NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

4.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mahasiswa mampu membuat algoritma bruteforce dan divide-conquer

2. Mahasiswa mampu menerapkan penggunaan algorima bruteforce dan divide-conquer

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

```java
package Minggu5;

public class Faktorial {
    int nilai;

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

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

```java
package Minggu5;

import java.util.Scanner;

public class MainFaktorial {
    public static void main(String[] args) {
        Scanner sc04 = new Scanner(System.in);
        System.out.println("================================");
        System.out.println("Masukkan jumlah elemen yang ingin dihitung: ");
        int iJml = sc04.nextInt();

        Faktorial[] fk = new Faktorial[10];
        for (int i = 0; i < iJml; i++) {
            fk[i] = new Faktorial();
            System.out.println("Masukkan nilai data ke-" + (i + 1) + " : ");
            int iNilai = sc04.nextInt();
            fk[i].nilai = iNilai;
        }
        System.out.println("HASIL - BRUTE FORCE");
        for (int i = 0; i < iJml; i++) {
            System.out.println("Hasil penghitungan faktorial menggunakn Brute Force adalah " + fk[i].faktorialBF(fk[i].nilai));
        }
        System.out.println("HASIL - DIVIDE CONQUER");
        for (int i = 0; i < iJml; i++) {
            System.out.println("Hasil penghitungan faktorial menggunakan Divide Conquer adalah " + fk[i].faktorialDC(fk[i].nilai));

        }
        sc04.close();
    }
}
```

```
Masukkan jumlah elemen yang ingin dihitung:
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 menggunakn Brute Force adalah 120
Hasil penghitungan faktorial menggunakn Brute Force adalah 40320
Hasil penghitungan faktorial menggunakn Brute Force adalah 6
HASIL - DIVIDE CONQUER
Hasil penghitungan faktorial menggunakan Divide Conquer adalah 120
Hasil penghitungan faktorial menggunakan Divide Conquer adalah 40320
Hasil penghitungan faktorial menggunakan Divide Conquer adalah 6
```

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

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!

Jawaban:

***Pada bagian* If,** *memeriksa apakah n adalah bilangan ganjil dengan menggunakan operasi modulo (%). Jika n adalah bilangan ganjil, maka akan melakukan perhitungan pangkat dengan rumus PangkatDC(a, n/2) * PangkatDC(a, n/2) * a. memanggil fungsi PangkatDC secara rekursif untuk menghitung pangkat dari a dengan eksponen n/2, dan kemudian mengalikan hasilnya dengan a,*

***Sedangkan* else** *memeriksa apakah n bukanlah bilangan ganjil, yang berarti n adalah bilangan genap. Jika demikian, akan melakukan perhitungan pangkat dengan rumus PangkatDC(a, n/2) * PangkatDC(a, n/2).*

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

Jawaban:

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

Memungkinkan dengan merubah perulangan yang semula for menjadi while

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

Jawaban:

fakto *= i; adalah operasi penugasan gabungan yang mengalikan nilai fakto dengan i dan kemudian menyimpan hasilnya kembali ke fakto. Dengan demikian, fakto akan berubah nilainya seiring dengan peningkatan nilai i hingga mencapai n.

Sedangkan , int fakto = n * faktorialDC(n-1); adalah pemanggilan rekursif terhadap fungsi faktorialDC. Pada setiap pemanggilan rekursif, nilai n akan dikalikan dengan hasil dari pemanggilan rekursif faktorialDC(n-1) yang berarti bahwa fungsi faktorialDC akan dipanggil berulang dengan nilai n yang semakin berkurang hingga mencapai kondisi dasar n == 1.

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

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

```java
package Minggu5;

import java.util.Scanner;

public class MainPangkat {
    Run|Debug
    public static void main(String[] args) {
        Scanner sc04 = new Scanner(System.in);
        System.out.println(x:"================================");
        System.out.println(x:"Masukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc04.nextInt();

        Pangkat[] png = new Pangkat[elemen];
        for (int i = 0; i < elemen; i++) {
            png[i] = new Pangkat();
            System.out.println(x:"Masukkan nilai yang hendak dipangkatkan: ");
            int nilai = sc04.nextInt();
            System.out.println(x:"Masukkan nilai pemangkat: ");
            int pangkat = sc04.nextInt();
            png[i].nilai = nilai;
            png[i].pangkat = pangkat;
        }
        System.out.println(x:"HASIL - BRUTE FORCE");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
                    + png[i].PangkatBF(png[i].nilai, png[i].pangkat));
        }
        System.out.println(x:"HASIL - DIVIDE CONQUER");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
                    + png[i].PangkatDC(png[i].nilai, png[i].pangkat));

