

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET 4**



**NAMA : DIMAS ADI BAYU SAMUDRA**

**KELAS : 1A**

**NO. ABSEN : 08**

**NIM : 2341720169**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## PRAKTIKUM

### PERCOBAAN 1

#### Code factorial

```
package jobsheet4.BruteForceDivideConquer08;

public class Faktorial08 {
    int nilai;

    public Faktorial08() {
        this.nilai = 1;
    }

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

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

#### Code main

```
package jobsheet4.BruteForceDivideConquer08;

import java.util.Scanner;

public class mainFaktorial08 {

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

        System.out.println("Masukkan jumlah elemen: ");
        int jml = sc.nextInt();

        Faktorial08[] fk = new Faktorial08[jml];
        for (int i = 0; i < jml; i++) {
            fk[i] = new Faktorial08();

            System.out.println("Masukkan nilai data ke-" + (i + 1) + ": ");
            int iNilai = sc.nextInt();
            fk[i].nilai = iNilai;
        }
    }
}
```

```

    }

    System.out.println("\nMETODE BRUTE FORCE");
    for (int i = 0; i < jml; i++) {
        System.out.println("Hasil perhitungan faktorial " + fk[i].nilai
+ " adalah " + fk[i].faktorialBF(fk[i].nilai));
    }

    System.out.println("\nMETODE DIVIDE AND CONQUER");
    for (int i = 0; i < jml; i++) {
        System.out.println("Hasil perhitungan faktorial " + fk[i].nilai
+ " adalah " + fk[i].faktorialDC(fk[i].nilai));
    }
}
}

```

### Hasil run

```

Masukkan jumlah elemen:
3
Masukkan nilai data ke-1:
5
Masukkan nilai data ke-2:
8
Masukkan nilai data ke-3:
3

METODE BRUTE FORCE
Hasil perhitungan faktorial 5 adalah 120
Hasil perhitungan faktorial 8 adalah 40320
Hasil perhitungan faktorial 3 adalah 6

METODE DIVIDE AND CONQUER
Hasil perhitungan faktorial 5 adalah 120
Hasil perhitungan faktorial 8 adalah 40320
Hasil perhitungan faktorial 3 adalah 6

```

### 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. Kondisi if:

- Mengecek apakah n sama dengan 0.
- Jika ya, berhenti dan langsung kembalikan nilai 1 (karena  $0! = 1$ ).

Kondisi else:

- Dilakukan ketika n tidak sama dengan 0.
- Hitung faktorial:  $n * \text{faktorial}(n - 1)$ .
- Algoritma membagi masalah (faktorial n) menjadi 2 sub-masalah:
- Faktorial (n - 1)
- Perkalian n dengan hasil faktorial (n - 1)

2. mungkin saja , kita bisa menggunakan while loop, do while, ataupun rekursi (tanpa loop)

Contoh dengan while loop

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

3.

- fakto \*= i: Digunakan dalam perulangan untuk menghitung faktorial secara iteratif (metode Brute Force).
- int fakto = n \* faktorialDC(n-1): Digunakan dalam rekursi untuk menghitung faktorial secara rekursif (metode Divide and Conquer).

## PERCOBAAN 2

Code pangkat

```
package jobsheet4.Pangkat;  
  
public class Pangkat08 {  
    public void setNilai(int nilai) {  
        this.nilai = nilai;  
    }  
  
    public void setPangkat(int pangkat) {  
        this.pangkat = pangkat;  
    }  
  
    public int nilai, pangkat;  
}
```

```

    public int pangkatBF(int nilai, int pangkat){
        int hasil = 1;
        for (int i=0; i<pangkat;i++){
            hasil = hasil*nilai;
        }
        return hasil;
    }

    public int pangkatDC(int nilai, int pangkat){
        if(pangkat==0){
            return 1;
        }
        else {
            if (pangkat%2==1)//bil ganjil
                return (pangkatDC(nilai,pangkat/2)*pangkatDC(nilai,
pangkat/2)*nilai);
            else //bil. genap
                return (pangkatDC(nilai,pangkat/2)*pangkatDC(nilai,
pangkat/2));
        }
    }
}

```

#### Code main

