**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET 5**



**Oleh:**

**DZULFIKAR MUHAMMAD AL GHIFARI**

**NIM. 2341760071**

**SIB-1F / 08**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

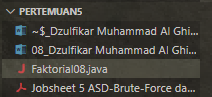
**POLITEKNIK NEGERI MALANG**

**PRAKTIKUM 5**

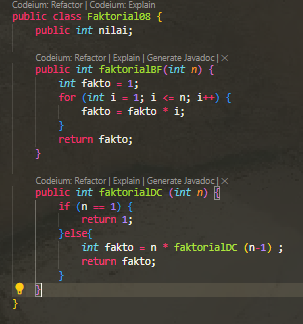
**4.2 PERCOBAAN**

**4.2.1 LANGKAH LANGKAH PERCOBAAN**

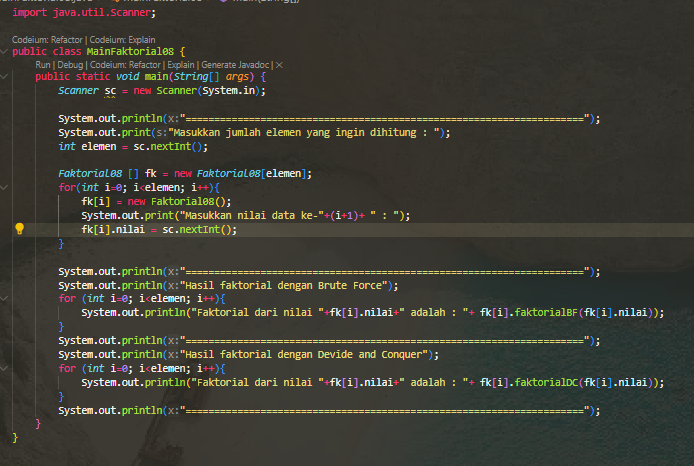
1. Buatlah class baru dengan nama Faktorial



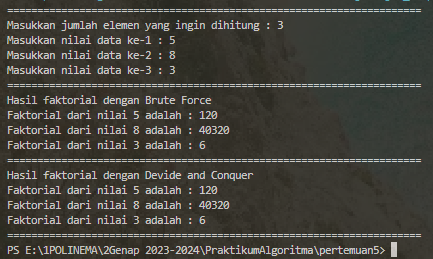
2. Lengkapi class Faktorial dengan atribut dan method



3. run MainFaktorial



**4.2.2 VERIFIKASI PERCOBAAN**



**4.2.3 PERTANYAAN**

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

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

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

4. Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!

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

Jawaban

1. Yang pertama nilai yang di kirimkan akan di validasi apakah sama dengan 1 atau tidak, jika sama dengan satu maka fungsi akan mengembalikan nilai dengan value 1. Jika tidak makanilai akan dikalikan dengan pemanggilan fungsi dengan parameter -1

N=3

3-faktorialDC(2)

2-faktorialDC(1)

1

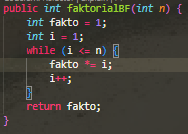
1. Didive yaitu

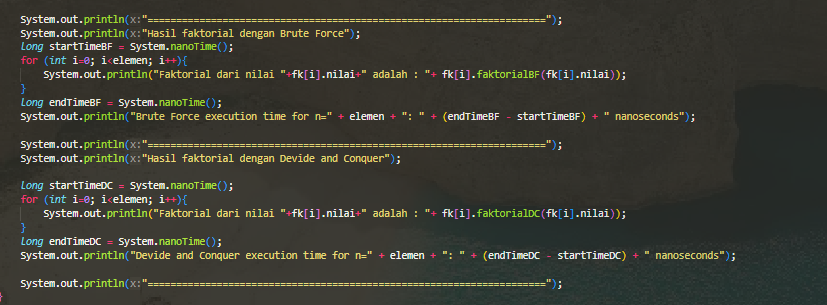


**Conquer yaitu**

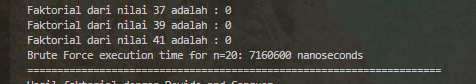


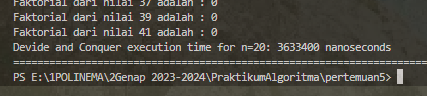
Combine tidak ada

1. Memungkinkan menggunakan perulangan while  
   



1. Didive and Conquer mengeksekusi 2x lebih cepat dibandingkan BruteForce

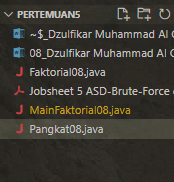




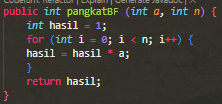
**4.3 MENGHITUNG HASIL PANGKAT DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE AND CONQUER**

