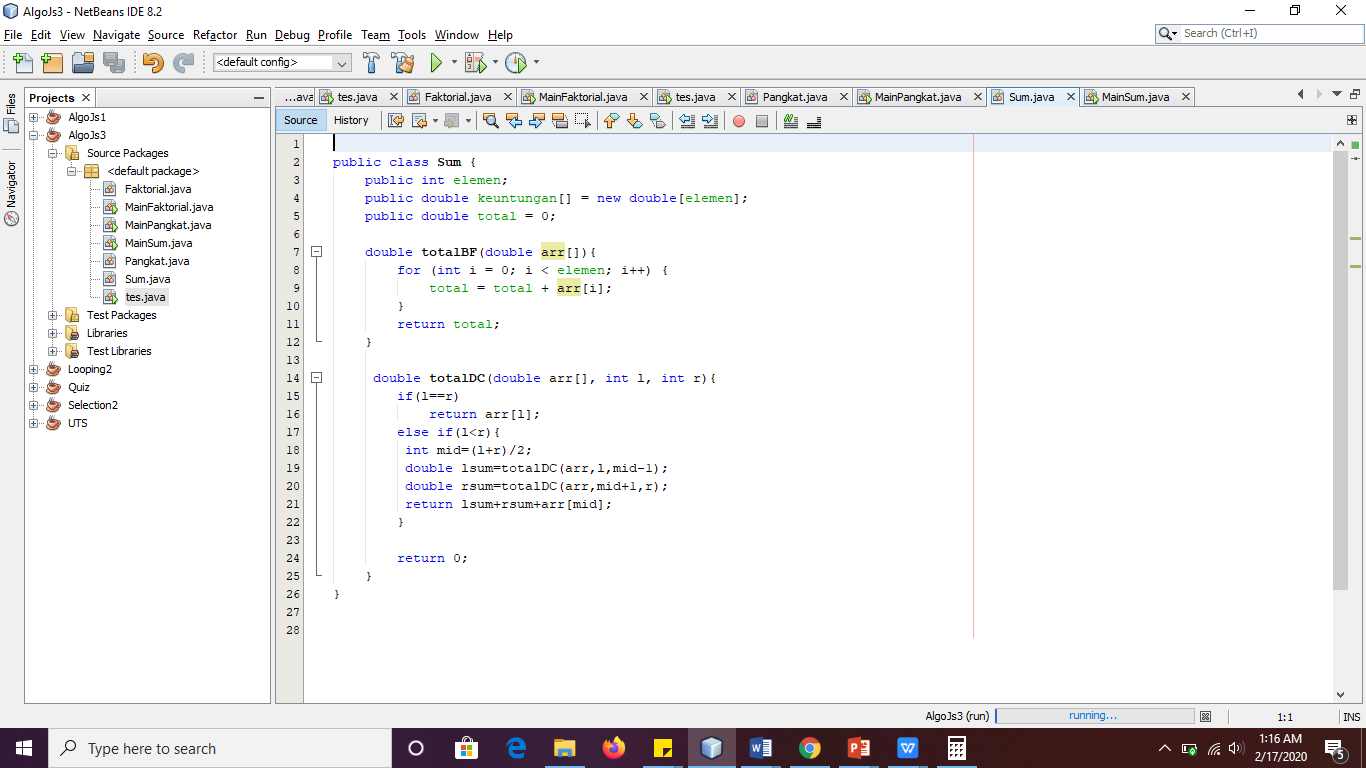




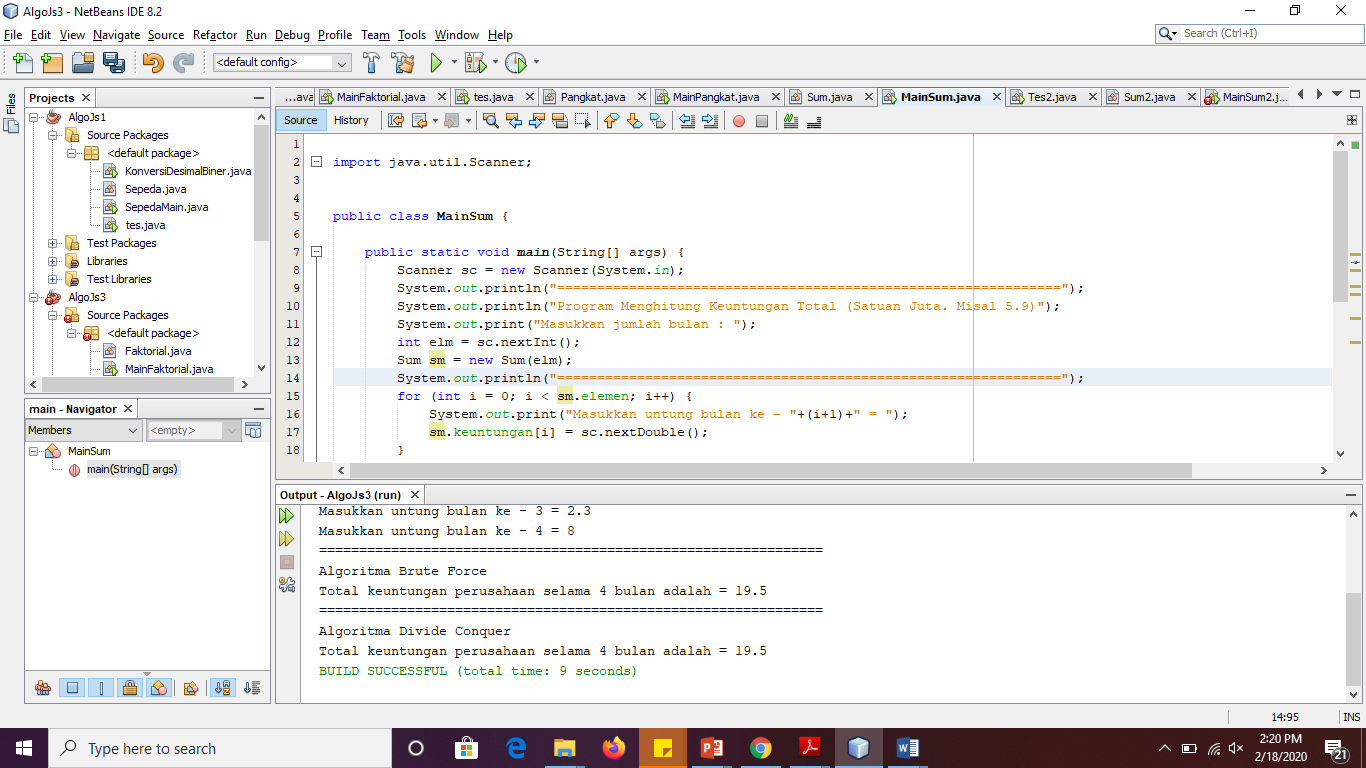
1. Tambahkan method TotalBF() yang akan menghitung total nilai array dengan cara *iterative*.



