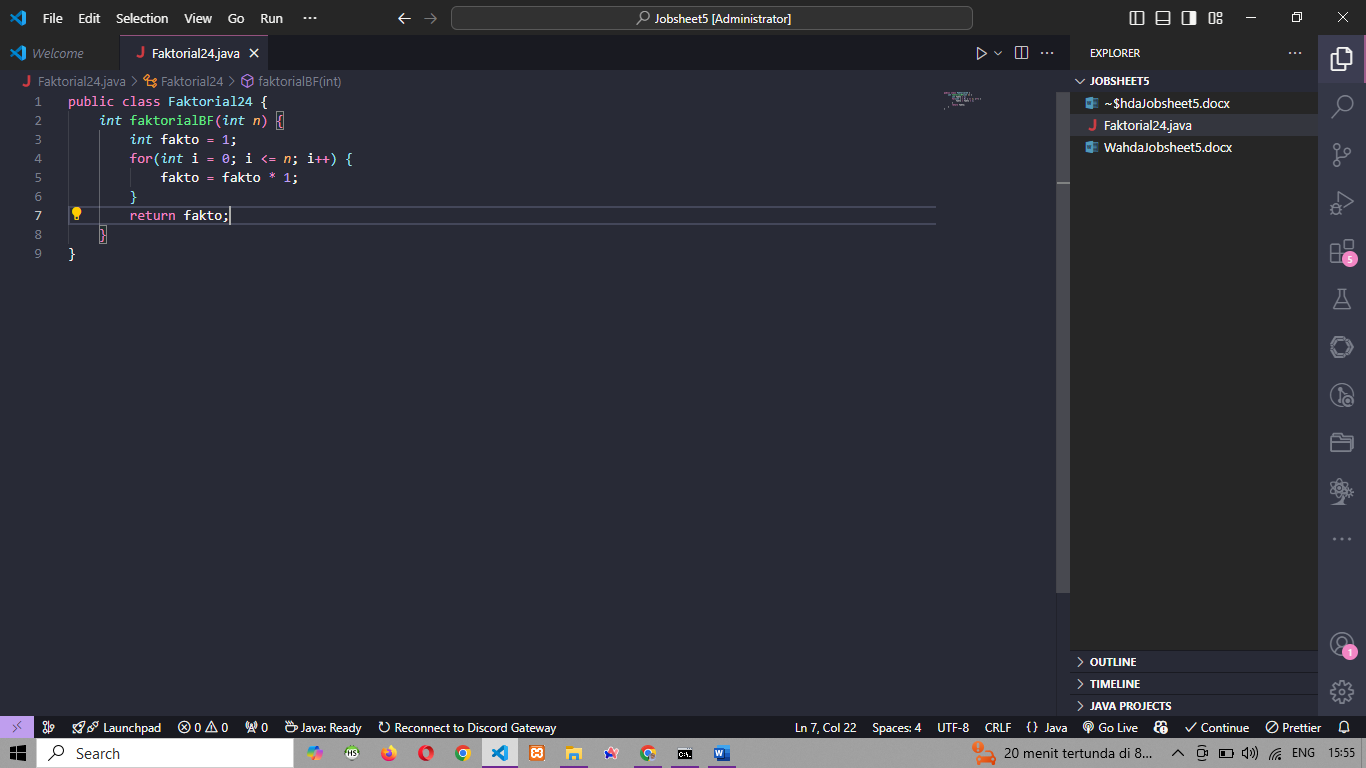
Nama: Wahda Adella Putri Febriana

Kelas: 1B / 24

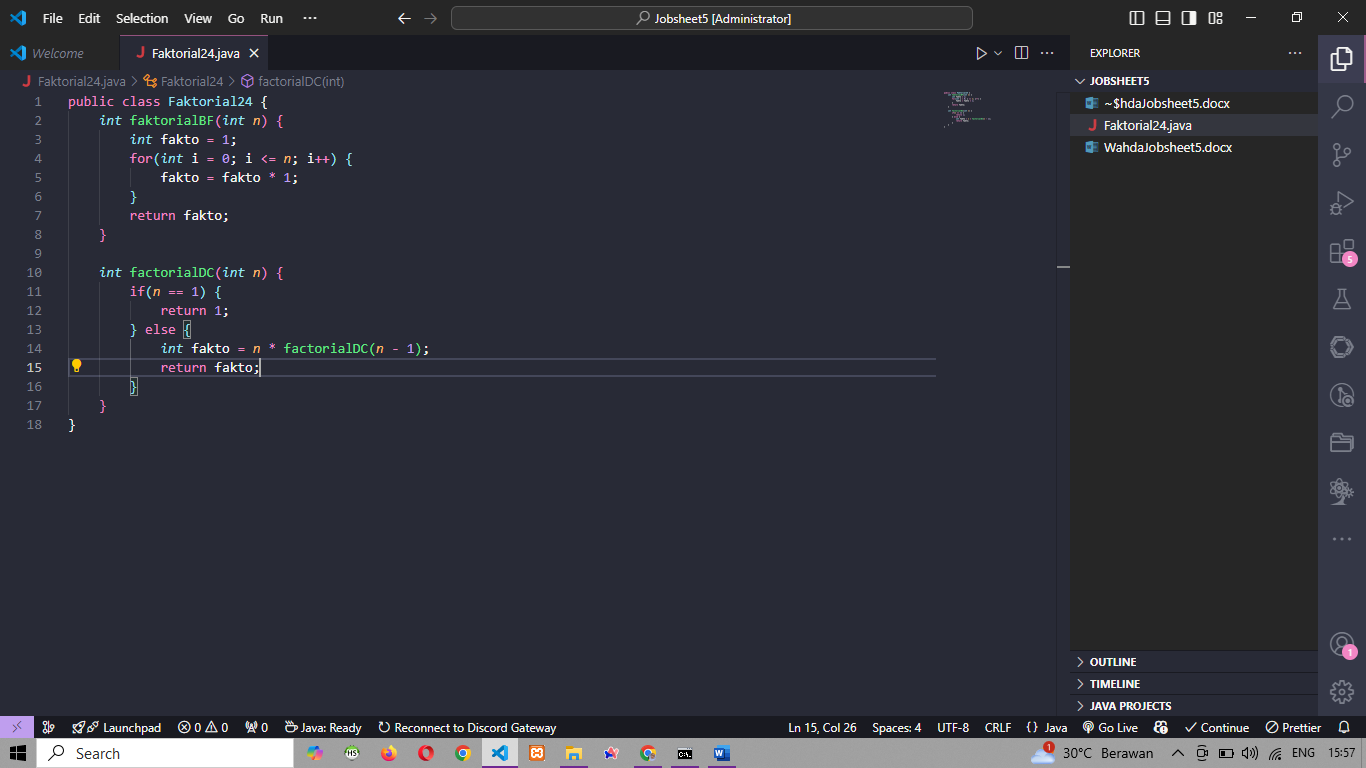
