

## Laporan Praktikum ASD Jobsheet 05

Nama: Gegas Anugrah Derajat

Kelas: SIB-1F

Absen: 11

### Percobaan 4.2

Membuat class dan atribut

```
public class Faktorial11 {  
    public int nilai;
```

Menambahkan method

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

Menambahkan method

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

pada fungsi main ditambahkan input nilai

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

membuat array of objek

```
Faktorial11 [] fk = new Faktorial11[elemen];  
for(int i=0; i<elemen; i++){  
    fk[i] = new Faktorial11();  
    System.out.print("Masukkan nilai data ke-"+(i+1)+ " : ");  
    fk[i].nilai = sc.nextInt();  
}
```

menampilkan hasil

```
System.out.println(x:"=====");
System.out.println(x:"Hasil faktorial dengan Brute 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(x:"=====");
System.out.println(x:"Hasil faktorial dengan Devide 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(x:"=====");
}
```

hasil

```
=====
Masukkan jumlah elemen yang ingin dihitung : 3
Masukkan nilai data ke-1 : 5
Masukkan nilai data ke-2 : 8
Masukkan nilai data ke-3 : 3
=====
Hasil faktorial dengan Brute Force
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 3 adalah : 6
=====
Hasil faktorial dengan Devide and Conquer
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 3 adalah : 6
=====
```

#### 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. Nilai pertama yang dikirimkan akan di cek apakah sama dengan 1 atau tidak, jika nilai sama dengan 1 maka nilai akan dikembalikan dengan value 1. Jika tidak sama dengan satu maka akan dikalikan dengan -1

2. divide

```
if (n == 1) {
    return 1;
}
```

conquer

```
int fakto = n * faktorialDC (n-1) ;  
return fakto;
```

### 3. menggunakan while

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

### 4.

```
System.out.println(x:"=====");  
System.out.println(x:"Hasil faktorial dengan Brute Force");  
long startTimeBF = System.nanoTime();  
for (int i=0; i<elemen; i++){  
    System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+ fk[i].faktorialBF(fk[i].nilai));  
}  
long endTimeBF = System.nanoTime();  
System.out.println("Brute Force execution time for n=" + elemen + ": " + (endTimeBF - startTimeBF) + " nanoseconds");  
  
System.out.println(x:"=====");  
System.out.println(x:"Hasil faktorial dengan Devide and Conquer");  
  
long startTimeDC = System.nanoTime();  
for (int i=0; i<elemen; i++){  
    System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+ fk[i].faktorialDC(fk[i].nilai));  
}  
long endTimeDC = System.nanoTime();  
System.out.println("Devide and Conquer execution time for n=" + elemen + ": " + (endTimeDC - startTimeDC) + " nanoseconds");  
  
System.out.println(x:"=====");
```

### 5. divide conquer lebih cepat dibandingkan dengan brute force

```
Faktorial dari nilai 51 adalah : 0  
Faktorial dari nilai 52 adalah : 0  
Brute Force execution time for n= 20: 7667200 nanoseconds
```

```
Faktorial dari nilai 51 adalah : 0  
Faktorial dari nilai 52 adalah : 0  
Devide and Conquer execution time for n=20: 1629600 nanoseconds
```

### Percobaan 4.3

membuat class dan atribut

```
public class Pangkat11 {  
    public int nilai, pangkat;
```

menambahkan method BF dan DC

```
public int pangkatBF(int a, int n) {  
    int hasil = 1;  
    for (int i = 0; i < n; i++) {  
        hasil = hasil * a;  
    }  
    return hasil;  
}
```

```
public int pangkatDC(int a, int n) {  
    if (n == 0) {  
        return 1;  
    } else {  
        if (n % 2 == 1) {  
            return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a);  
        } else {  
            return (pangkatDC(a, n/2) * pangkatDC(a, n/2));  
        }  
    }  
}
```

membuat class main

```
public class MainPangkat11 {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int pilihan = 0;  
  
        System.out.println(x:"=====");  
        System.out.print(s:"Masukkan jumlah elemen yang ingin dihitung : ");  
        int elemen = sc.nextInt();  
  
        Pangkat11[] pangkatArr = new Pangkat11[elemen];  
  
        for (int i = 0; i < elemen; i++) {  
            System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i + 1) + " : ");  
            int nilai = sc.nextInt();  
            System.out.print("Masukkan nilai pemangkat ke-" + (i + 1) + " : ");  
            int pangkat = sc.nextInt();  
  
            pangkatArr[i] = new Pangkat11(nilai, pangkat);  
        }  
    }  
}
```



