PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET PERTEMUAN KE-4

RIO TRI PRAYOGO

TI 1A

26

2341720236

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Brute Force dan Divide Conquer

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

Percobaan :

Main

```
package minggu5;


import java.util.Scanner;


public class MainFaktorial26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("-----------------");
        System.out.println("Masukkan jumlah elemen: ");
        int iJml = scan.nextInt();
        Faktorial[] fk = new Faktorial[iJml];
        for (int i = 0; i < iJml; i++) {
            fk[i] = new Faktorial();
            System.out.println("Masukkan nilai data ke-" + (i + 1) + ":");
            fk[i].nilai = scan.nextInt();
        }
        System.out.println("HASIL - BRUTE FORCE");
        for (int i = 0; i < iJml; i++) {
            System.out.println(
                    "Hasil penghitungan faktorial menggunakan Brute Force
adalah " + fk[i].faktorialBF(fk[i].nilai));
        }
        System.out.println("HASIL - DIVIDE AND CONQUER");
        for (int i = 0; i < iJml; i++) {
            System.out.println(
                    "Hasil penghitungan faktorial menggunakan Divide and
Conquer adalah "
                            + fk[i].faktorialDC(fk[i].nilai));
        }
    }
}
```

Object

```
package minggu5;


public class Faktorial {
    public int nilai;
    int faktorialBF(int n) {
        int fakto = 1;
        for (int i = 1; i <= n; i++) {
            fakto = fakto * i;
        }
        return fakto;
    }

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

Output :

```
----------------
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!
   = Pada *if* pertama berfungsi untuk mengecek apakah parameter *n* bernilai 1(satu), jika bernilai satu maka akan langsung melakukan return nilai faktorial berupa 1(satu) jika tidak maka akan masuk ke dalam *else*. Pada bagian ini parameter *n* dihitung dengan memanggil fungsi `faktorialDC()` secara rekursif dengan parameter *(n-1)* dan dikalikan dengan *n*. Parameter *(n-1)* berfungsi untuk membagi/memisahkan submasalah menjadi lebih kecil sehingga terjadi pemisahan pada setiap rekursif.
2. Apakah memungkinkan perulangan pada method `faktorialBF()` dirubah selain menggunakan for?Buktikan!
   = Bisa, perulangan tersebut bisa diubah menggunakan *do-while* seperti pada kode dibawah ini:

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

3. Jelaskan perbedaan antara **fakto *= i;** dan **int fakto = n * faktorialDC(n-1);** !
   - **fakto *= i;** = Baris tersebut menghitung nilai faktorial secara iteratif dimana menghitung setiap faktor hingga n secara langsung. Biasa digunakan dalam algoritma Brute Force.
   - **int fakto = n * faktorialDC(n-1);** = Baris tersebut menghitung nilai faktorial secara rekursif dimana perhitungan dilakukan dengan membagi masalah menjadi beberapa bagian yang lebih kecil melalui parameter *(n-1)*. Biasa digunakan dalam algoritma Divide and Conquer.

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

Percobaan :

Main

```
package minggu5;

import java.util.Scanner;

public class MainPangkat26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
```

```
        System.out.println("=======================================");
        System.out.println("Masukkan jumlah elemen yang dihitung: ");
        int elemen = scan.nextInt();

        Pangkat[] png = new Pangkat[elemen];
        for (int i = 0; i < elemen; i++) {
            png[i] = new Pangkat();
            System.out.println("Masukkan nilai yang hendak dipangkatkan:
");
            png[i].nilai = scan.nextInt();
            System.out.println("Masukkan nilai pangkat: ");
            png[i].pangkat = scan.nextInt();
        }

        System.out.println("HASIL PANGKAT - BRUTFORCE");
        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("HASIL PANGKAT - DIVIDE AND 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));
        }
    }
}
```

Object

```
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 == 0) {
            return 1;
        } 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);
            }
        }
    }
}
```

Output :

```
==========================================
Masukkan jumlah elemen yang dihitung:
2
Masukkan nilai yang hendak dipangkatkan:
6
Masukkan nilai pangkat:
2
Masukkan nilai yang hendak dipangkatkan:
4
Masukkan nilai pangkat:
3
HASIL PANGKAT - BRUTFORCE
 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
```

Pertanyaan :

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu `PangkatBF()` dan `PangkatDC()`!
   = Perbedaan kedua method tersebut terletak pada cara bagaimana menghitung pangkat. Method `PangkatBF()` menghitungnya dengan cara iteratif dimana melakukan pengulangan secara langsung sebanyak nilai yang diinginkan (*n*) untuk mengalikan *a* dengan dirinya sendiri. Sedangkan method `PangkatDC()` menghitung pangkat dengan cara rekursif dan membagi masalah menjadi beberapa bagian menjadi lebih kecil melalui parameter *(n-2)* sehingga mengurangi jumlah operasi yang diperlukan.
2. Apakah tahap *combine* sudah termasuk dalam kode tersebut?Tunjukkan!
   = Tahap *combine* sudah termasuk dalam kode diatas dan bisa ditunjukkan pada line:

```
return pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a;
```

