



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

## PRAKTIKUM 4 - BRUTE FORCE DAN DIVIDE CONQUER

### PERCOBAAN 1

#### 4.2 Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

```
1 package Jobsheet_4.Minggu_5.src.BruteForceDivideConquer;
2
3 public class Faktorial {
4     public int nilai;
5
6     int faktorialBF(int n) {
7         int fakto = 1;
8         for (int i = 1; i <= n; i++) {
9             fakto *= i;
10        }
11        return fakto;
12    }
13    int faktorialDC(int n) {
14        if (n == 1) {
15            return 1;
16        }
17        else {
18            int fakto = n * faktorialDC(n - 1);
19            return fakto;
20        }
21    }
22 }
```

```
1 package Jobsheet_4.Minggu_5.src.BruteForceDivideConquer;
2 import java.util.Scanner;
3
4 public class MainFaktorial {
5     public static void main(String[] args) {
6         Scanner ayleen07 = new Scanner(System.in);
7
8         System.out.println("-----");
9         System.out.print("Masukkan jumlah elemen : ");
10        int ijml = ayleen07.nextInt();
11
12        Faktorial[] fk = new Faktorial[10];
13        for (int i = 0; i < ijml; i++) {
14            fk[i] = new Faktorial();
15            System.out.print("Masukkan nilai data ke-" + (i+1) + " : ");
16            fk[i].nilai = ayleen07.nextInt();
17        }
18        System.out.println("HASIL - BRUTE FORCE");
19        for (int i = 0; i < ijml; i++) {
20            System.out.println("Hasil perhitungan faktorial menggunakan Brute Force adalah " + fk[i].faktorialBF(fk[i].nilai));
21        }
22        System.out.println("HASIL - DIVIDE AND CONQUER");
23        for (int i = 0; i < ijml; i++) {
24            System.out.println("Hasil perhitungan faktorial menggunakan Divide and Conquer adalah " + fk[i].faktorialDC(fk[i].nilai));
25        }
26    }
27 }
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
-----  
Masukkan jumlah elemen : 3  
Masukkan nilai data ke-1 : 5  
Masukkan nilai data ke-2 : 8  
Masukkan nilai data ke-3 : 3  
HASIL - BRUTE FORCE  
Hasil perhitungan faktorial menggunakan Brute Force adalah 120  
Hasil perhitungan faktorial menggunakan Brute Force adalah 40320  
Hasil perhitungan faktorial menggunakan Brute Force adalah 6  
HASIL - DIVIDE AND CONQUER  
Hasil perhitungan faktorial menggunakan Divide and Conquer adalah 120  
Hasil perhitungan faktorial menggunakan Divide and Conquer adalah 40320  
Hasil perhitungan faktorial menggunakan Divide and Conquer adalah 6  
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester 2\7. Prati-  
m_AlgoritmaStrukturData_07> █
```

### Pertanyaan

1. Pada baseline Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!

Jawab :

- **if** bagian ini merupakan kasus basis (base case) dari rekursif, yaitu **kondisi yang akan menghentikan rekursif**. Bagian ini digunakan untuk menghitung faktorial dari 1.
- **else** kasus rekursif (recursive case), **kasus ini digunakan untuk melanjutkan rekursif** di mana nilai faktorial dihitung dengan memanggil fungsi faktorial itu sendiri sehingga memecah masalah faktorial menjadi sub-masalah yang lebih kecil.

2. Apakah memungkinkan perulangan pada method faktorialBF() diubah selain menggunakan for? Buktikan!

Jawab : Perulangan pada method faktorialBF() dapat diubah selain menggunakan perulangan for, yaitu dengan menggunakan perulangan while atau do-while. Jika dijalankan, maka outputnya tetap sama.

Contoh :

```
1 int faktorialBF_while(int n) {  
2     int fakto = 1;  
3     int i = 1;  
4     while (i <= n) {  
5         fakto *= i;  
6         i++;  
7     }  
8     return fakto;  
9 }
```

```
1 int faktorialBF_dowhile(int n) {  
2     int fakto = 1;  
3     int i = 1;  
4     do {  
5         fakto *= i;  
6         i++;  
7     } while (i <= n);  
8     return fakto;  
9 }
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
-----  
Masukkan jumlah elemen : 3  
Masukkan nilai data ke-1 : 5  
Masukkan nilai data ke-2 : 8  
Masukkan nilai data ke-2 : 8  
Masukkan nilai data ke-3 : 3  
HASIL - BRUTE FORCE WHILE  
Hasil perhitungan faktorial menggunakan Brute Force While adalah 120  
Hasil perhitungan faktorial menggunakan Brute Force While adalah 40320  
Hasil perhitungan faktorial menggunakan Brute Force While adalah 6  
HASIL - BRUTE FORCE DO - WHILE  
Hasil perhitungan faktorial menggunakan Brute Force Do-While adalah 120  
Hasil perhitungan faktorial menggunakan Brute Force Do-While adalah 40320  
Hasil perhitungan faktorial menggunakan Brute Force Do-While adalah 6  
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester 2\7. Prakt
```

