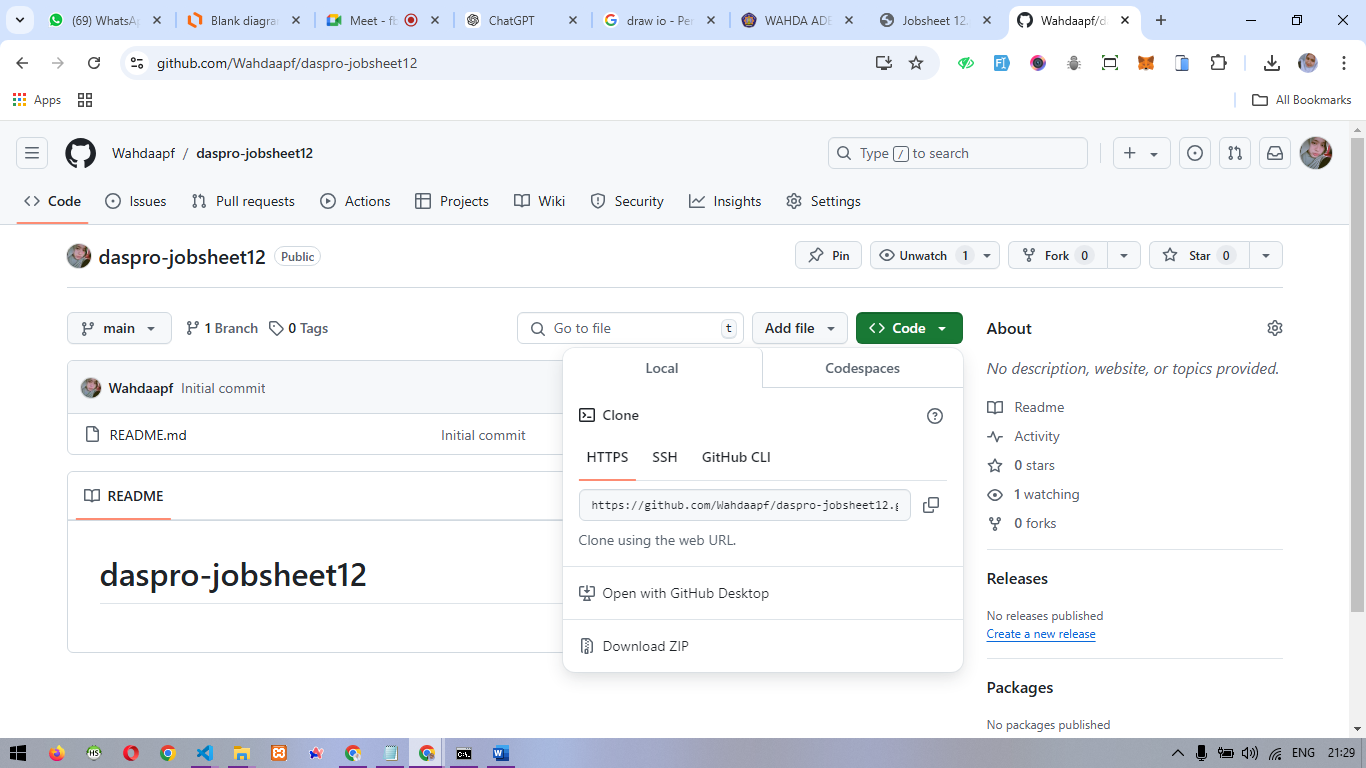
Nama: Wahda Adella Putri Febriana

NIM : 244107020156

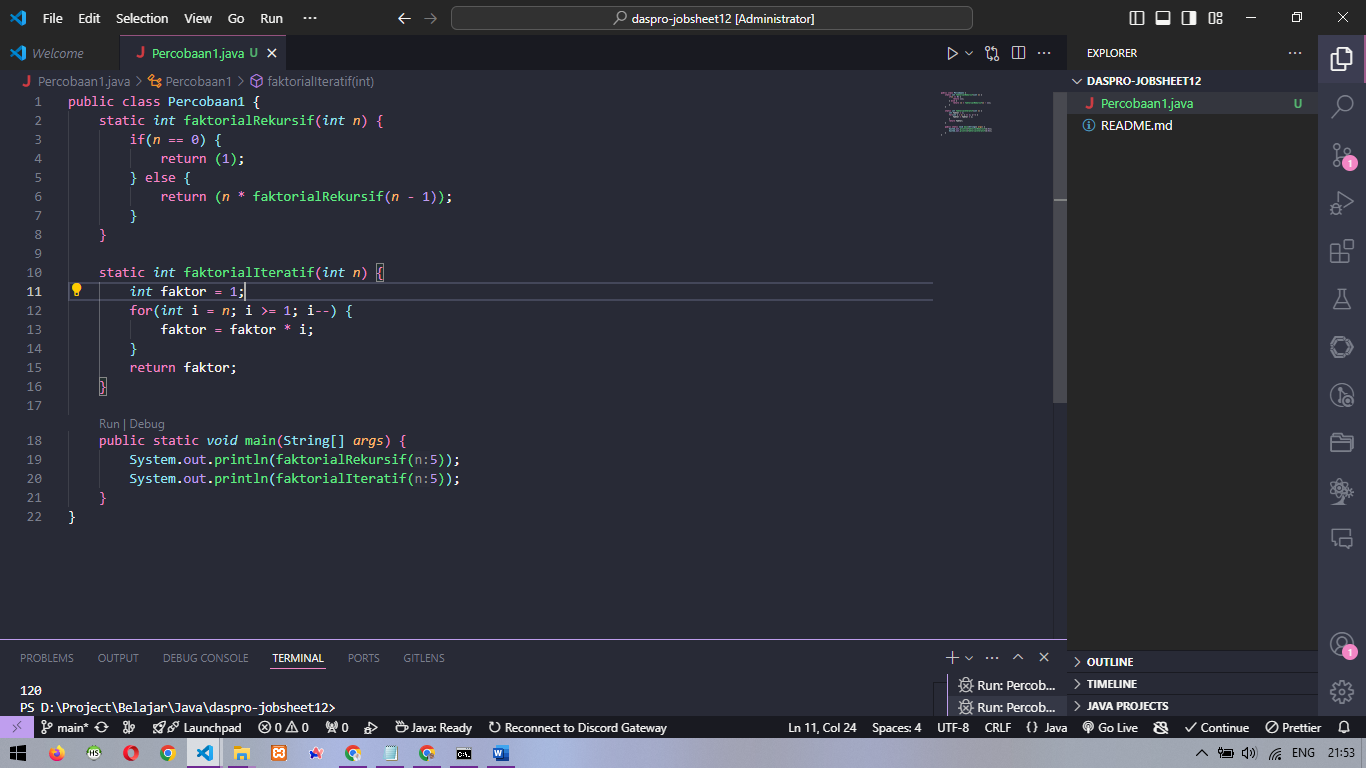
Kelas: 1B

**Percobaan 1**

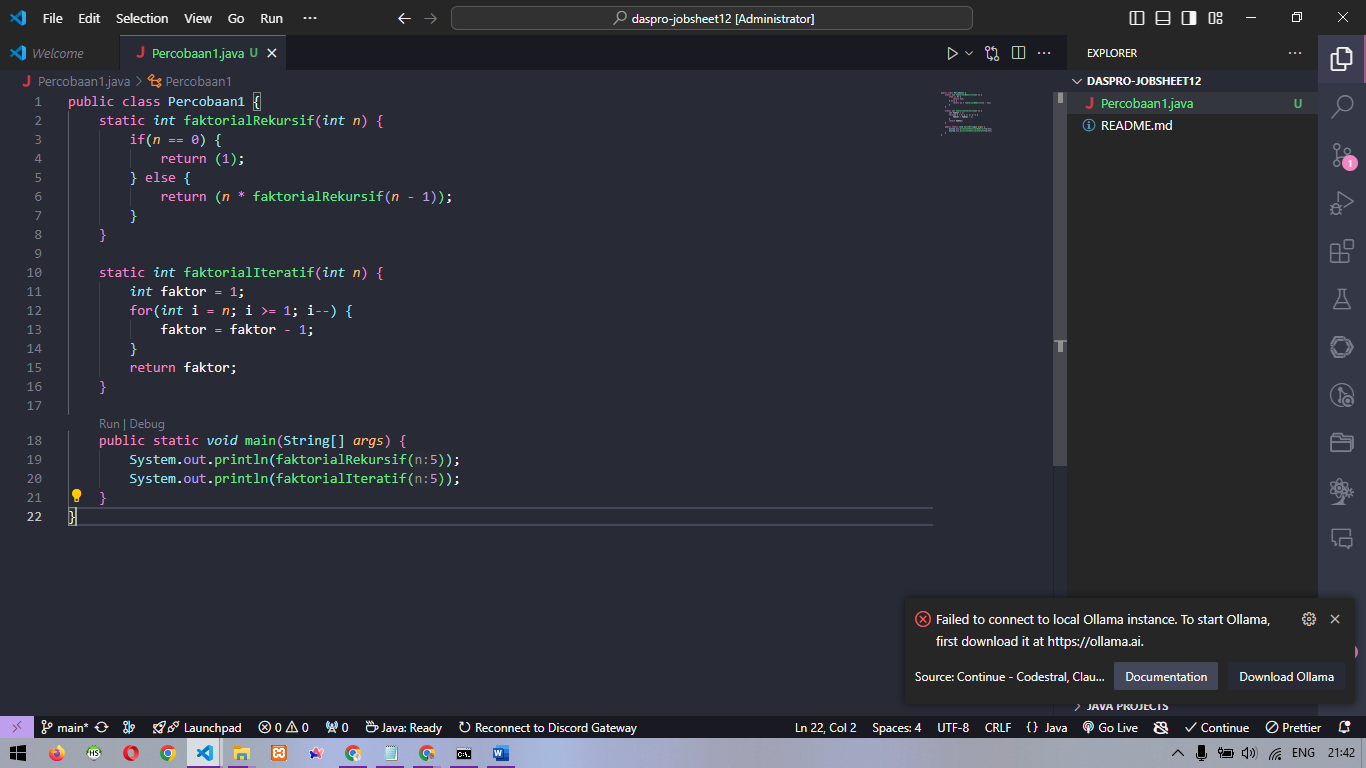
1. Buat Repository “daspro-jobsheet12”



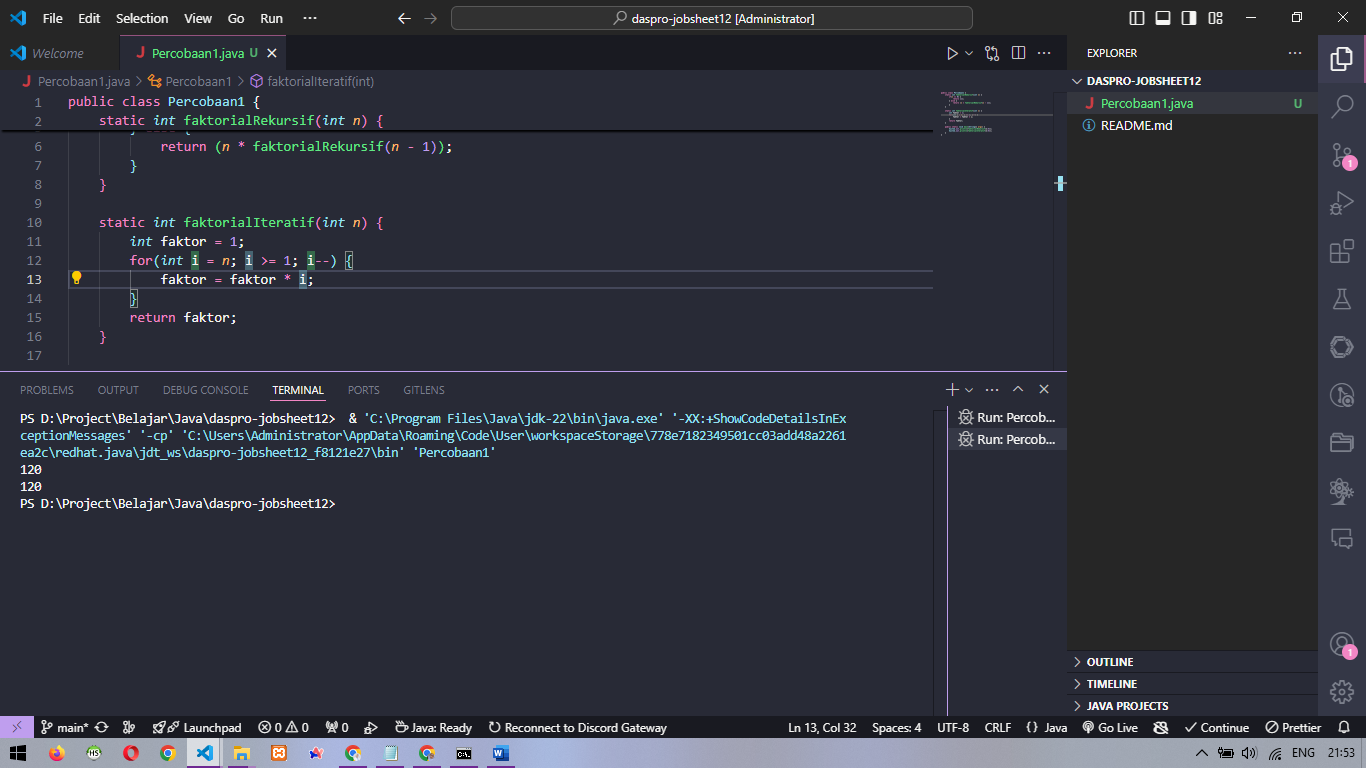
1. Clone dan Open di VSC, buat file Percobaan1.java, dan fungsi faktorialRekursif(), faktorialIteratif()