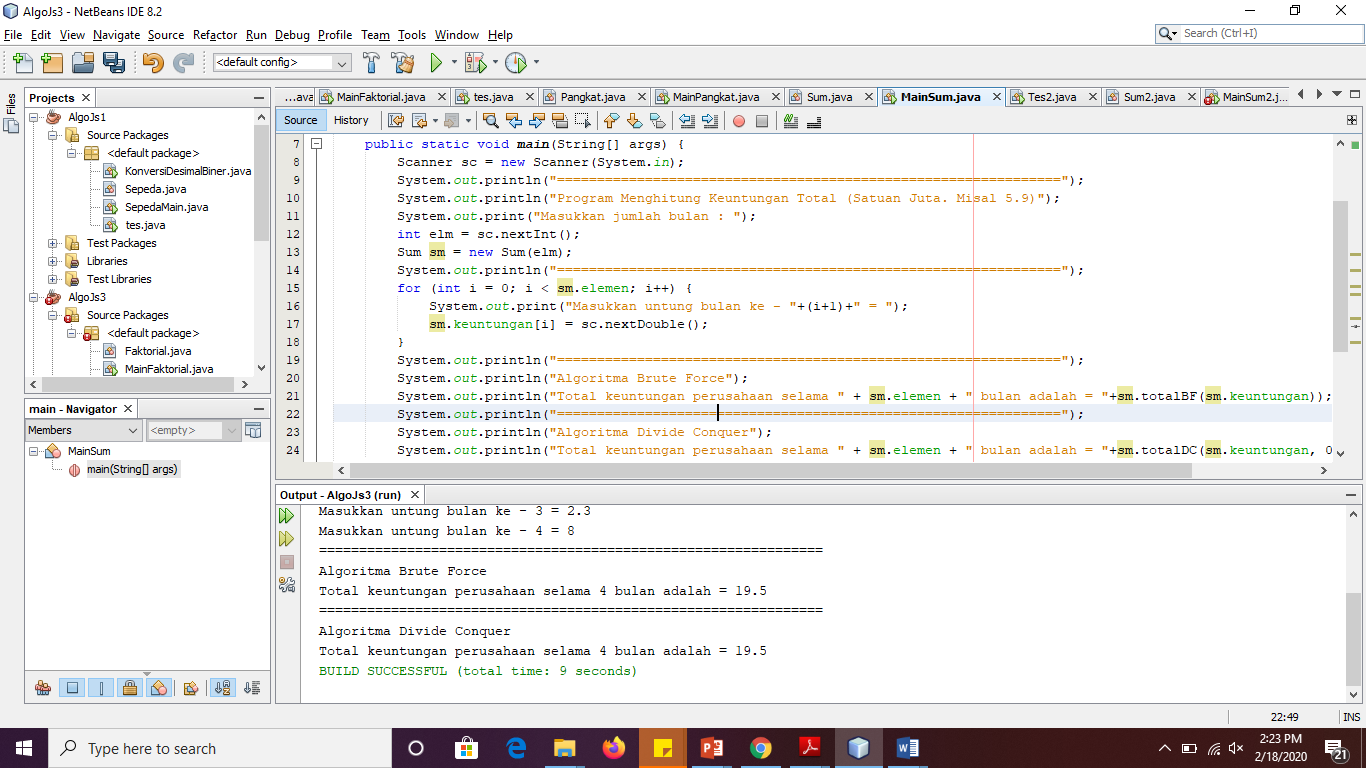
1. Tambahkan pula method TotalDC() untuk implementasi perhitungan nilai total array menggunakan algoritma Divide and Conquer



