



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

## Percobaan 1: Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

```
1 package P4.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 = fakto * i;
10        }
11        return fakto;
12    }
13
14    int faktorialDC(int n) {
15        if (n == 1) {
16            return 1;
17        } else {
18            int fakto = n * faktorialDC(n - 1);
19            return fakto;
20        }
21    }
22 }
23
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1 package P4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainFaktorial {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("-----");
9         System.out.print("Masukkan jumlah elemen: ");
10        int iJml = sc.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            int iNilai = sc.nextInt();
17            fk[i].nilai = iNilai;
18        }
19
20        System.out.println("HASIL - BRUTE FORCE");
21        for (int i = 0; i < iJml; i++) {
22            System.out.println(
23                "Hasil perhitungan faktorial menggunakan Brute Forece adalah " + fk[i].faktorialBF(fk[i].nilai));
24        }
25        System.out.println("Hasil DIVIDE AND CONQUER");
26        for (int i = 0; i < iJml; i++) {
27            System.out.println("Hasil perhitungan faktorial menggunakan Divide and Conquer adalah "
28                + fk[i].faktorialDC(fk[i].nilai));
29        }
30    }
31 }
32
33
```

```
-----
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 Forece adalah 120
Hasil perhitungan faktorial menggunakan Brute Forece adalah 40320
Hasil perhitungan faktorial menggunakan Brute Forece 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:\project\AlgoritmaStrukturData_1G_28\AlgoritmaStrukturData_1G_28>
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

## Pertanyaan

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

Jawab: pada function faktorialDC menggunakan konsep fungsi recursif yaitu kode pada penggunaan if adalah base case ketika nilai  $n == 1$  maka function tersebut mengembalikan nilai  $n$  dimana nilai  $n$  adalah 1 sedangkan kode pada penggunaan else adalah kode yang dijalankan ketika nilai  $n != 1$  dan pada kondisi ini terjadi pemanggilan recursif dengan argumen  $n-1$ .

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

Jawab:

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

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

Jawab:

Perbedaan antara  $\text{fakto} *= i$ ; dan  $\text{int fakto} = n * \text{faktorialDC}(n-1)$  jika dijabarkan adalah:

$\text{fakto} *= i \Rightarrow \text{fakto} = \text{fakto} * i$ ; dimana nilai  $i$  adalah 1 di dalam perulangan for pada function faktorialBF lalu perulangan tersebut akan berhenti ketika nilai  $i > n$ .

sedangkan

$\text{int fakto} = n * \text{faktorialDC}(n-1) \Rightarrow$  jika dimisalkan  $n$  adalah 5, maka

$\text{int fakto} = 5 * \text{faktorialDC}(5-1)$ , dimana nilai  $n$  adalah argumen dari function faktorialDC, sehingga pada baris kode tersebut akan terjadi pemanggilan recursif dan akan berhenti ketika  $n == 1$  (base case).



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

## Percobaan 2: Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

```
1 package P4.BruteForceDivideConquer;
2
3 public class Pangkat {
4     public int nilai;
5     public int pangkat;
6
7     int pangkatBF(int a, int n) {
8         int hasil = 1;
9         for (int i = 0; i < n; i++) {
10             hasil *= a;
11         }
12         return hasil;
13     }
14
15     int pangkatDC(int a, int n) {
16         if (n == 1) {
17             return 1;
18         } else {
19             if (n % 2 == 1) {
20                 return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);
21             } else {
22                 return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
23             }
24         }
25     }
26 }
27
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1 package p4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainPangkat {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.print("Masukkan jumlah elemen yang dihitung: ");
10        int elemen = sc.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            int nilai = sc.nextInt();
17            png[i].nilai = nilai;
18            System.out.print("Masukkan nilai pemangkat: ");
19            int pangkat = sc.nextInt();
20            png[i].pangkat = pangkat;
21        }
22        System.out.println("HASIL PANGKAT - BRUTE FORCE");
23        for (int i = 0; i < elemen; i++) {
24            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
25                               + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
26        }
27
28        System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
29        for (int i = 0; i < elemen; i++) {
30            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
31                               + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
32        }
33    }
34 }
35
```

```
=====
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
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

## Pertanyaan

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

Jawab:

Metode PangkatBF menggunakan pendekatan Brute Force untuk menghitung pangkat suatu bilangan yaitu dengan melakukan operasi perkalian sebanyak  $n$  kali terhadap bilangan  $a$ . Metode PangkatDC menggunakan pendekatan Divide and Conquer untuk memecah masalah menjadi sub-masalah yang lebih kecil yaitu dengan membagi bilangan  $n$  menjadi dua bagian yang lebih kecil.

