



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

2. 1 Percobaan 1

```
1 package P5.BruteForceDivideConquer;
2
3 public class Faktorial {
4     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
```

```
1 package P5.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
9         System.out.println(x:"=====");
10        System.out.print(s:"Masukkan jumlah elemen : ");
11        int iJml = sc.nextInt();
12
13        Faktorial[] fk = new Faktorial[10];
14        for (int i = 0; i < iJml; i++) {
15            fk[i] = new Faktorial();
16            System.out.printf(format:"Masukkan nilai data ke-%d : ", i + 1);
17            fk[i].nilai = sc.nextInt();
18        }
19
20        System.out.println(x:"\nHASIL - BRUTE FORCE");
21        for (int i = 0; i < iJml; i++) {
22            System.out.printf(format:"Hasil penghitungan faktorial menggunakan Brute Force adalah %d\n",
23                fk[i].faktorialBF(fk[i].nilai));
24        }
25
26        System.out.println(x:"\nHASIL - DIVIDE AND CONQUER");
27        for (int i = 0; i < iJml; i++) {
28            System.out.printf(format:"Hasil penghitungan faktorial menggunakan Divide and Conquer adalah %d\n",
29                fk[i].faktorialDC(fk[i].nilai));
30        }
31    }
32 }
33
```



NAMA : OKTAVIAN EKA RAMADHAN
NIM : 2341720117
KELAS : 1G
MATERI : 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 penghitungan faktorial menggunakan Brute Force adalah 120
Hasil penghitungan faktorial menggunakan Brute Force adalah 40320
Hasil penghitungan faktorial menggunakan Brute Force adalah 6

HASIL - DIVIDE AND CONQUER
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 120
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 40320
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 6
```

Pertanyaan :

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

Jawab : Bagian kode if ($n==1$) merupakan nilai batas dari sebuah fungsi rekursif. Bagian ini berfungsi untuk melakukan pengecekan apakah nilai n sudah sesuai dengan nilai batas atau belum, jika sudah maka akan melakukan return untuk mengembalikan semua fungsi dan menghentikan fungsi rekursif.

Sedangkan bagian kode else berfungsi untuk melakukan pemanggilan fungsi itu kembali dengan nilai n yang berbeda dengan sebelumnya.

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

Jawab : Perulangan pada method tersebut memungkinkan untuk diubah menjadi model perulangan lainnya

```
6      int faktorialBF(int n) {
7          int fakto = 1;
8          // for (int i = 1; i <= n; i++) {
9          // fakto = fakto * i;
10         // }
11
12         int i = 1;
13         while (i <= n) {
14             fakto *= i;
15             i++;
16         }
17
18         return fakto;
19     }
```



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : 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 penghitungan faktorial menggunakan Brute Force adalah 120
Hasil penghitungan faktorial menggunakan Brute Force adalah 40320
Hasil penghitungan faktorial menggunakan Brute Force adalah 6

HASIL - DIVIDE AND CONQUER
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 120
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 40320
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 6
PS D:\Vian\kuliah\sem-2\AlgoritmaStrukturData>
```

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

Jawab : $fakto *= i$ dapat diartikan memberikan nilai pada variabel fakto dari hasil perkalian fakto itu sendiri dengan i. Sintaks ini biasanya digunakan pada perulangan faktorial, sehingga tiap hasil bilangan sebelumnya akan dikalikan dengan i+1 hingga perulangan berakhir.

Sedangkan $fakto = n * faktorialDC(n-1);$ adalah sintaks untuk memanggil fungsi itu kembali (fungsi rekursif) dan akan mengembalikan nilai jika sintaks tersebut tidak lagi dipanggil (batas akhir). Hal ini dilakukan dengan ketentuan nilai n akan selalu berkurang hingga menemukan batas akhir dan mengembalikan nilai tertentu