NIM: 2440700156

**PERCOBAAN 1**

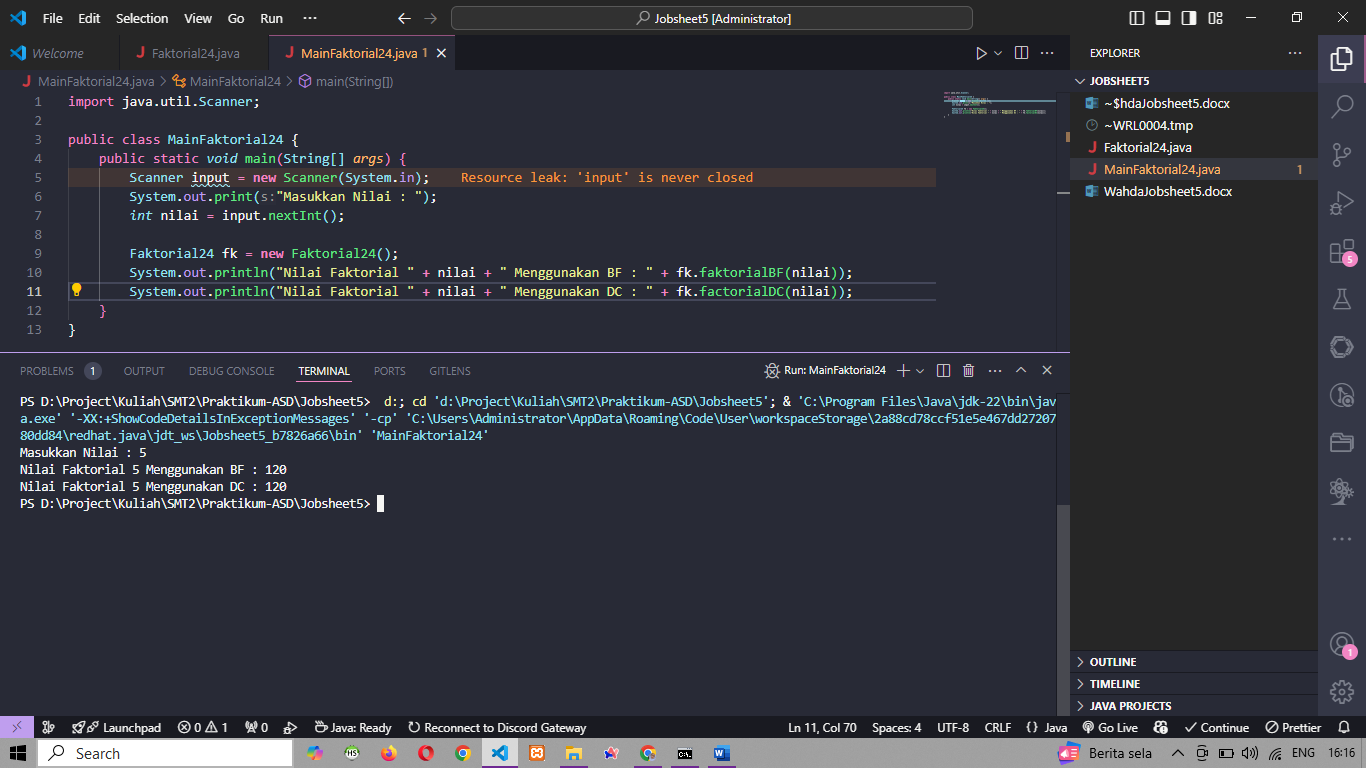
1. Buat file factorial24 dan buat fungsi faktorialBF



