

Nama : Muhammad Navis Abdillah

Kelas : TI 1F

NIM : 1941720147

### Jawaban 2.3.3

1. Base lain Algoritma Divide and Conquer adalah pencarian nilai factorial yang didasarkan pada perhitungan rekursif dan membagi masalah menjadi beberapa bagian yang disebut upa-masalah.

#### 2. 2. - Divide

```
if (n == 1) {  
    return 1;  
- Conquer  
} else {  
    int fakto = n * faktorialDC(n - 1);  
- combine  
    return fakto;  
}
```

#### 3. public int faktorialBF(int n) {

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

#### 4.

```
} public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("=====");  
    System.out.println("Masukkan jumlah elemen yang ingin dihitung : ");  
    int elemen = sc.nextInt();  
    Faktorial [] fk = new Faktorial[elemen];  
    for (int i = 0; i < elemen; i++){  
        fk[i] = new Faktorial ();  
        System.out.println("Masukkan nilai data ke- " + (i+1)+ " : ");  
        fk[i].nilai = sc.nextInt();  
    }  
    System.out.println("=====");  
    System.out.println("Hasil Faktorial dengan Bruce Force");  
    for (int i = 0; i < elemen; i++){  
        System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialBF(fk[i].nilai));  
    }  
    System.out.println("=====");  
    System.out.println("Hasil Faktorial dengan Divide and Conquer");  
    for (int i = 0; i < elemen; i++){  
        System.out.println("Faktorial dari nilai "+fk [i].nilai+ " adalah : " + fk[i].faktorialDC(fk[i].nilai) );  
    }  
    System.out.println("=====");  
}
```

---

Masukkan jumlah elemen yang ingin dihitung :

25

Masukkan nilai data ke- 1 :

1

Masukkan nilai data ke- 2 :

2

Masukkan nilai data ke- 3 :

3

Masukkan nilai data ke- 4 :

4

Masukkan nilai data ke- 5 :

5

Masukkan nilai data ke- 6 :

6

Masukkan nilai data ke- 7 :

7

Masukkan nilai data ke- 8 :

5

Masukkan nilai data ke- 9 :

3

Masukkan nilai data ke- 10 :

4

Masukkan nilai data ke- 11 :

5

Masukkan nilai data ke- 12 :

6

Masukkan nilai data ke- 13 :

4

Masukkan nilai data ke- 14 :

5

Masukkan nilai data ke- 15 :

6

Masukkan nilai data ke- 16 :

2

Masukkan nilai data ke- 17 :

3

Masukkan nilai data ke- 18 :

4

---

Masukkan nilai data ke- 19 :

5

Masukkan nilai data ke- 20 :

6

Masukkan nilai data ke- 21 :

2

Masukkan nilai data ke- 22 :

3

Masukkan nilai data ke- 23 :

4

Masukkan nilai data ke- 24 :

1

Masukkan nilai data ke- 25 :

3

Hasil Faktorial dengan Bruce Force

Faktorial dari nilai 1 adalah : 1  
Faktorial dari nilai 2 adalah : 1  
Faktorial dari nilai 3 adalah : 1  
Faktorial dari nilai 4 adalah : 1  
Faktorial dari nilai 5 adalah : 1  
Faktorial dari nilai 6 adalah : 1  
Faktorial dari nilai 7 adalah : 1  
Faktorial dari nilai 5 adalah : 1  
Faktorial dari nilai 3 adalah : 1  
Faktorial dari nilai 4 adalah : 1  
Faktorial dari nilai 5 adalah : 1  
Faktorial dari nilai 6 adalah : 1  
Faktorial dari nilai 4 adalah : 1  
Faktorial dari nilai 5 adalah : 1  
Faktorial dari nilai 6 adalah : 1  
Faktorial dari nilai 2 adalah : 1  
Faktorial dari nilai 3 adalah : 1  
Faktorial dari nilai 4 adalah : 1  
Faktorial dari nilai 5 adalah : 1  
Faktorial dari nilai 6 adalah : 1  
Faktorial dari nilai 2 adalah : 1  
Faktorial dari nilai 3 adalah : 1  
Faktorial dari nilai 4 adalah : 1  
Faktorial dari nilai 1 adalah : 1  
Faktorial dari nilai 3 adalah : 1

=====

Hasil Faktorial dengan Divide and Conquer

Faktorial dari nilai 1 adalah : 1  
Faktorial dari nilai 2 adalah : 2  
Faktorial dari nilai 3 adalah : 6  
Faktorial dari nilai 4 adalah : 24  
Faktorial dari nilai 5 adalah : 120  
Faktorial dari nilai 6 adalah : 720  
Faktorial dari nilai 7 adalah : 5040  
Faktorial dari nilai 5 adalah : 120  
Faktorial dari nilai 3 adalah : 6  
Faktorial dari nilai 4 adalah : 24  
Faktorial dari nilai 5 adalah : 120  
Faktorial dari nilai 6 adalah : 720  
Faktorial dari nilai 4 adalah : 24  
Faktorial dari nilai 5 adalah : 120  
Faktorial dari nilai 6 adalah : 720  
Faktorial dari nilai 2 adalah : 2  
Faktorial dari nilai 3 adalah : 6  
Faktorial dari nilai 4 adalah : 24  
Faktorial dari nilai 5 adalah : 120  
Faktorial dari nilai 6 adalah : 720  
Faktorial dari nilai 2 adalah : 2  
Faktorial dari nilai 3 adalah : 6  
Faktorial dari nilai 4 adalah : 24  
Faktorial dari nilai 1 adalah : 1  
Faktorial dari nilai 3 adalah : 6