```

package jobsheet4.Pangkat;

import java.util.Scanner;

public class mainPangkat08 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

        Pangkat08[] pangkat = new Pangkat08[elemen];
        for (int i = 0; i < elemen; i++) {

            pangkat[i] = new Pangkat08();
            System.out.println("Masukkan nilai yang hendak dipangkatkan: ");
            int nilai = sc.nextInt();
            System.out.println("Masukkan nilai pemangkat: ");
            int pangkatN = sc.nextInt();

            pangkat[i].setNilai(nilai);
            pangkat[i].setPangkat(pangkatN);
        }

        for (int i = 0; i < elemen; i++) {
            pangkat[i].pangkatBF(pangkat[i].nilai, pangkat[i].pangkat);
            pangkat[i].pangkatDC(pangkat[i].nilai, pangkat[i].pangkat);
        }
    }
}

```

```

    }
    System.out.println("HASIL PANGKAT BRUTE FORCE");
    for (int i = 0; i < elemen; i++) {
        System.out.println("Hasil dari " + pangkat[i].nilai + " pangkat " +
pangkat[i].pangkat + " adalah " + pangkat[i].pangkatBF(pangkat[i].nilai,
pangkat[i].pangkat));
    }

    System.out.println("HASIL PANGKAT DIVIDE AND CONQUER");
    for (int i = 0; i < elemen; i++) {
        System.out.println("Hasil dari " + pangkat[i].nilai + " pangkat " +
pangkat[i].pangkat + " adalah " + pangkat[i].pangkatDC(pangkat[i].nilai,
pangkat[i].pangkat));
    }
}
}

```

Hasil run

```

=====
Masukkan jumlah elemen yang akan dihitung :
2
Masukkan nilai yang hendak dipangkatkan:
6
Masukkan nilai pemangkat:
2
Masukkan nilai yang hendak dipangkatkan:
4
Masukkan nilai pemangkat:
3
HASIL PANGKAT BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
HASIL PANGKAT DIVIDE AND 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()!
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.

PangkatBF():

- Menggunakan perulangan untuk menghitung pangkat secara iteratif.
- Mengalikan basis dengan dirinya sendiri sebanyak nilai pangkat.
- Lebih efisien untuk menghitung pangkat dengan nilai pangkat kecil.
- Rumus:  $a^n = a * a * \dots * a$  (n kali)
- Menghitung pangkat secara langsung dengan perulangan.

PangkatDC():

- Menggunakan rekursi untuk menghitung pangkat secara rekursif.
- Membagi masalah menjadi sub-masalah yang lebih kecil dan menghitungnya secara rekursif.
- Lebih efisien untuk menghitung pangkat dengan nilai pangkat besar.
- Rumus:  $a^0 = 1$
- $a^n = a * a^{(n-1)}$  (n ganjil)
- $a^n = (a^{(n/2)}) * (a^{(n/2)})$  (n genap)
- Membagi masalah menjadi dua sub-masalah dan menghitungnya secara rekursif.

2.

```
if (pangkat%2==1)//bil ganjil
return (pangkatDC(nilai,pangkat/2)*pangkatDC(nilai, pangkat/2)*nilai);
else //bil. genap
return (pangkatDC(nilai,pangkat/2)*pangkatDC(nilai, pangkat/2));
```

3. Code pangkat

```
public class Pangkat08 {

    private int nilai;
    private int pangkat;

    public Pangkat08(int nilai, int pangkat) {
        this.nilai = nilai;
        this.pangkat = pangkat;
    }

    public int pangkatBF() {
        int hasil = 1;
        for (int i = 0; i < pangkat; i++) {
            hasil *= nilai;
        }
        return hasil;
    }

    public int pangkatDC() {
        if (pangkat == 0) {
```

```

        return 1;
    } else {
        if (pangkat % 2 == 1) {
            return (pangkatDC() * pangkatDC() * nilai);
        } else {
            return (pangkatDC() * pangkatDC());
        }
    }
}
}

