

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET PERTEMUAN KE-4**



**NAMA : ALVINO VALERIAN D.R**

**KELAS : 1A**

**NO. ABSEN : 05**

**NIM : 2341720027**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2024**

```

package minggu5;

import java.util.Scanner;

public class MainFaktorial {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("=====
        ==");
        System.out.print("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.print("Masukkan Nilai Data Ke-"+(i+1)+" : ");
            fk[i].nilai = sc.nextInt();
        }

        System.out.println("=====
        ==");
        System.out.println("Hasil Faktorial Dengan Brute Force");
        for(int i=0 ; i < elemen ;i++){
            System.out.println("Faktorial Dari Nilai"+fk[i].nilai+"
        Adalah : "+fk[i].faktorialIBF(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("=====
    ==");

    sc.close();
}
}

```

```

package minggu5;

public class Faktorial {
    public int nilai;

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

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

```

```
ming\Code\User\workspaceStorage\658e11305cdc710f8051e8962d122877\
=====
Masukkan Jumlah Elemen Yang Ingin Dihitung : 3
Masukkan Nilai Data Ke-1 : 8
Masukkan Nilai Data Ke-2 : 5
Masukkan Nilai Data Ke-3 : 3
=====
Hasil Faktorial Dengan Brute Force
Faktorial Dari Nilai8 Adalah : 40320
Faktorial Dari Nilai5 Adalah : 120
Faktorial Dari Nilai3 Adalah : 6
=====
Hasil Faktorial Dengan Divide and Conquer
Faktorial Dari Nilai8 Adalah : 40320
Faktorial Dari Nilai5 Adalah : 120
Faktorial Dari Nilai3 Adalah : 6
=====
D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet4>
```

#### 4.2.3 Pertanyaan

1. Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!
2. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for?Buktikan!
3. Jelaskan perbedaan antara **fakto \*= i;** dan **int fakto = n \* faktorialDC(n-1); !**

JAWAB:

1. Ketika  $n == 1$ , maka proses rekursi akan dihentikan. Selama  $n != 1$ , maka fungsi faktorialDC akan memanggil dirinya dengan  $n-1$  sampai mencapai  $n==1$ . Ketika  $n==1$ , maka faktorialDC(n) akan mengembalikan nilai  $n$  saat itu. Kemudian nilai  $1$  akan mengembalikan nilai faktorialDC(n) sebelumnya. Kemudian nilai faktorialDC(n) tersebut akan dikalikan dengan nilai sebelumnya.
2. Memungkinkan. Menggunakan while loop

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

```

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

```

3. Perbedaannya ialah, ketika fakto \*= i maka perkaliannya akan runtut dari awal. Sedangkan untuk int fakto = n\*faktorialDC(n-1) dibagi menjadi sub masalah kecil, selama n != 1, maka akan n-1. Ketika sudah mencapai 1, maka secara rekursif akan mengalikan n \* factorial(n-1) sampai ke n awal. Kemudian hasil perkalian tersebut akan dicombine.

```

package Pekan5;

public class Pangkat05 {
    public int nilai, pangkat;

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

```

```

package Pekan5;

import java.util.Scanner;

```

```

public class PangkatMain05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=====");
        System.out.print("Masukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc.nextInt();

        Pangkat05[] png = new Pangkat05[elemen];

        System.out.println("=====");
        for (int i = 0; i < elemen; i++) {
            png[i] = new Pangkat15();
            System.out.print("Masukkan nilai yang akan dipangkatkan ke-
" + (i+1) + " : ");
            png[i].nilai = sc.nextInt();
            System.out.print("Masukkan nilai pemangkat ke-" + (i+1) + " : ");
            png[i].pangkat = sc.nextInt();
        }
        sc.close();

        System.out.println("=====");
        System.out.println("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));
        }

        System.out.println("=====");
        System.out.println("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));
        }
    }
}

```

```

=====
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 - 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 1
Hasil dari 4 pangkat 3 adalah 64

```

#### 4.3.3 Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu PangkatBF() dan PangkatDC()!
2. Apakah tahap *combine* sudah termasuk dalam kode tersebut? Tunjukkan!
3. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.
4. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan switch-case!