1. Tambahkan factorialDC

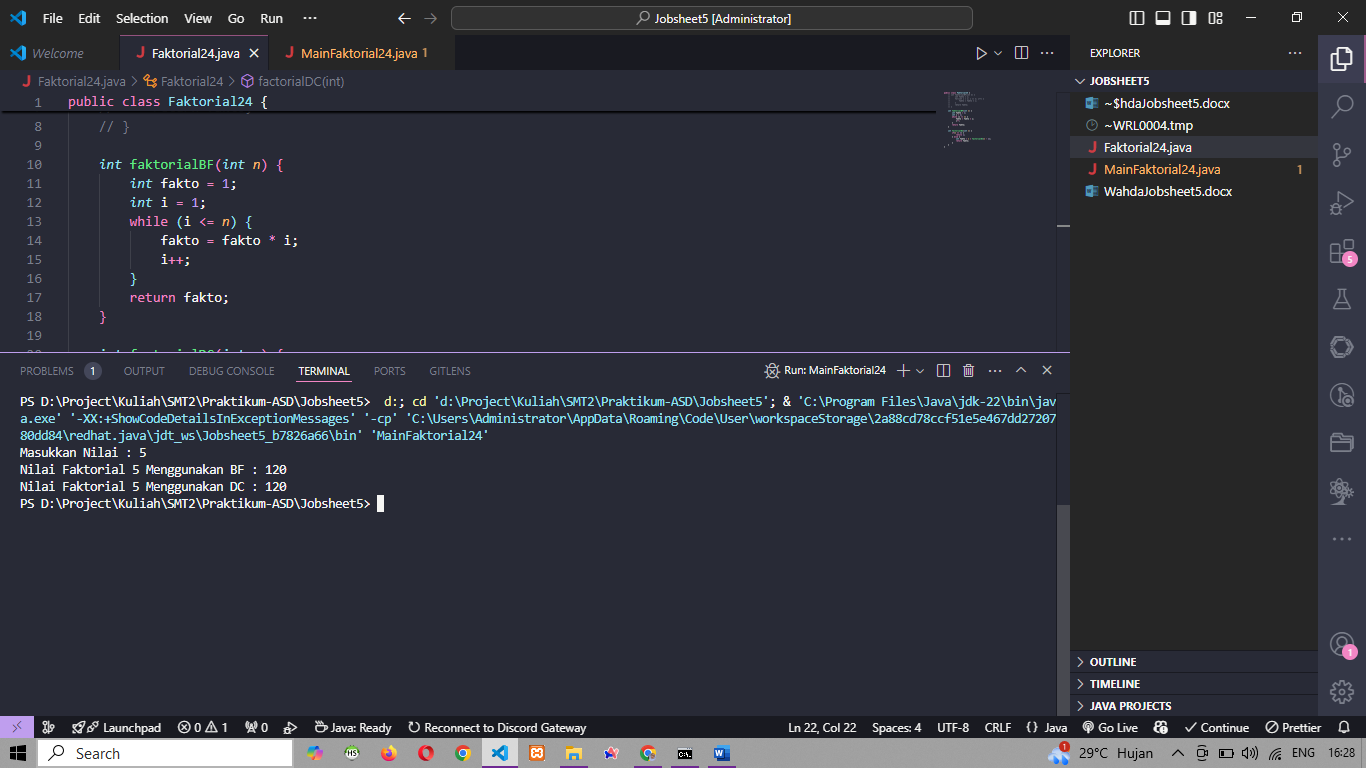


1. Panggil fungsi nya di main dan run



**PERTANYAAN**

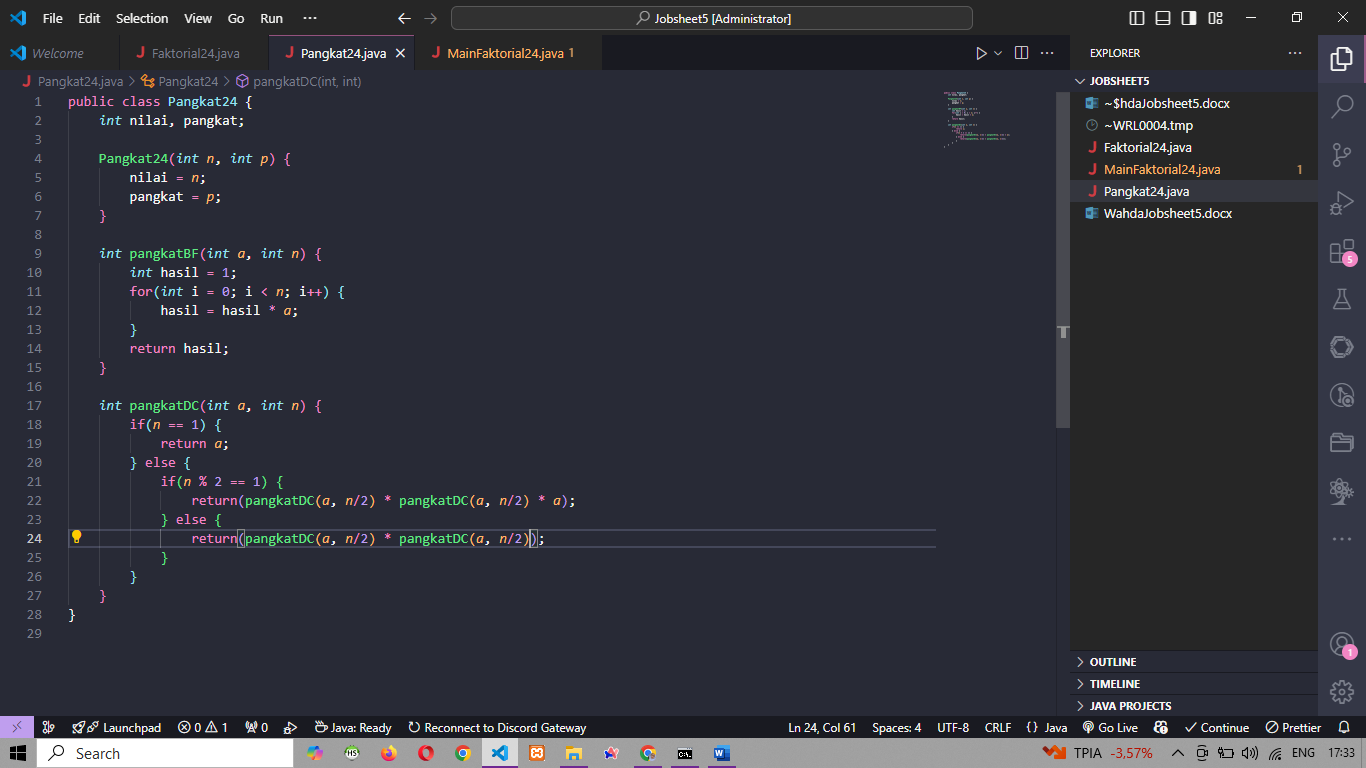
1. If nya adalah base case nya dan else nya adalah fungsi rekursif nya
2. Bisa, contoh saya menggunakan while



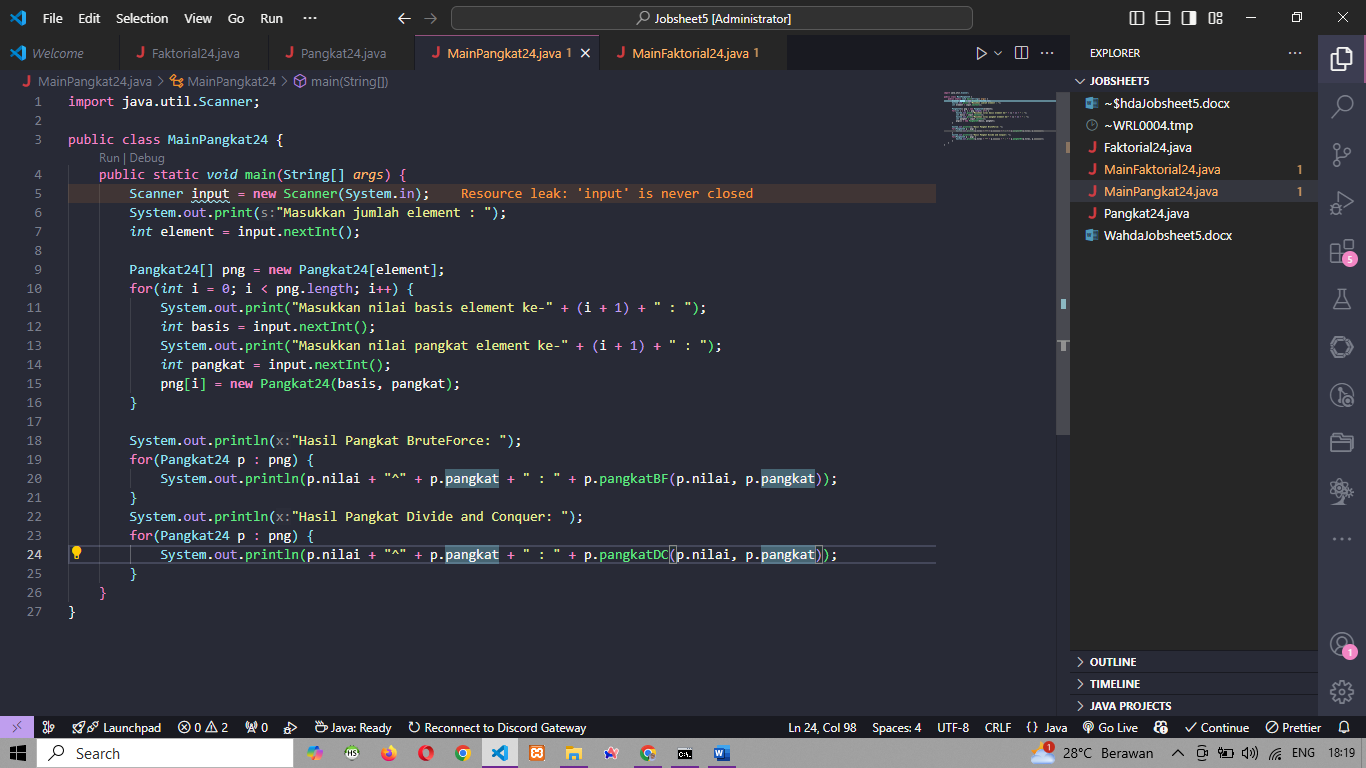
1. fakto \*= i; digunakan untuk mengkalikan fakto setiap iterasi perulangan. int fakto = n \* faktorialDC(n - 1); sama namun ini memanggil fungsinya sendiri dengan menghitung n – 1 dulu baru di kali n
2. factorialBF menggunakan Brute Force (Iteratif) menggunakan for dilakukan bertahap dari 1 hingga n dengan mengalikan nilai satu persatu. faktorialDC menggunakan Divide and Conquer (Rekursif) dengan memanggil fungsinya sendiri, memecah masalah menjadi lebih kecil sampai mencapai base case.