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

2.2 Percobaan 2

```
1 package P5.BruteForceDivideConquer;
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 = 0; i < n; i++) {
9             hasil *= a;
10        }
11        return hasil;
12    }
13
14    int pangkatDC(int a, int n) {
15        if (n == 1) {
16            return a;
17        } else {
18            if (n % 2 == 1) {
19                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);
20            } else {
21                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
22            }
23        }
24    }
25 }
26
```

```
1 package P5.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(x:"=====");
9         System.out.print(s:"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(s:"Masukkan nilai yang hendak dipangkatkan : ");
16            png[i].nilai = sc.nextInt();
17            System.out.print(s:"Masukkan nilai pemangkat : ");
18            png[i].pangkat = sc.nextInt();
19        }
20
21        System.out.println(x:"\nHASIL PANGKAT - BRUTE FORCE");
22        for (int i = 0; i < elemen; i++) {
23            System.out.printf(format:"Hasil dari %d pangkat %d adalah %d\n", png[i].nilai, png[i].pangkat,
24                               png[i].pangkatBF(png[i].nilai, png[i].pangkat));
25        }
26        System.out.println(x:"\nHASIL PANGKAT - DIVIDE AND CONQUER");
27        for (int i = 0; i < elemen; i++) {
28            System.out.printf(format:"Hasil dari %d pangkat %d adalah %d\n", png[i].nilai, png[i].pangkat,
29                               png[i].pangkatDC(png[i].nilai, png[i].pangkat));
30        }
31    }
32 }
33
```



NAMA : OKTAVIAN EKA RAMADHAN
NIM : 2341720117
KELAS : 1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
=====
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 D:\Vian\kuliah\sem-2\AlgoritmaStrukturData>
```

Pertanyaan :

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

Jawab : PangkatBF melakukan perulangan untuk setiap hasil kali, sedangkan PangkatDC melakukan pemanggilan fungsi berulang(fungsi rekursif) sehingga hasil akan dikalikan dengan hasil kali fungsi.

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

Jawab : Sudah, tahap combine dijalankan ketika batas akhir telah dicapai, dan fungsi tidak lagi memanggil dirinya sendiri. Sehingga yang terjadi adalah seluruh fungsi yang dipanggil melakukan return untuk melengkapi return-return sebelumnya.

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

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



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

Jawab :

```
public int nilai, pangkat;

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

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

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

for (int i = 0; i < elemen; i++) {
    System.out.print(s:"Masukkan nilai yang hendak dipangkatkan : ");
    int Nilai = sc.nextInt();
    System.out.print(s:"Masukkan nilai pemangkat : ");
    int Pangkat = sc.nextInt();
    png[i] = new Pangkat(Nilai, Pangkat);
}
```

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



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

Jawab :

```
20
21     System.out.println(x:"Pilih metode:\n1.Brute Force\n2.Divide Conquer");
22     System.out.print(s:"Masukkan angka: ");
23     int pil = sc.nextInt();
24
25     int hasil[] = new int[elemen];
26
27     switch (pil) {
28         case 1:
29             System.out.println(x:"\nHASIL PANGKAT - BRUTE FORCE");
30             for (int i = 0; i < elemen; i++) {
31                 hasil[i] = png[i].pangkatBF(png[i].nilai, png[i].pangkat);
32             }
33             break;
34         case 2:
35             System.out.println(x:"\nHASIL PANGKAT - DIVIDE AND CONQUER");
36             for (int i = 0; i < elemen; i++) {
37                 hasil[i] = png[i].pangkatDC(png[i].nilai, png[i].pangkat);
38             }
39             break;
40         default:
41             break;
42     }
43
44     for (int i = 0; i < elemen; i++) {
45         System.out.printf(format:"Hasil dari %d pangkat %d adalah %d\n",
46             png[i].nilai, png[i].pangkat, hasil[i]);
47     }
48
```

```
=====
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
```

```
Pilih metode:
1.Brute Force
2.Divide Conquer
Masukkan angka: 1
```

```
HASIL PANGKAT - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
```



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

2.3 Percobaan 3

```
1 package P5.BruteForceDivideConquer;
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
13    double totalBF(double[] arr) {
14        for (int i = 0; i < elemen; i++) {
15            total += arr[i];
16        }
17        return total;
18    }
19
20    double totalDC(double[] arr, int l, int r) {
21        if (l == r) {
22            return arr[l];
23        }
24        int mid = (l + r) / 2;
25        double lsum = totalDC(arr, l, mid);
26        double rsum = totalDC(arr, mid + 1, r);
27        return lsum + rsum;
28    }
29 }
```

