**Laporan Praktikum Algoritma dan Struktur Data**

**Jobsheet 5 – Brute Force dan Divide Conquer**

**Dosen Pengampu : Ibu Triana Fatmawati, S.T., M.T**



**Nama : Annisa**

**Nim : 2341760032**

**Prodi : D-IV Sistem Informasi Bisnis**

**Absen : 04**

**JURUSAN TEKNOLOGI INFORMASI**

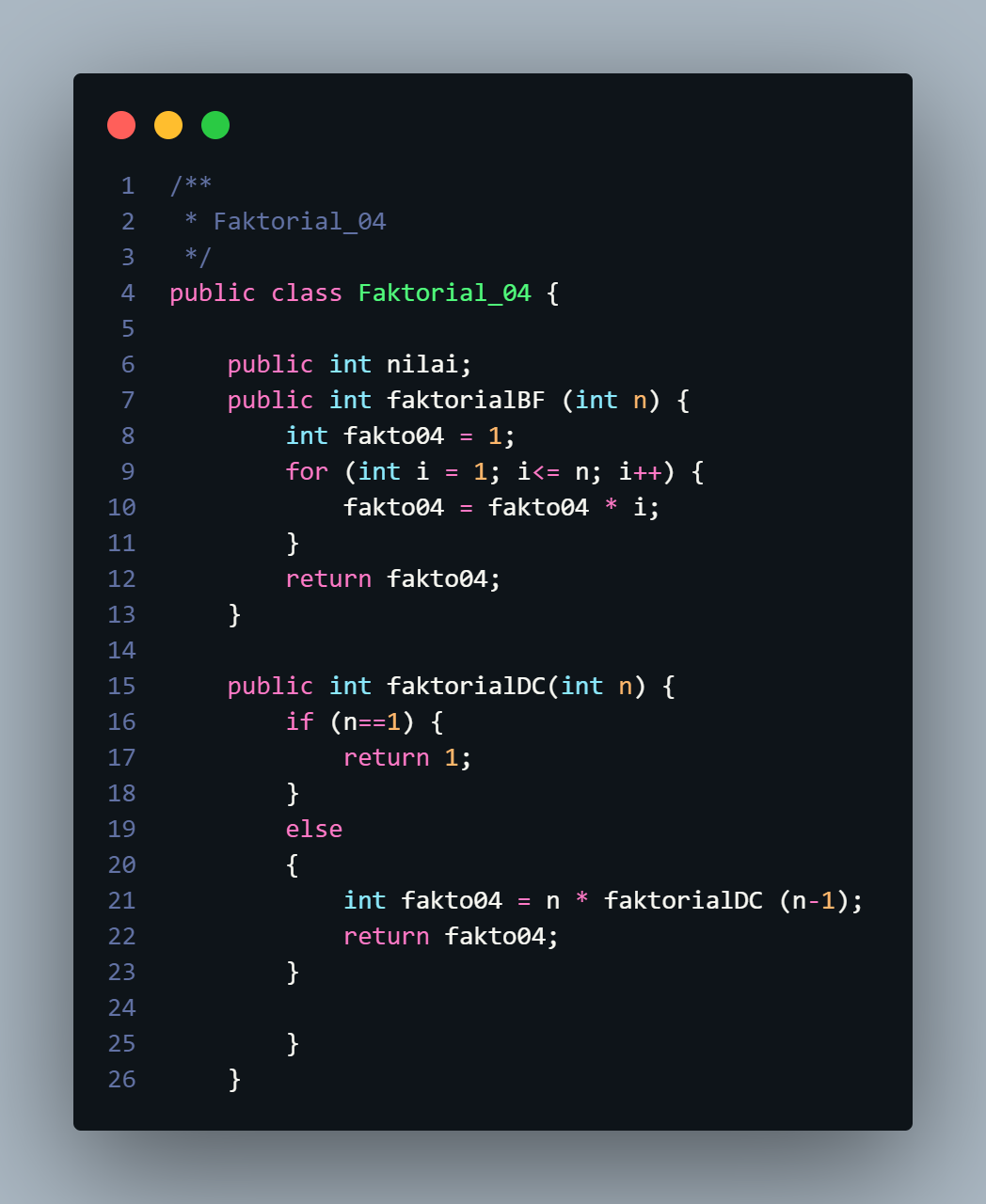
**POLITEKNIK NEGERI MALANG**

**2023/2024**

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

**4.2.1 Langkah-langkah Percobaan**

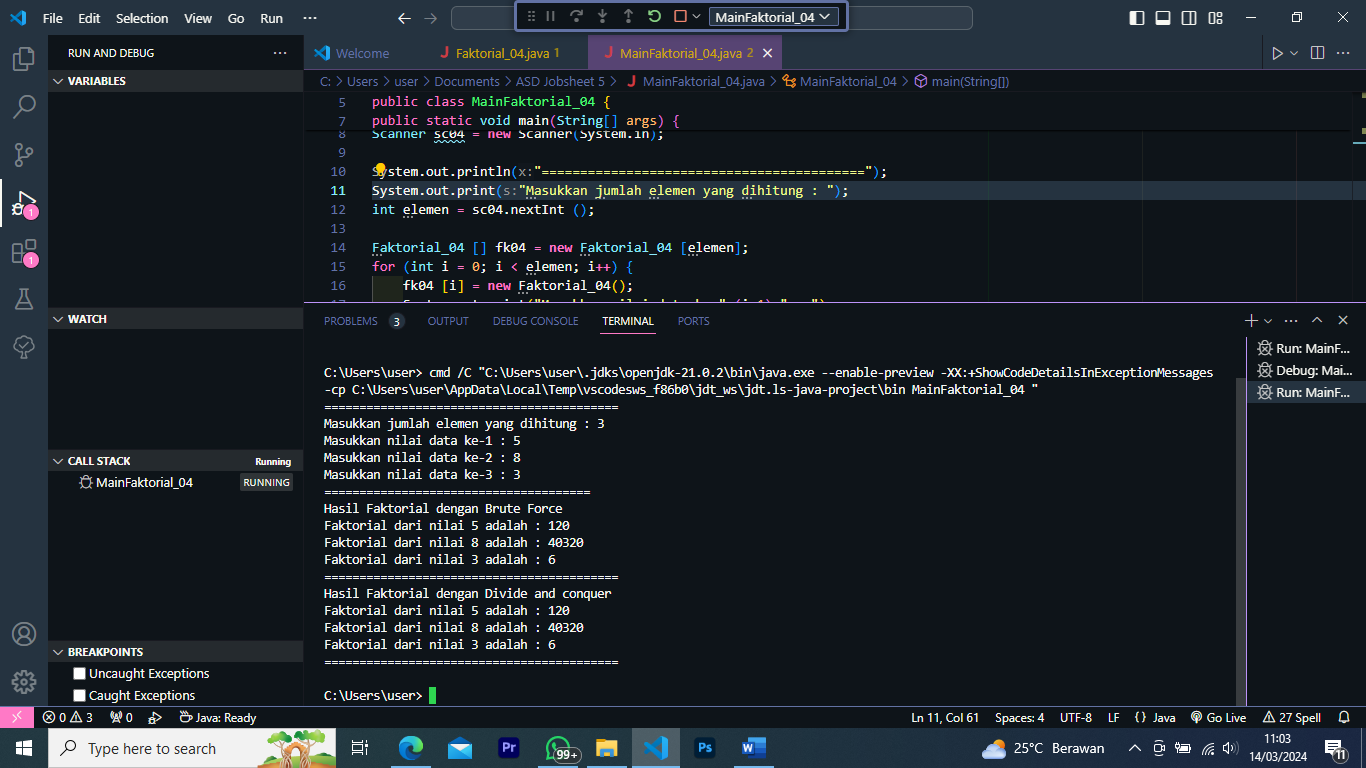
Berikut merupakan kode program dari class Faktorial

****

Berikut merupakan kode program dari class MainFaktorial



**4.2.2 Verifikasi Hasil Percobaan**



**4.2.3 Pertanyaan**

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

Base line pada algoritma Divide Conquer pada pencarian nilai factorial berada pada

line 15 di class Faktorial. Karena pada line tersebut merupakan sebuah tanda akhir

dari pemanggilan fungsi rekursif.

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

Lengkap. Dimana pembagiannya :

a. Divide ; pada if else, dimana setiap case akan dibagi

b. Conquer : untuk conquer pada sintask “faktorialDC(n-1)”

c. Combine : pada sintaks “int fakto = n \* faktorialDC(n-1)”

