



**Nama**

Raudhil Firdaus Naufal

**NIM**

2341720164

**Kelas**

1E

**Jurusan**

Teknologi Informasi

**Prodi**

Teknik Informatika

## JOBSHEET IV

### BRUTE FORCE AND DIVIDE CONQUER

#### Percobaan 1:

#### Hasil Kode Program:

```
-----  
Masukkan jumlah elemen:  
3  
Masukkan nilai data ke-1:  
5  
Masukkan nilai data ke-2:  
8  
Masukkan nilai data ke-3:  
3  
HASIL - BRUTE FORCE  
Hasil penghitungan faktorial menggunakan Brute Force adalah 120  
Hasil penghitungan faktorial menggunakan Brute Force adalah 40320  
Hasil penghitungan faktorial menggunakan Brute Force adalah 6  
HASIL - DIVIDE AND CONQUER  
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 120  
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 40320  
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 6
```

#### Pertanyaan:

1. Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!  
Perbedaannya adalah, jika pada if nilai n dari parameter adalah 1, maka akan mengembalikan nilai 1, sementara jika else, yaitu parameter selain 1, maka akan melakukan pengulangan rekursif dalam fungsi method faktorial dc yang dimana n akan terus berkurang 1 di setiap pengulangannya, dan saat di perulangan terakhir mencapai 1, maka akan mengembalikan nilai 1, dan semua perulangan dikalikan dan dikembalikan nilai fakto yaitu hasil dari semua perkalian pengulangan rekursif tersebut.

2. Apakah memungkinkan perulangan pada method faktorialBF() diubah selain menggunakan for? Buktikan!

Ya bisa, di sini saya menggunakan perulangan while.

```
int faktorialBF(int n) {  
    int fakto = 1;  
    int i = 1;  
    while (i <= n) {  
        fakto = fakto * i;  
        i++;  
    }  
    return fakto;  
}
```

3. Jelaskan perbedaan antara fakto \*= i; dan int fakto = n \* faktorialDC(n-1); !

Perbedaan antara fakto \*= i adalah setiap i bertambah, maka fakto akan terus dikalikan dengan i hingga mencapai batas dari perulangan, sehingga akan terhitung dari setiap perulangan. Sementara fakto = n \* faktorialDC(n-1) adalah n dikali dengan rekursif dari faktorialDC dimana n akan terus berkurang, tetapi jumlah n akan terus bertambah hingga batas rekursif yang ditentukan, jadi akan dihitung dalam 1 baris.

## Percobaan 2:

### Hasil Kode Program:

```
HASIL PANGKAT - BRUTE FORCE  
Hasil dari 6 pangkat 2 adalah 36  
Hasil dari 4 pangkat 3 adalah 64  
HASIL PANGKAT - DIVIDE CONQUER  
Hasil dari 6 pangkat 2 adalah 36  
Hasil dari 4 pangkat 3 adalah 64
```

### Pertanyaan:

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

Pada PangkatBF, hasil akan terus dikalikan dengan int nilai didalam perulangan yang dibatasi dengan jumlah pangkat. Sedangkan PangkatDC melakukan pengulangan rekursif jika pangkat bernilai genap atau ganjil kecuali 1, saat pangkat ganjil maka akan melakukan pengulangan rekursif pangkatDC dengan parameter nilai dan pangkat dibagi 2 dikali rekursif pangkatDC dengan parameter yang sama, lalu dikali kan dengan nilai, dan saat pangkat menjadi genap, maka akan dihitung pangkatDC berparameter nilai dan pangkat dibagi 2 dikali pangkatDC dengan parameter yang sama, jika pangkat bernilai 1, maka akan melakukan pengembalian nilai yaitu nilai itu sendiri.

2. Apakah tahap combine sudah termasuk dalam kode tersebut? Tunjukkan!  
Sudah seperti di kode ini.

```
if (n%2==1) {  
    return(pangkatDC(a,n/2)*pangkatDC(a,n/2)*a);  
} else{  
    return (pangkatDC(a,n/2) * pangkatDC(a,n/2));  
}
```

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

```
for (int i = 0; i < elemen; i++) {  
    png[i] = new Pangkat();  
    System.out.println(x:"Masukkan nilai yang hendak  
    dipangkatkan: ");  
    int nilai = sc.nextInt();  
    png[i].nilai = nilai;  
    System.out.println(x:"Masukkan nilai pemangkat: ");  
    int pangkat = sc.nextInt();  
    png[i].pangkat= pangkat;  
    png[i].pangkatBF(png[i].nilai,png[i].pangkat);  
}
```

4. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan switch-case!

```
System.out.println(x:"Tentukan Jenis Operasi yang ingin
dilakukan:\n1. BRUTE FORCE\n2. DIVIDE CONQUER");
int choice = sc.nextInt();

switch (choice) {
    case 1:
        System.out.println(x:"HASIL PANGKAT - BRUTE FORCE");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + png[i].nilai + "
            pangkat " + png[i].pangkat + " adalah "
            + png[i].pangkatBF(png[i].nilai, png[i].
            pangkat));
        }
        break;
    case 2:
        System.out.println(x:"HASIL PANGKAT - DIVIDE CONQUER");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + png[i].nilai + "
            pangkat " + png[i].pangkat + " adalah "
            + png[i].pangkatDC(png[i].nilai, png[i].
            pangkat));
        }
        break;
    default:
        System.out.println(x:"Silahkan pilih antara 1 atau 2");
        break;
}
```

## Latihan Praktikum:

1. Sebuah showroom memiliki daftar mobil dengan data sesuai tabel di jobsheet, tentukan:

- top\_acceleration tertinggi menggunakan Divide and Conquer!

```
int topSpeed(int top_ac[],int l,int r){
    if (l == r) {
        top = top_ac[l];
    } else if (top_ac[l] > top_ac[r]) {
        topSpeed(top_ac, l, r + 1);
    } else {
        topSpeed(top_ac, l - 1, r);
    }

    return top;
}
```

- top\_acceleration terendah menggunakan Divide and Conquer!

```
int bottomSpeed(int bot_ac[],int l,int r){
    if (l == r) {
        bottom = bot_ac[r];
    } else if (bot_ac[l] < bot_ac[r]) {
        bottomSpeed(bot_ac, l, r + 1);
    } else{
        bottomSpeed(bot_ac, l - 1, r);
    }

    return bottom;
}
```

- Rata-rata top\_power dari seluruh mobil menggunakan Brute Force!

```
int averageSpeed(int power_ac[]){
    int sum = 0;
    for (int i = 0; i < power_ac.length; i++) {
        sum += power_ac[i];
        if (i == power_ac.length - 1) {
            sum = sum/power_ac.length;
        }
    }
    return sum;
}
```

## Hasil Kode Program

```
Top acceleration tertinggiya adalah 6816  
Top acceleration terendahnya adalah 3700  
Rata rata top powernya adalah 633
```