```
1 package P5.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainSum {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         System.out.println(x:"=====");
10        System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
11        System.out.print(s:"Masukkan jumlah bulan : ");
12        int elm = sc.nextInt();
13
14        Sum sm = new Sum(elm);
15        System.out.println(x:"=====");
16        for (int i = 0; i < sm.elemen; i++) {
17            System.out.printf(format:"Masukkan untung bulan ke-%d : ", i + 1);
18            sm.keuntungan[i] = sc.nextDouble();
19        }
20
21        System.out.println(x:"=====");
22        System.out.println(x:"Algoritma Brute Force");
23        System.out.printf(format:"Total keuntungan perusahaan selama %d bulan adalah = %.2f\n", sm.elemen,
24            sm.totalBF(sm.keuntungan));
25        System.out.println(x:"=====");
26        System.out.println(x:"Algoritma Divide Conquer");
27        System.out.printf(format:"Total keuntungan perusahaan selama %d bulan adalah %.2f\n", sm.elemen,
28            sm.totalDC(sm.keuntungan, 1:0, (sm.elemen - 1)));
29    }
30 }
```




NAMA : OKTAVIAN EKA RAMADHAN
NIM : 2341720117
KELAS : 1G
MATERI : 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 = 40.34
=====
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];
```

Jawab : Untuk melakukan combine pada fungsi yang telah dibagi

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

Jawab : Untuk menjadi titik tengah dari fungsi yang dijalankan, sehingga fungsi dapat dibagi dua dan mengoptimalkan proses yang berjalan.

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!



NAMA : OKTAVIAN EKA RAMADHAN

NIM : 2341720117

KELAS : 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

Jawab :

```
8
9      System.out.print(s:"Masukkan jumlah perusahaan: ");
10     int jmlPer = sc.nextInt();
11     Sum perusahaan[] = new Sum[jmlPer];
12
13     for (int i = 0; i < jmlPer; i++) {
14         System.out.println("\nPERUSAHAAN " + (i + 1));
15         System.out.println(x:"=====");
16         System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
17         System.out.print(s:"Masukkan jumlah bulan : ");
18         int elm = sc.nextInt();
19
20         perusahaan[i] = new Sum(elm);
21
22         System.out.println(x:"=====");
23         for (int j = 0; j < perusahaan[i].elemen; j++) {
24             System.out.printf(format:"Masukkan untung bulan ke-%d : ", j + 1);
25             perusahaan[i].keuntungan[j] = sc.nextDouble();
26         }
27     }
28
29     for (int i = 0; i < jmlPer; i++) {
30         System.out.println("\nPERUSAHAAN " + (i + 1));
31         System.out.println(x:"=====");
32         System.out.println(x:"Algoritma Brute Force");
33         System.out.printf(format:"Total keuntungan perusahaan selama %d bulan adalah %.2f\n", perusahaan[i].elemen,
34             perusahaan[i].totalBF(perusahaan[i].keuntungan));
35         System.out.println(x:"=====");
36         System.out.println(x:"Algoritma Divide Conquer");
37         System.out.printf(format:"Total keuntungan perusahaan selama %d bulan adalah %.2f\n", perusahaan[i].elemen,
38             perusahaan[i].totalDC(perusahaan[i].keuntungan, 1:0, (perusahaan[i].elemen - 1)));
39     }
40 }
41 }
```

Masukkan jumlah perusahaan: 2

PERUSAHAAN 1

=====

Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)

Masukkan jumlah bulan : 3

=====

Masukkan untung bulan ke-1 : 1.4

Masukkan untung bulan ke-2 : 6.8

Masukkan untung bulan ke-3 : 3.5

PERUSAHAAN 2

=====

Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)

Masukkan jumlah bulan : 4

=====

Masukkan untung bulan ke-1 : 1.4

Masukkan untung bulan ke-2 : 2.6

Masukkan untung bulan ke-3 : 7.3

Masukkan untung bulan ke-4 : 8.8

PERUSAHAAN 1

=====

Algoritma Brute Force

Total keuntungan perusahaan selama 3 bulan adalah 11.70

=====

Algoritma Divide Conquer

Total keuntungan perusahaan selama 3 bulan adalah 11.70

PERUSAHAAN 2

=====

Algoritma Brute Force

Total keuntungan perusahaan selama 4 bulan adalah 20.10

=====

Algoritma Divide Conquer

Total keuntungan perusahaan selama 4 bulan adalah 20.10

=====

Algoritma Divide Conquer

Total keuntungan perusahaan selama 4 bulan adalah 20.10