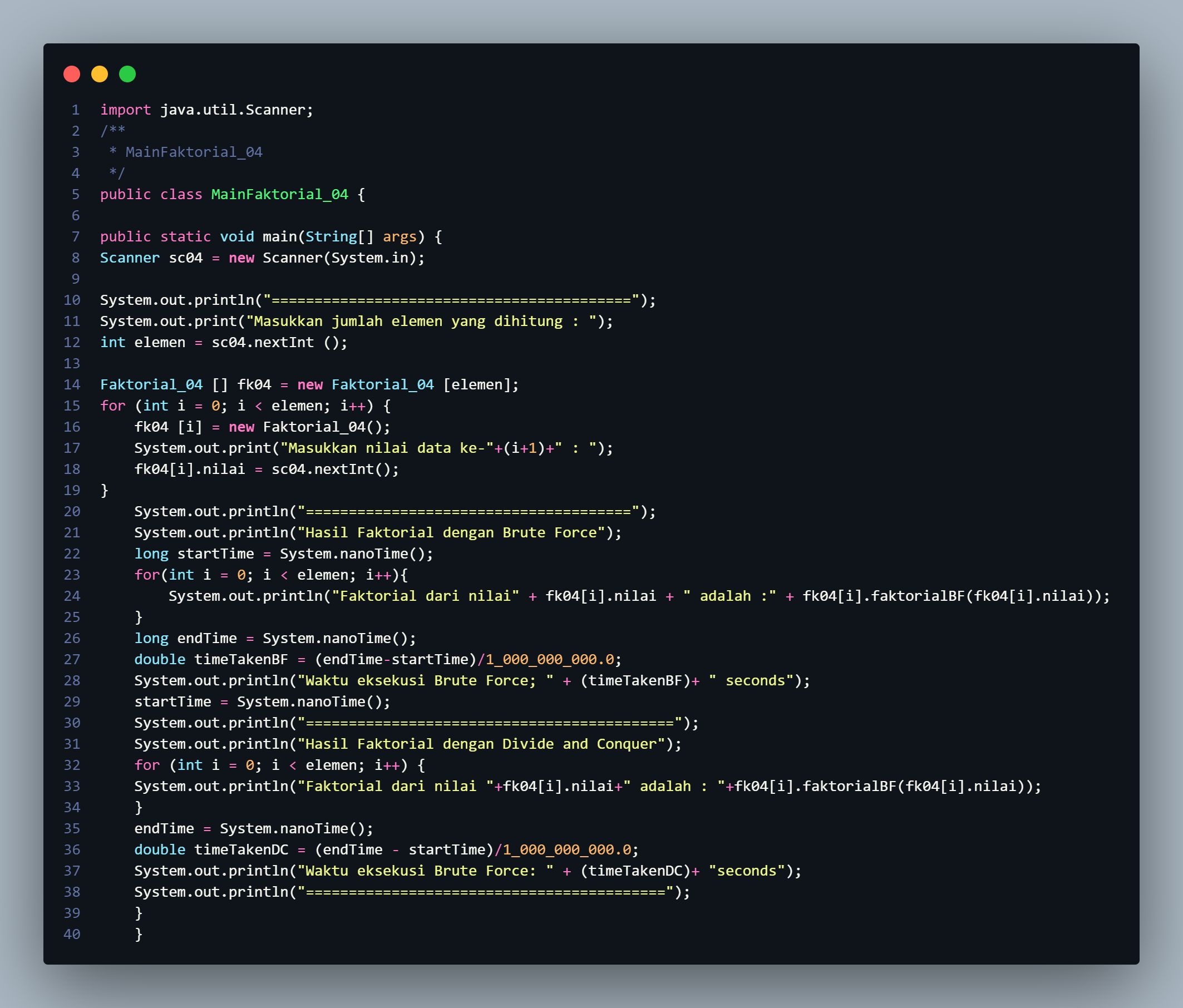
**3. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for?Buktikan!**