**PERCOBAAN 2**

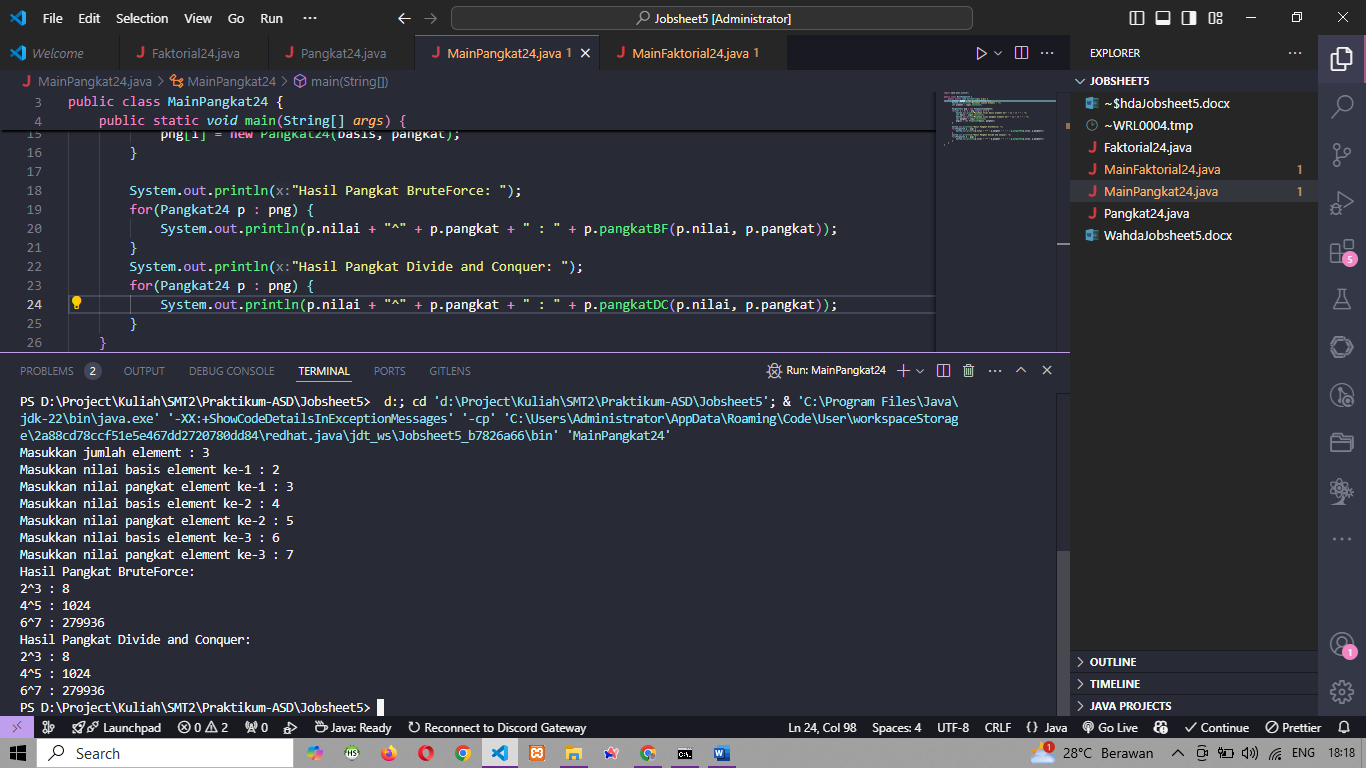
1. Buatkan file Pangkat24.java dan fungsi fungsinya