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

Jawab:

Ya sudah termasuk, tahap combine terdapat pada code = >

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

Jawab:



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1 package P4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainPangkat {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.print("Masukkan jumlah elemen yang dihitung: ");
10        int elemen = sc.nextInt();
11
12        Pangkat[] png = new Pangkat[elemen];
13        for (int i = 0; i < elemen; i++) {
14            System.out.print("Masukkan nilai yang hendak dipangkatkan: ");
15            int nilai = sc.nextInt();
16            System.out.print("Masukkan nilai pemangkat: ");
17            int pangkat = sc.nextInt();
18            png[i] = new Pangkat(nilai, pangkat);
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 "
23                + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
24        }
25
26        System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
27        for (int i = 0; i < elemen; i++) {
28            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
29                + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
30        }
31    }
32 }
33
```

```
1 package P4.BruteForceDivideConquer;
2
3 public class Pangkat {
4     public int nilai;
5     public int pangkat;
6
7     Pangkat(int nilai, int pangkat) {
8         this.nilai = nilai;
9         this.pangkat = pangkat;
10    }
11
12    int pangkatBF(int a, int n) {
13        int hasil = 1;
14        for (int i = 0; i < n; i++) {
15            hasil *= a;
16        }
17        return hasil;
18    }
19
20    int pangkatDC(int a, int n) {
21        if (n == 1) {
22            return 1;
23        } else {
24            if (n % 2 == 1) {
25                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);
26            } else {
27                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
28            }
29        }
30    }
31 }
32
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
=====
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
```

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

Jawab:





NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1 package P4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainPangkat {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.print("Masukkan jumlah elemen yang dihitung: ");
10        int elemen = sc.nextInt();
11
12        Pangkat[] png = new Pangkat[elemen];
13        for (int i = 0; i < elemen; i++) {
14            System.out.print("Masukkan nilai yang hendak dipangkatkan: ");
15            int nilai = sc.nextInt();
16            System.out.print("Masukkan nilai pangkat: ");
17            int pangkat = sc.nextInt();
18            png[i] = new Pangkat(nilai, pangkat);
19        }
20
21        System.out.println("Pilih Metode: ");
22        System.out.println("1. Brute Force");
23        System.out.println("2. Divide and Conquer");
24        int choice = sc.nextInt();
25        switch (choice) {
26            case 1:
27                System.out.println("HASIL PANGKAT - BRUTE FORCE");
28                for (int i = 0; i < elemen; i++) {
29                    System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
30                        + png[i].pangkat8f(png[i].nilai, png[i].pangkat));
31                }
32                break;
33            case 2:
34                System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
35                for (int i = 0; i < elemen; i++) {
36                    System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
37                        + png[i].pangkat8f(png[i].nilai, png[i].pangkat));
38                }
39                break;
40            default:
41                System.out.println("Pilihan salah");
42                System.out.println("Program berhenti");
43                break;
44        }
45    }
46 }
47
48 }
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

### Percobaan 3: Menghitung Sum Array dengan Algoritma Brute Force dan Divide and 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
```

```
1 package P4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainSum {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
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 = sc.nextInt();
12        Sum sm = new Sum(elm);
13        System.out.println("=====");
14        for (int i = 0; i < sm.elemen; i++) {
15            System.out.print("Masukkan untung bulan ke-" + (i + 1) + " = ");
16            sm.keuntungan[i] = sc.nextDouble();
17        }
18        System.out.println("=====");
19        System.out.println("Algoritma Brute Force");
20        System.out.println(
21            "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 = "
25            + Math.round(sm.totalDC(sm.keuntungan, 0, sm.elemen - 1) * 100.0) / 100.0);
26    }
27 }
28
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1  package P4.BruteForceDivideConquer;
2
3  public class Sum {
4      int elemen;
5      double keuntungan[];
6      double total;
7
8      Sum(int elemen) {
9          this.elemen = elemen;
10         this.keuntungan = new double[elemen];
11         this.total = 0;
12     }
13
14     double totalBF(double arr[]) {
15         for (int i = 0; i < elemen; i++) {
16             total = total + arr[i];
17         }
18         return total;
19     }
20
21     double totalDC(double arr[], int l, int r) {
22         if (l == r) {
23             return arr[l];
24         } else if (l < r) {
25             int mid = (l + r) / 2;
26             double lsum = totalDC(arr, l, mid - 1);
27             double rsum = totalDC(arr, mid + 1, r);
28             return lsum + rsum + arr[mid];
29         }
30         return 0;
31     }
32 }
33
```