Bisa, berikut bukti screenshot dengan perulangan while

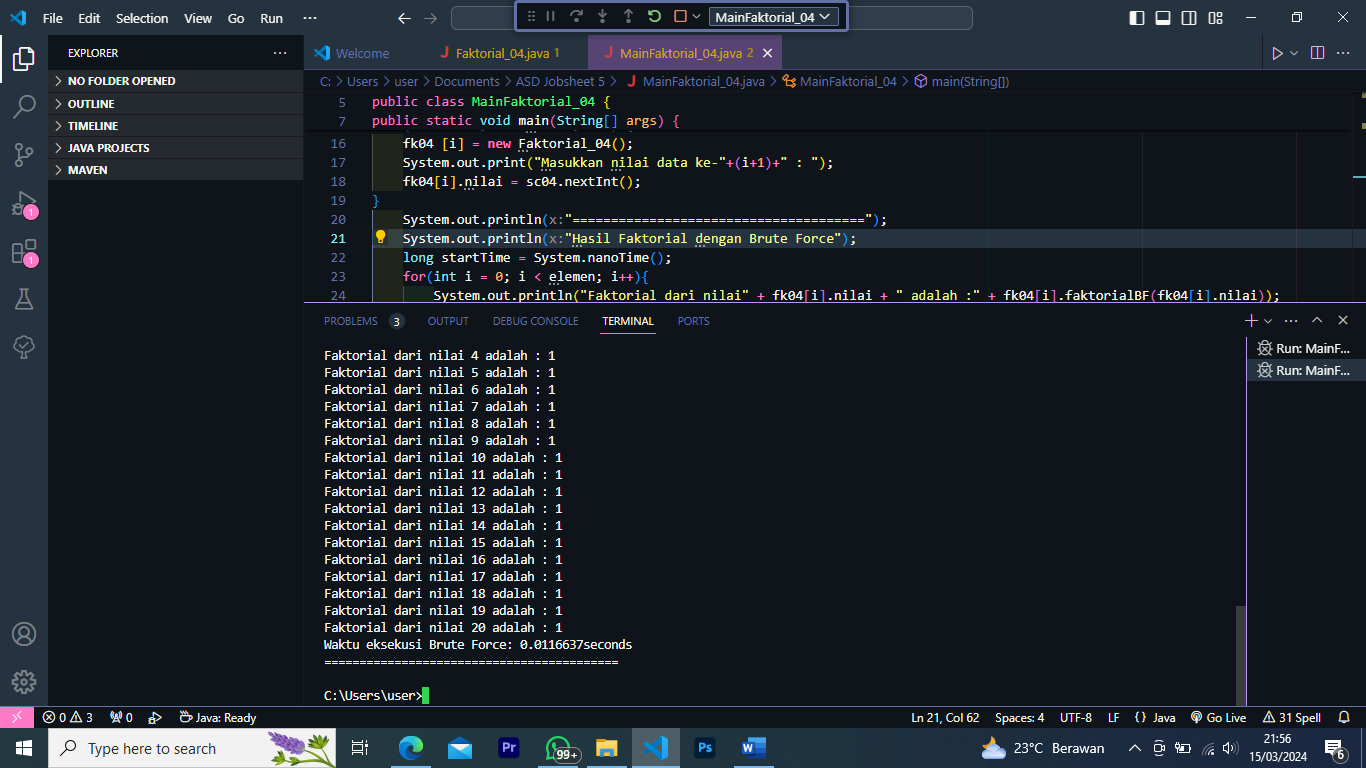


4. **Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!**

Berikut merupakan kode program yang sudah ditambah pengecekan waktu eksekusi

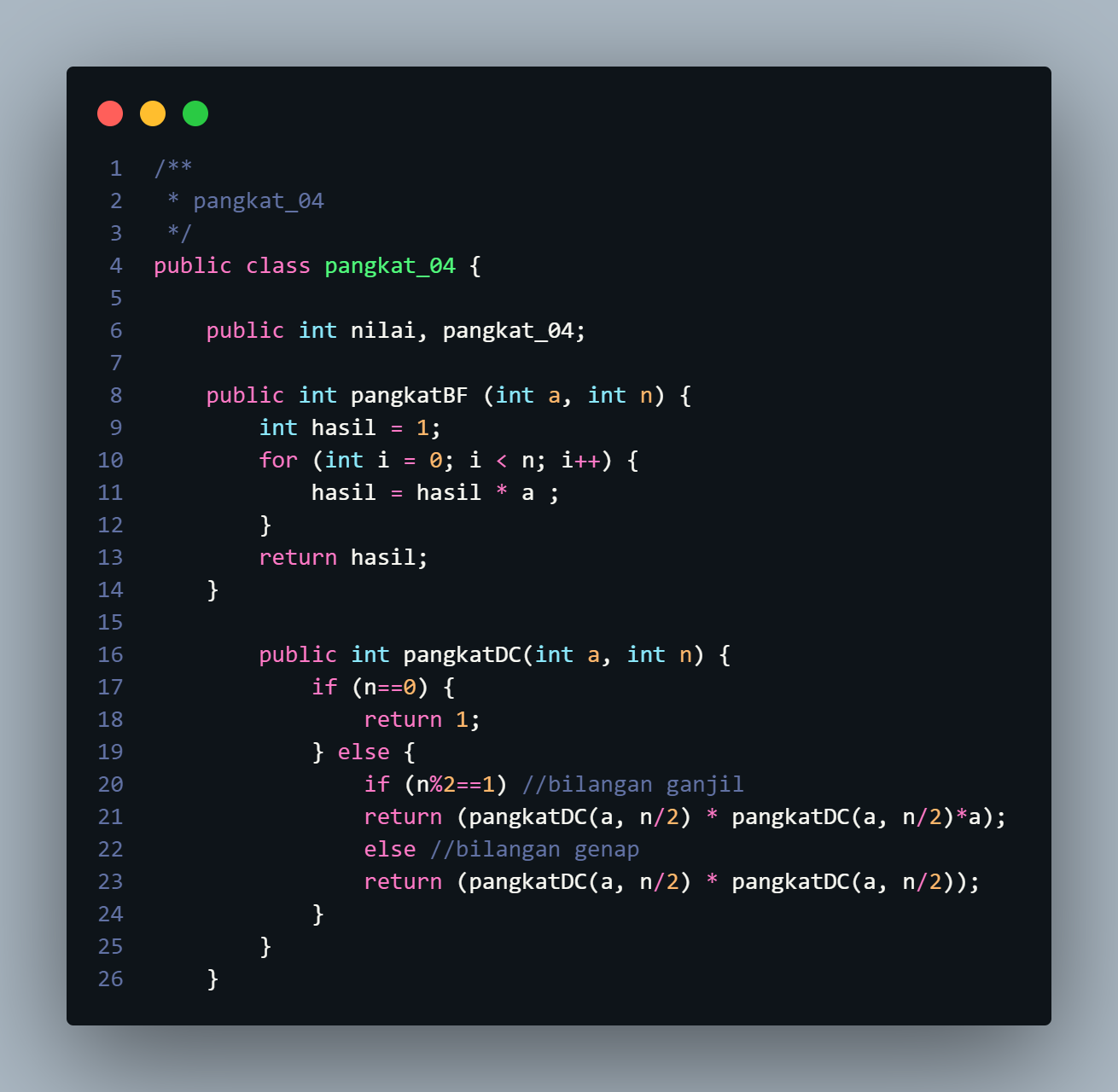
****

**5. Buktikan dengan inputan elemen yang di atas 20 angka, apakah ada perbedaan waktu eksekusi?**

****

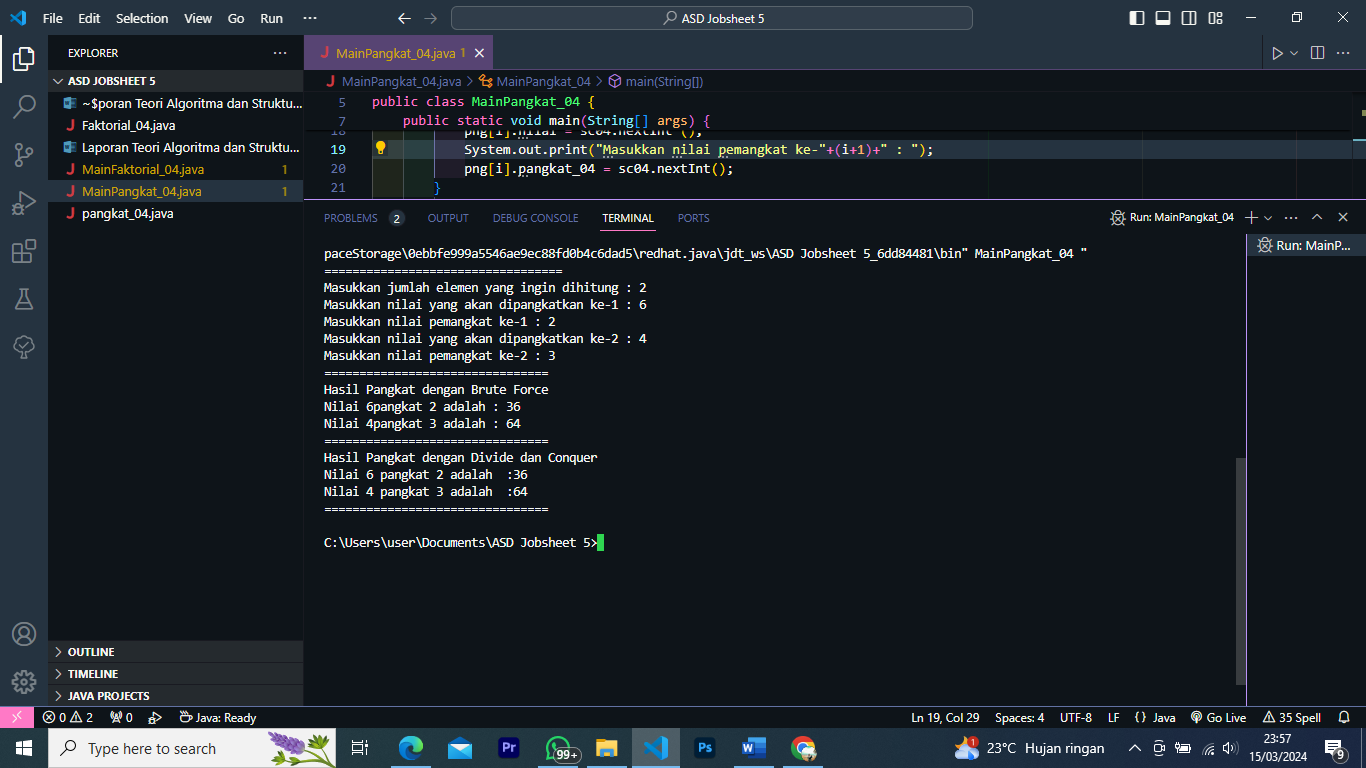
**4.3 Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer**

**4.3.1 Langkah-langkah Percobaan**

****

****

**4.3.2 Verifikasi Hasil Percobaan**

****

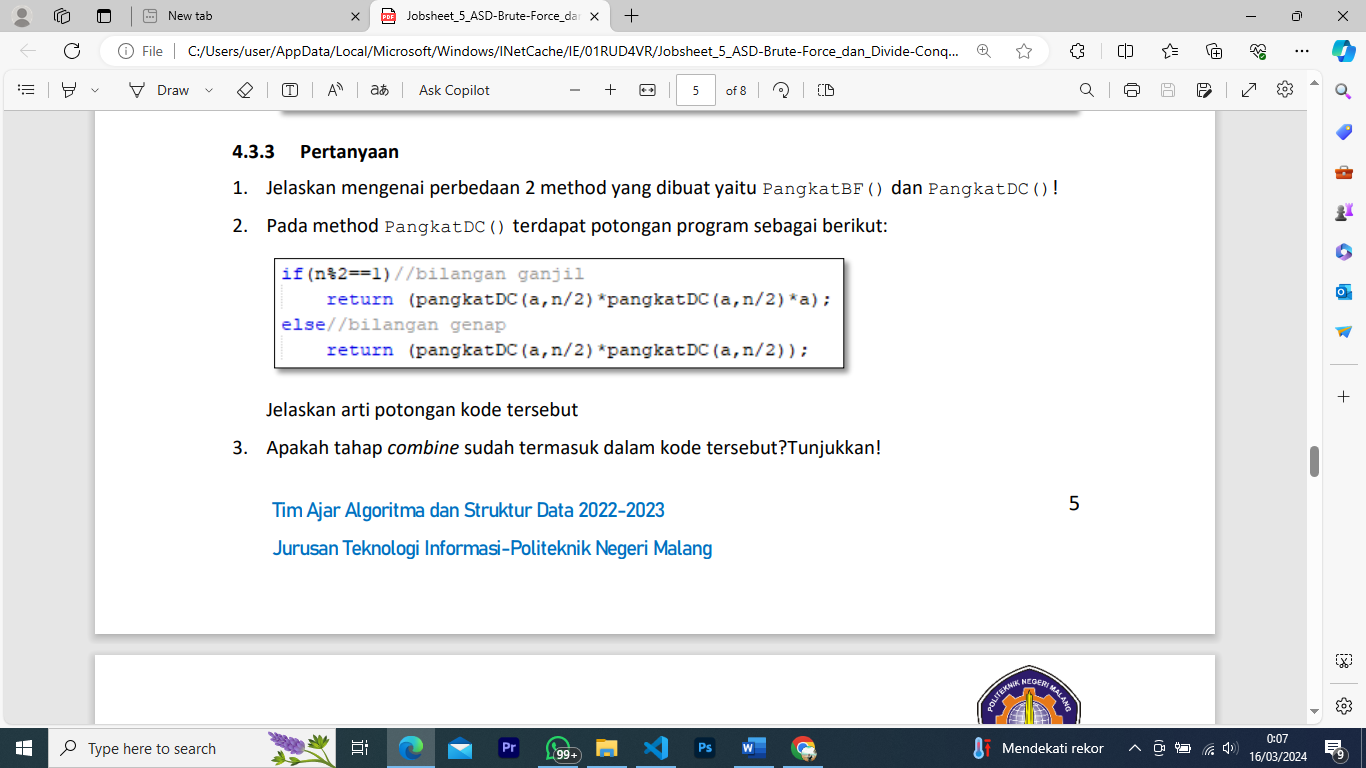
**4.3.3 Pertanyaan**

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

Method PangkatBF() merupakan method menggunakan perulangan kalkulasinya, sehingga ketika ada data yang lebih banyak durasi akan lebih lama.

Method PangkatDC() merupakan method yang menggunakan fungsi rekursif dengan melakukan pembagian case hingga menemukan solusi lalu dicombine sehingga mendapatkan solusi akhir yang lebih cepat dibandingkan dengan method PangkatBF().

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

****

**Jelaskan arti potongan kode tersebut**

Potongan kode tersebut akan dilakukan sebuah pembagian case dengan menggunakan if else ketika kondisi n%2==1 bernilai true maka program akan menjalankan statement pangkatDC (a, n/2) \* pangkatDC (a, n/2)\*a. Dimana statement tersebut merupakan solusi pertama ketika kondisi awal bernilai true, lalu akan dilakukan pemanggilan fungsi lagi jika kondisi n bernilai false dan menjalankan statement pangkatDC (a, n/2)\*pangkatDC(a, n/2) sebaga solusi ke-2 hingga terakhir. Pada n yang dimaksut pada pemanggian fungsi tersebut terdapat perbedaan nilai yang kiri dan kanan dimana n kiri akan dicari dulu lalu ke n kanan dengan nilai yang berbeda.

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

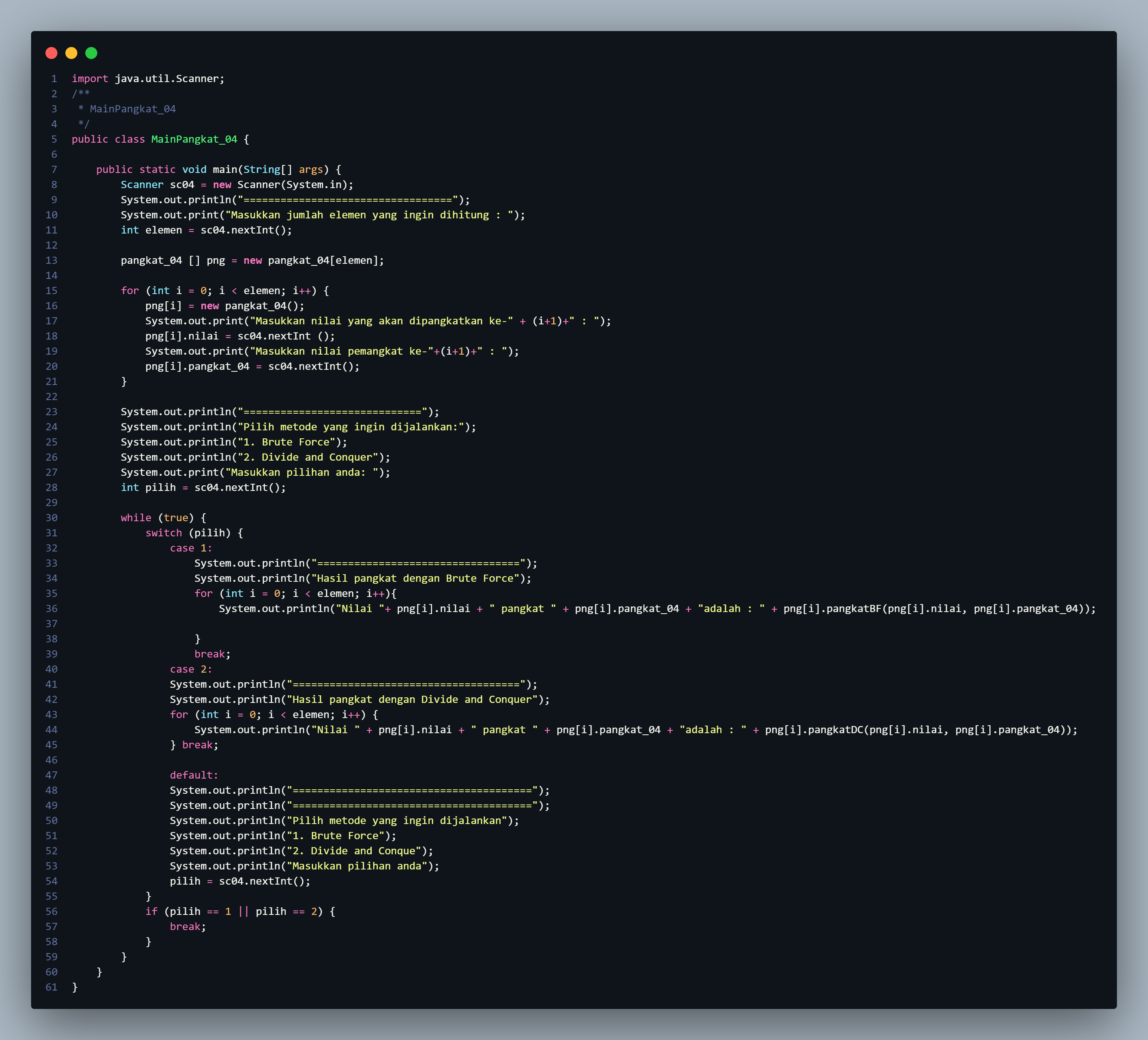
Sudah, tahap combine terdapat pada kode:  
return (pangkatDC (a, n/2)\*a);