=====

5. Ada terjadi perbedaan, jika menggunakan DC akan lebih lama karena pengulangan akan dilakukan terus menerus sesuai dengan faktorial yang diinputkan. Brute force lebih cepat daripada DC

#### Jawaban 2.4.3

1. -PangkatBF menggunakan Algoritma Brute Force yaitu dengan melakukan perulangan secara berurutan.  
- PangkatDC menggunakan Algoritma Divide and Conquer yaitu dengan membagi tugas nya sehingga menjadi permasalahan yang lebih kecil lalu men combine nya kembali, memanggil fungsinya sendiri.
2. Cara mengeksekusi nya dengan membagi tugas nya terlebih dahulu menjadi lebih kecil (Algoritma Divide and Conquer). Pada saat membagi tugas nya program akan membagi elemen menjadi dua bagian, sedangkan pada bilangan ganjil tidak dapat dibagi dua, sehingga dibuat lah seperti ini : semisal  $23 = 2 \times 2 \times 2$  akan dibagi 2 bagian menjadi :  $(2 \times 2)(2)$
3. Pada Return

```
    return 1;

    } else {

    }

    if (n % 2 == 1) { //bilangan ganjil
        return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);

    } else { //bilangan genap
        return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
    }
}
```

4.

```
/**
 *
 * @author Navis
 */
public class Pangkat {
    public int nilai, pangkat;

    Pangkat() {
        this.nilai = nilai;
        this.pangkat = pangkat;
    }
}
```

5.

```
public class Mainpangkat {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("=====");
        System.out.println("Masukkan jumlah elemen yang ingin dihitung : ");
        int elemen = sc.nextInt();

        Pangkat [] png = new Pangkat[elemen];
        for (int i = 0; i < elemen; i++){
            png[i] = new Pangkat();
            System.out.println("Masukkan nilai yang akan dipangkatkan ke -" + (i+1) + " : ");
            png[i].nilai = sc.nextInt();
            System.out.println("Masukkan nilai pangkat ke - " + (i+1) + " : ");
            png[i].pangkat = sc.nextInt();
        }
        System.out.println("=====");
        System.out.println("Hasil Pangkat dengan Bruce Force");
        for (int i = 0; i < elemen; i++){
            System.out.println("Nilai " + png[i].nilai + "pangkat " + png[i].pangkat + " adalah : " + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
        }
        System.out.println("=====");
        System.out.println("Hasil Pangkat dengan Divide and Conquer ");
        for (int i = 0; i < elemen; i++){
            System.out.println("Nilai " + png[i].nilai + "pangkat " + png[i].pangkat + " adalah : " + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
        }
        System.out.println("=====");
    }
}
```

#### Jawaban 2.5.3

- Perbedaan perhitungan
  - TotalBF() menggunakan for untuk melakukan perulangan dimana untuk menghitung keuntungan for(i=0; i<=elemen; i++) akan bertambah sampai terpenuhi yang akan dijabarkan dibawah ini :
    - i=0 => total = 0 + keuntungan 1
    - i=1 => total = (0 + keuntungan 1) + keuntungan 2
    - i=2 => total = ((0 + keuntungan 1) + (keuntungan 2) + keuntungan elemen)
  - TotalDC() menggunakan Divide and Conquer Misalkan bulan yang kita masukkan = 5, keuntungan = {1, 2, 3, 4, 5}. Jadi perhitungan keuntungan dibagi menjadi 2 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 lsum + rsum + arr[mid] = 3 + 9 + 3 = 15 (hasil akhir)
- Cara membatasinya dengan System.out.printf
- 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.
- Variable 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.
- public class MainSum { public static void main(String[] args) {

```

Scanner sc = new Scanner(System.in);
System.out.println("=====
=====");
System.out.println(" Progam Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
System.out.println("=====
=====");
System.out.print("Masukkan jumlah perusahaan : ");
int elm = sc.nextInt();
for(int i=0; i) {
System.out.println("=====
=====");
System.out.println("Perusahaan ke-" + (i+1)); System.out.println("Masukkan jumlah bulan :
"); int bln = sc.nextInt();
}
Sum sm = new Sum(elm);
System.out.println("=====
=====");
for(int i=0; i<sm.elemen; i++){
System.out.print("Masukkan untung bulan ke-" + (i+1) + " = ");
sm.keuntungan[i] = sc.nextDouble();
}
System.out.println("=====
=====");
System.out.println(" Algoritma Brute Force");
System.out.println("=====
=====");
System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "
+ sm.totalBF(sm.keuntungan));
System.out.println("=====
====="); System.out.println(" Algoritma Divide and
Conquer");
System.out.println("=====
=====");
System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "
+ sm.totalDC(sm.keuntungan, 0, sm.elemen-1));
}
}

```