1. Buat file MainPangkat24.java dan panggil pangkat24.java

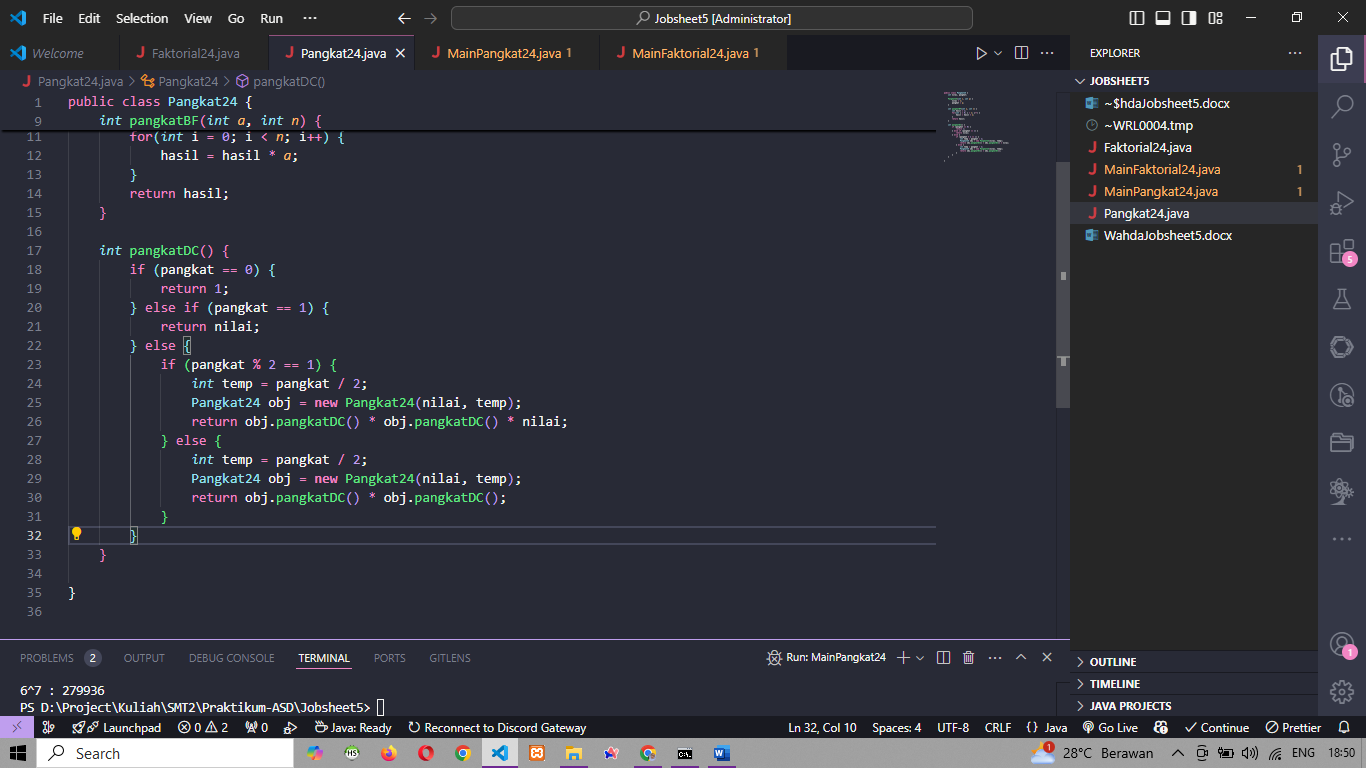


1. Hasil Run



**PERTANYAAN**

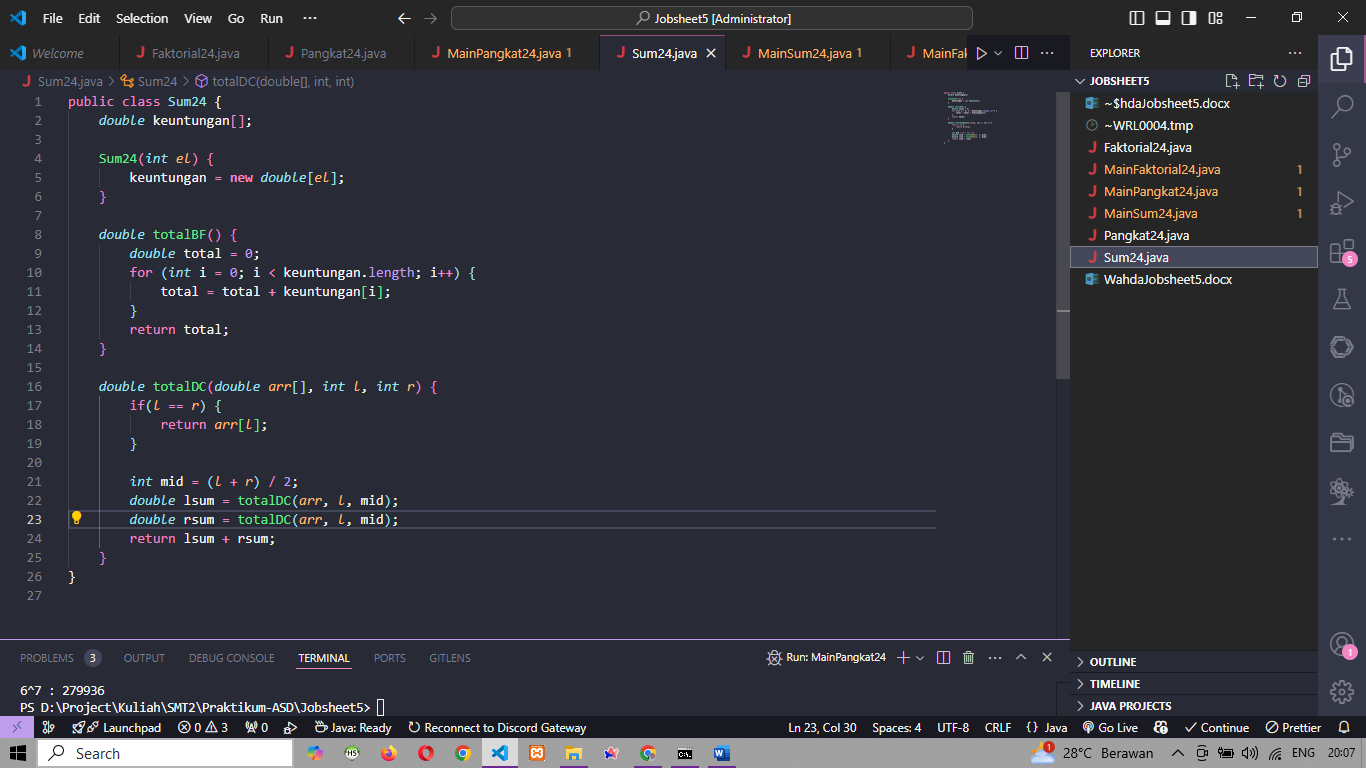
1. pangkatBF menggunakan brute force, pangkatDC menggunakan divide dan qoncuer
2. Sudah, return(pangkatDC(a, n/2) \* pangkatDC(a, n/2) \* a);