1. Panggil function di main



1. Hasil run



**Pertanyaan 1**

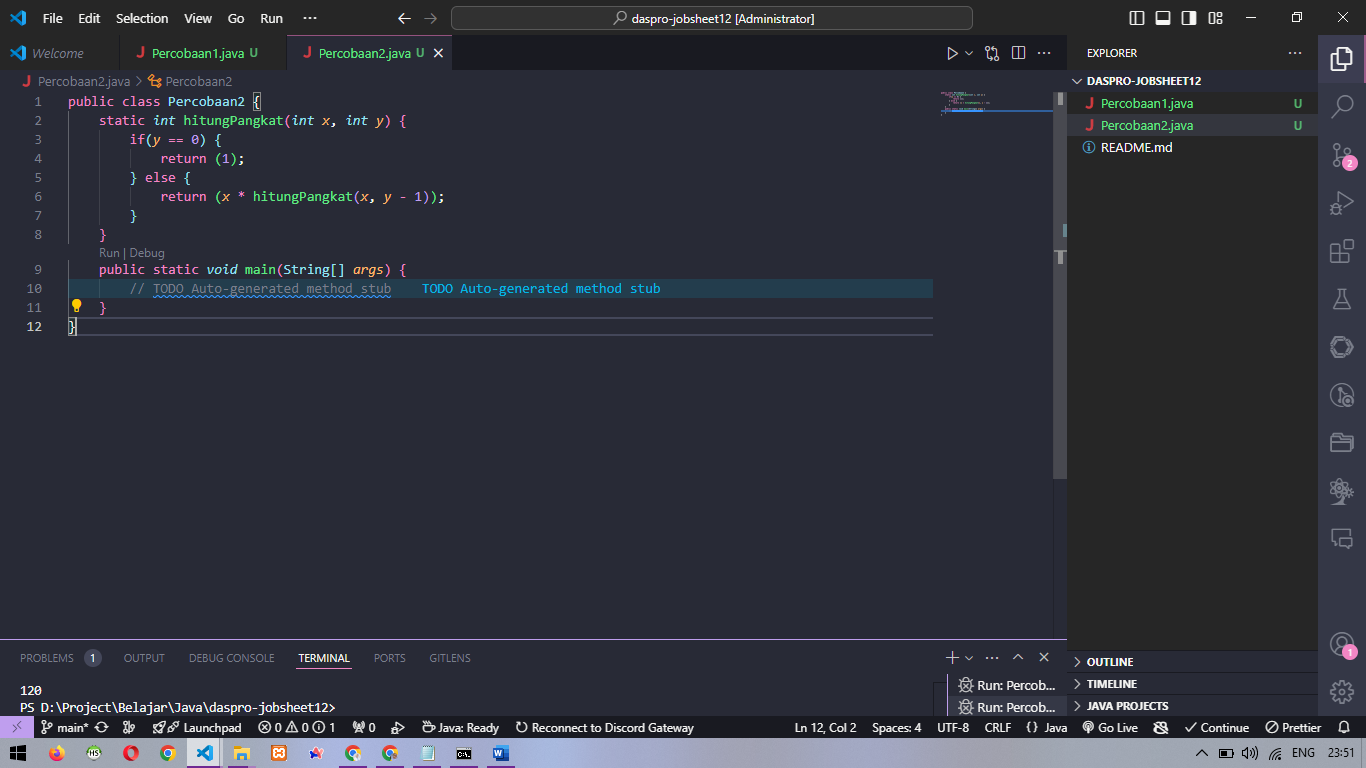
1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama eksekusi untuk menyelesaikan suatu masalah dengan cara membaginya menjadi submasalah yang lebih kecil dan serupa.
2. a. menghitung nilai faktorial

