



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

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

4.2.1 Langkah-langkah Percobaan

```
1 package P5.BruteForceDivideConquer;
2
3 public class Faktorial10 {
4     public int nilai;
5
6     int faktorialBF(int n) {
7         int faktor = 1;
8         for (int i = 1; i <= n; i++) {
9             faktor = faktor * i;
10        }
11        return faktor;
12    }
13
14    int faktorialDC(int n) {
15        if (n == 1) {
16            return 1;
17        } else {
18            int faktor = n * faktorialDC(n - 1);
19            return faktor;
20        }
21    }
22 }
```

```
1 package P5.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainFaktorial10 {
6     public static void main(String[] args) {
7         Scanner sc10 = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.print("Masukkan Jumlah Elemen: ");
10        int iJml = sc10.nextInt();
11
12        System.out.println("=====");
13        Faktorial10[] fk = new Faktorial10[iJml];
14        for (int i = 0; i < iJml; i++) {
15            fk[i] = new Faktorial10();
16            System.out.print("Masukkan nilai data ke-" + (i + 1) + " :");
17            fk[i].nilai = sc10.nextInt();
18        }
19
20        System.out.println("=====");
21        System.out.println("HASIL - BRUTE FORCE");
22        for (int i = 0; i < iJml; i++) {
23            System.out.println(
24                "Hasil perhitungan faktorial menggunakan brute force adalah " + fk[i].faktorialBF(fk[i].nilai));
25        }
26
27        System.out.println("=====");
28        System.out.println("HASIL - DIVIDE AND CONQUER");
29        for (int i = 0; i < iJml; i++) {
30            System.out.println(
31                "Hasil perhitungan faktorial menggunakan Divide and Conquer adalah "
32                + fk[i].faktorialDC(fk[i].nilai));
33        }
34        sc10.close();
35    }
36 }
```



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 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 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
```

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!

Jawab:

- Pada kode if, memeriksa apakah nilai n adalah 1. Jika ya, mengembalikan 1 sebagai hasil faktorial. Merupakan base case untuk rekursif.
- Pada kode else, jika $n > 1$ akan mengalikan nilai n dengan hasil faktorial dari nilai $n-1$. Kemudian mengembalikan hasil ini sebagai hasil faktorial dari nilai n.(recursion call)
- Sehingga pada Algoritma Divide Conquer menggunakan rekursif untuk memecahkan masalah menjadi beberapa bagian yang lebih kecil

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

Jawab:Bisa menggunakan while

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



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 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 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
```

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

Jawab:

- fakto *= i
melakukan perkalian antara nilai variabel fakto dengan variabel i dan hasil perkalian disimpan kembali ke variabel fakto. Dilakukan dalam perulangan for dan menghasilkan faktorial dari suatu bilangan menggunakan metode Brute Force.
- int fakto = n * faktorialDC(n-1);
melakukan perkalian antara nilai variabel n dengan hasil dari pemanggilan fungsi faktorialDC(n - 1). Hasil perkalian disimpan ke variabel fakto dan dilakukan dalam metode Divide and Conquer untuk menghitung faktorial.

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

4.3.1 Langkah-langkah Percobaan

```
1 package P5.BruteForceDivideConquer;
2
3 public class Pangkat10 {
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 == 0) {
16            return 1;
17        } else {
18            if (n % 2 == 1) { // bilangan ganjil
19                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);
20            } else { // bilangan genap
21                return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
22            }
23        }
24    }
25 }
```



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
1 package P5.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainPangkat10 {
6     public static void main(String[] args) {
7         Scanner sc10 = new Scanner(System.in);
8
9         System.out.println("=====");
10        System.out.print("Masukkan jumlah elemen yang dihitung: ");
11        int elemen = sc10.nextInt();
12
13        Pangkat10[] png = new Pangkat10[elemen];
14        for (int i = 0; i < elemen; i++) {
15            png[i] = new Pangkat10();
16            System.out.print("Masukkan nilai yang hendak dipangkatkan: ");
17            png[i].nilai = sc10.nextInt();
18            System.out.print("Masukkan nilai pemangkat: ");
19            png[i].pangkat = sc10.nextInt();
20        }
21
22        System.out.println("=====");
23        System.out.println("HASIL PANGKAT - BRUTE FORCE");
24        for (int i = 0; i < elemen; i++) {
25            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
26                + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
27        }
28
29        System.out.println("=====");
30        System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
31        for (int i = 0; i < elemen; i++) {
32            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
33                + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
34        }
35
36        sc10.close();
37    }
38 }
```