NAMA : Roy Wijaya  
NIM : 2341720120  
KELAS : 1-G  
MATERI : Brute Force and Divide Conquer (Jobsheet 4)

## Pertanyaan

1. Mengapa terdapat formulasi return value berikut? Jelaskan!

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

Jawab:

Return lsum + rsum + arr[mid] digunakan untuk menggabungkan solusi dari dua submasalah dengan elemen pada titik tengah untuk mendapatkan solusi total keseluruhan yang sesuai dengan konsep divide and conquer.

2. Kenapa dibutuhkan variable mid pada method TotalDC()?  
jawab: variabel mid digunakan untuk membagi array yang sedang diproses menjadi dua bagian sehingga code dapat diproses secara efisien
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:

```
=====
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)
Banyak Perusahaan: 2
Perusahaan ke 1
Masukkan jumlah bulan: 3
=====
Masukkan untung bulan ke-1 = 3,1
Masukkan untung bulan ke-1 = 5,3
Masukkan untung bulan ke-1 = 2,2
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 3 bulan adalah = 10.600000000000001
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 3 bulan adalah = 10.6
Perusahaan ke 2
Masukkan jumlah bulan: 4
=====
Masukkan untung bulan ke-2 = 2,5
Masukkan untung bulan ke-2 = 2,1
Masukkan untung bulan ke-2 = 2,5
Masukkan untung bulan ke-2 = 2,1
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 4 bulan adalah = 9.2
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 4 bulan adalah = 9.2
```



NAMA : Roy Wijaya

NIM : 2341720120

KELAS : 1-G

MATERI : Brute Force and Divide Conquer (Jobsheet 4)

```
1 package P4.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainSum {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");
10        System.out.print("Banyak Perusahaan: ");
11        int jml = sc.nextInt();
12        Sum sm[] = new Sum[jml];
13        for (int i = 0; i < jml; i++) {
14            System.out.println("Perusahaan ke " + (i + 1));
15            System.out.print("Masukkan jumlah bulan: ");
16            int elm = sc.nextInt();
17            sm[i] = new Sum(elm);
18
19            System.out.println("=====");
20            for (int j = 0; j < sm[i].elemen; j++) {
21                System.out.print("Masukkan untung bulan ke-" + (i + 1) + " = ");
22                sm[i].keuntungan[j] = sc.nextDouble();
23            }
24            System.out.println("=====");
25            System.out.println("Algoritma Brute Force");
26            System.out.println(
27                "Total keuntungan perusahaan selama " + sm[i].elemen + " bulan adalah = "
28                + sm[i].totalBF(sm[i].keuntungan));
29
30            System.out.println("=====");
31            System.out.println("Algoritma Divide Conquer");
32            System.out.println("Total keuntungan perusahaan selama " + sm[i].elemen + " bulan adalah = "
33                + Math.round(sm[i].totalDC(sm[i].keuntungan, 0, sm[i].elemen - 1) * 100.0) / 100.0);
34        }
35    }
36 }
37 }
38 }
```

LINK GITHUB: [https://github.com/RoyW12/AlgoritmaStrukturData\\_1G\\_28](https://github.com/RoyW12/AlgoritmaStrukturData_1G_28)