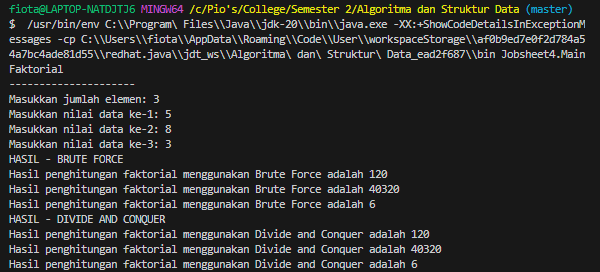
**LAPORAN TUGAS PRAKTIKUM   
BRUTE FORCE DAN DIVIDE CONQUER**



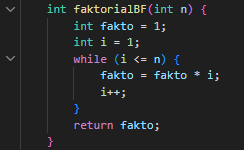
**Oleh:  
FAWWAZ ALIFIO FARSA  
NIM. 2341720128  
TI-1E / 10**

**D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

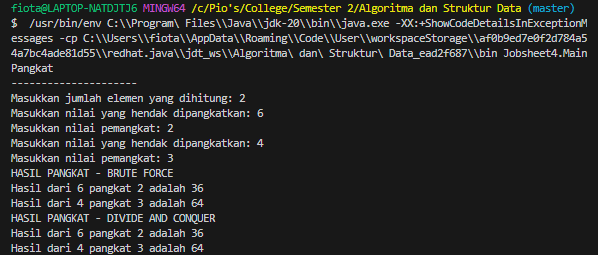
**Praktikum Percobaan 1**

Verifikasi Hasil Percobaan  


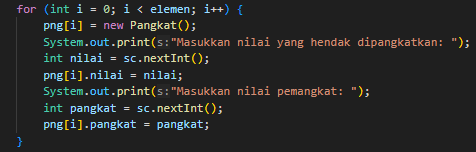
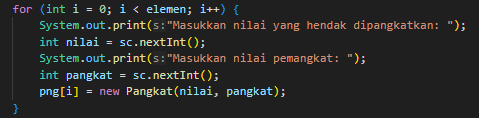
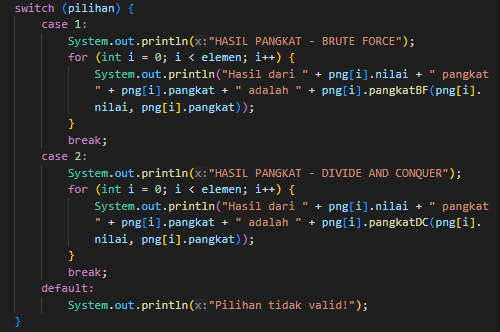
**Jawaban Pertanyaan Praktikum Percobaan 1**

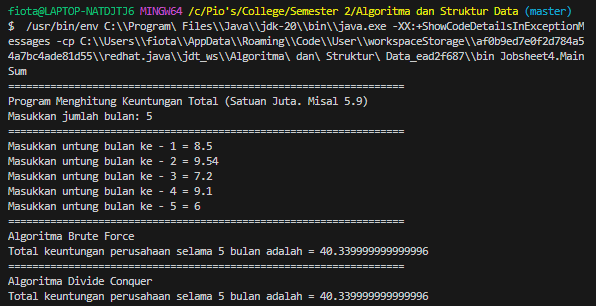
1. Pada bagian base line algoritma Divide Conquer terdapat perbedaan signifikan dalam penggunaan **if** dan **else**. Kondisi **if** memeriksa apakah nilai **n** sama dengan 1, dan jika ya, maka akan mengembalikan nilai 1 sebagai hasil faktorial. Kondisi **else** dilakukan ketika nilai **n** tidak sama dengan 1. Kondisi **else** akan menghitung **fakto** dengan mengalikan **n** dengan hasil rekursif dari **faktorialDC(n-1)**, dan pada akhirnya akan mengembalikan nilai **fakto** sebagai hasil faktorial.
2. Ya, perulangan pada method **faktorialBF()** dapat diubah selain menggunakan for, sebagai contoh alternatifnya dengan menggunakan while.  
   