```

#### Code main

```

import java.util.Scanner;

public class mainPangkat08 {

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

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

        Pangkat08[] pangkat = new Pangkat08[elemen];
        for (int i = 0; i < elemen; i++) {
            System.out.println("Masukkan nilai yang hendak dipangkatkan: ");
            int nilai = sc.nextInt();
            System.out.println("Masukkan nilai pemangkat: ");
            int pangkatN = sc.nextInt();

            pangkat[i] = new Pangkat08(nilai, pangkatN);
        }

        for (int i = 0; i < elemen; i++) {
            pangkat[i].pangkatBF();
            pangkat[i].pangkatDC();
        }

        System.out.println("HASIL PANGKAT BRUTE FORCE");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + pangkat[i].nilai + " pangkat "
                + pangkat[i].pangkat + " adalah " + pangkat[i].pangkatBF());
        }

        System.out.println("HASIL PANGKAT DIVIDE AND CONQUER");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + pangkat[i].nilai + " pangkat "
                + pangkat[i].pangkat + " adalah " + pangkat[i].pangkatDC());
        }
    }
}

```



#### 4. Code pangkat

```
package jobsheet4.Pangkat;

public class Pangkat08 {
    public int nilai, pangkat;

    Pangkat08(int n, int p){
        nilai = n;
        pangkat = p;
    }

    public void menu(){

    }

    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)//bilangan ganjil
                return (PangkatDC(a, n / 2) * PangkatDC(a, n / 2) * a);
            else
                return (PangkatDC(a, n / 2) * PangkatDC(a, n / 2));
        }
    }
}
```

#### Code main

```
package jobsheet4.Pangkat;

import java.util.Scanner;

public class mainPangkat08 {

    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();

        Pangkat08[] png = new Pangkat08[elemen];
        for (int i = 0; i < elemen; i++) {

            System.out.println("Masukkan nilai yang akan dipangkatkan ke- "
+ (i + 1) + " : ");
```

```

        int nilai = sc.nextInt();
        System.out.println("Masukkan nilai pemangkat ke- " + (i + 1) + "
: ");

        int pang = sc.nextInt();
        png[i] = new Pangkat08(nilai, pang);
    }
    for (int i = 0; i < elemen; i++) {
        png[i].menu();
        do {

System.out.println("=====");
            System.out.println("Silahkan Pilih Menu :");
            System.out.println("1. Brute Force");
            System.out.println("2. Devide and Conquer");
            System.out.println("3. Exit");
            System.out.println("");
            System.out.print("Pilihan Menu Anda (1/2/3): ");
            int menu = sc.nextInt();
            switch (menu) {
                case 1: {
                    System.out.println("=====");
                    System.out.println("Hasil Pangkat dengan Brute Force
");
                        for (int j = 0; j < elemen; j++) {
                            System.out.println("Nilai " + png[j].nilai + "
pangkat " + png[j].pangkat + " adalah : " + png[j].PangkatBF(png[j].nilai,
png[j].pangkat));
                        }
                    }
                case 2: {
                    System.out.println("=====");
                    System.out.println("Hasil Pangkat dengan Devide and
Conquer ");
                        for (int j = 0; j < elemen; j++) {
                            System.out.println("Nilai " + png[j].nilai + "
pangkat " + png[j].pangkat + " adalah : " + png[j].PangkatDC(png[j].nilai,
png[j].pangkat));
                        }
                    }
                case 3: {
                    break;
                }
            }
        } while (true);
    }
    System.out.println("=====");
}
}

```

### PERCOBAAN 3

#### Code sum

```
package jobsheet4.Sum;

public class Sum08 {
    int elemen;
    double keuntungan[], total;

    Sum08(int elemen){
        this.elemen = elemen;
        this.keuntungan = new double[elemen];
        this.total = 0;
    }

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

    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;
    }
}
```