menambahkan pemanggilan method

```
System.out.println(x:"=====");
System.out.println(x:"Hasil Pangkat dengan Brute Force");
for (int i = 0; i < elemen; i++) {
    System.out.println("Nilai " + pangkatArr[i].nilai + " pangkat " + pangkatArr[i].pangkat + " adalah : " + pangkatArr[i].pangkatBF(pang
);
```

```
System.out.println(x:"=====");
System.out.println(x:"Hasil Pangkat dengan Divide and Conquer");
for (int i = 0; i < elemen; i++) {
    System.out.println("Nilai " + pangkatArr[i].nilai + " pangkat " + pangkatArr[i].pangkat + " adalah : " + pangkatArr[i].pangkatDC(pang
);
```

hasil

```
Masukkan jumlah elemen yang ingin dihitung : 2
Masukkan nilai yang akan dipangkatkan ke-1 : 6
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 4
Masukkan nilai pemangkat ke-2 : 3
```

```
=====
Hasil Pangkat dengan Brute Force
Nilai 6 pangkat 2 adalah : 64
Nilai 4 pangkat 3 adalah : 81
```

```
=====
Hasil Pangkat dengan Divide and Conquer
Nilai 6 pangkat 2 adalah : 64
Nilai 4 pangkat 3 adalah : 81
```

#### 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

```
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));
```

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 melakukan perulangan secara runtut dan setiap iterasinya akan mengalikan hasil bilangan dengan pangkat. Method divide conquer menggunakan metode rekursif, dengan melakukan cek apakah pangkat bernilai 0 atau tidak dan mengecek juga apakah bilangan ganjil atau tidak
2. Pada tahap awal akan di cek apakah bilangan ganjil atau tidak, jika ganjil maka fungsi akan menghitung pangkat dan dikalikan imbuhan diakhir akan dilakukan dengan nilai. jika genap maka akan fungsi menghitung pangkat dikalikan dengan fungsi penghitung pangkat
- 3.

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

4.

```
public int pangkatBF(int a, int n) {  
    int hasil = 1;  
    for (int i = 0; i < n; i++) {  
        hasil = hasil * a;  
    }  
    return hasil;  
}
```

5.

```
while (true) {  
    System.out.println();  
    System.out.println(x:"=====");  
    System.out.println(x:"Menu Perhitungan Pangkat");  
    System.out.println(x:"=====");  
  
    System.out.println(x:"1. Brute Force");  
    System.out.println(x:"2. Divide and Conquer");  
    System.out.println(x:"3. Keluar");  
    System.out.println(x:"-----");  
    System.out.print(s:"Masukkan pilihan Anda (1/2/3): ");  
    pilihan = sc.nextInt();  
}
```

```
Masukkan jumlah elemen yang ingin dihitung : 2  
Masukkan nilai yang akan dipangkatkan ke-1 : 6  
Masukkan nilai pemangkat ke-1 : 2  
Masukkan nilai yang akan dipangkatkan ke-2 : 4  
Masukkan nilai pemangkat ke-2 : 3
```

```
=====
```

```
Menu Perhitungan Pangkat
```

```
=====
```

```
1. Brute Force  
2. Divide and Conquer  
3. Keluar
```

```
-----
```

```
Masukkan pilihan Anda (1/2/3): 2
```

```
=====
```

```
Hasil Pangkat dengan Divide and Conquer  
Nilai 6 pangkat 2 adalah : 64  
Nilai 4 pangkat 3 adalah : 81
```

#### Percobaan 4.4

membuat class, atribut, dan konstruktor

```
public class Sum11 {  
    public int elemen;  
    public double keuntungan[];  
    public double total;  
  
    Sum11(int elemen) {  
        this.elemen = elemen;  
        this.keuntungan = new double[elemen];  
        this.total = 0;  
    }  
}
```

menambahkan method

```
double totalBF(double arr[]) {  
    for (int i = 0; i < elemen; i++) {  
        total = total + arr[i];  
    }  
    return total;  
}
```

menambahkan method

```
double totalDC(double arr[], int l, int r)  
{  
    if (l == r) {  
        return arr[l];  
    } else if (l < r) {  
        int mid = (l + r) / 2;  
        double lsum = totalDC(arr, l, mid);  
        double rsum = totalDC(arr, mid + 1, r);  
        return lsum + rsum;  
    }  
    return 0;  
}
```