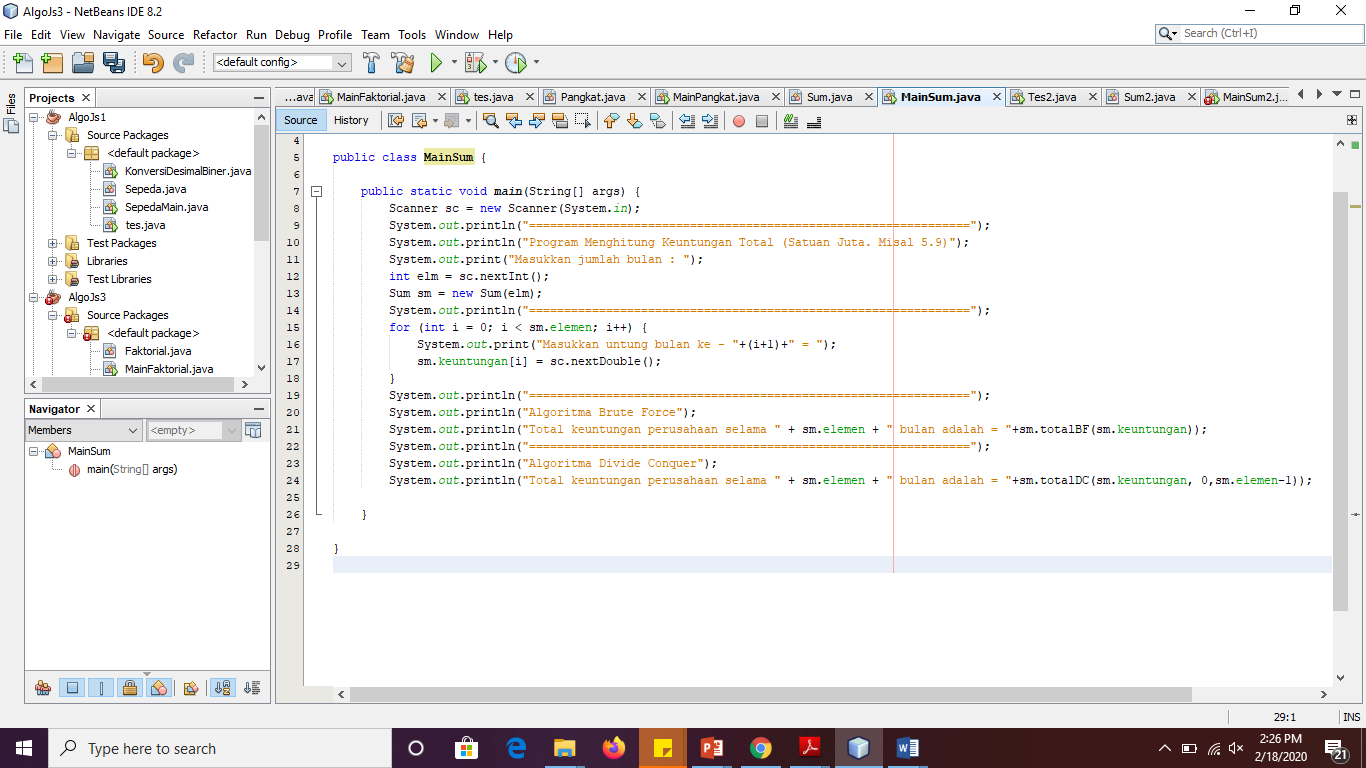
1. Buat class baru yaitu MainSum. Di dalam kelas ini terdapat method main. Pada method ini user dapat menuliskan berapa bulan keuntungan yang akan dihitung. Dalam kelas ini sekaligus dibuat instansiasi objek untuk memanggil atribut ataupun fungsi pada class Sum



1. Karena yang akan dihitung adalah total nilai keuntungan, maka ditambahkan pula pada method main mana array yang akan dihitung. Array tersebut merupakan atribut yang terdapat di class Sum, maka dari itu dibutuhkan pembuatan objek Sum terlebih dahulu.