return (pangkatDC (a, n)\*pangkatDC(a, n/2));

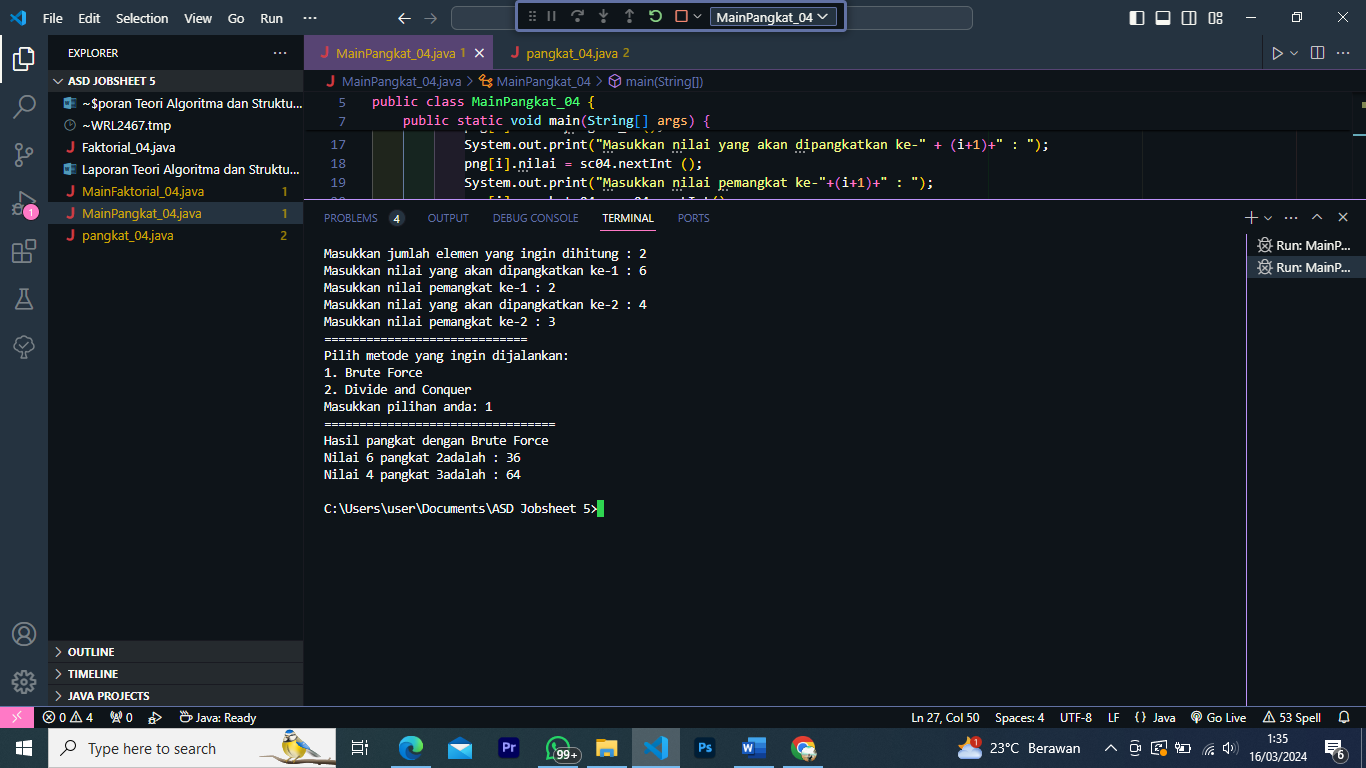
4**. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.**