b. Fibonacci

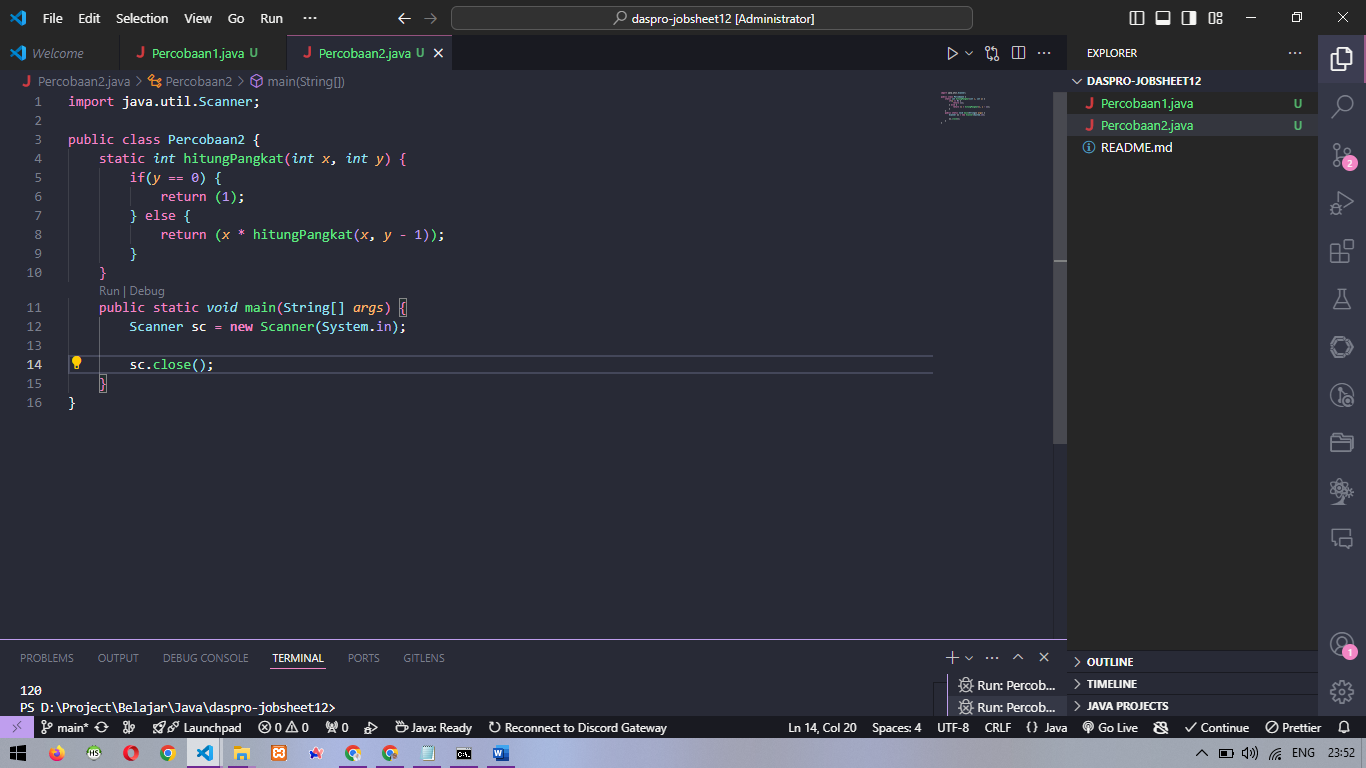
1. Sama, fungsi faktorialRekursif memanggil dirinya sendiri secara terus-menerus hingga mencapai n == 0, fungsi berhenti memanggil dirinya sendiri dan mulai "kembali" (return) hasil perhitungan secara berurutan ke pemanggil sebelumnya, fungsi faktorialIteratif menggunakan perulangan for untuk menghitung faktorial. Variabel faktor digunakan untuk menyimpan hasil perhitungan sementara, dan loop mengalikan nilai faktor dengan angka yang menurun dari n hingga 1.

**Percobaan 2**

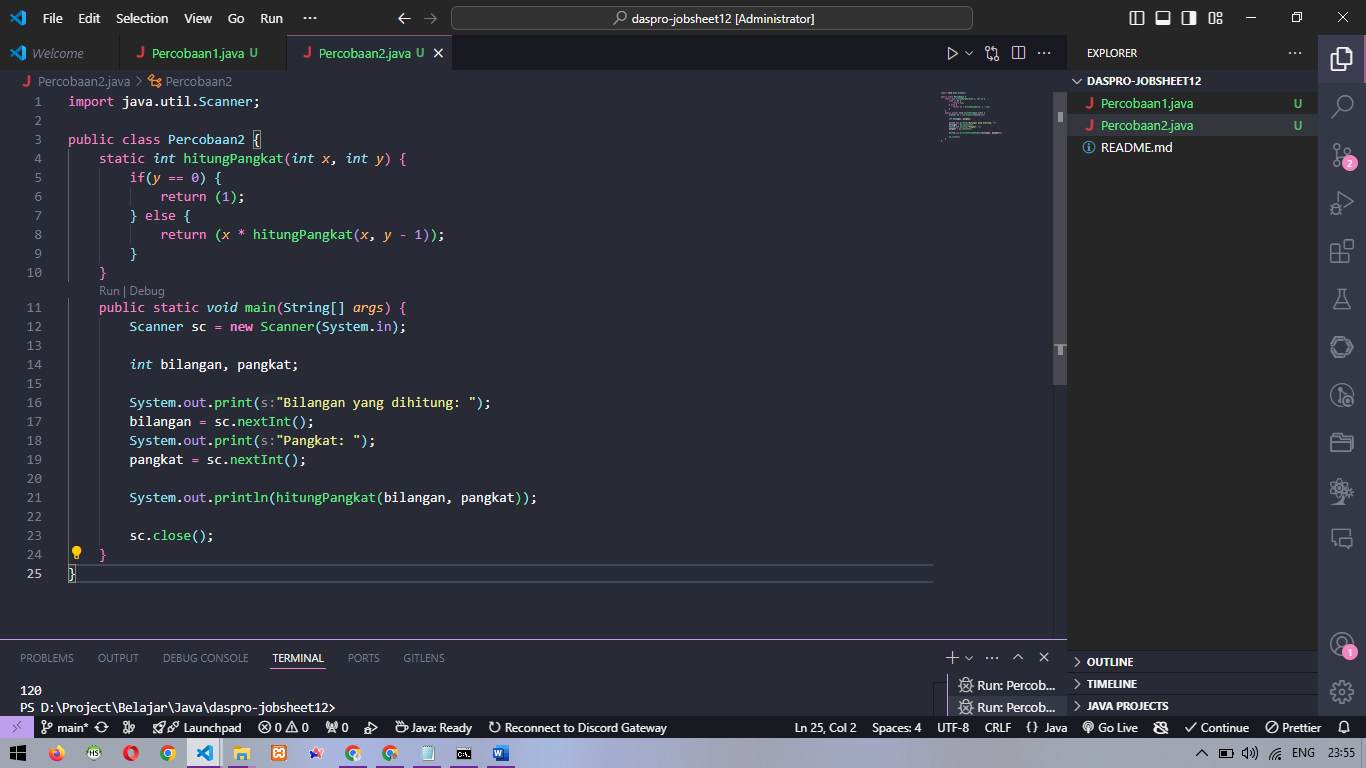
1. Buat file Percobaan2.java, dan fungsi hitungPangkat