**4.3.1 LANGKAH LANGKAH PERCOBAAN**

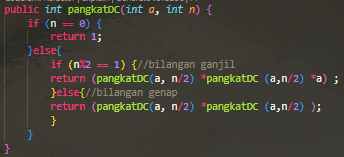
1. Buatlah class baru dengan nama Pangkat



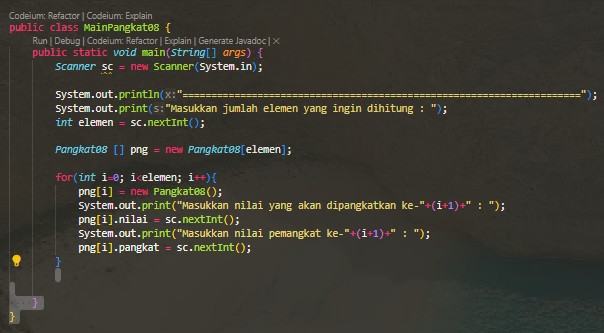
2. Menambahkan method pangkatBF



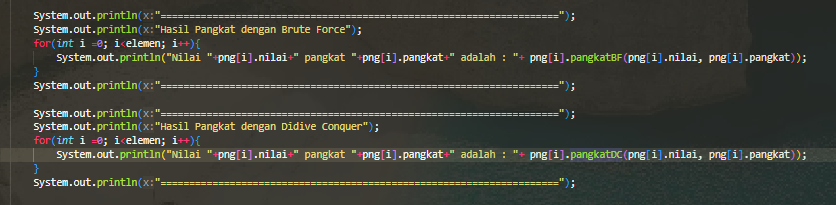
3. Menambahkan method pangkatDC



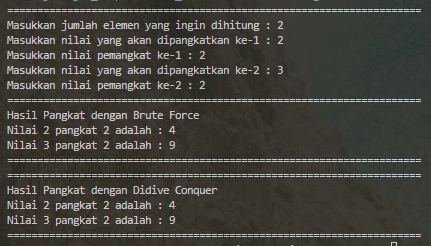
4. Menambahkan class MainPangkat



1. Pemanggilan method pangkatBF dan pangkatDC



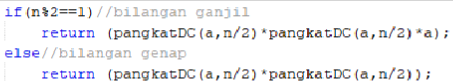
**4.3.2 VERIFIKASI PERCOBAAN**



**4.2.3 PERTANYAAN**

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

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



Jelaskan arti potongan kode tersebut

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

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

Konstruktor

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

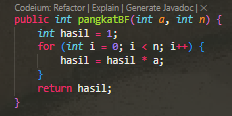
Jawaban

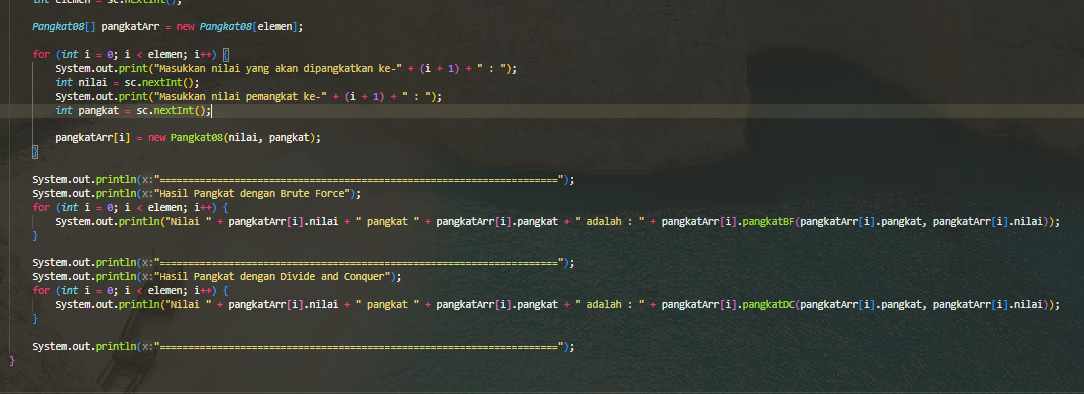
1. method Brute Force menggunakan perulangan secara runtut dan di setiap literasinya akan mengkalikan hasil bilangan dengan pangkat. Method Divide Conquer akan menggunakan medote secara rekursif, dengan memberi validasi di awal apakah pangkat bernilai 0, jika ya maka akan mengembalikan 1, jika tidak maka akan memvalidasi apakah bilangan ganjil atau tidak, kemudian akan melakukan rekursif

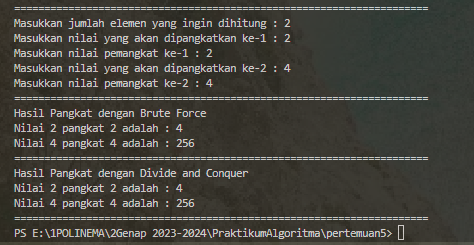
2.di awal akan Memvalidasi bilangan apakah ganjil atau tidak, kemudian jika ganjil maka fungsi menghitung pangkat akan dikalikan dengan fungsi mengitung pangkat dengan imbuhan di akhir dikalikan dengan nilai. Jika genap maka akan menjalankan fungsi menghitung pangkat dikalikan dengan fungsi menghitung pangkat.