#### Code main

```
package jobsheet4.Sum;

import java.util.Scanner;

public class mainSum08 {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("=====");
        System.out.println("Program menghitung Keuntungan Total");
        System.out.print("Masukkan jumlah bulan :");
        int elm = sc.nextInt();

        Sum08 sm = new Sum08(elm);
        System.out.println("=====");
        for (int i = 0; i < sm.elemen; i++) {
            System.out.print("Masukkan keuntungan bulan ke - " + (i + 1)+ "
= ");
        }
    }
}
```

```

        sm.keuntungan[i] = sc.nextDouble();
    }

    System.out.println("=====
=====");
        System.out.println("ALGORITMA BRUCE FORCE");
        System.out.println("Total Keuntungan perusahaan selama " + sm.elemen
+ " bulan adalah " + sm.totalBF(sm.keuntungan));

    System.out.println("=====
=====");
        System.out.println("Algoritma Divide Conquer");
        System.out.println("Total keuntungan perusahaan selama " + sm.elemen
+ " bulan adalah "+sm.totalDC(sm.keuntungan, 0, sm.elemen-1));

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

```

Hasil run

```

=====
Program menghitung Keuntungan Total
Masukkan jumlah bulan :5
=====
Masukkan keuntungan bulan ke - 1 = 8.5
Masukkan keuntungan bulan ke - 2 = 9.54
Masukkan keuntungan bulan ke - 3 = 7.2
Masukkan keuntungan bulan ke - 4 = 9.1
Masukkan keuntungan bulan ke - 5 = 6
=====
ALGORITMA BRUCE FORCE
Total Keuntungan perusahaan selama 5 bulan adalah 40.339999999999996
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah 40.34
=====

```

Pertanyaan

1. Mengapa terdapat formulasi return value berikut?Jelaskan!

```

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

```

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. i: Variabel indeks yang digunakan dalam algoritma. Perannya tergantung pada implementasi spesifik Divide and Conquer.

rsum: Merupakan total nilai elemen pada sub-array kanan. Dalam Divide and Conquer, masalah dibagi menjadi sub-masalah, dan "right sum" mengacu pada total elemen di sub-array kanan.

arr[mid]: Elemen pada indeks tengah array.

2.

- Membagi Array
- Menghitung Sub-Array
- Menggabungkan Hasil
- Menghindari Perulangan Berlebihan

3. Code sum

```
package jobsheet4.Sum;

public class Sum08 {
    int elemen;
    double keuntungan[];
    double total;

    Sum08(int elemen){
        this.elemen = elemen;
        this.keuntungan = new double[elemen];
        this.total = 0;
    }

    Sum08() {

    }

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

    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;
    }
}
```

```
}  
}
```

#### Code main

```
package jobsheet4.Sum;  
  
import java.util.Scanner;  
  
public class mainSum08 {  
  
    public static void main(String [] args){  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("===== "  
        "====");  
        System.out.println("Program Menghitung Keuntungan Total (Satuan  
        Juta, Misal 5.9)");  
        System.out.println("Masukkan jumlah Perusahaan : ");  
        int perusahaan = sc.nextInt();  
  
        Sum08[] sm = new Sum08[perusahaan];  
        for (int i=0; i<perusahaan; i++){  
            sm[i] = new Sum08();  
  
        System.out.println("===== "  
        "====");  
            System.out.println("Perusahaan ke-"+(i+1));  
            System.out.println("Masukkan jumlah bulan : ");  
            sm[i].elemen = sc.nextInt();  
  
        System.out.println("===== "  
        "====");  
            sm[i].keuntungan = new double [sm[i].elemen];  
            for (int j=0; j<perusahaan; j++){  
                System.out.println("Masukkan untung bulan ke-"+(j+1)+" =  
            ");  
                sm[i].keuntungan[j] = sc.nextDouble();  
            }  
        }  
  
        System.out.println("===== "  
        "====");  
  
        Sum08 dat = new Sum08();  
        for(int i=0; i<perusahaan; i++){  
            System.out.println("Perusahaan ke-"+(i+1));  
            System.out.println("Jumlah Bulan : "+ sm[i].elemen);  
            System.out.println("");  
            System.out.println("Algoritma Brute Force");  
            System.out.printf("Total keuntungan perusahaan selama "+  
        sm[i].elemen  
            + " bulan adalah = %.2f ", dat.totalBF(sm[i].keuntungan));  
        }  
    }  
}
```

```

                System.out.println("\n");
                System.out.println("Algoritma Divide Conquer");
                System.out.printf("Total keuntungan perusahaan selama "+
sm[i].elemen
                + " bulan adalah = %.2f ", dat.totalDC(sm[i].keuntungan, 0,
sm[i].elemen-1));

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

        }

    }

}