Berikut kode program hasil modifikasi

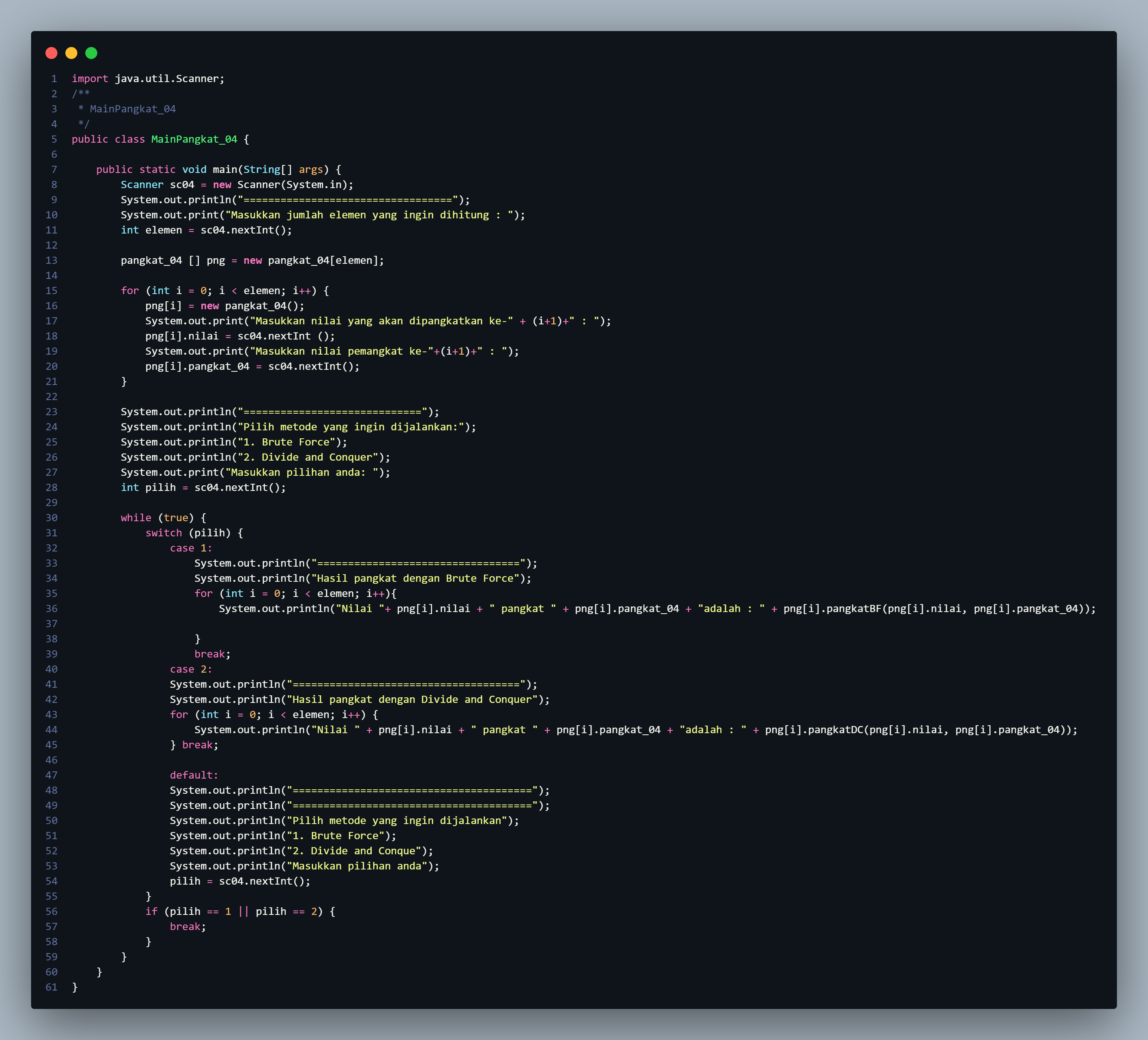
****

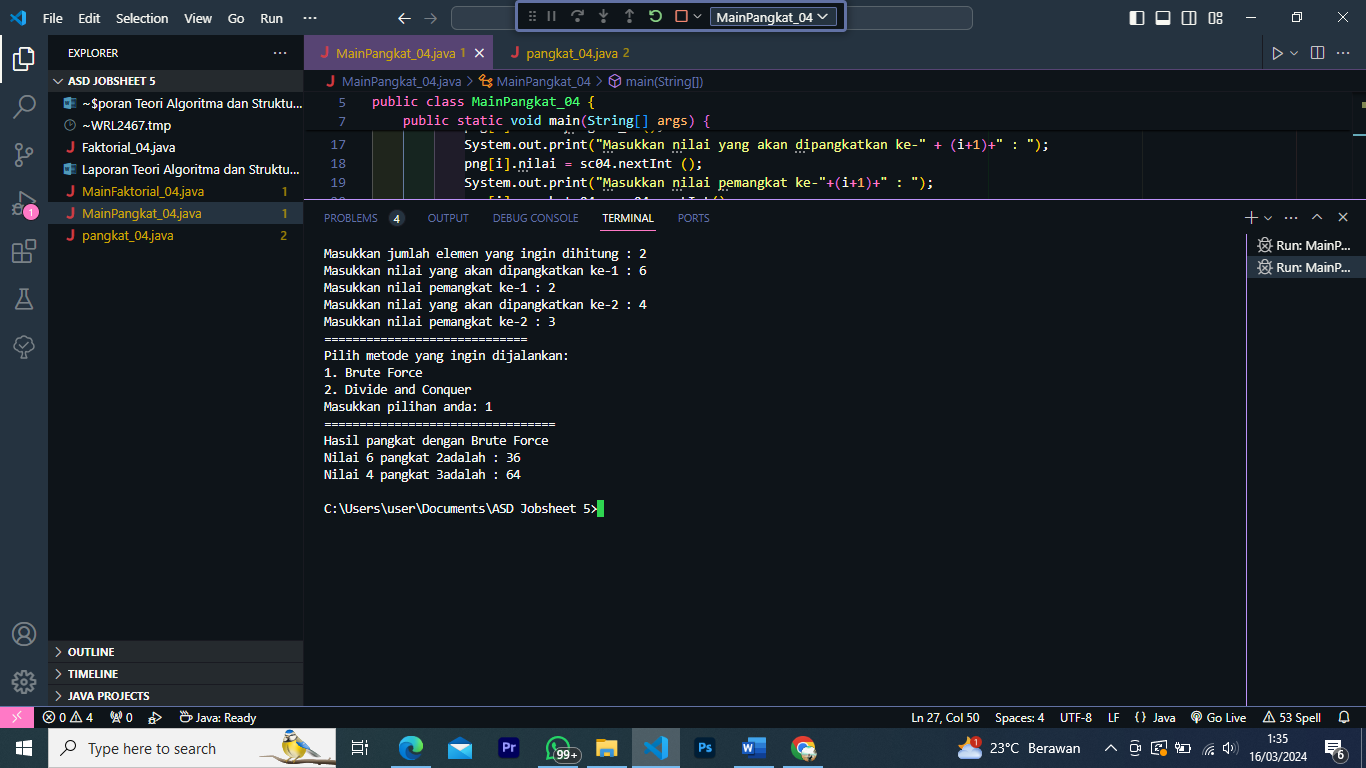
****

Berikut merupakan hasil running dari kode program di atas

****

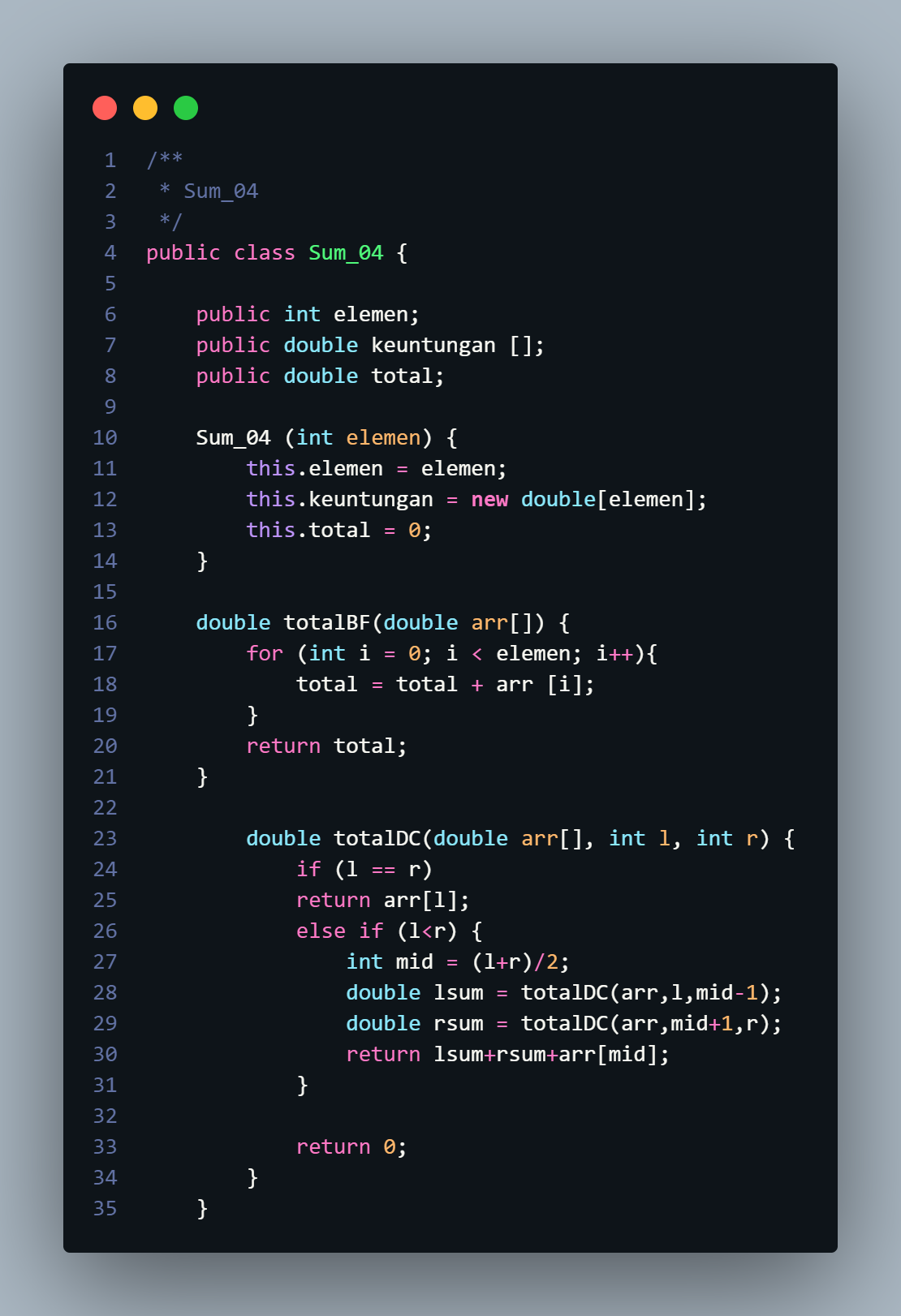
**5. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan!**

****

****

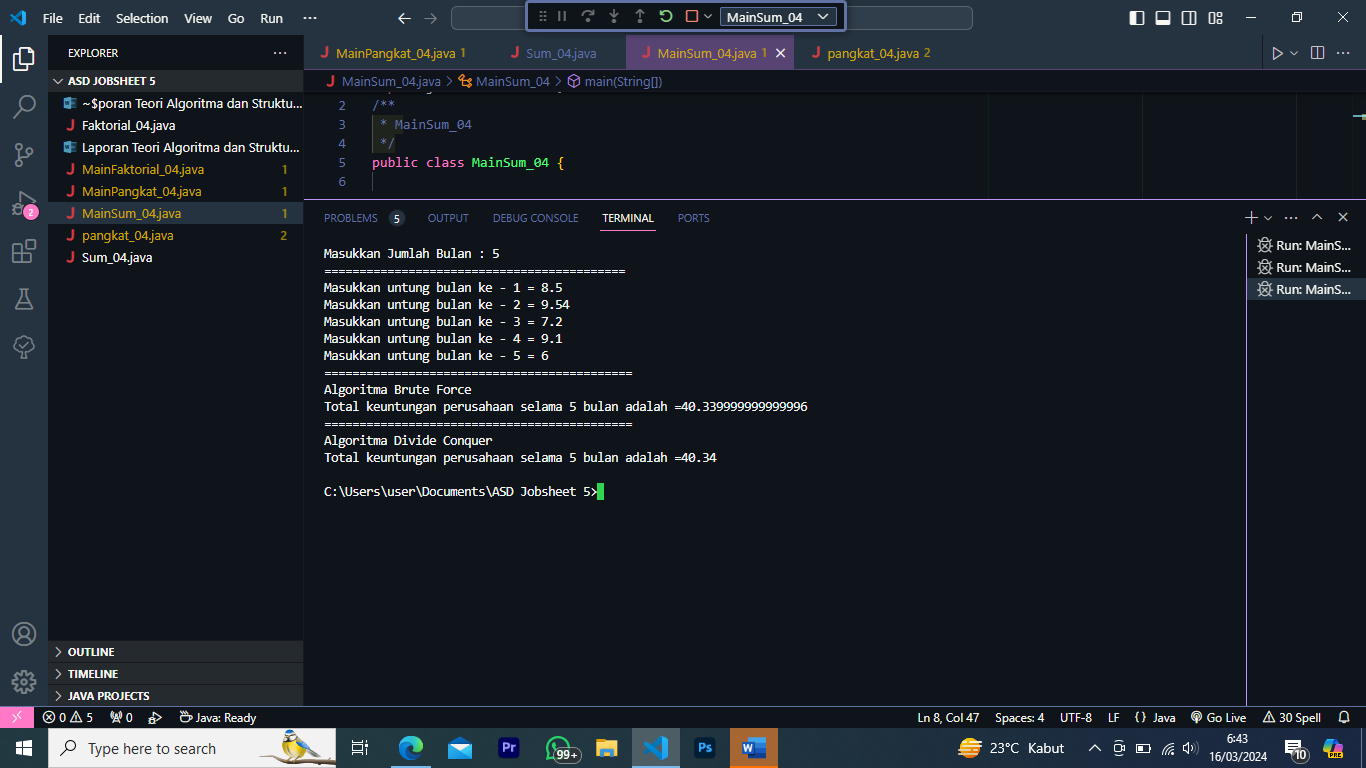
**4.4 Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer**