3. 

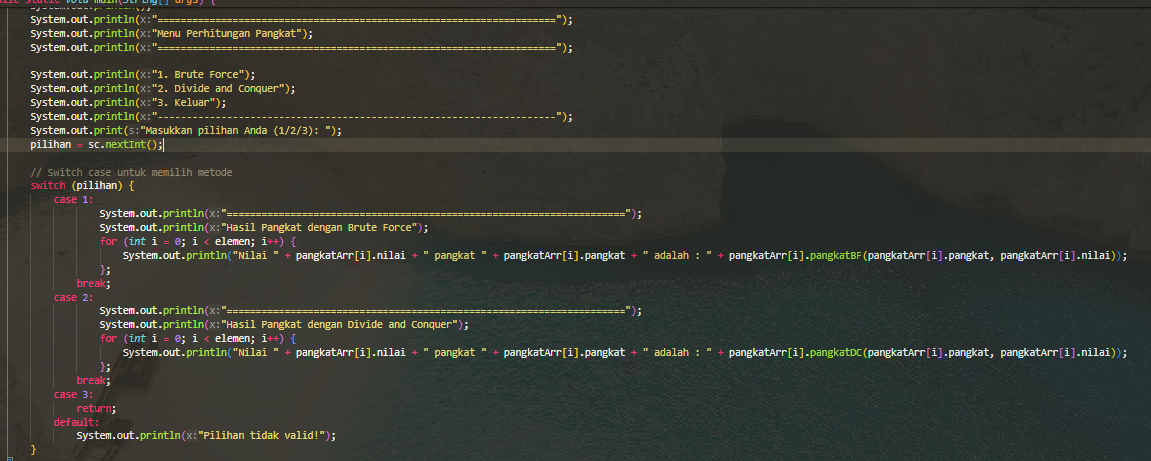
4.

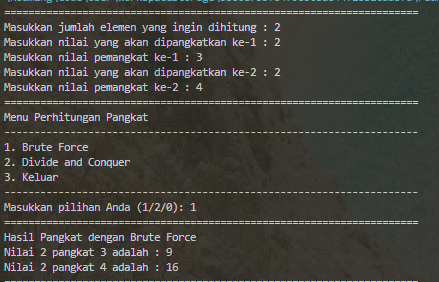






5.

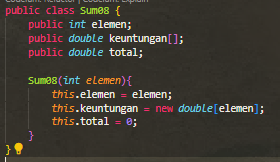




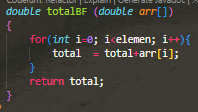
**4.4 MENGHITUNG SUM ARRAY DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE AND CONQUER**

**4.4.1 LANGKAH LANGKAH PERCOBAAN**

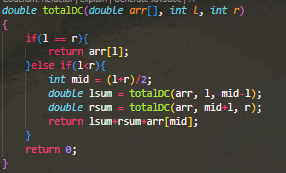
1. Membuat class sum, menambahkan elemen, menambahkan constructor