3. Relevan, bisa tapi menjadi ribet karena rekursif butuh nilai n yang berubah tiap Langkah



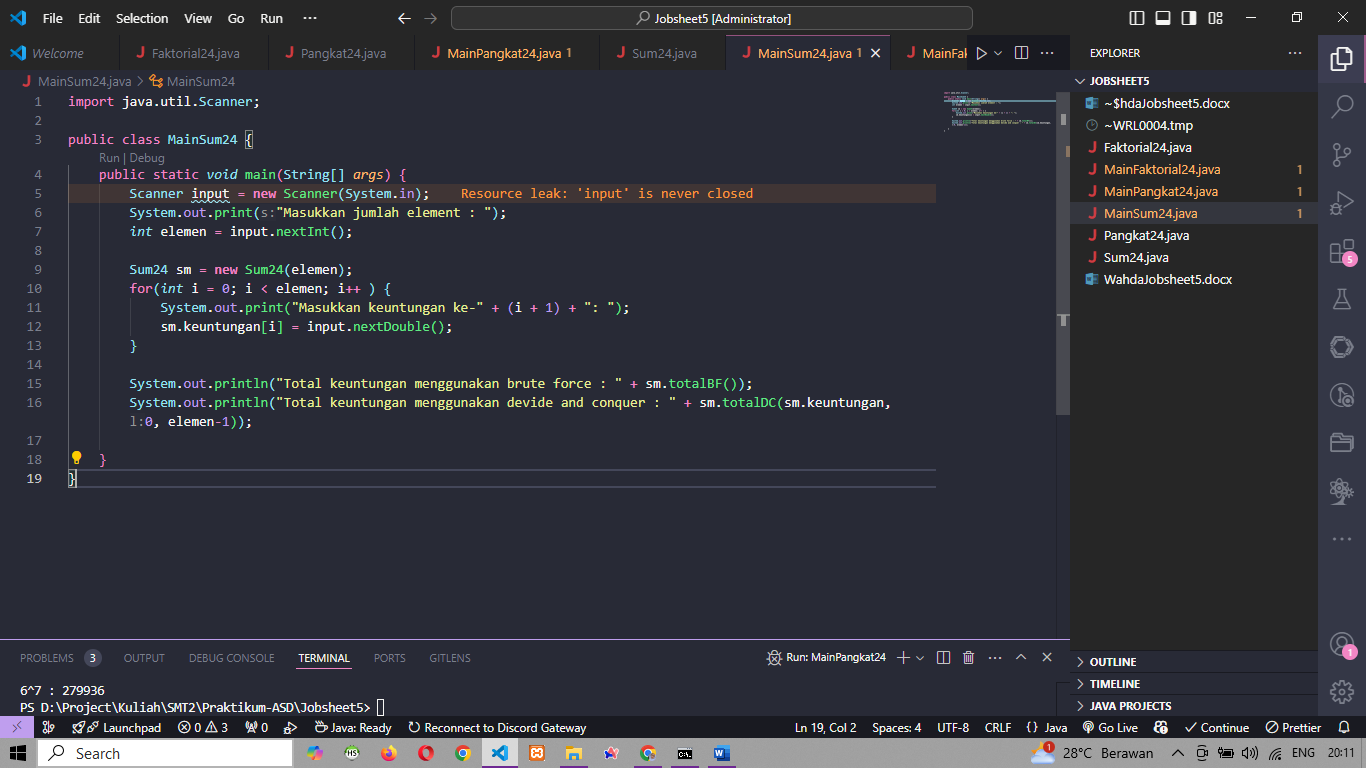
1. pangkatBF menggunakan brute force mengalikan sebanyak n dalam sebuah loop, pangkatDC menggunakan divide dan qoncuer membagi masalah menjadi 2 bagian lebih kecil