```

## Latihan Praktikum

### Code latihan

```

package jobsheet4.latihan1;

public class latihan08 {
    int[] Accel;
    int[] topPower;

    latihan08(int[] Accel, int[] topPower){
        this.Accel = Accel;
        this.topPower = topPower;
    }

    int highestAccel(){
        if (Accel.length == 1) {
            return Accel[0];
        }
        return highestAccel(Accel, 0, Accel.length -1);
    }

    int highestAccel(int[] data, int l, int r){
        if (l == r) {
            return data[l];
        }

        int mid = (r + l) / 2;
        int lAccel = highestAccel(data, l, mid);
        int rAccel = highestAccel(data, mid + 1, r);

        if (lAccel > rAccel) {
            return lAccel;
        } else {
            return rAccel;
        }
    }

    int lowAccel(){

```

```

        if (Accel.length == 1) {
            return Accel[0];
        }
        return lowAccel(Accel, 0, Accel.length - 1);
    }

    int lowAccel(int[] data, int l, int r){
        if (l == r) {
            return data[l];
        }

        int mid = (r + l) / 2;
        int lAccel = lowAccel(data, l, mid);
        int rAccel = lowAccel(data, mid + 1, r);

        if (lAccel > rAccel) {
            return rAccel;
        } else {
            return lAccel;
        }
    }

    int avgPower(){
        int sum = 0;
        for (int i : topPower) {
            sum += i;
        }
        sum /= topPower.length;
        return sum;
    }
}

```

#### Code main

```

package jobsheet4.latihan1;

public class mainLatihan08 {
    public static void main(String[] args) {
        int[] Accel = {6816,3921,4360,4058,6255,3700,4180,4180};
        int[] topPower = {728,575,657,609,703,553,609,631};
        latihan08 latihan = new latihan08(Accel, topPower);

        System.out.println("_____");
        System.out.println("|Merk           | Tipe           | Tahun |");
        System.out.println("top_acceleration | top_power |");

        System.out.println("_____");
        System.out.println("|BMW           | M2 Coupe       | 2016  | 6816");
        System.out.println("| 728           |");
        System.out.println("|Ford          | Fiesta ST      | 2014  | 3921");
        System.out.println("| 575           |");
    }
}

```



```

        System.out.println("|Nissan          | 370Z          | 2009 | 4360
| 657          |");
        System.out.println("|Subaru        | BRZ           | 2014 | 4058
| 609          |");
        System.out.println("|Subaru        | Impreza WRX STI | 2013 | 6255
| 703          |");
        System.out.println("|Toyota        | AE86 Trueno    | 1986 | 3700
| 553          |");
        System.out.println("|Toyota        | 86/GT86        | 2014 | 4180
| 609          |");
        System.out.println("|Volkswagen    | Golf GTI       | 2014 | 4180
| 631          |");

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

        System.out.println();

        System.out.println("Top Acceleration adalah " +
latihan.highestAccel());
        System.out.println("Low Acceleataion adalah " + latihan.lowAccel());
        System.out.println("Rata Rata power adalah " + latihan.avgPower());
    }
}

```

Hasil run

Merk	Tipe	Tahun	top_acceleration	top_power
BMW	M2 Coupe	2016	6816	728
Ford	Fiesta ST	2014	3921	575
Nissan	370Z	2009	4360	657
Subaru	BRZ	2014	4058	609
Subaru	Impreza WRX STI	2013	6255	703
Toyota	AE86 Trueno	1986	3700	553
Toyota	86/GT86	2014	4180	609
Volkswagen	Golf GTI	2014	4180	631

=====

Top Acceleration adalah 6816  
Low Acceleataion adalah 3700  
Rata Rata power adalah 633