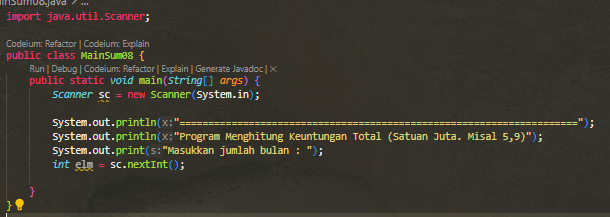
2. Menambahkan method totalBF



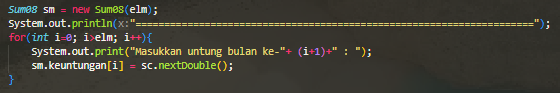
3. Menambahkan method totalDC



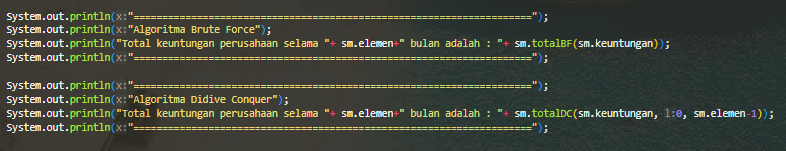
4. Menambahkan mainSum



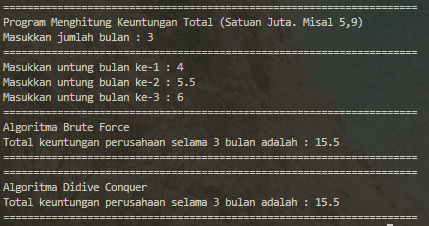
5. Membuat objek sum



6. Mnampilkan hasil Brute Force dan Didice Connquer



**4.4.2 VERIFIKASI PERCOBAAN**



**4.4.3 PERTANYAAN**

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

TotalDC()

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.

3. Mengapa terdapat formulasi return value berikut?Jelaskan!



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

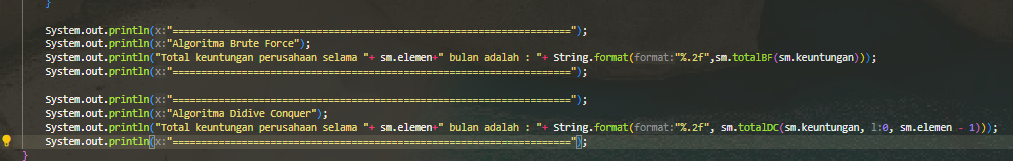
5. Program perhitungan keuntungan suatu perusahaan ini hanya untuk satu perusahaan saja. Bagaimana cara menghitung sekaligus keuntungan beberapa bulan untuk beberapa

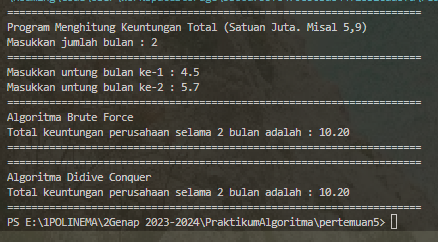
perusahaan.(Setiap perusahaan bisa saja memiliki jumlah bulan berbeda-beda)? Buktikan

dengan program!

Jawaban

1. method totalBF menggunakan perulangan dan di setiap literasi akan menambahkan total dari tiap tiap array key ke var total, sehingga setelah proses perulangan var total berisi total dari value array. Method totalDC memanfaatkan fungsi rekursif, dengan terlebih dahulu memberi validasi 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 validasi lagi apakah l < r, jika tidak maka akan mengembalikan 0, jika yam aka akan melakukan kalkulasi rekursif hingga selesai, setelah itu akan mengembalikan hasil dari kalkulasi.

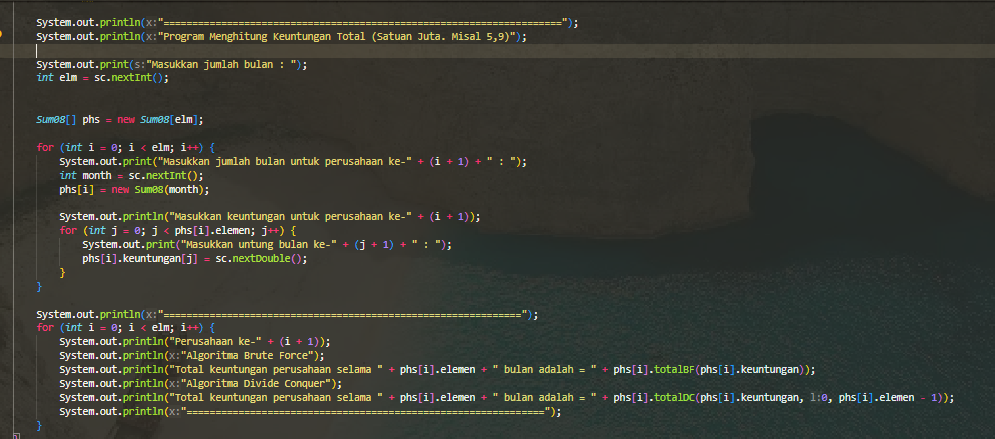
2. 

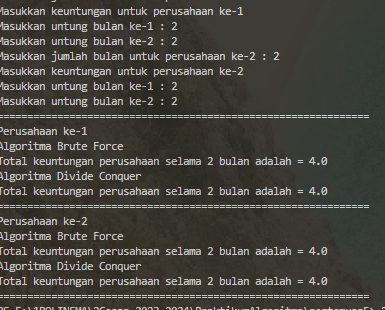


3. untuk menghitung jumlah dari total elemen yang dihitung

4. mid digunakan untuk menampung nilai tengah dan menentukan batas rekursif

5.





**4.4 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.