**PERCOBAAN 3**

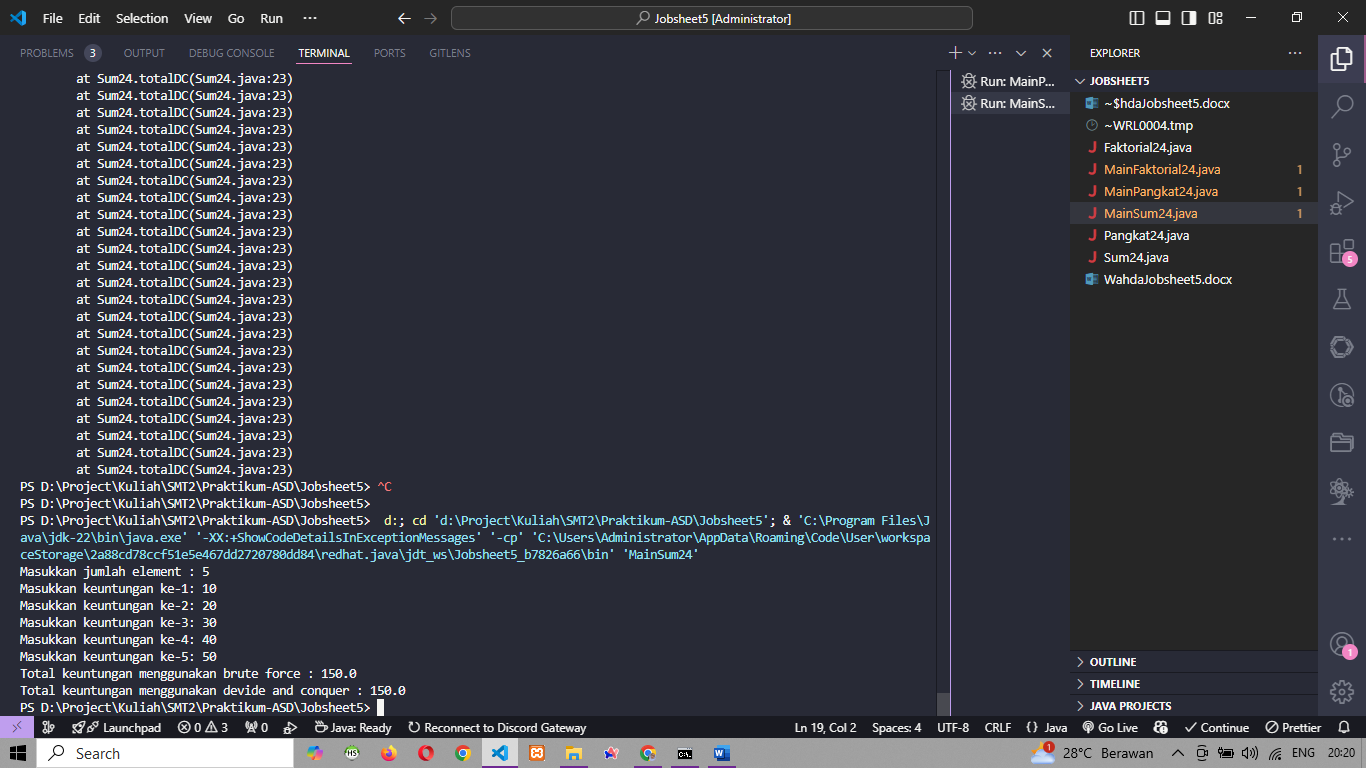
1. Buat file Sum24.java



1. File mainSum



1. Hasil Run



**PERTANYAAN**

1. Mid nerfungsi untuk membagi array menjadi 2 bagian
2. Kedua bars tersebut merupakan inti dari rekursi diguakan untuk menghitung jumlah total element array
3. Untuk menggabungkan hasil dari perhitungan dua bagian array yang telah dibagi sebelumnya
4. Base case nya yang ini

if (l == r) {

return arr[l];

}

1. Pertama array dibagi menjadi 2 bagian menggunakan index Tengah, lalu bagian yang di bagi di selesaikan secara rekursif dengan memanggil fungsi itu lagi. Lalu setelah itu hasilnya digabung dengan cara dijumlah