1. Deklarasi scanner

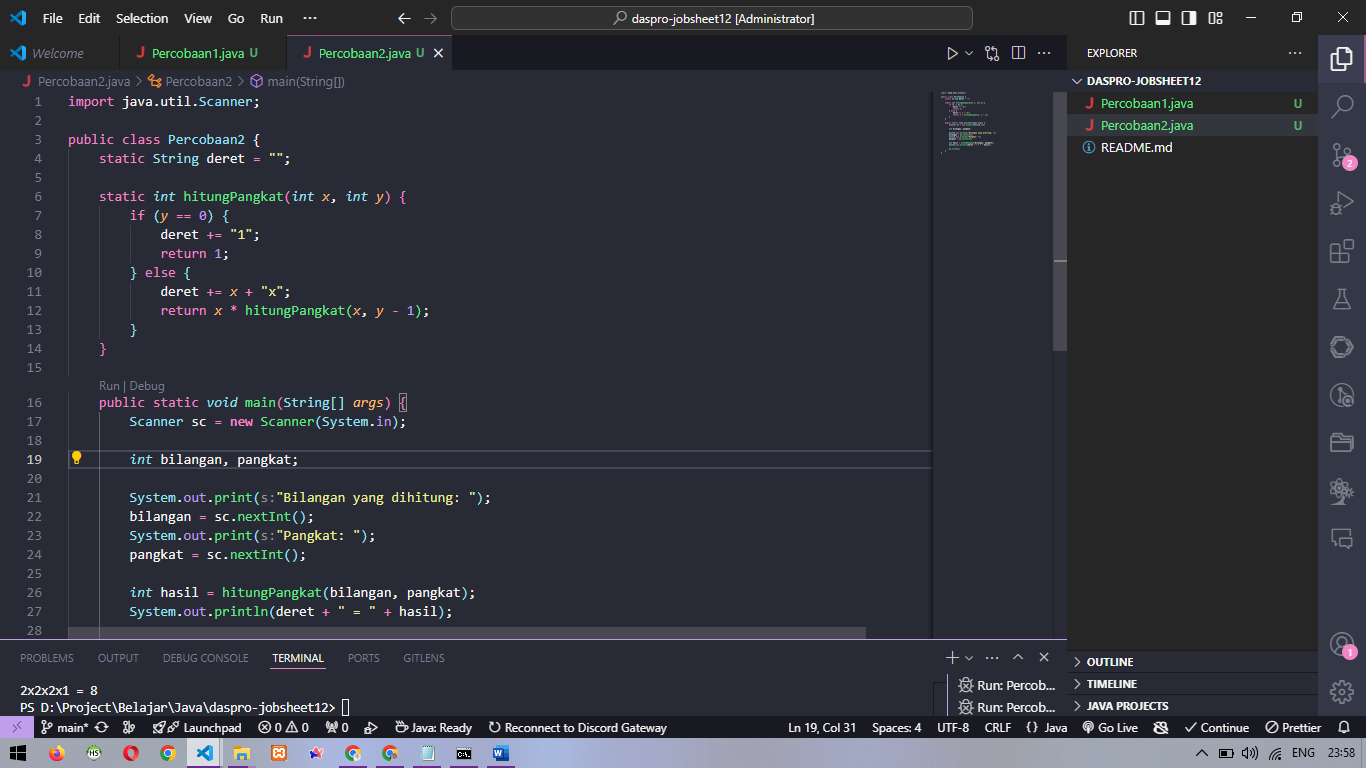


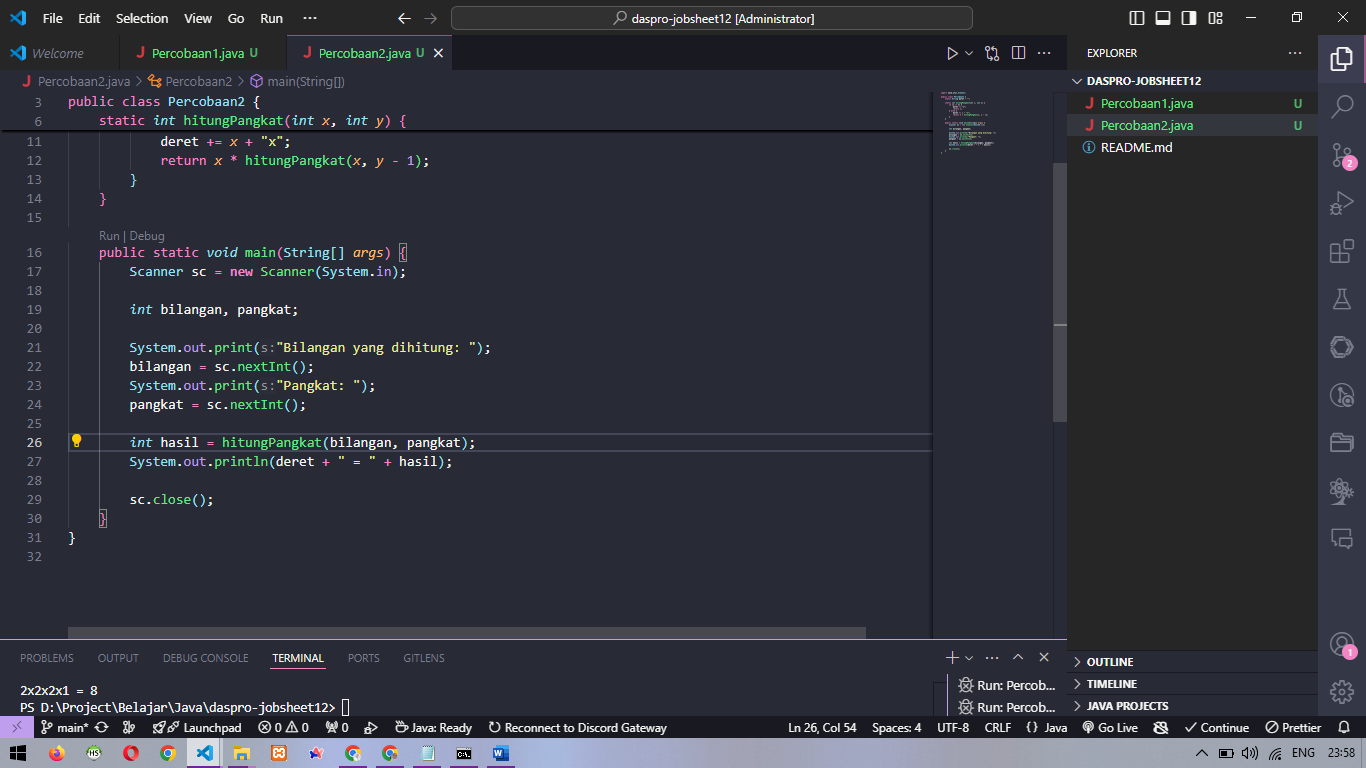
1. Tambahkan kodingan di main



**Pertanyaan 2**

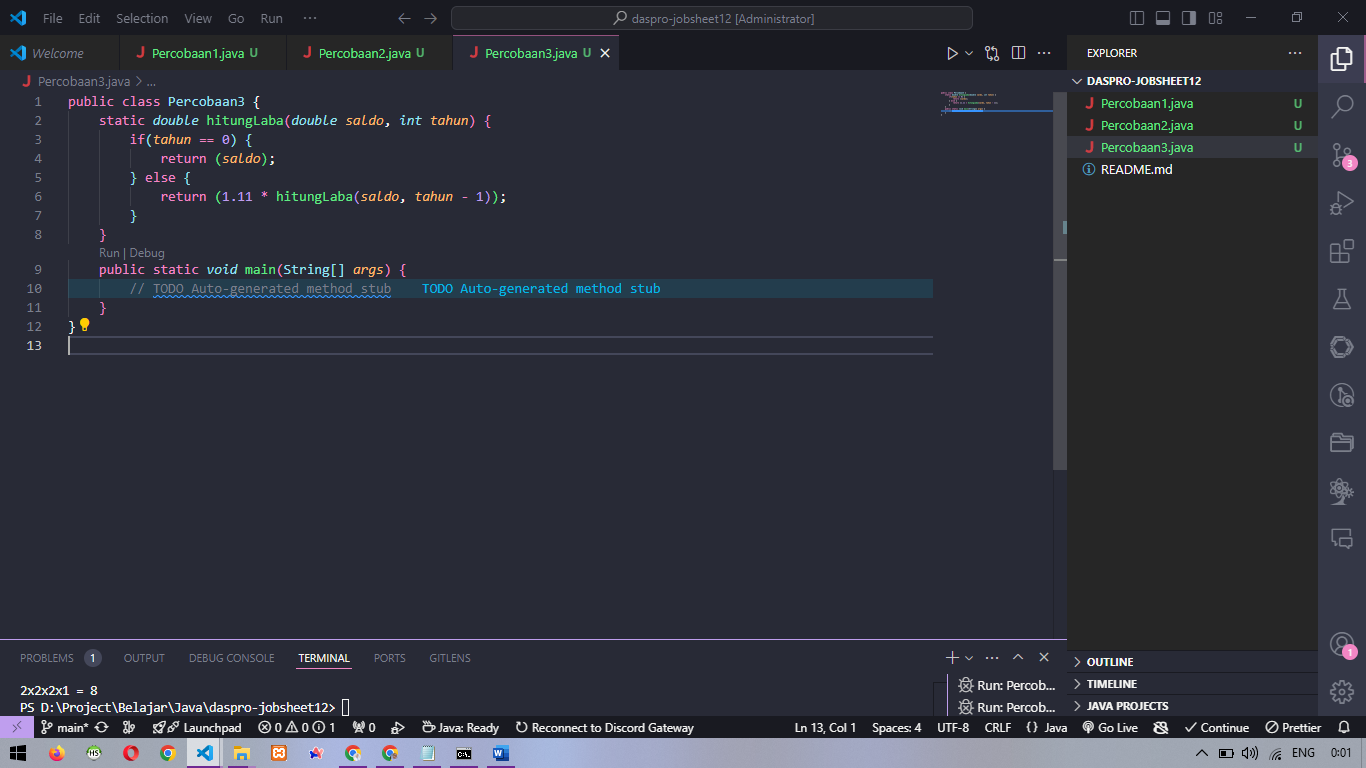
1. Pada program Percobaan2, fungsi hitungPangkat(x, y) dipanggil secara rekursif hingga mencapai kondisi base case, yaitu saat y == 0. Pada titik ini, fungsi berhenti memanggil dirinya sendiri dan mulai mengembalikan nilai hasil perhitungan.
