

Nama : Muhammad Fauzan

NIM : 1941720171

Kelas : TI 1F

2.3.3

JAWAB

1) Base lain yang mencari nilai factorial

```
20 public int faktorialDC(int n) {
21     if (n==1) {
22         return 1;
23     }
24     else {
25         int fakto = n * faktorialDC(n-1);
26         return fakto;
27     }
```

Base ini akan mencari nilai faktorial, divide dan conquer lebih natural dengan skema rekursif yang dilakukan saat pencarian nilai faktorial dari program tersebut, Maksudnya jika nilai $n=1$ maka akan di return jika salah akan masuk ke pencarian nilai fakto = n dikalikan (*) dengan faktorialDC (n-1); Dan akan di return nilai fakto nya.

2) 1.

```
21     if (n==1) {
22         return 1;
23     }
```

2.

```
24     else {
25         int fakto = n * faktorialDC(n-1);
```

3.

```
26         return fakto;
```

3)

```
12 public int faktorialBF(int n) {
13     int fakto = 1;
14     int i = 1;
15     while(i < fakto) {
16         fakto = fakto * i;
17         i++;
18     }
```

4)

```

49     for (int i=0; i<elemen; i++) {
50         System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+fk[i].faktorialBF(fk[i].nilai));
51         double start = System.nanoTime();
52         double end = (double) System.nanoTime();
53         System.out.println("Waktu Eksekusi = "+(end-start)/1000000+ " milisecond ");
54     }
55     System.out.println("=====");
56     System.out.println("Hasil Faktorial dengan Divide and Conquer");
57     for (int i=0; i<elemen; i++) {
58         System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+fk[i].faktorialDC(fk[i].nilai));
59         double start = System.nanoTime();
60         double end = (double) System.nanoTime();
61         System.out.println("Waktu Eksekusi = "+(end-start)/1000000+ " milisecond ");

```

5)

```

=====
MENGHITUNG NILAI FAKTORIAL
=====
Masukan jumlah elemen yang ingin dihitung =25
masukan nilai data ke-1 = 3
masukan nilai data ke-2 = 2
masukan nilai data ke-3 = 3
masukan nilai data ke-4 = 4
masukan nilai data ke-5 = 5
masukan nilai data ke-6 = 6
masukan nilai data ke-7 = 67
masukan nilai data ke-8 = 89
masukan nilai data ke-9 = 23
masukan nilai data ke-10 = 45
masukan nilai data ke-11 = 67
masukan nilai data ke-12 = 89
masukan nilai data ke-13 = 90
masukan nilai data ke-14 = 12
masukan nilai data ke-15 = 34
masukan nilai data ke-16 = 56
masukan nilai data ke-17 = 75
masukan nilai data ke-18 = 43
masukan nilai data ke-19 = 46
masukan nilai data ke-20 = 23
masukan nilai data ke-21 = 12
masukan nilai data ke-22 = 23
masukan nilai data ke-23 = 78
masukan nilai data ke-24 = 45
masukan nilai data ke-25 = 34
=====
HASIL FAKTORIAL DENGAN BRUTE FORCE
Faktorial dari nilai3adalah = 6
Faktorial dari nilai2adalah = 2
Faktorial dari nilai3adalah = 6
Faktorial dari nilai4adalah = 24
Faktorial dari nilai5adalah = 120
Faktorial dari nilai6adalah = 720
Faktorial dari nilai67adalah = 0
Faktorial dari nilai89adalah = 0
Faktorial dari nilai23adalah = 862453760
Faktorial dari nilai45adalah = 0
Faktorial dari nilai67adalah = 0
Faktorial dari nilai89adalah = 0
Faktorial dari nilai90adalah = 0
Faktorial dari nilai12adalah = 479001600
Faktorial dari nilai34adalah = 0
Faktorial dari nilai56adalah = 0
Faktorial dari nilai75adalah = 0
Faktorial dari nilai43adalah = 0
Faktorial dari nilai46adalah = 0
Faktorial dari nilai23adalah = 862453760
Faktorial dari nilai12adalah = 479001600
Faktorial dari nilai23adalah = 862453760
Faktorial dari nilai78adalah = 0
Faktorial dari nilai45adalah = 0
Faktorial dari nilai34adalah = 0
Waktu eksekusi = 1 milisecond

```

```

=====
HASIL FAKTORIAL DENGAN DIVIDE DAN CONQUER
Faktorial dari nilai 3adalah : 6
Faktorial dari nilai 2adalah : 2
Faktorial dari nilai 3adalah : 6
Faktorial dari nilai 4adalah : 24
Faktorial dari nilai 5adalah : 120
Faktorial dari nilai 6adalah : 720
Faktorial dari nilai 67adalah : 0
Faktorial dari nilai 89adalah : 0
Faktorial dari nilai 23adalah : 862453760
Faktorial dari nilai 45adalah : 0
Faktorial dari nilai 67adalah : 0
Faktorial dari nilai 89adalah : 0
Faktorial dari nilai 90adalah : 0
Faktorial dari nilai 12adalah : 479001600
Faktorial dari nilai 34adalah : 0
Faktorial dari nilai 56adalah : 0
Faktorial dari nilai 75adalah : 0
Faktorial dari nilai 43adalah : 0
Faktorial dari nilai 46adalah : 0
Faktorial dari nilai 23adalah : 862453760
Faktorial dari nilai 12adalah : 479001600
Faktorial dari nilai 23adalah : 862453760
Faktorial dari nilai 78adalah : 0
Faktorial dari nilai 45adalah : 0
Faktorial dari nilai 34adalah : 0
=====
Waktu eksekusi = 10 milisecond