JAWAB:

1. Perbedaan 2 method tersebut terletak pada proses alurnya dimana pada proses pangkatBF akan dilakukan secara berurutan mulai dari awal hingga syaratnya terpenuhi, sedangkan pada PangkatDC setiap nilai dibedakan menjadi proses tertentu seperti contoh pada saat  $n==1$  akan mereturn  $a$ , pada saat  $n\%2==1$  akan mereturn  $\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2) * a$
2. Ada namun tidak ter jelaskan secara eksplisit, tahap combine terletak pada Panggilan rekursif menangani bagian "bagi" dengan membagi eksponen ( $n$ ) menjadi separuh. Penggandaan nilai yang dikembalikan dari panggilan rekursif  $((a^{(n/2)})^2$  atau  $(a^{(n/2)})^2 * a$  pada dasarnya menggabungkan hasil untuk mendapatkan jawaban akhir  $a^n$ .

```

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. Terjadi perubahan pada main

```

package Pekan5;

import java.util.Scanner;

public class PangkatMain05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

System.out.println("=====");
System.out.print("Masukkan jumlah elemen yang ingin dihitung: ");
int elemen = sc.nextInt();

Pangkat05[] png = new Pangkat05[elemen];

System.out.println("=====");
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 angkat = sc.nextInt();

    png[i] = new Pangkat05(nilai, angkat);
}
sc.close();

System.out.println("=====");
System.out.println("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));
}

System.out.println("=====");
System.out.println("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));
}
}
}

```



```

=====
Masukkan jumlah elemen yang ingin dihitung: 3
=====
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
Masukkan nilai yang akan dipangkatkan ke-3 : 2
Masukkan nilai pemangkat ke-3 : 5
=====
HASIL PANGKAT - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
Hasil dari 2 pangkat 5 adalah 32
=====
HASIL PANGKAT - DIVIDE CONQUER
Hasil dari 6 pangkat 2 adalah 1
Hasil dari 4 pangkat 3 adalah 64
Hasil dari 2 pangkat 5 adalah 2
=====

```

4. package pekan5;

```

import java.util.Scanner;

public class MainPangkat05 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("=====");
        System.out.println("Menghitung Pangkat:");
        System.out.println("1. Brute Force");
        System.out.println("2. Divide Conquer");
        System.out.println("=====");
        System.out.print("Masukkan Pilihan: ");

        int pilihan = sc.nextInt();

        System.out.println("=====");
        System.out.print("Masukkan jumlah elemen yang ingin dihitung: ");
    }
}

```

```

int elemen = sc.nextInt();

Pangkat05[] png = new Pangkat05[elemen];

switch (pilihan) {
    case 1:
        System.out.println("=====");
        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 angkat = sc.nextInt();

            png[i] = new Pangkat05(nilai, angkat);
        }

        System.out.println("=====");
        System.out.println("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("=====");
        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 angkat = sc.nextInt();

        png[i] = new Pangkat05(nilai, angkat);
    }

    System.out.println("=====");
    System.out.println("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:
        sc.close();
        break;
    }
}

@Override
public String toString() {
    return "MainPangkat05 []";
}
}

```

```

1. Brute Force(+ Shift+U)
2. Divide Conquer
=====
Masukkan Pilihan: 1
=====
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 - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet4>

```

1. Mengapa terdapat formulasi *return value* berikut?Jelaskan
2. Kenapa dibutuhkan variable mid pada method TotalDC()?
3. 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!

JAWAB:

1. Karena command tersebut didalam methodnya digunakan untuk menggabungkan hasil parsial yaitu lsum dan rsum dari panggilan rekursif secara efisien
2. Karena sesuai dengan system kerja dari algoritmanya dimana algoritma tersebut akan membagi data menjadi beberapa bagian yang kemudia di hitung masing masing yang nantinya akan disatukan
- 3.

```

package minggu5;

import java.util.Scanner;

public class sumMain05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=====");
        System.out.print("Jumlah perusahaan yang akan dihitung: ");
        int prh = sc.nextInt();

        for (int i = 0; i < prh; i++) {

```

```

System.out.println("=====");

        System.out.println("Perusahaan ke-"+(i+1));

        System.out.println("Progres Menghitung Keuntungan Total
(Satuan Juta, Misal 5.9)");

        System.out.print("Masukkan jumlah bulan: ");

        int elemen = sc.nextInt();

        sum05 sm = new sum05(elemen);

System.out.println("=====");

        for (int j = 0; j < sm.elemn; j++) {
            System.out.print("Masukkan untung bulan ke - " +
(j+1) + " = ");

            sm.keuntungan[j] = sc.nextDouble();