2. Kode



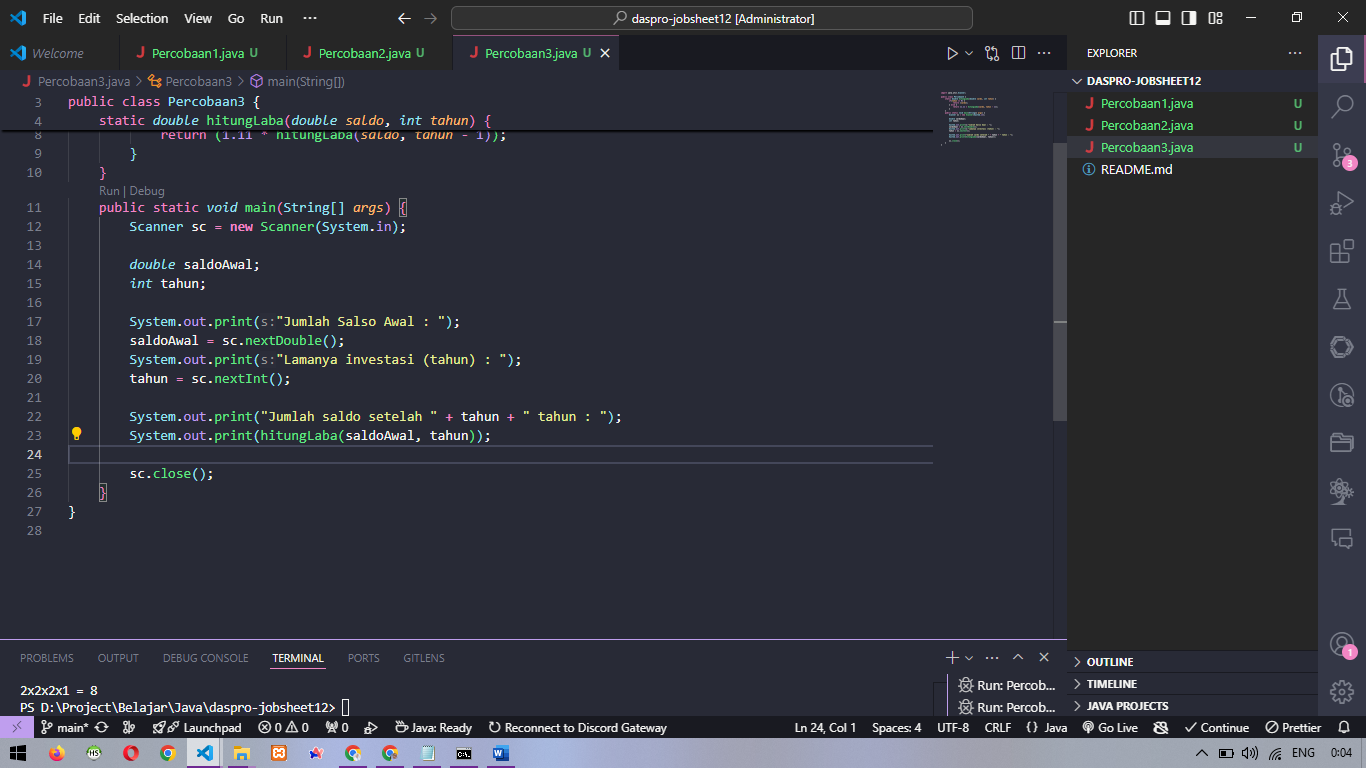


**Percobaan 3**

1. Buat file Percobaan3.java dan fungsi hitungLaba



1. Tambah kodingan ini di main



**Pertanyaan**

1. Base case nya

if (tahun == 0) {

return (saldo);

}

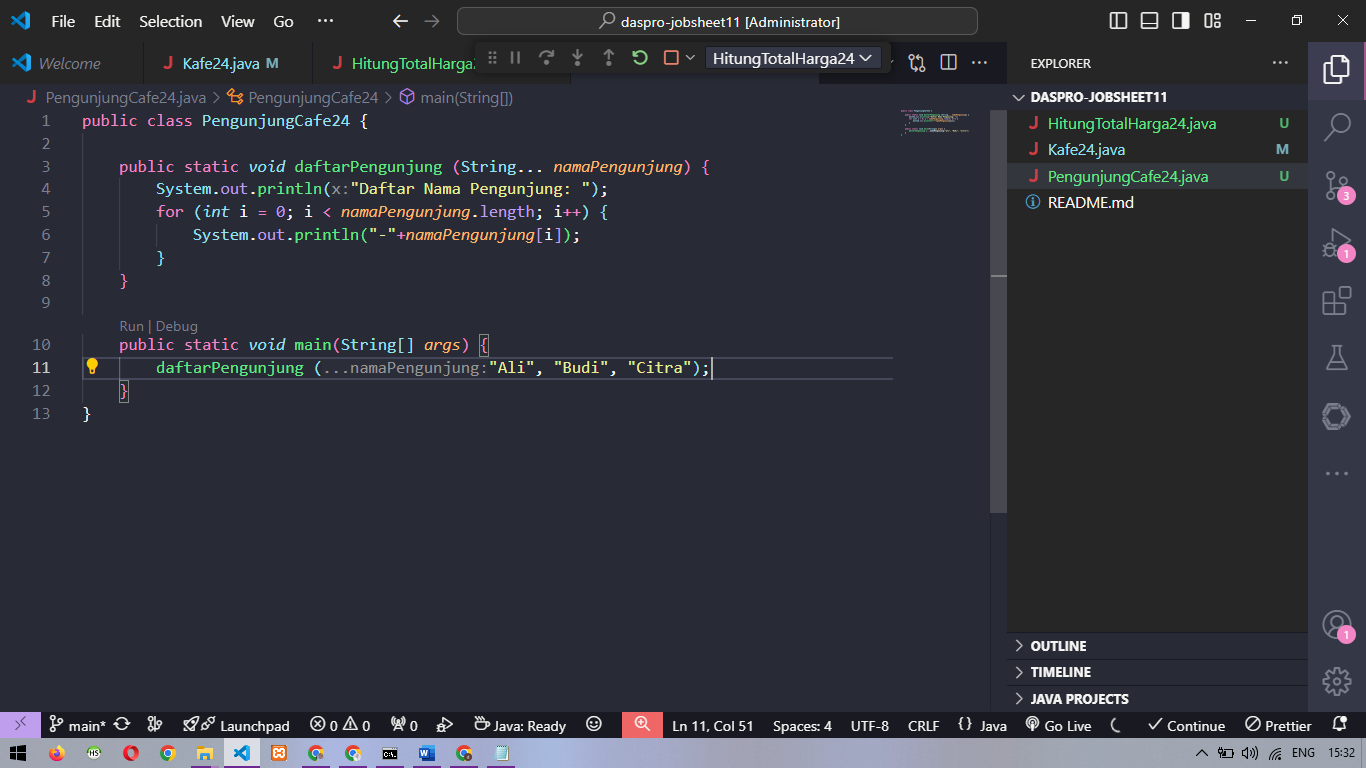
Recursion

return (1.11 \* hitungLaba(saldo, tahun - 1));