**4.4.1 Langkah-langkah Percobaan**

****

****

**4.4.2 Verifikasi Hasil Percobaan**

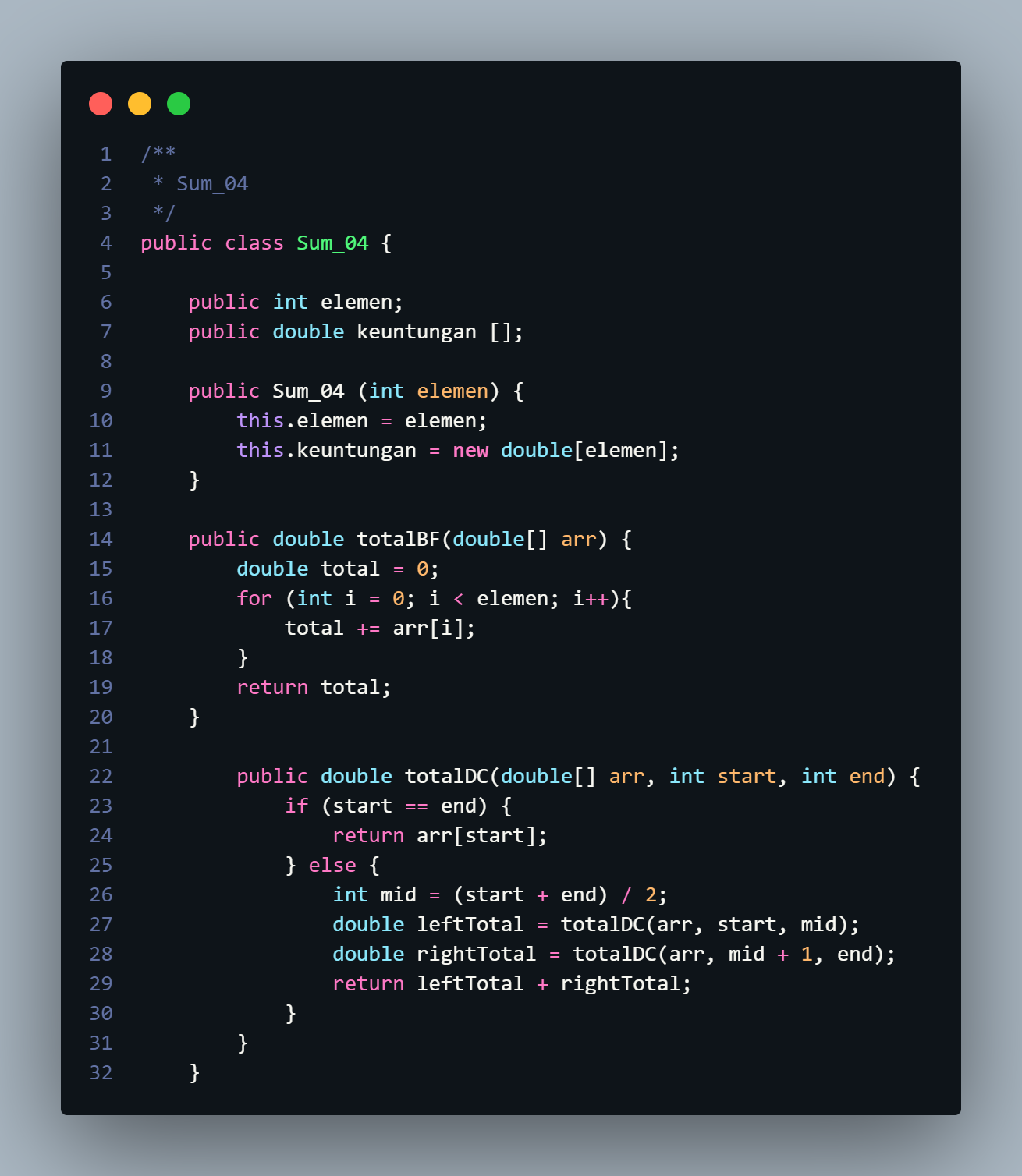
****

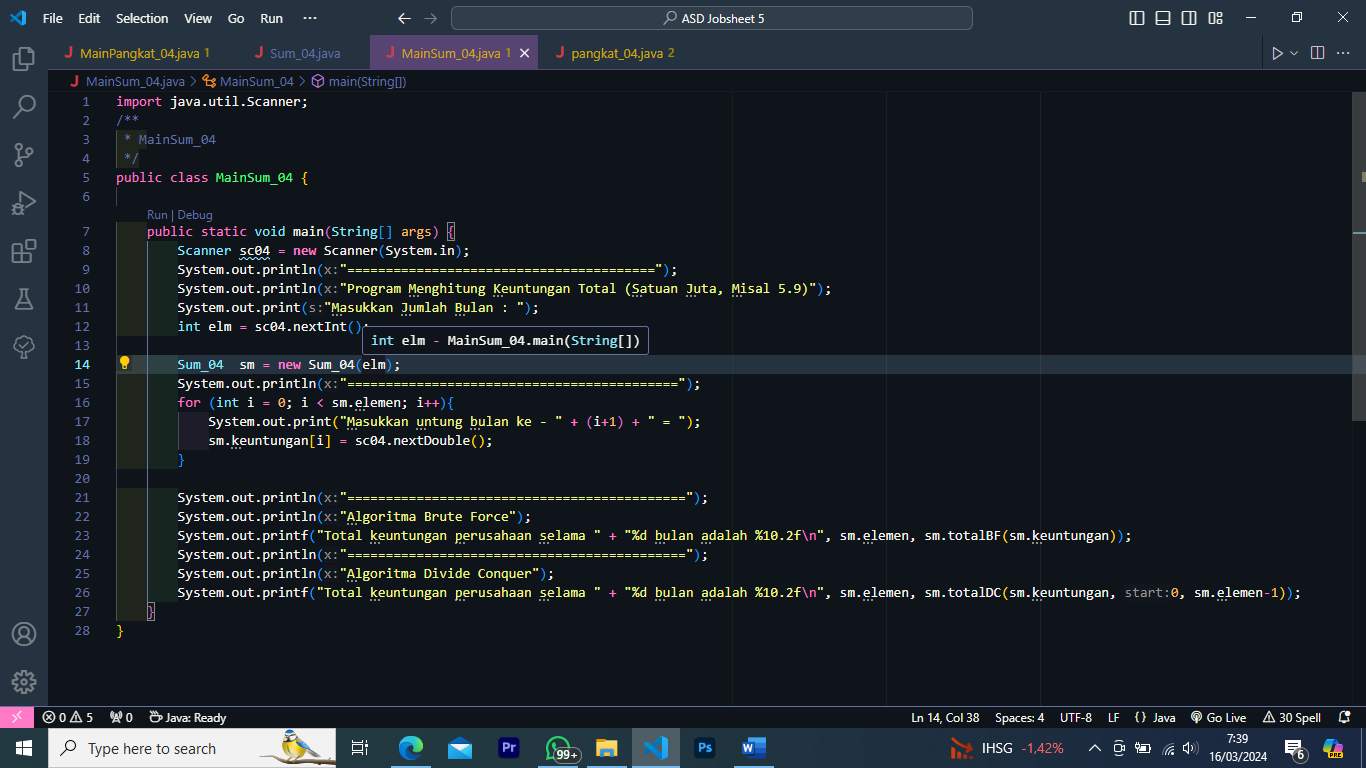
**4.4.3 Pertanyaan**

**1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method TotalBF() ataupun TotalDC()**

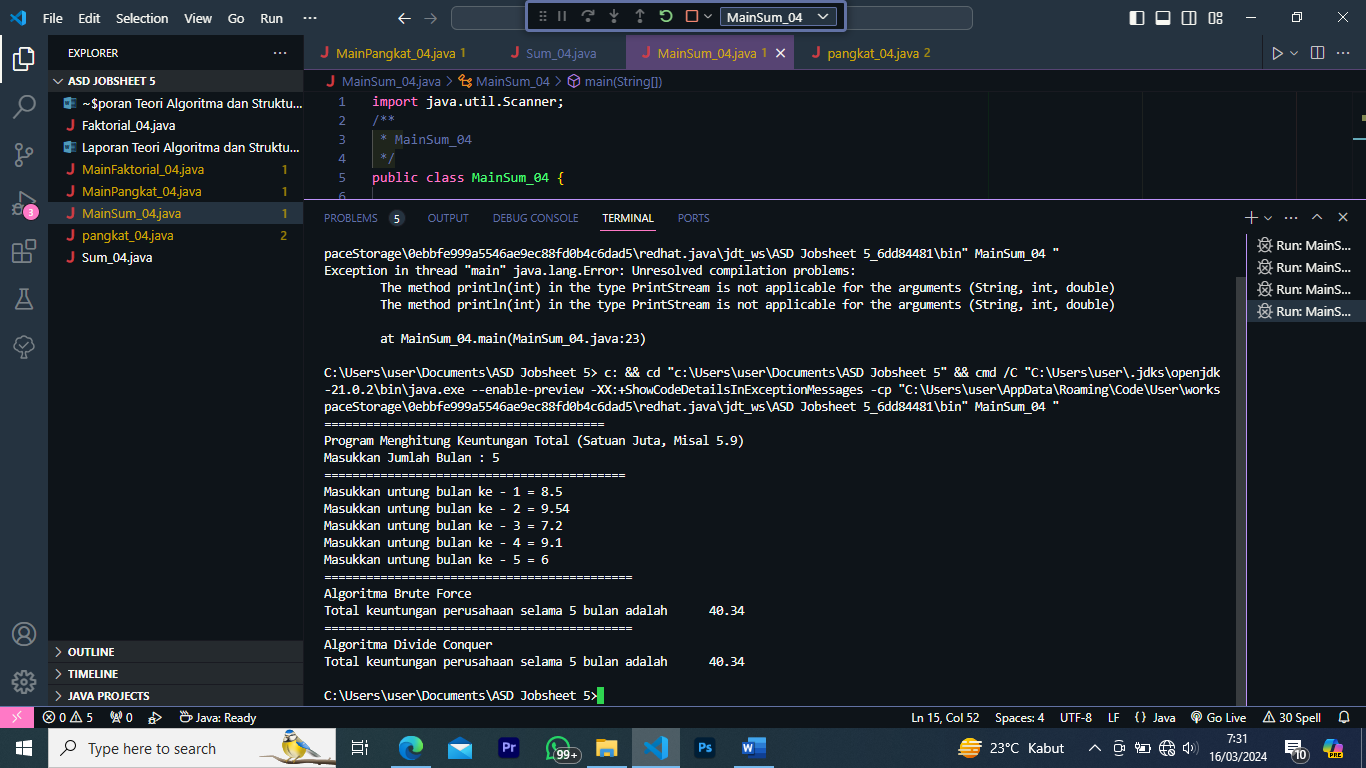
Keuntungan menghitung dengan menggunakan method TotalBF() adalah lebih mudah untuk menghitung kalkulasi dan dapat mempermudah membuat program karena tidak terlalu sulit untuk merancang programnya. Sedangkan method TotalDC() lebih cepat dalam proses kalkulasi ketika dijalankan.