        }

System.out.println("=====");

        System.out.println("Algoritma Brute Force");

        System.out.println("Total keuntungan perusahaan selama "
+ sm.elemn + " bulan adalah " + sm.totalBF(sm.keuntungan));

System.out.println("=====");

        System.out.println("Algoritma Divide Conquer");

        System.out.println("Total keuntungan perusahaan selama "
+ sm.elemn + " bulan adalah " + sm.totalDC(sm.keuntungan, 0,
sm.elemn-1)+"\n");

        }

        sc.close();

    }

}

```

```
package minggu5;

public class sum05 {
    int elemen;
    double[] keuntungan;
    double total;

    public sum05(int elemen) {
        this.elemen = elemen;
        this.keuntungan = new double[elemen];
        this.total = 0;
    }

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

    public 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-1);
            double rsum = totalDC(arr, mid+1, r);
            return lsum+rsum+arr[mid];
        }
        return 0;
    }
}
```

## LATIHAN PRAKTIKUM

```
package Praktikum;

public class mobil05 {
    public int akselerasi, kekuatan;

    int cariMaksimumDQ(int arr[], int s, int f){
        if (s == f) {
            return arr[s];
        }
        int mid = (s + f) / 2;
        int lmax = cariMaksimumDQ(arr, s, mid);
        int rmax = cariMaksimumDQ(arr, mid + 1, f);
        int max = 0;
        if (lmax > max) {
            max = lmax;
        }else if (rmax > max) {
            max = rmax;
        }
        return max;
    }

    int cariMinimumDQ(int arr[], int s, int f){
        if (s == f) {
            return arr[s];
        }
        int mid = (s + f) / 2;
        int lmin = cariMinimumDQ(arr, s, mid);
        int rmin = cariMinimumDQ(arr, mid + 1, f);
        if (lmin < rmin) {
            return lmin;
        } else {
```

```

        return rmin;

    }

}

double hitungRerataBF(int[] n, int elm) {
    int total = 0;
    for (int i = 0; i < elm; i++) {
        total += n[i];
    }
    return (double) total / elm;
}
}

```

```

package Praktikum;

import java.util.Scanner;

public class mobilMain05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] aks = {6816, 3921, 4360, 4058, 6255, 3700, 4180, 4180};
        int[] pow = {728, 575, 657, 609, 703, 553, 609, 631};

        mobil05[] mbl = new mobil05[7];
        for (int i = 0; i < mbl.length; i++) {
            mbl[i] = new mobil05();
            mbl[i].akselerasi = aks[i];
            mbl[i].kekuatan = pow[i];
        }

        System.out.println("Pilih Menu: ");
        System.out.println("1. Cari Minimum DQ");
        System.out.println("2. Cari Maximum DQ");
        System.out.println("3. Hitung Rerata DQ");
        System.out.println("=====");
    }
}

```



```

int pil = sc.nextInt();
switch (pil) {
    case 1:
        int akselerasiminim = 0;
        for (int i = 0; i < mbl.length; i++) {
            akselerasiminim = mbl[i].cariMinimumDQ(aks, 0,
aks.length-1);

        }
        System.out.println("top speed minimum adalah " +
akselerasiminim);

        break;
    case 2 :
        int akselerasimaksim = 0;
        for (int i = 0; i < mbl.length; i++) {
            akselerasimaksim = mbl[i].cariMaksimumDQ(aks, 0,
aks.length-1);

        }
        System.out.println("top speed maksimum adalah " +
akselerasimaksim);
        break;
    case 3:
        double rerata = mbl[0].hitungRerataBF(pow, pow.length);
        System.out.println("Rerata kekuatan mobil adalah: " +
rerata);

        break;
    default:
        System.out.println("Pilihan Salah");
        break;
}
sc.close();
}
}

```

1. Cari Minimum DQ
2. Cari Maximum DQ
3. Hitung Rerata DQ

=====

1

top speed minimum adalah 3700

D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet4>

1. Cari Minimum DQ
2. Cari Maximum DQ
3. Hitung Rerata DQ

=====

2

top speed maksimum adalah 6816

D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet4>

Pilih Menu:

1. Cari Minimum DQ
2. Cari Maximum DQ
3. Hitung Rerata DQ

=====

3

Rerata kekuatan mobil adalah: 633.125

D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet4>