1. Fase Ekspansi (Memecah Masalah)
2. 1.11 x hitungLaba(100000, 3) = 136761
3. 1.11 x hitungLaba(100000, 2) = 123210
4. 1.11 x hitungLaba(100000, 1) = 111000
5. hitungLaba(100000, 0) = 100000

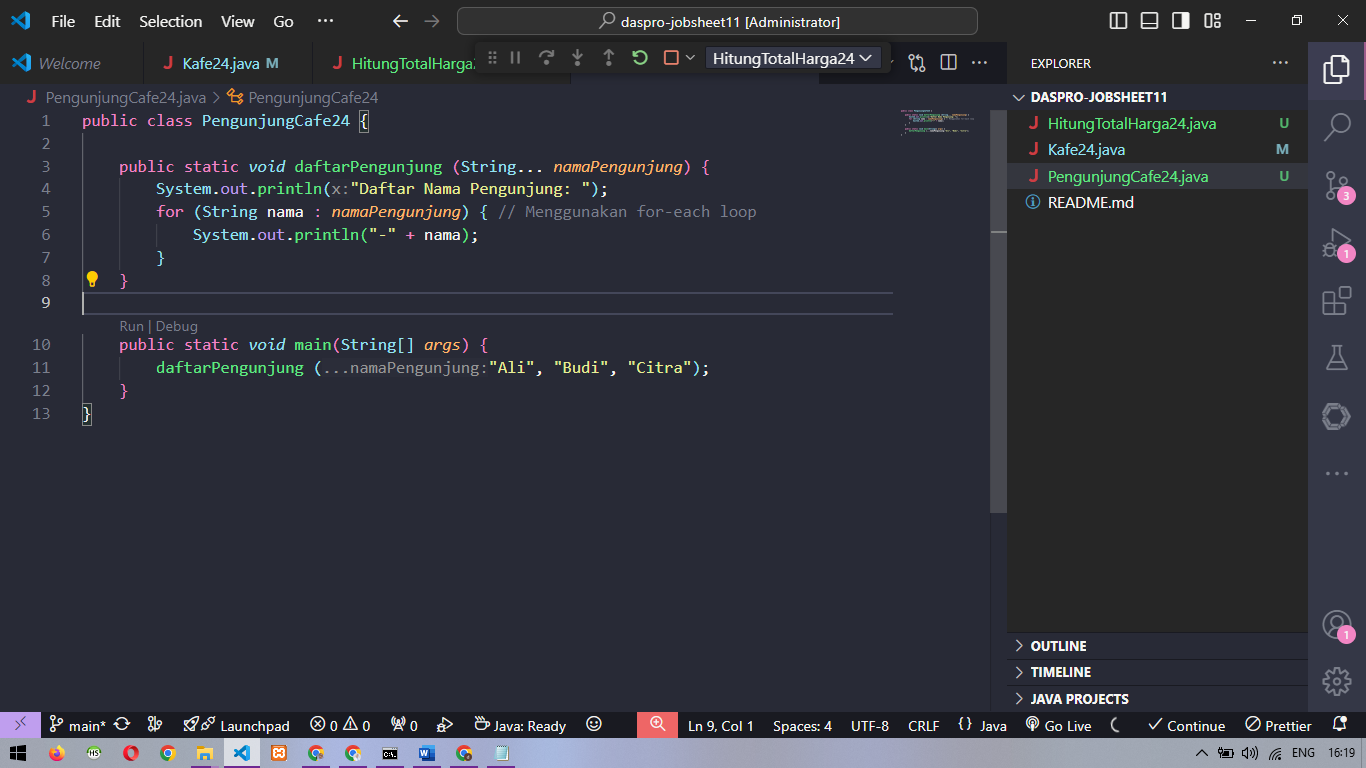
**Percobaan 4**

1. Buat file pengunjungKafeNo24.java, dan tambahkan kodingan berikut



**Pertanyaan**

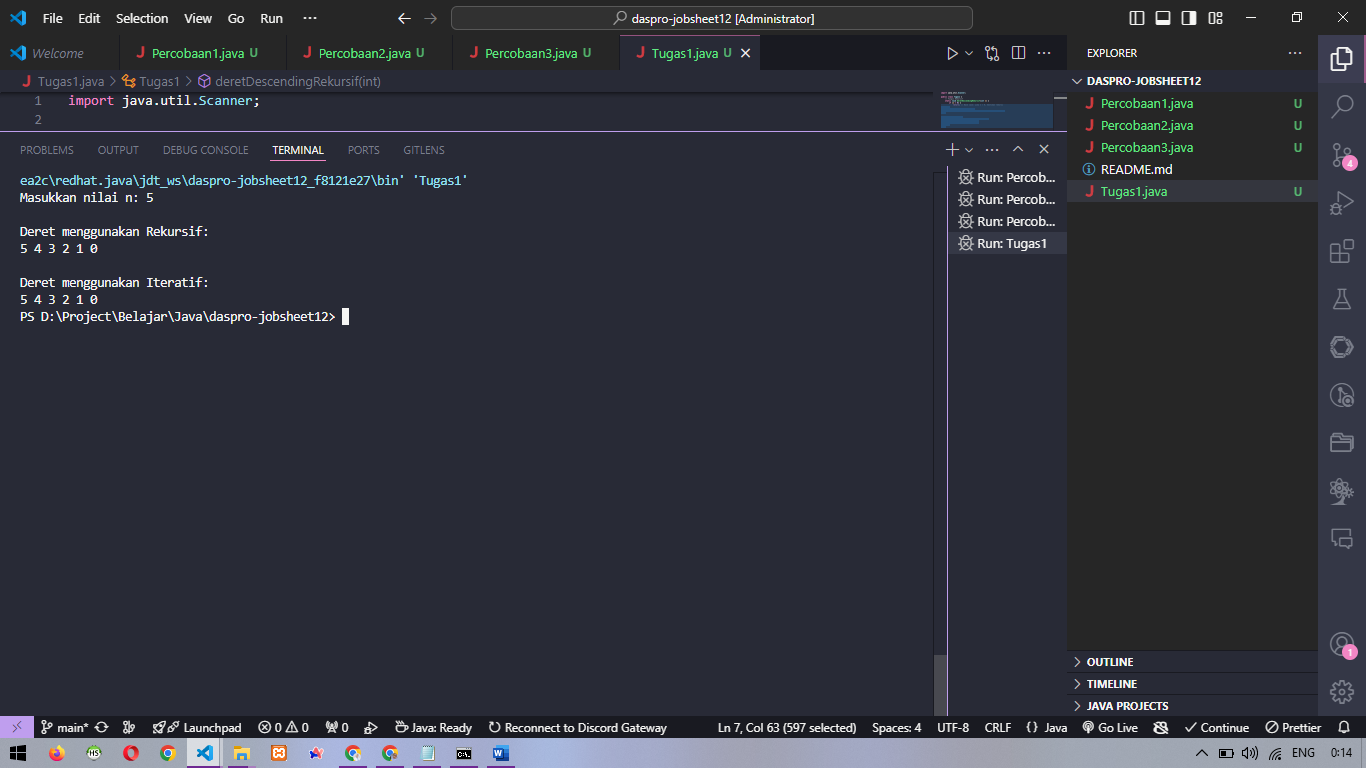
1. bertujuan untuk memungkinkan metode menerima **jumlah argumen yang tidak ditentukan sebelumnya**.
2. Ubah menjadi for each