```
=====
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
```

4.3.3 Pertanyaan

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

Jawab:

- PangkatBF() : metode yang menggunakan Brute Force dan pemrosesan ini tidak efisien untuk mencari nilai pangkat dari sebuah bilangan. Dengan menggunakan



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

perulangan, PangkatBF() memanggil dirinya sendiri sehingga membutuhkan waktu yang lebih lama dan tidak efisien.

- PangkatDC() : metode yang menggunakan Divide and Conquer atau pembagian masalah. Dengan menggunakan Divide dan Conquer, PangkatDC() dapat memecah masalah menjadi beberapa masalah kecil yang lebih mudah diproses. Setelah itu, PangkatDC() akan menggabungkan kembali hasil dari setiap masalah kecil tersebut. PangkatDC() atau Divide and Conquer lebih efisien dan mudah dibaca.

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

Jawab: Sudah, terdapat pada method pangkatDC. Tahap combine sudah terintegrasi dalam langkah recursive call.. Kode akan mengulangi recursive call hingga memenuhi kondisi tertentu, yang akhirnya akan menggabungkan hasil dari setiap sub-masalah kecil itu.

```
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 { // bilangan genap  
            return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));  
        }  
    }  
}
```

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

```
public Pangkat10(int nilai, int pangkat) {  
    this.nilai = nilai;  
    this.pangkat = pangkat;  
}
```

Jawab:

```
Pangkat10[] png = new Pangkat10[elemen];  
for (int i = 0; i < elemen; i++) {  
    // png[i] = new Pangkat10();  
    System.out.print("Masukkan nilai yang hendak dipangkatkan: ");  
    int nilai = sc10.nextInt();  
    // png[i].nilai = sc10.nextInt();  
    System.out.print("Masukkan nilai pemangkat: ");  
    int pangkat = sc10.nextInt();  
    // png[i].pangkat = sc10.nextInt();  
    png[i] = new Pangkat10(nilai, pangkat);  
}
```

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

Jawab:



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
25     int menu;
26     System.out.println("=====");
27     System.out.println("Pilih metode yang digunakan:");
28     System.out.println("1. Brute Force");
29     System.out.println("2. Divide and Conquer");
30     System.out.print("Masukkan pilihan anda: ");
31     menu = sc10.nextInt();
32
33     switch (menu) {
34     case 1:
35         System.out.println("=====");
36         System.out.println("HASIL PANGKAT - BRUTE FORCE");
37         for (int i = 0; i < elemen; i++) {
38             System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
39                 + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
40         }
41         break;
42     case 2:
43         System.out.println("=====");
44         System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
45         for (int i = 0; i < elemen; i++) {
46             System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
47                 + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
48         }
49         break;
50     default:
51         System.out.println("Pilihan tidak tersedia.");
52     }
53     sc10.close();
54 }
55 }
```

```
=====
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 yang digunakan:
1. Brute Force
2. Divide and Conquer
Masukkan pilihan anda: 1
=====
HASIL PANGKAT - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
```

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

4.4.1 Langkah-langkah Percobaan



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
1 package P5.BruteForceDivideConquer;
2
3 public class Sum10 {
4     int elemen;
5     double keuntungan[], total;
6
7     Sum10(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 = 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        } else if (l < r) {
24            int mid = (l + r) / 2;
25            double lsum = totalDC(arr, l, mid - 1);
26            double rsum = totalDC(arr, mid + 1, r);
27            return lsum + rsum + arr[mid];
28        }
29        return 0;
30    }
31 }
```

```
1 package P5.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainSum10 {
6     public static void main(String[] args) {
7         Scanner sc10 = 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 = sc10.nextInt();
12
13        Sum10 sm = new Sum10(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] = sc10.nextDouble();
18        }
19
20        System.out.println("=====");
21        System.out.println("Algoritma Brute Force");
22        System.out.println(
23            "Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));
24        System.out.println("=====");
25        System.out.println("Algoritma Divide Conquer");
26        System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "
27            + Math.round(sm.totalDC(sm.keuntungan, 0, sm.elemen - 1) * 100.0) / 100.0);
28        System.out.println("=====");
29        sc10.close();
30    }
31 }
```



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 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 = 40.339999999999996
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah = 40.34
=====
```