```

2.4.3

JAWAB

- 1) a. PangkatBF()
Menggunakan perulangan
b. PangkatDC()
menggunakan fungsi rekursif
- 2) Dalam kode program tersebut ada penyeleksian angka dimana angka n, jika $(n\%2) = 1$ maka n adalah bilangan ganjil dan akan dikembalikan ke nilai return berupa rekursif yaitu $(\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2) * a)$ dan jika salah maka angka n merupakan bilangan genap sehingga akan mengembalikan nilai return rekursif pada $(\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2))$;

- 3) Ya

```

26 |         return (pangkatDC(a,n/2)*pangkatDC(a,n/2));

```

- 4)

```

10 | Pangkat() {
    |     this.nilai = nilai;
    |     this.pangkat = pangkat;
13 | }

```

5)

```
14 public static void main(String[] args) {
15     Scanner sc = new Scanner(System.in);
16     System.out.println("=====");
17     System.out.println("1. BRUTE FORCE");
18     System.out.println("2. DIVIDE AND CONQUER");
19     System.out.print("Silahkan Masukkan Pilihan Anda : ");
20     int pil = sc.nextInt();
21     System.out.println("=====");
22     System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
23     int elemen = sc.nextInt();
24
25     Pangkat [] png = new Pangkat[elemen];
26
27     for (int i=0; i<elemen; i++) {
28         png[i] = new Pangkat();
29         System.out.print("Masukkan nilai yang akan dipangkatkan ke-"+(i+1)+" : ");
30         png[i].nilai = sc.nextInt();
31         System.out.print("Masukkan nilai pemangkat ke-"+(i+1)+" : ");
32         png[i].pangkat = sc.nextInt();
33     }
34
35     switch (pil) {
36         case 1:
37             System.out.println("=====");
38             System.out.println("Hasil Faktorial dengan Brute Force");
39             for (int i=0; i<elemen; i++) {
40                 System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat+" adalah : "+png[i].pangkatBF(png[i].nilai, png[i].pangkat));
41             }
42             break;
43         case 2:
44             System.out.println("=====");
45             System.out.println("Hasil Faktorial dengan divide and Conquer");
46             for (int i=0; i<elemen; i++) {
47                 System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat+" adalah : "+png[i].pangkatDC(png[i].nilai, png[i].pangkat));
48             }
49             break;
50     }
51     System.out.println("=====");
52 }
```

Output :

```
compile-single:
run-single:
=====
1. BRUTE FORCE
2. DIVIDE AND CONQUER
Silahkan Masukkan Pilihan Anda : 2
=====
Masukkan jumlah elemen yang ingin dihitung : 1
Masukkan nilai yang akan dipangkatkan ke-1 : 5
Masukkan nilai pemangkat ke-1 : 7
=====
Hasil Faktorial dengan divide and Conquer
Nilai 5 pangkat 7 adalah : 78125
=====
BUILD SUCCESSFUL (total time: 24 seconds)
```

2.5.3

JAWAB

1. Perbedaan perhitungan

- a. TotalBF() menggunakan menggunakan for untuk melakukan perulangan dimana untuk menghitung keuntungan for ($i = 0; i \leq \text{elemen}; i$) akan bertambah sampai terpenuhi yang akan dijabarkan dibawah ini :

- $i = 0 \Rightarrow \text{total} = 0 + \text{keuntungan 1}$
- $i = 1 \Rightarrow \text{total} = (0 + \text{keuntungan 1}) + \text{keuntungan 2}$
- $i = 2 \Rightarrow \text{total} = ((0 + \text{keuntungan 1}) + (\text{keuntungan 2}) + \text{keuntungan elemen})$

- b. TotalDC() menggunakan Divide and Conquer

Misalnya bulan yang kita masukan = 5, keuntungan = {1,2,3,4,5} Jadi perhitungan keuntungan dibagi menjadi 2 bagian bagian kiri lsum dan bagian kanan rsum ,yang akan saya jabarkan sebagai berikut :

- Lsum = 1, 2 dan rsum = 4, 5, kemudian nilai mid (tengah) = 3
- Di lsum melakukan perhitungan $1+2 = 3$
- Di rsum melakukan perhitungan $4+5 = 9$

Kemudian direturn dengan menjumlahkan $\text{lsum} + \text{rsum} + \text{arr}[\text{mid}] = 3 + 9 + 3 = 15$ (hasil akhir).

2. Cara membatasinya dengan System.out.printf

```
29 | System.out.printf("total keuntungan perusahaan selama "+sm.element+" bulan adalah = "+sm.totalBF(sm.keuntungan));
```

3. Kode program tersebut adalah combine dari divide and conquer yaitu fungsi TotalDC itu sendiri, dimana fungsi digunakan untuk menggabungkan kembali pemecahan masalah tadi yang telah di conquer sebelumnya.(Hasil akhir dari proses tersebut dan untuk menampilkan outputnya).
4. Variabel mid berfungsi untuk mengetahui nilai tengah dari array arr[] yang dimana fungsi program mid ini yang mencari keuntungan perbulan yang didapatkan.(keuntungan perbulan yang didapatkan).