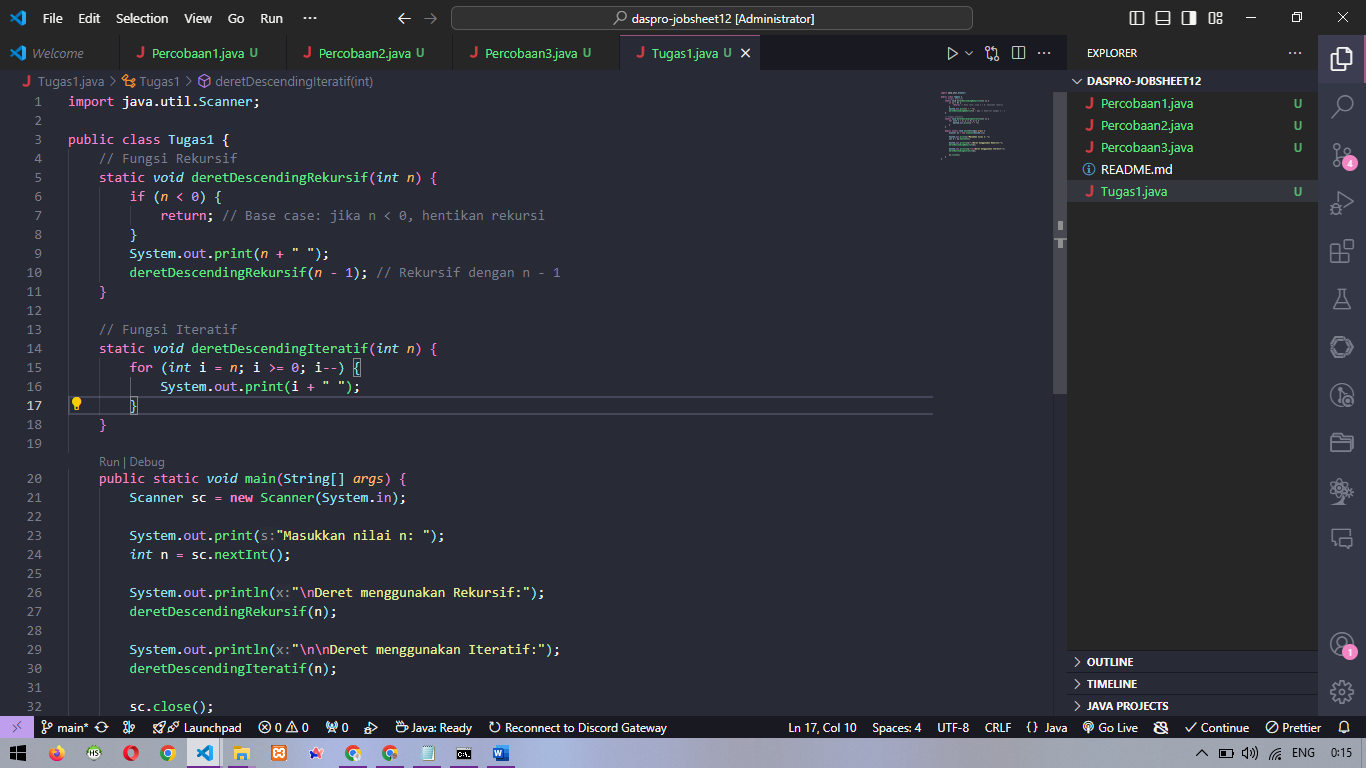
1. Tidak, tidak dapat menggunakan dua parameter varargs (**...**) dalam satu fungsi. Karena parameter varargs akan menangkap semua argumen yang sesuai, sehingga akan membingungkan compiler mengenai argumen mana yang harus masuk ke parameter varargs tertentu.