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.**

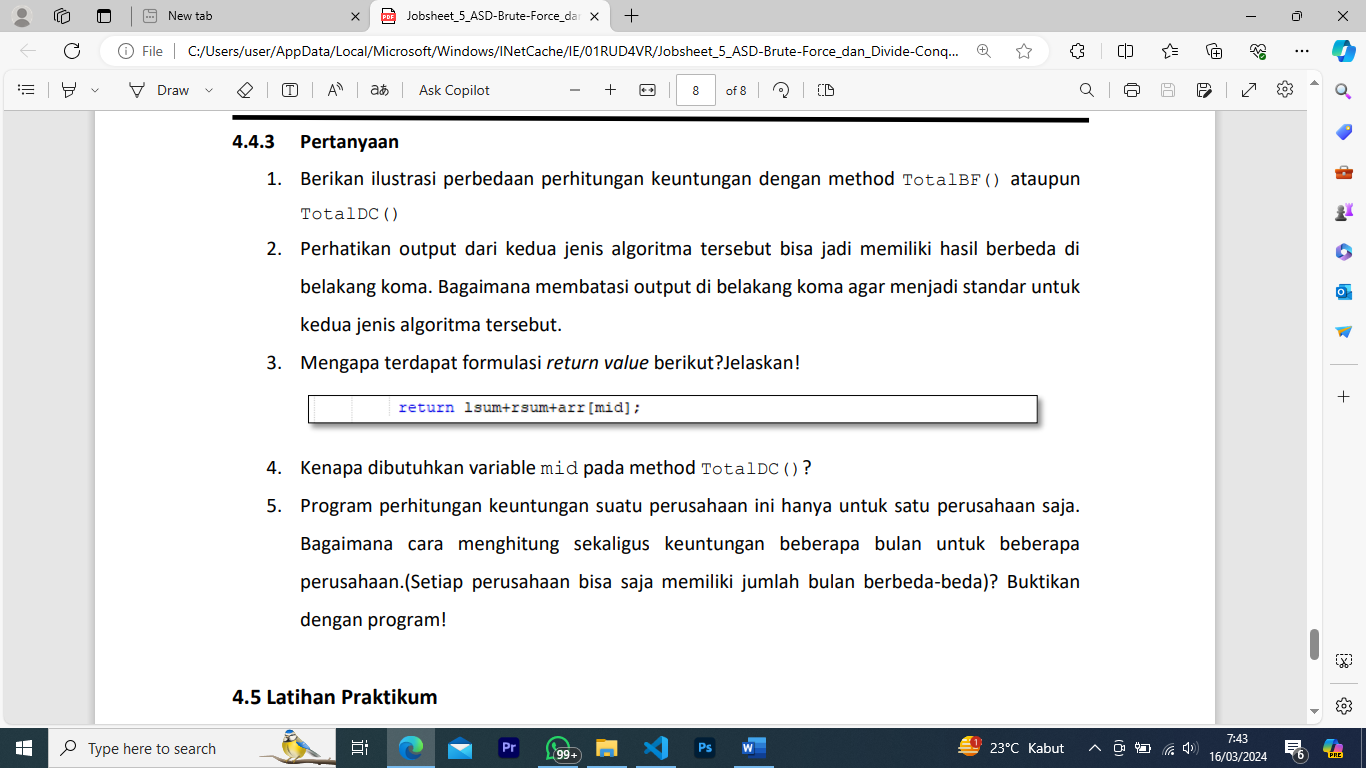




**(Maaf bu untuk screenshot nya tidak sama karena codesnap nya tiba-tiba tidak bisa🙏🏻)**



**3. Mengapa terdapat formulasi return value berikut?Jelaskan!**

****

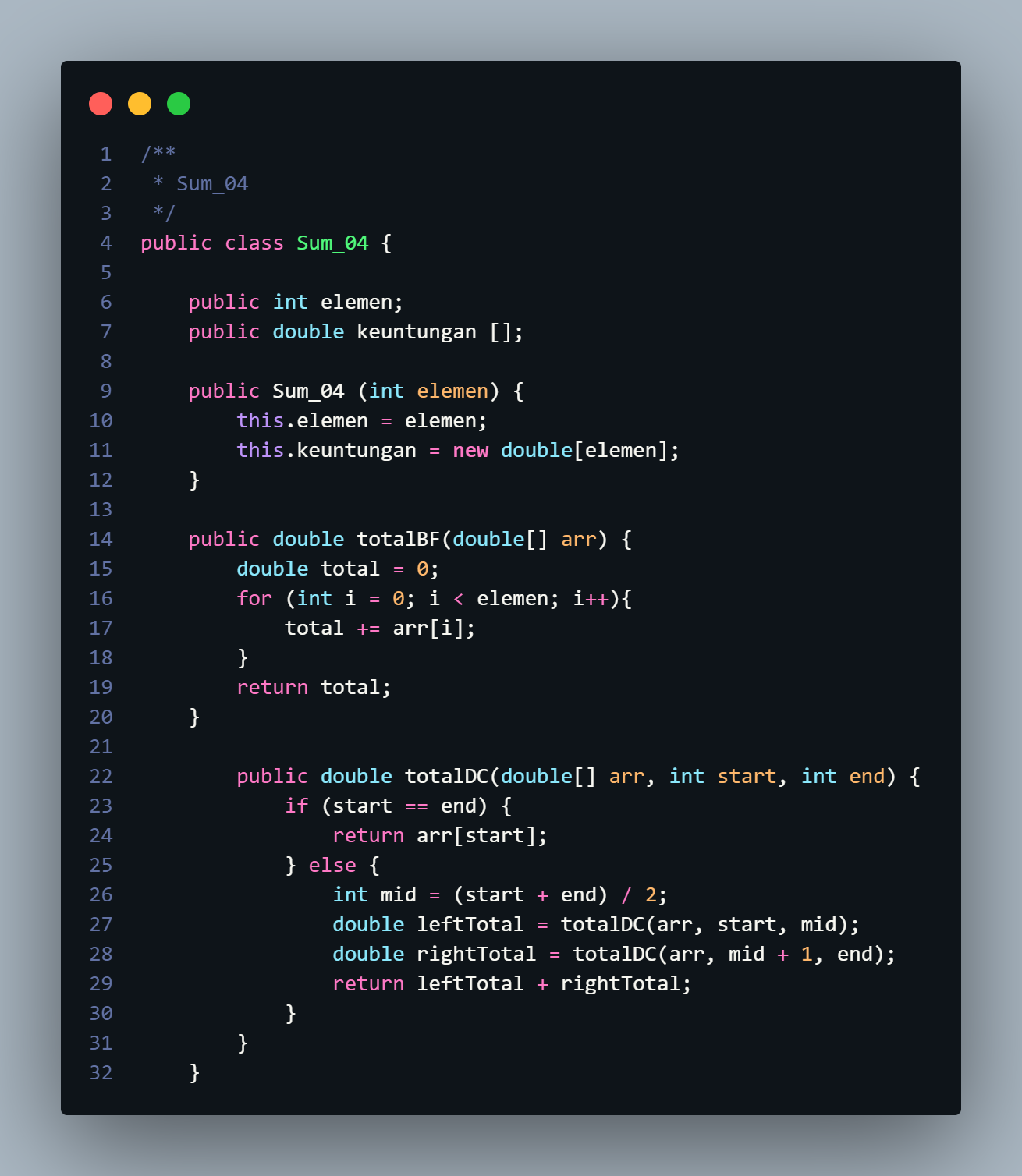
Formulasi return tersebut merupakan combine dari Divide and Conquer dimana pada program tersebut mengkalkulasikan semua hasil dari proses sebelumnya untuk mendapatkan hasil akhir yang diinginkan

**4. Kenapa dibutuhkan variable mid pada method TotalDC()?**

Karena variable mid merupakan pembatas pada saat pembagian atau divide proses, sehingga proses perhitungan akan dilakukan sesuai dengan per bagian yang telah dibagi.