4.4.3 Pertanyaan

1. Mengapa terdapat formulasi return value berikut?Jelaskan!

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

Jawab: Terdapat formulasi return value `return lsum + rsum + arr[mid];` dalam fungsi `totalDC(double arr[], int l, int r)` untuk menggabungkan hasil penjumlahan dari sub-array / mengembalikan hasil penjumlahan dari semua elemen array yang berada di sebelah kiri, kanan, dan tengah.(combine pada Divide conquer)

- `lsum`: Merupakan hasil penjumlahan dari semua elemen array yang berada di sebelah kiri dari elemen tengah.
- `rsum`: Merupakan hasil penjumlahan dari semua elemen array yang berada di sebelah kanan dari elemen tengah.
- `arr[mid]`: Merupakan elemen tengah yang bukan termasuk dalam `lsum` dan `rsum`, sehingga harus dijumlahkan secara terpisah.

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

Jawab: Variable `mid` digunakan dalam method `totalDC()` untuk membagi array menjadi dua bagian(divide pada Divide Conquer), yaitu bagian kiri dan kanan, di mana bagian kiri terdiri dari elemen dari index awal (`l`) hingga sebelum index `mid`, sedangkan bagian kanan terdiri dari elemen dari index `mid` hingga index akhir (`r`).

Variable `mid` digunakan sebagai indikator untuk menandakan titik tengah dari array. Sehingga, `mid` digunakan untuk membagi array menjadi bagian kiri dan kanan, dan kemudian menggabungkan hasil dari setiap bagian tersebut dalam mencapai total.

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 : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
1 package P5.BruteForceDivideConquer;
2
3 import java.util.Scanner;
4
5 public class MainSum10 {
6     public static void main(String[] args) {
7         Scanner sc10 = new Scanner(System.in);
8         System.out.print("Masukkan Banyak Perusahaan: ");
9         int perusahaan = sc10.nextInt();
10        System.out.println();
11
12        for (int i = 0; i < perusahaan; i++) {
13            System.out.println("Data Keuntungan Perusahaan ke-" + (i + 1));
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 = sc10.nextInt();
18
19            Sum10 sm = new Sum10(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] = sc10.nextDouble();
24            }
25
26            System.out.println("=====");
27            System.out.println("Algoritma Brute Force");
28            System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));
29            System.out.println("=====");
30            System.out.println("Algoritma Divide Conquer");
31            System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "
32                + Math.round(sm.totalDC(sm.keuntungan, 0, sm.elemen - 1) * 100.0) / 100.0);
33            System.out.println("=====");
34        }
35        sc10.close();
36    }
37 }
38 }
```

Masukkan Banyak Perusahaan: 2

Data Keuntungan Perusahaan ke-1

=====

Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)

Masukkan jumlah bulan : 3

=====

Masukkan untung bulan ke- 1 = 6,8

Masukkan untung bulan ke- 2 = 7,5

Masukkan untung bulan ke- 3 = 8,1

=====

Algoritma Brute Force

Total keuntungan perusahaan selama 3 bulan adalah = 22.4

=====

Algoritma Divide Conquer

Total keuntungan perusahaan selama 3 bulan adalah = 22.4

=====

Data Keuntungan Perusahaan ke-2

=====

Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)

Masukkan jumlah bulan : 3

=====

Masukkan untung bulan ke- 1 = 9,1

Masukkan untung bulan ke- 2 = 8,6

Masukkan untung bulan ke- 3 = 7,5

=====

Algoritma Brute Force

Total keuntungan perusahaan selama 3 bulan adalah = 25.2

=====

Algoritma Divide Conquer

Total keuntungan perusahaan selama 3 bulan adalah = 25.2

=====



NAMA : Dina Rahmawati

NIM : 2341720050

KELAS : TI 1G

MATERI : BRUTE FORCE DAN DIVIDE CONQUER

4.5 Latihan Praktikum

1. Sebuah showroom memiliki daftar mobil dengan data sesuai tabel di bawah ini

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

Tentukan:

- a) top_acceleration tertinggi menggunakan Divide and Conquer!
- b) top_acceleration terendah menggunakan Divide and Conquer!
- c) Rata-rata top_power dari seluruh mobil menggunakan Brute Force!

https://github.com/dinaarh/Prak_Algoritma_Struktur_Data10.git