3. Jelaskan perbedaan antara **fakto \*= i**; dan **int fakto = n \* faktorialDC(n-1)** ; !

Jawab :

Perbedaan antara kedua baris kode tersebut terletak pada pendekatan untuk menghitung faktorial. **fakto \*= i** lebih mudah dipahami, sedangkan **int fakto = n \* faktorialDC(n-1)** lebih efisien dalam penggunaan memori.

- **fakto \*= i** : method yang menggunakan perulangan untuk menyimpan nilai **i** ke dalam variabel **fakto**.
- **int fakto = n \* faktorialDC(n-1)** : method yang menggunakan rekursif untuk menghitung nilai dari faktorial **n** dengan mengalikan **n** dari hasil faktorial (**n - 1**).

## PERCOBAAN 2

### 4.3 Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

```
1 package Jobsheet_4.Minggu_5.src.Pangkat;  
2  
3 public class Pangkat {  
4     public int nilai, pangkat;  
5  
6     int pangkatBF(int a, int n) {  
7         int hasil = 1;  
8         for (int i = 1; i <= n; i++) {  
9             hasil *= a;  
10        }  
11        return hasil;  
12    }  
13    int pangkatDC(int a, int n) {  
14        if (n == 0) {  
15            return 1;  
16        } else {  
17            if (n % 2 == 1) {  
18                return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a);  
19            } else {  
20                return (pangkatDC(a, n/2) * pangkatDC(a, n/2));  
21            }  
22        }  
23    }  
24 }
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
1 package Jobsheet_4.Minggu_5.src.Pangkat;  
2 import java.util.Scanner;  
3  
4 public class MainPangkat {  
5     public static void main(String[] args) {  
6         Scanner ayleen07 = new Scanner(System.in);  
7  
8         System.out.println("=====");  
9         System.out.print("Masukkan jumlah elemen yang dihitung : ");  
10        int elemen = ayleen07.nextInt();  
11  
12        Pangkat[] png = new Pangkat[elemen];  
13        for (int i = 0; i < elemen; i++) {  
14            png[i] = new Pangkat();  
15            System.out.print("Masukkan nilai yang hendak dipangkatkan : ");  
16            png[i].nilai = ayleen07.nextInt();  
17            System.out.print("Masukkan nilai pemangkat : ");  
18            png[i].pangkat = ayleen07.nextInt();  
19        }  
20        System.out.println("HASIL PANGKAT - BRUTE FORCE");  
21        for (int i = 0; i < elemen; i++) {  
22            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatBF(png[i].nilai, png[i].pangkat));  
23        }  
24        System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");  
25        for (int i = 0; i < elemen; i++) {  
26            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatDC(png[i].nilai, png[i].pangkat));  
27        }  
28    }  
29 }
```

```
=====  
Masukkan jumlah elemen yang 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  
PS C:\Users\Admin\OneDrive - ypt.or.id\Document
```

## Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu **PangkatBF()** dan **PangkatDC()**!

Jawab :

- Metode **pangkatBF()** mengimplementasikan algoritma Brute Force untuk menghitung pangkat suatu bilangan. Metode ini menggunakan perulangan for untuk mengalikan bilangan dengan dirinya sendiri sebanyak nilai pangkat yang diberikan. Pendekatan ini kurang efisien untuk nilai pangkat yang besar karena harus melakukan perhitungan berulang kali.
- Metode **pangkatDC()** mengimplementasikan algoritma Divide and Conquer untuk menghitung pangkat suatu bilangan. Metode ini menggunakan pendekatan rekursif dengan membagi masalah menjadi submasalah yang lebih kecil. Jika pangkat bernilai



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

ganjil, maka hasil diperoleh dengan mengalikan hasil rekursif dari  $n/2$  dengan hasil rekursif dari  $n/2$  dan bilangan itu sendiri. Jika pangkat bernilai genap, maka hasil diperoleh dengan mengalikan hasil rekursif dari  $n/2$  dengan hasil rekursif dari  $n/2$ . Pendekatan ini lebih efisien daripada Brute Force untuk nilai pangkat yang lebih besar.