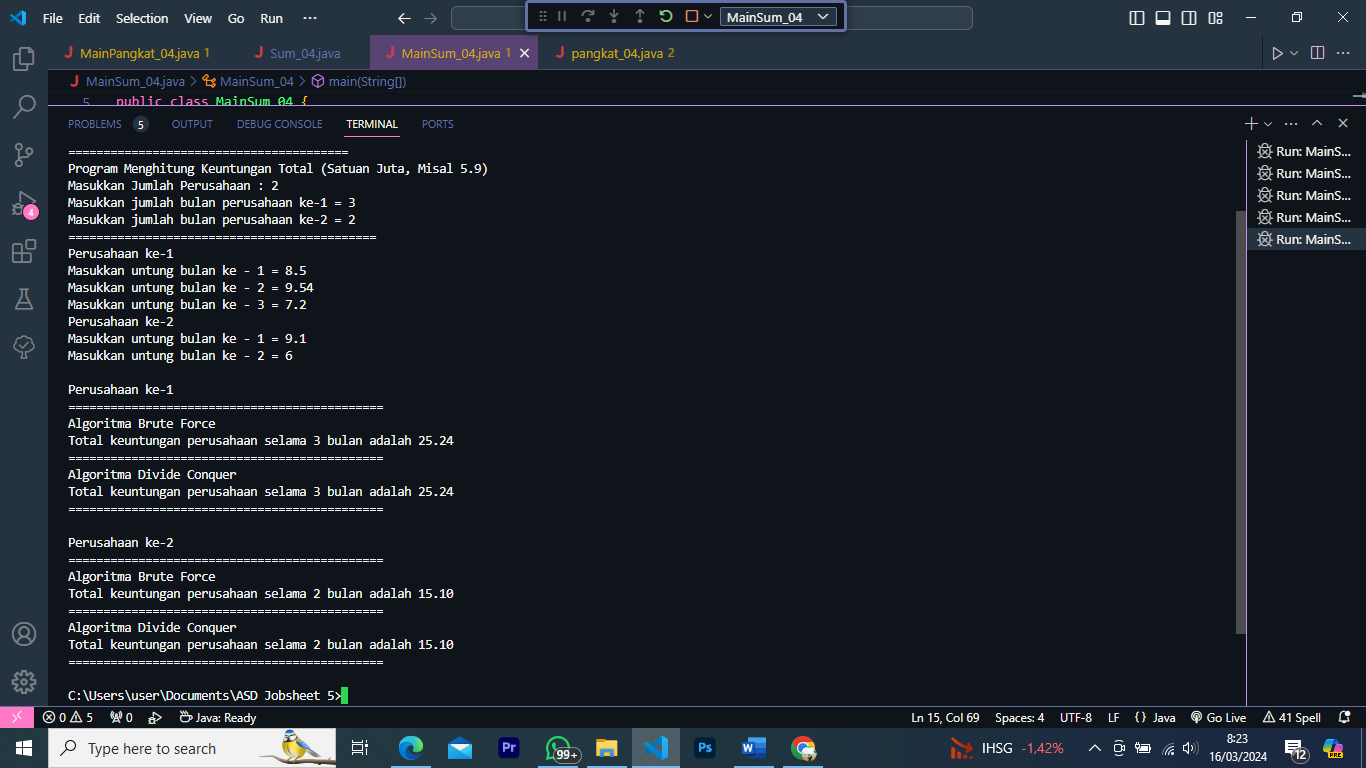
**5. Program perhitungan keuntungan suatu perusahaan ini hanya untuk satu perusahaan saja. Bagaimana cara menghitung sekaligus keuntungan beberapa bulan untuk beberapa perusahaan.(Setiap perusahaan bisa saja memiliki jumlah bulan berbeda-beda)? Buktikan dengan program!**

Dengan menambahkan input untuk memasukkan jumlah perusahaan dan menambahkan array objek untuk perusahaan yang berbeda





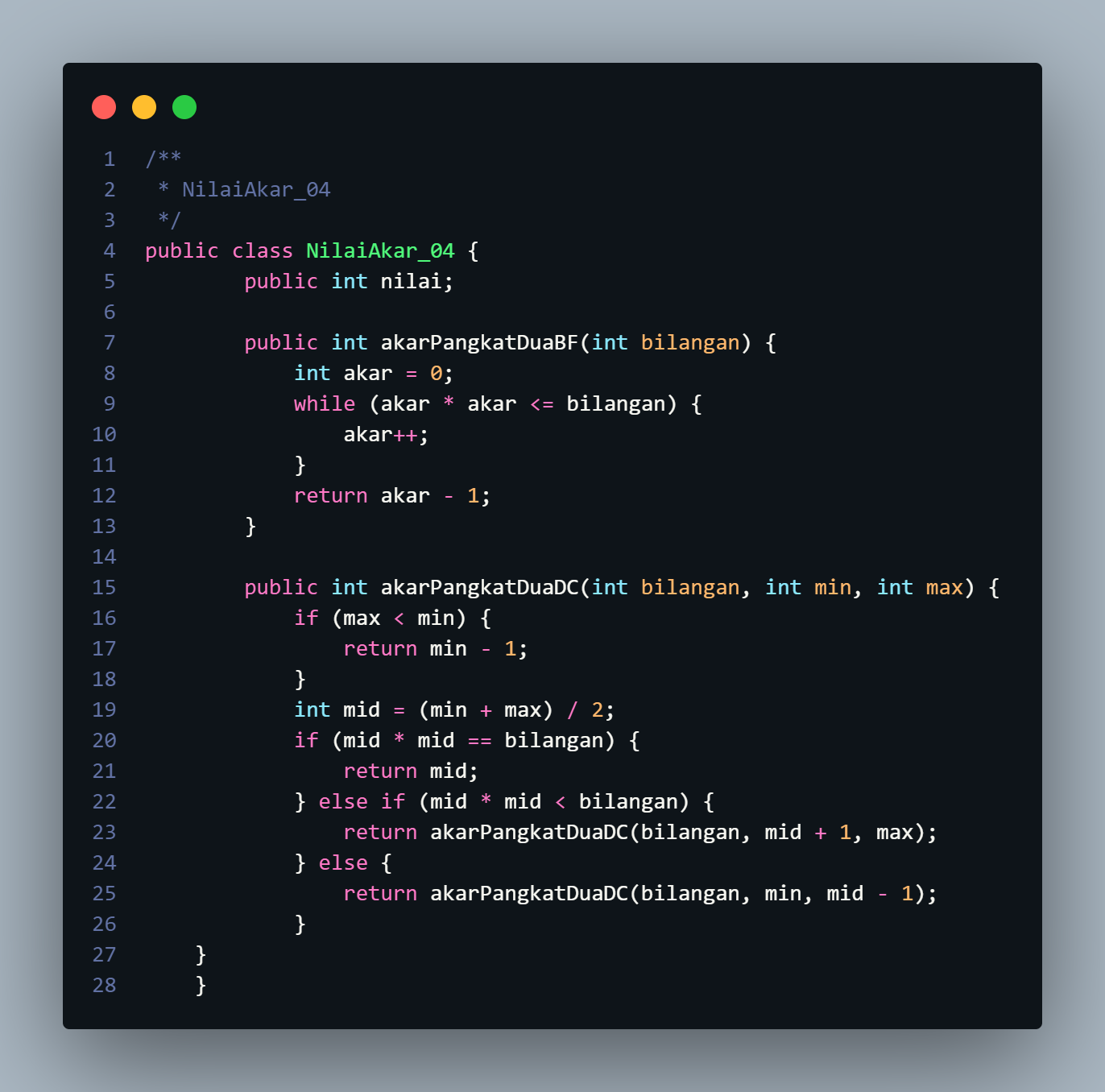
Berikut merupakan hasil running dari kode program di atas



**4.5** **Latihan Praktikum**

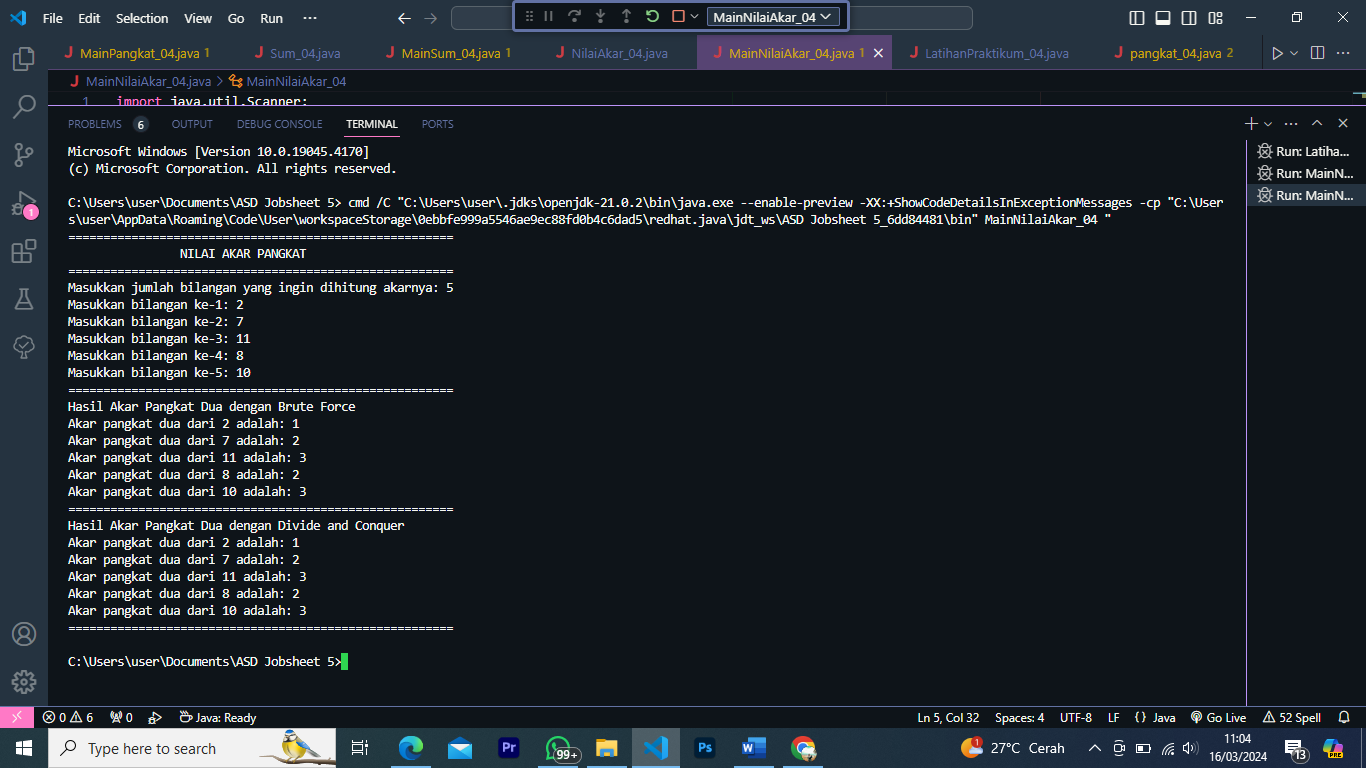
**Buatlah kode program untuk menghitung nilai akar dari suatu bilangan dengan algoritma Brute Force dan Divide Conquer! Jika bilangan tersebut bukan merupakan kuadrat sempurna, bulatkan angka ke bawah.**

Berikut merupakan kode program untuk menghitung nilai akar dengan BF dan DC dengan menginputkan angka kuadrat sempurna dan tidak sempurna.

****

****

Berikut merupakan hasil running dari kode program di atas

****