        }
        sc04.close();

    }
}
```

```java
package Minggu5;

public class Pangkat {
    public int nilai, pangkat;

    int PangkatBF(int a, int n) {
        int hasil = 1 ;
        for (int i = 0; i < n; i++) {
            hasil *= a;
        }
        return hasil;
    }

    int PangkatDC(int a, int n) {
        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));
            }
        }
    }
}
```

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

Jawaban:

Metode PangkatDC() membagi masalah menjadi sub-masalah yang lebih kecil, yaitu menghitung pangkat dari setengah nilai n dan mengalikan hasilnya. Sebaliknya, metode PangkatBF() menggunakan pendekatan iteratif sederhana, yaitu

melakukan perulangan sebanyak n kali untuk mengalikan basis dengan dirinya sendiri n kali.

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

Jawaban:

Tahap combine tidak termasuk dalam kode tersebut. Dalam kode yang diberikan, hanya terdapat implementasi dari metode PangkatBF dan PangkatDC untuk menghitung pangkat bilangan. Tahap combine tidak ada dalam kode tersebut.

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

Jawaban:

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

```
png[i] = new Pangkat(nilai, pangkat);
png[i].nilai = nilai;
png[i].pangkat = pangkat;
```

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

Jawaban:

```
        System.out.println("Pilih metode yang ingin digunakan:");
        System.out.println("1. Brute Force");
        System.out.println("2. Divide Conquer");
        System.out.println("================================"+ "\nMasukkan nomor metode yang ingin digunakan: ");
        int metode = sc04.nextInt();

        switch (metode) {
            case 1:
                System.out.println("HASIL - BRUTE FORCE");
                for (int i = 0; i < elemen; i++) {
                    System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
                            + png[i].PangkatBF(png[i].nilai, png[i].pangkat));
                }
                break;
            case 2:
                System.out.println("HASIL - DIVIDE CONQUER");
                for (int i = 0; i < elemen; i++) {
                    System.out.println("Hasil dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah "
                            + png[i].PangkatDC(png[i].nilai, png[i].pangkat));
                }
                break;
            default:
                System.out.println("Metode yang dipilih tidak valid.");
                break;
        }

        sc04.close();
    }
}
```

```
================================
Masukkan jumlah elemen yang ingin dihitung:
2
Masukkan nilai yang hendak dipangkatkan:
5
Masukkan nilai pemangkat:
2
Masukkan nilai yang hendak dipangkatkan:
4
Masukkan nilai pemangkat:
2
Pilih metode yang ingin digunakan:
1. Brute Force
2. Divide Conquer
================================
Masukkan nomor metode yang ingin digunakan:
1
HASIL - BRUTE FORCE
Hasil dari 5 pangkat 2 adalah 25
Hasil dari 4 pangkat 2 adalah 16
```

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

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

```java
package Minggu5;

public class Sum {
    int elemen;
    double Keuntungan[],total;

    Sum(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);
            double rsum = totalDC(arr, mid + 1, r);
            return lsum + rsum;
        } else {
            return 0;
        }
    }
}
```

```java
package Minggu5;

import java.util.Scanner;

public class MainSum {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc04 = new Scanner(System.in);
        System.out.println(x:"================================");
        System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9M)");
        System.out.print(s:"Masukkan Jumlah Bulan: ");
        int elm = sc04.nextInt();

        Sum sm = new Sum(elm);
        System.out.println(x:"================================");
        for (int i = 0; i < elm; i++) {
            System.out.print("Masukkan untung bulan ke-" + (i + 1) + " = ");
            sm.Keuntungan[i] = sc04.nextDouble();
        }
        System.out.println(x:"================================");
        System.out.println(x:"Algoritma Brute Force");
        System.out.println("Total keuntungan perusahaan selama " + elm + " bulan adalah = " + sm.totalBF(sm.Keuntungan));
        System.out.println(x:"================================");
        System.out.println(x:"Algoritma Divide Conquer");
        System.out.println("Total keuntungan perusahaan selama " + elm + " bulan adalah = " + sm.totalDC(sm.Keuntungan, l:0, elm - 1));
        sc04.close();
    }
}
```

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

```
================================
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9M)
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 = 40.339999999999996
```

4.4.3 Pertanyaan

1. Mengapa terdapat formulasi return value berikut? Jelaskan

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

Jawaban:

Karena Formulasi return lsum + rsum + arr[mid]; digunakan untuk menghitung total dari elemen-elemen dalam array arr yang berada di antara indeks l dan r, termasuk elemen pada indeks mid.

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

Jawaban:

Variabel mid digunakan dalam metode totalDC() untuk menentukan titik tengah dari rentang array yang sedang diproses. Rentang array ini dibagi menjadi dua bagian: bagian kiri dan bagian kanan.

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

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
KELAS : TI-1G
MATERI : BRUTE FORCE DAN DIVIDE CONQUER

berbeda-beda)? Buktikan dengan program!