2. Apakah tahap *combine* sudah termasuk dalam kode tersebut? Tunjukkan!

Jawab : Tahap combine sudah termasuk dalam kode tersebut, tahap ini terjadi melalui proses rekursif dalam method `pangkatDC()`. Sehingga hasil diperoleh dari sub-masalah yang lebih kecil yang digabungkan kembali untuk mendapatkan hasil akhir pangkat.

Tahap combine terjadi pada kode berikut :

```
1  if (n % 2 == 1) {  
2      return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a);  
3  } else {  
4      return (pangkatDC(a, n/2) * pangkatDC(a, n/2));  
5  }
```

3. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.

Jawab :

```
1  public class Pangkat {  
2      public int nilai, pangkat;  
3  
4      Pangkat (int nilai, int pangkat) {  
5          this.nilai = nilai;  
6          this.pangkat = pangkat;  
7      }  
}
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
1 Pangkat[] png = new Pangkat[elemen];  
2 for (int i = 0; i < elemen; i++) {  
3     System.out.print("Masukkan nilai yang hendak dipangkatkan : ");  
4     int nilai = ayleen07.nextInt();  
5     System.out.print("Masukkan nilai pemangkat : ");  
6     int pangkat = ayleen07.nextInt();  
7     png[i] = new Pangkat(nilai, pangkat);  
8 }
```

4. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan switch-case!

Jawab :

```
1 System.out.println("Pilih method yang ingin dijalankan :");  
2 System.out.println("1. BRUTE FORCE");  
3 System.out.println("2. DIVIDE AND CONQUER");  
4 System.out.print("Pilih method : ");  
5 int menu = ayleen07.nextInt();  
6  
7 switch (menu) {  
8     case 1:  
9         System.out.println("HASIL PANGKAT - BRUTE FORCE");  
10        for (int i = 0; i < elemen; i++) {  
11            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatBF(png[i].nilai, png[i].pangkat));  
12        }  
13        break;  
14        case 2:  
15            System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");  
16            for (int i = 0; i < elemen; i++) {  
17                System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatDC(png[i].nilai, png[i].pangkat));  
18            }  
19            break;  
20            default:  
21                System.out.println("Pilihan menu tidak valid!");  
22        }
```

```
=====  
Masukkan jumlah elemen yang dihitung : 2  
Masukkan nilai yang hendak dipangkatkan : 2  
Masukkan nilai pemangkat : 6  
Masukkan nilai yang hendak dipangkatkan : 4  
Masukkan nilai pemangkat : 3  
Pilih method yang ingin dijalankan :  
1. BRUTE FORCE  
2. DIVIDE AND CONQUER  
Pilih method : 1  
HASIL PANGKAT - BRUTE FORCE  
Hasil dari 2 pangkat 6 adalah 64  
Hasil dari 4 pangkat 3 adalah 64  
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

### PERCOBAAN 3

#### 4.4 Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer

```
1 package Jobsheet_4.Minggu_5.src.Sum;
2
3 public class Sum {
4     int elemen;
5     double keuntungan[], total;
6
7     Sum(int elemen) {
8         this.elemen = elemen;
9         this.keuntungan = new double[elemen];
10        this.total = 0;
11    }
12    double totalBF(double arr[]) {
13        for (int i = 0; i < elemen; i++) {
14            total = total + arr[i];
15        }
16        return total;
17    }
18    double totalDC(double arr[], int l, int r) {
19        if (l == r) {
20            return arr[l];
21        } else if (l < r) {
22            int mid = (l + r) / 2;
23            double lsum = totalDC(arr, l, mid - 1);
24            double rsum = totalDC(arr, mid + 1, r);
25            return lsum + rsum + arr[mid];
26        }
27        return 0;
28    }
29 }
```

```
1 package Jobsheet_4.Minggu_5.src.Sum;
2 import java.util.Scanner;
3
4 public class MainSum {
5     public static void main(String[] args) {
6         Scanner ayleen07 = new Scanner(System.in);
7
8         System.out.println("=====");
9         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");
10        System.out.print("Masukkan jumlah bulan : ");
11        int elm = ayleen07.nextInt();
12
13        Sum sm = new Sum(elm);
14        System.out.println("=====");
15        for (int i = 0; i < sm.elemen; i++) {
16            System.out.print("Masukkan untung bulan ke-" + (i + 1) + " = ");
17            sm.keuntungan[i] = ayleen07.nextDouble();
18        }
19        System.out.println("=====");
20        System.out.println("Algoritma Brute Force");
21        System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));
22        System.out.println("=====");
23        System.out.println("Algoritma Divide Conquer");
24        System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + Math.round(sm.totalDC(sm.keuntungan, 0, sm.elemen-1) * 100.0) / 100.0);
25    }
26 }
```



NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
=====
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 Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah = 40.34
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester 2\7. Pra
```

## Pertanyaan

1. Mengapa terdapat formulasi *return value* berikut? Jelaskan!

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

Jawab : Formulasi return value **lsum + rsum + arr[mid]** digunakan dalam method **totalDC** karena method tersebut mengimplementasikan algoritma divide and conquer untuk menghitung jumlah total dari semua elemen dalam array secara rekursif.