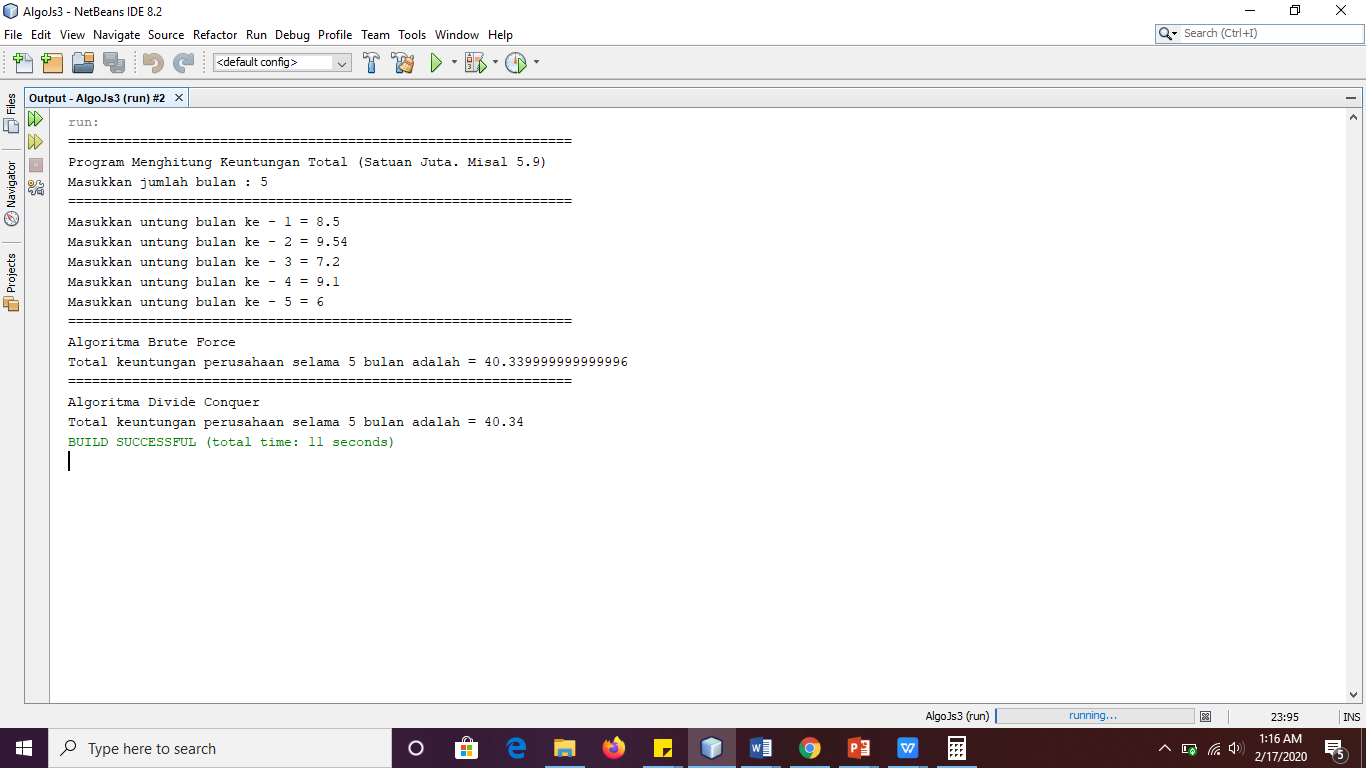


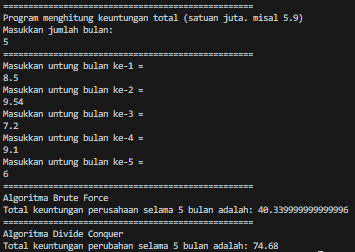
1. Tampilkan hasil perhitungan melalui objek yang telah dibuat untuk kedua cara yang ada (Brute