3. Perbedaan utamnya terdapat pada cara dalam memperbarui nilai **fakto**. Kode **fakto \*= i;** tidak melibatkan rekursi, sedangkan kode **int fakto = n \* faktorialDC(n-1);** menggunakan rekursi.

**Praktikum Percobaan 2**

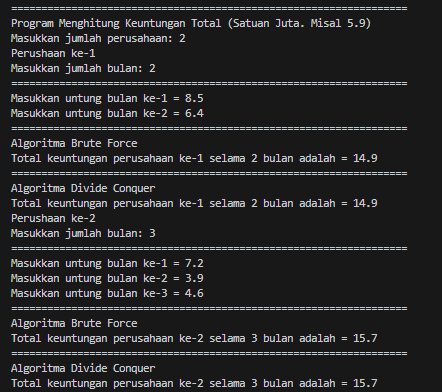
Verifikasi Hasil Percobaan  


**Jawaban Pertanyaan Praktikum Percobaan 2**

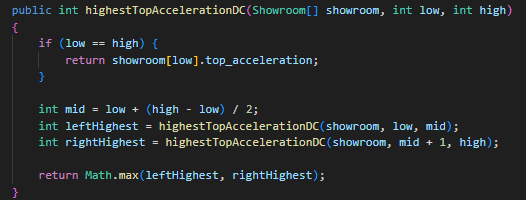
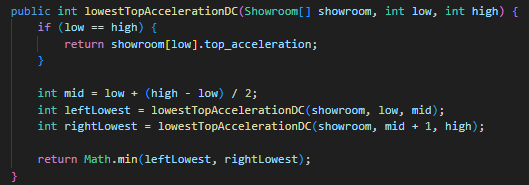
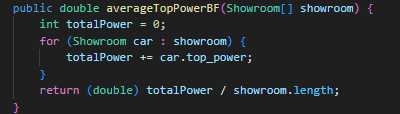
1. Perbedaan 2 method yaitu **pangkatBF()** menghitung pangkat secara langsung dengan perulangan dan mengalikan bilangan dasar dengan dirinya sendiri sebanyak nilai pangkat (**O(n)**), sedangkan **pangkatDC()** membagi masalah menjadi sub-masalah yang lebih kecil dan menghitung pangkat dengan rekursi (**O(log n)**).
2. Ya, terdapat tahap combine di dalam kode tersebut.  
   
3. Sebelum modifikasi  
     
   Setelah modifikasi agar pengisian atribut dilakukan dengan konstruktor:  
   
4. Menambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan **switch-case**.  
   

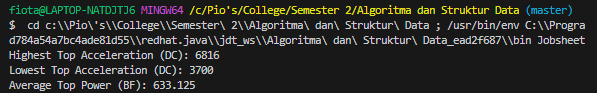
**Praktikum Percobaan 3**  
Verifikasi Hasil Percobaan  


**Jawaban Pertanyaan Praktikum Percobaan 3**

1. Formulasi kode **return lsum + arr[mid] + rsum;** adalah dengan menghitung jumlah elemen di sub-array kiri (**lsum**) yang diperoleh dari panggilan rekursif **totalDC(arr, l, mid-1)** dan sub-array kanan (**rsum**) yang diperoleh dari panggilan rekursif **totalDC(arr, mid+1, r)**, dan kemudian menggabungkan jumlah sub-array ini dengan elemen tengah (**arr[mid]**) untuk mendapatkan jumlah total seluruh sub-array.
2. Variabel **mid** dalam fungsi **totalDC()** dibutuhkan karena merupakan inti dari pendekatan yang digunakan untuk menentukan titik tengah dari sub-array yang sedang diproses.
3. Output setelah modifikasi:  
   

**Tugas**

1. Menentukan:
2. **top\_acceleration** tertinggi menggunakan Divide and Conquer  
   ****
3. **top\_acceleration** terendah menggunakan Divide and Conquer
4. Rata – rata **top\_power** dari seluruh mobil menggunakan Brute Force  
   ****

Output:  
****