Dalam algoritma divide and conquer, suatu masalah dibagi menjadi dua sub-masalah yang lebih kecil, yaitu dengan menghitung total keuntungan pada bagian kiri array **lsum** dan menghitung total keuntungan pada bagian kanan array **rsum**. Kemudian, hasil dari kedua sub-masalah tersebut digabungkan dengan dijumlahkan dan ditambahkan dengan nilai keuntungan pada indeks tengah **arr[mid]**.

2. Kenapa dibutuhkan variable **mid** pada method **TotalDC()**?

Jawab : Variabel **mid** dibutuhkan dalam metode **totalDC()** karena metode ini menggunakan pendekatan divide and conquer untuk menghitung total keuntungan. Variabel **mid** digunakan untuk membagi array menjadi dua bagian sub-array, yaitu sub-array kiri dan sub-array kanan pada setiap tahap rekursif.

Kemudian method ini digunakan untuk menghitung total dari sub-array kiri dan sub-array kanan. Hasilnya, yaitu **lsum** dan **rsum**, digabungkan dengan nilai elemen di mid untuk mendapatkan total keseluruhan dari array. Sehingga nilai **mid** akan menjadi indeks tengah dari array tersebut. Pembagian ini dilakukan untuk menghitung total keuntungan secara rekursif pada kedua sub-array.

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 :





NAMA : AYLEEN RUHUL QISTHY  
NIM : 2341720012  
KELAS : 1G  
MATERI : JOBSHEET IV - BRUTE FORCE DAN DIVIDE CONQUER

```
1 package Jobsheet_4.Minggu_5.src.Sum;  
2 import java.util.Scanner;  
3  
4 public class MainSum {  
5     public static void main(String[] args) {  
6         Scanner ayleen07 = new Scanner(System.in);  
7  
8         System.out.print("Masukkan jumlah perusahaan : ");  
9         int jumlahPerusahaan = ayleen07.nextInt();  
10  
11         for (int i = 0; i < jumlahPerusahaan; i++) {  
12             System.out.println("\nPerusahaan " + (i + 1));  
13  
14             System.out.println("=====");  
15             System.out.println("Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");  
16             System.out.print("Masukkan jumlah bulan : ");  
17             int elm = ayleen07.nextInt();  
18  
19             Sum sm = new Sum(elm);  
20             System.out.println("=====");  
21             for (int j = 0; j < sm.elemen; j++) {  
22                 System.out.print("Masukkan untung bulan ke-" + (j + 1) + " = ");  
23                 sm.keuntungan[j] = ayleen07.nextDouble();  
24             }  
25             System.out.println("=====");  
26             System.out.println("Algoritma Brute Force");  
27             System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));  
28             System.out.println("=====");  
29             System.out.println("Algoritma Divide Conquer");  
30             System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + Math.round(sm.totalDC(sm.keuntungan, 0, sm.elemen-1) * 100.0) / 100.0);  
31         }  
32     }  
33 }
```

```
Masukkan jumlah perusahaan : 2  
  
Perusahaan 1  
=====  
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)  
Masukkan jumlah bulan : 2  
=====  
Masukkan untung bulan ke-1 = 5,6  
Masukkan untung bulan ke-2 = 3,4  
=====  
Algoritma Brute Force  
Total keuntungan perusahaan selama 2 bulan adalah = 9.0  
=====  
Algoritma Divide Conquer  
Total keuntungan perusahaan selama 2 bulan adalah = 9.0  
  
Perusahaan 2  
=====  
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)  
Masukkan jumlah bulan : 3  
=====  
Masukkan untung bulan ke-1 = 4,1  
Masukkan untung bulan ke-2 = 3  
Masukkan untung bulan ke-3 = 2,7  
=====  
Algoritma Brute Force  
Total keuntungan perusahaan selama 3 bulan adalah = 9.8  
=====  
Algoritma Divide Conquer  
Total keuntungan perusahaan selama 3 bulan adalah = 9.8  
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester 2
```

Link GitHub Pribadi :

[https://github.com/ayleenrq/Prak\\_AlgoritmaStrukturData\\_07.git](https://github.com/ayleenrq/Prak_AlgoritmaStrukturData_07.git)