Force dan Divide and Conquer)



* + 1. Verifikasi Hasil Percobaan

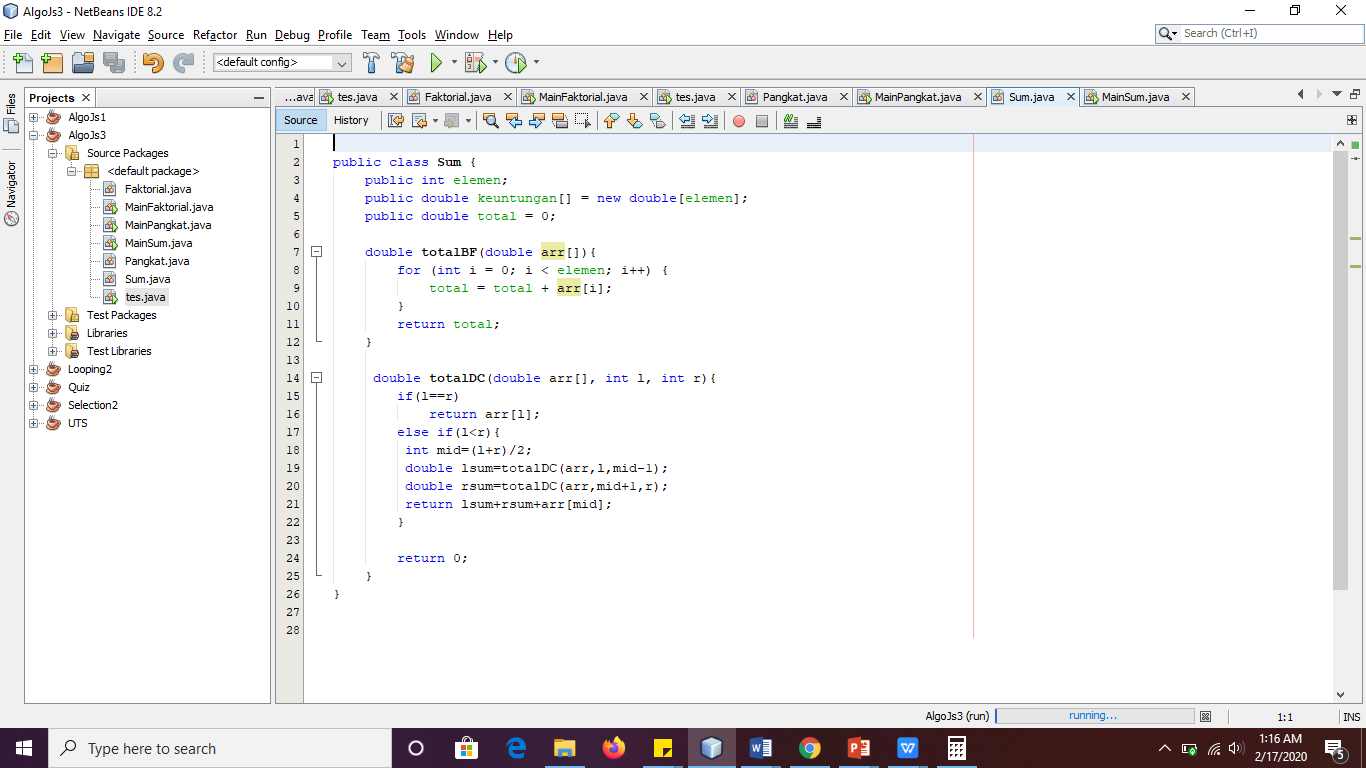
Cocokkan hasil compile kode program anda dengan gambar berikut ini.