```java
package Minggu5;

import java.util.Scanner;

public class MainSum {
    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 5.9M)");
        System.out.print("Masukkan Jumlah Perusahaan: ");
        int jumlahperusahaan = sc.nextInt();

        double[][] keuntunganPerusahaan = new double[jumlahperusahaan][];

        for (int i = 0; i < jumlahperusahaan; i++) {
            System.out.print("Masukkan Jumlah Bulan untuk Perusahaan ke-" + (i + 1) + ": ");
            int JumBulan = sc.nextInt();
            keuntunganPerusahaan[i] = new double[JumBulan];

            for (int j = 0; j < JumBulan; j++) {
                System.out.print("Masukkan keuntungan bulan ke-" + (j + 1) + " untuk Perusahaan ke-" + (i + 1) + " = ");
                keuntunganPerusahaan[i][j] = sc.nextDouble();
            }
        }

        System.out.println("================================");
        System.out.println("Algoritma Brute Force");

        for (int i = 0; i < jumlahperusahaan; i++) {
            double totalKeuntunganBF = Sum.totalBF(keuntunganPerusahaan[i]);
            System.out.println("Total keuntungan perusahaan ke-" + (i + 1) + " adalah = " + totalKeuntunganBF);
        }

        System.out.println("================================");
        System.out.println("Algoritma Divide Conquer");

        for (int i = 0; i < jumlahperusahaan; i++) {
            double totalKeuntunganDC = Sum.totalDC(keuntunganPerusahaan[i], 0, keuntunganPerusahaan[i].length - 1);
            System.out.println("Total keuntungan perusahaan ke-" + (i + 1) + " adalah = " + totalKeuntunganDC);
        }

        sc.close();
    }
}
```

```java
public class Sum {
    int elemen;
    double Keuntungan[],total;

    Sum(int elemen){
        this.elemen = elemen;
        this.Keuntungan = new double[elemen];
        this.total = 0;
    }
    public static double totalBF(double[] keuntungan) {
        double total = 0;
        for (int i = 0; i < keuntungan.length; i++) {
            total = total + keuntungan[i];
        }
        return total;       You, 1 hour ago • P5_ Percobaan 3_ 04 _ PASD
    }


    public static double totalDC(double[] keuntungan, int l, int r){
        if (l == r) {
            return keuntungan[l];
        } else if (l < r) {
            int mid = (l + r) / 2;
            double lsum = totalDC(keuntungan, l, mid-1);
            double rsum = totalDC(keuntungan, mid + 1, r);
            return lsum + rsum+keuntungan[mid];
        } else {
            return 0;
        }
    }
}
```

```
================================
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9M)
Masukkan Jumlah Perusahaan: 2
Masukkan Jumlah Bulan untuk Perusahaan ke-1: 5
Masukkan keuntungan bulan ke-1 untuk Perusahaan ke-1 = 8.5
Masukkan keuntungan bulan ke-2 untuk Perusahaan ke-1 = 9.54
Masukkan keuntungan bulan ke-3 untuk Perusahaan ke-1 = 7.2
Masukkan keuntungan bulan ke-4 untuk Perusahaan ke-1 = 9.1
Masukkan keuntungan bulan ke-5 untuk Perusahaan ke-1 = 6
Masukkan Jumlah Bulan untuk Perusahaan ke-2: 1
Masukkan keuntungan bulan ke-1 untuk Perusahaan ke-2 = 10.6
================================
Algoritma Brute Force
Total keuntungan perusahaan ke-1 adalah = 40.339999999999996
Total keuntungan perusahaan ke-2 adalah = 10.6
================================
Algoritma Divide Conquer
Total keuntungan perusahaan ke-1 adalah = 40.34
Total keuntungan perusahaan ke-2 adalah = 10.6
```

NAMA : Ahmad Fadlih Wahyu Sardana
NIM : 2341720069
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!