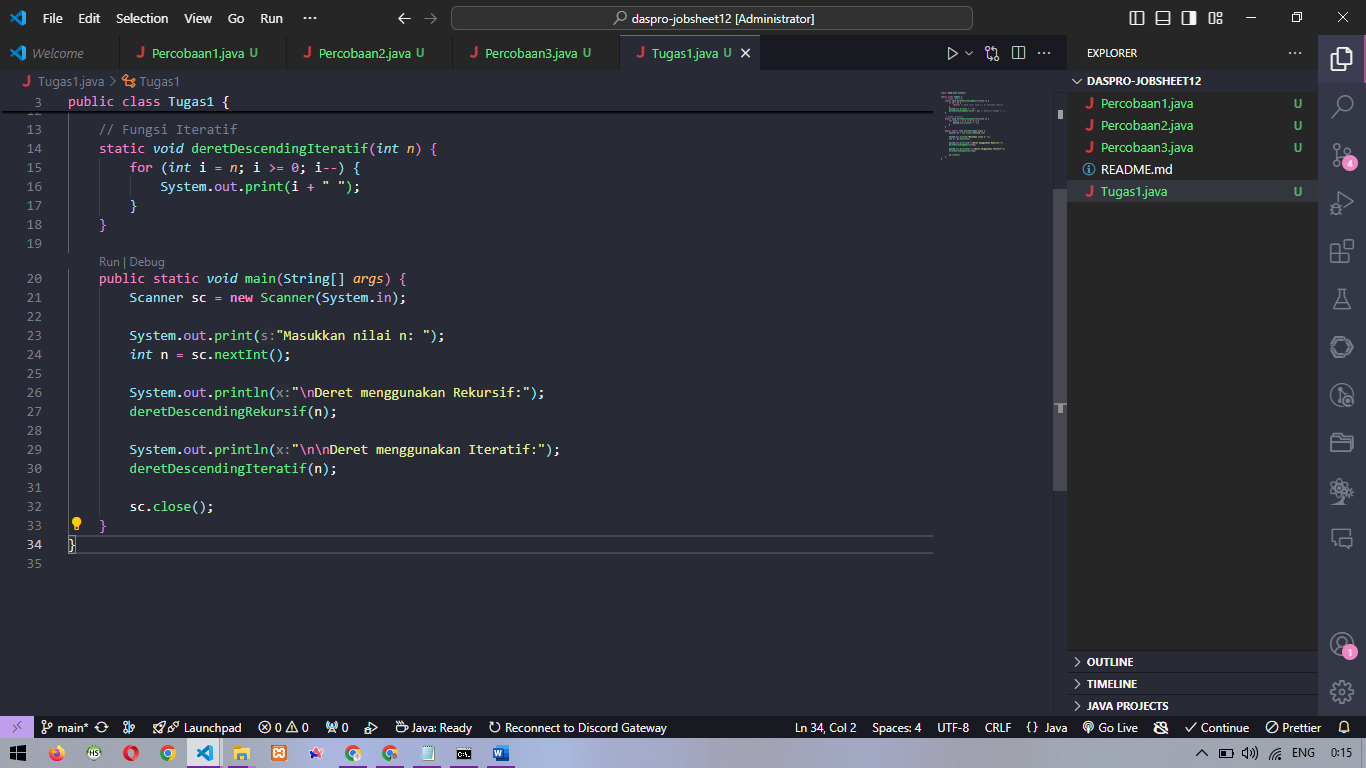
**Tugas**

1. Output

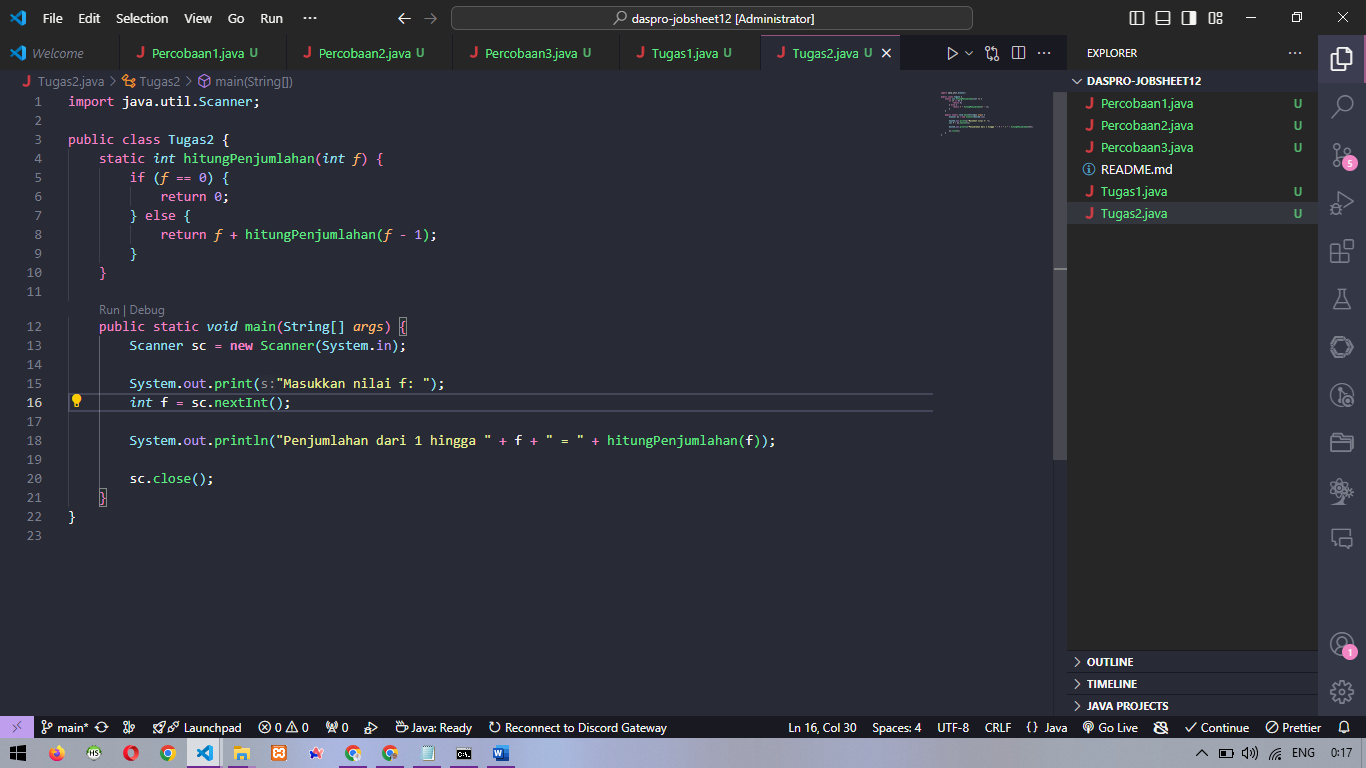


Code

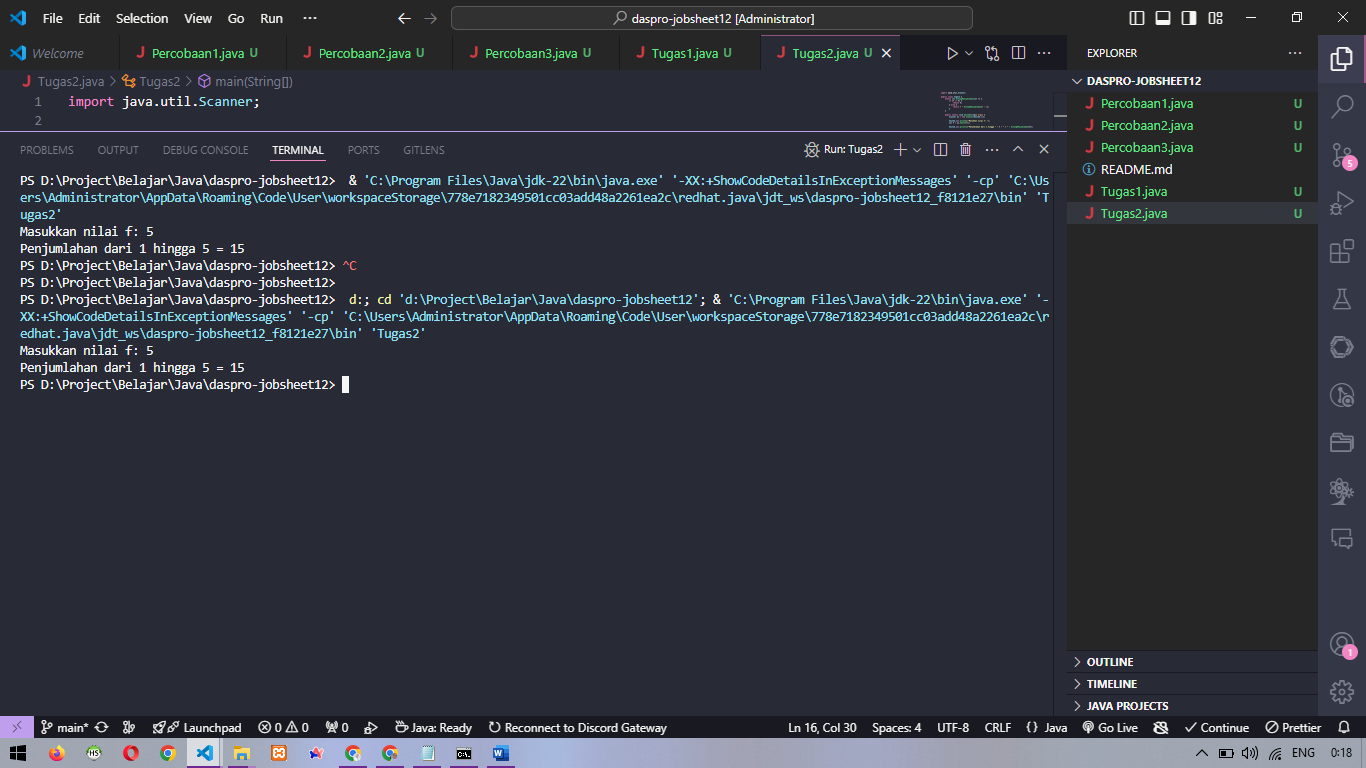




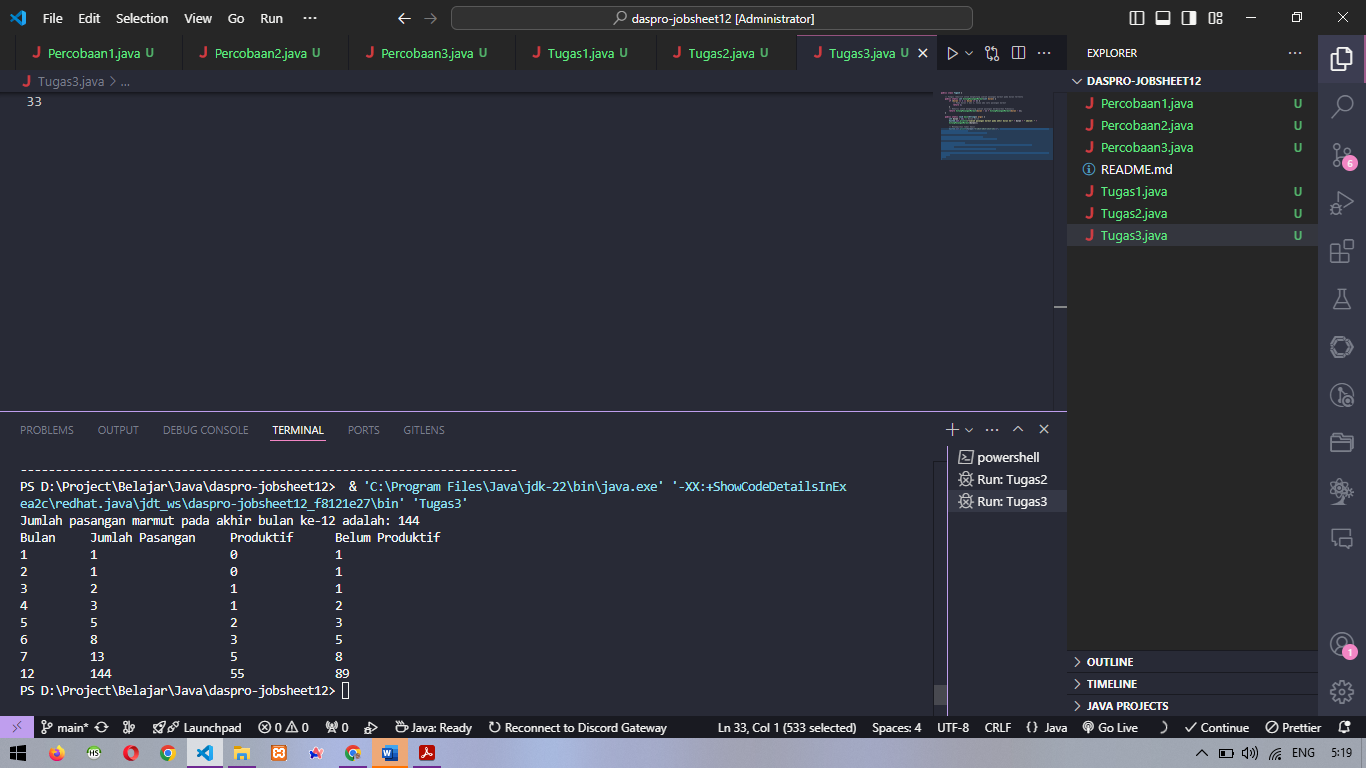
1. code



Ouput



1. Output



Kode