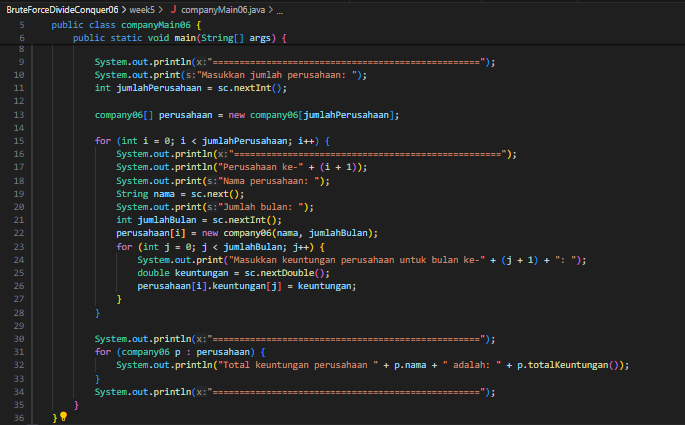
* + 1. Pertanyaan

1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method TotalBF() ataupun TotalDC()
   * TotalBF(): to perform monthly profit increments one by one using looping.
   * TotalDC(): to split the profit array into two and calculate the total profit of each part recursively, then combine both parts as well as the profit of the middle element.
2. Perhatikan output dari kedua jenis algoritma tersebut bisa jadi memiliki hasil berbeda di belakang koma. Bagaimana membatasi output di belakang koma agar menjadi standar untuk kedua jenis algoritma tersebut.
   * To limit the output behind a comma, you can use 'String.format'.
3. Mengapa terdapat formulasi *return* *value* berikut?Jelaskan!



* + Used to combine the total gain of the two parts of the array as well as the gain of the center element or combine stage.

1. Kenapa dibutuhkan variable mid pada method TotalDC()?
   * is used to mark the midpoint when dividing the profit array into two parts in the 'totalDC()' method, in order to divide the subproblems into smaller ones.
2. Program perhitungan keuntungan suatu perusahaan ini hanya untuk satu perusahaan saja. Bagaimana cara menghitung sekaligus keuntungan beberapa bulan untuk beberapa perusahaan.(Setiap perusahaan bisa saja memiliki jumlah bulan berbeda-beda)? Buktikan dengan program!



* 1. Latihan Praktikum

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.*