membuat class main

```
public class MainSum11 {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println(x:"=====");  
        System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta. Misal 5,9)");  
  
        System.out.print(s:"Masukkan jumlah bulan : ");  
        int elm = sc.nextInt();  
    }  
}
```

menampilkan hasil

```
System.out.println(x:"=====");
System.out.println(x:"Algoritma brute force");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+ " bulan adalah = "+ sm.totalBF(sm.keuntungan));
System.out.println(x:"=====");
System.out.println(x:"Algoritma didvide conquer");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+ " bulan adalah = "+ sm.totalDC(sm.keuntungan, 1:0, sm.elemen-1));
}
```

hasil

```
=====
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5,9)
Masukkan jumlah bulan : 5
=====
Masukkan untung bulan ke - 1 = 8.5
Masukkan untung bulan ke - 2 = 9.54
Masukkan untung bulan ke - 3 = 7.2
Masukkan untung bulan ke - 4 = 9.1
Masukkan untung bulan ke - 5 = 6
=====
Algoritma brute force
Total keuntungan perusahaan selama 5 bulan adalah = 40.339999999999996
=====
Algoritma didvide conquer
Total keuntungan perusahaan selama 5 bulan adalah = 40.339999999999996
```

#### 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!

```
return lsum+rsum+arr[mid];
```

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

2.

```
System.out.println(x:"=====");
System.out.println(x:"Algoritma brute force");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+ " bulan adalah = "+String.format(format:"%.2f", sm.totalBF(sm.keuntungan)) );
System.out.println(x:"=====");
System.out.println(x:"Algoritma didvide conquer");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+ " bulan adalah = "+String.format(format:"%.2f", sm.totalDC(sm.keuntungan, 1:0, sm
```



3. Menghitung jumlah dari total elemen yang dihitung
4. mid digunakan untuk menampung nilai tengah dan menentukan nilai batas rekursif
- 5.

```

Sum11[] perusa = new Sum11[elm];

for (int i = 0; i < elm; i++) {
    System.out.print("Masukkan jumlah bulan untuk perusahaan ke-" + (i + 1) + " : ");
    int month = sc.nextInt();
    perusa[i] = new Sum11(month);

    System.out.println("Masukkan keuntungan untuk perusahaan ke-" + (i + 1));
    for (int j = 0; j < perusa[i].elemen; j++) {
        System.out.print("Masukkan untung bulan ke-" + (j + 1) + " : ");
        perusa[i].keuntungan[j] = sc.nextDouble();
    }
}

System.out.println(x:"=====");
for (int i = 0; i < elm; i++) {
    System.out.println("Perusahaan ke-" + (i + 1));
    System.out.println(x:"Algoritma Brute Force");
    System.out.println("Total keuntungan perusahaan selama " + perusa[i].elemen + " bulan adalah - " + String.format(format:"%.2f",perusa[i].totalBF(perusa[i].keuntungan)));
    System.out.println(x:"Algoritma Divide Conquer");
    System.out.println("Total keuntungan perusahaan selama " + perusa[i].elemen + " bulan adalah - " + String.format(format:"%.2f",perusa[i].totalDC(perusa[i].keuntungan)));
    System.out.println(x:"=====");
}

```

#### Latihan 4.5

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

```

public class Akar11 {
    public double num;

    public double akarBF(double num) {
        double low = 0, high = num, mid;
        while (low <= high) {
            mid = low + (high - low) / 2;
            if (mid * mid == num) {
                return mid;
            } else if (mid * mid < num) {
                low = mid + 0.00001;
            } else {
                high = mid - 0.00001;
            }
        }
        return low;
    }

    public double akarDC(double num, double low, double high) {
        double mid = low + (high - low) / 2;
        if (high - low < 0.00001) {
            return mid;
        }
        if (mid * mid == num) {
            return mid;
        } else if (mid * mid < num) {
            return akarDC(num, mid, high);
        } else {
            return akarDC(num, low, mid);
        }
    }
}

```

```
import java.util.Scanner;

public class MainAkar11 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println(x: "=====");
        System.out.print(s: "Masukkan bilangan yang ingin dihitung akarnya : ");
        double num = sc.nextDouble();

        Akar11 akar = new Akar11();
        System.out.println(x: "=====");
        System.out.println(x: "Hasil akar dengan Brute Force");
        System.out.println("Akar dari " + num + " adalah : " + akar.akarBF(num));

        System.out.println(x: "=====");
        System.out.println(x: "Hasil akar dengan Divide and Conquer");
        System.out.println("Akar dari " + num + " adalah : " + akar.akarDC(num, low:0, num));
    }
}
```