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

```
package minggu5;


import java.util.Scanner;


public class MainPangkat26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);


        System.out.println("========================================");
        System.out.println("Masukkan jumlah elemen yang dihitung: ");
        int elemen = scan.nextInt();


        Pangkat[] png = new Pangkat[elemen];
        for (int i = 0; i < elemen; i++) {
            System.out.println("Masukkan nilai yang hendak dipangkatkan:
");
            int nilai = scan.nextInt();
            System.out.println("Masukkan nilai pangkat: ");
            int pangkat = scan.nextInt();
            png[i] = new Pangkat(nilai, pangkat);
        }


        System.out.println("HASIL PANGKAT - BRUTFORCE");
        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("HASIL PANGKAT - DIVIDE AND 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));
        }
    }
}
```

Object

```
package minggu5;

public class Pangkat {
    public int nilai, pangkat;

    public Pangkat(int a, int n) {
        nilai = a;
        pangkat = n;
    }

    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 == 0) {
            return 1;
```

```
        } 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);
            }
        }
    }
}
```

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

```
package minggu5;

import java.util.Scanner;

public class MainPangkat26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        boolean run = true;
        do {

            System.out.println("\n=======================================
==");
            System.out.println("Program penghitung pangkat");
            System.out.println("=======================================
");
            System.out.println("Masukkan jumlah elemen yang dihitung:
");
            int elemen = scan.nextInt();

            Pangkat[] png = new Pangkat[elemen];
            for (int i = 0; i < elemen; i++) {
                System.out.println("Masukkan nilai yang hendak
dipangkatkan: ");
                int nilai = scan.nextInt();
                System.out.println("Masukkan nilai pangkat: ");
                int pangkat = scan.nextInt();
```

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

            System.out.println("========================================
");
            System.out.println("Pilih algoritma untuk menghitung");
            System.out.println("========================================
");
            System.out.println("1. Brute Force");
            System.out.println("2. Divide and Conquer");
            System.out.println("0. Keluar");
            System.out.print("(1/2/0): ");
            int input = scan.nextInt();
            switch (input) {
                case 1:
                    System.out.println("================================
========");
                    System.out.println("HASIL PANGKAT - BRUTFORCE");
                    System.out.println("================================
========");
                    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("================================
========");
                    System.out.println("HASIL PANGKAT - DIVIDE AND
CONQUER");
                    System.out.println("================================
========");
                    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;
                    case 0:
                        System.out.println("================================
========");
                        System.out.println("Terima Kasih!");
                        run = false;
                        break;
                    default:
                        System.out.println("================================
========");
                        System.out.println("Pilihan tidak valid");
                        break;
                }
            } while (run);
        }
}
```

Object : Tetap sama dan tidak ada perubahan

Output

```
========================================
Program penghitung pangkat
========================================
Masukkan jumlah elemen yang dihitung:
1
Masukkan nilai yang hendak dipangkatkan:
2
Masukkan nilai pangkat:
6
========================================
Pilih algoritma untuk menghitung
========================================
1. Brute Force
2. Divide and Conquer
0. Keluar
(1/2/0): 1
========================================
HASIL PANGKAT - BRUTFORCE
========================================
Hasil dari 2 pangkat 6 adalah 64
```

```
========================================
Program penghitung pangkat
========================================
Masukkan jumlah elemen yang dihitung:
1
Masukkan nilai yang hendak dipangkatkan:
2
Masukkan nilai pangkat:
6
========================================
Pilih algoritma untuk menghitung
========================================
1. Brute Force
2. Divide and Conquer
0. Keluar
(1/2/0): 2
========================================
HASIL PANGKAT - DIVIDE AND CONQUER
========================================
Hasil dari 2 pangkat 6 adalah 64
```

Praktikum 3 : Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer

Percobaan :

Main

```
package minggu5;


import java.util.Scanner;


public class MainSum26 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("==========================================
================");

        System.out.println("Program Menghitung Keuntungan Total (Satuan
Juta. Misal 5.9)");

        System.out.print("Masukkan jumlah bulan: ");

        int elm = scan.nextInt();

        Sum sm = new Sum(elm);

        System.out.println("==========================================
============");

        for (int i = 0; i < sm.elemen; i++) {

            System.out.print("Masukkan untung bulan ke- " + (i + 1) + " =
");

            sm.keuntungan[i] = scan.nextDouble();

        }

        System.out.println("------------------------------------------
---------------");

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

        System.out.println(

                "Total keuntungan perusahaan selama " + sm.elemen + " bulan
adalah = " + sm.totalBF(sm.keuntungan));

        System.out.println("==========================================
===============");

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

        System.out.println(

                "Total keuntungan perusahaan selama " + sm.elemen + " bulan
adalah = "

                        + sm.totalDC(sm.keuntungan, 0, sm.elemen - 1));

    }

}
```

Object

```
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 - 1);
            double rsum = totalDC(arr, mid + 1, r);
            return lsum + rsum + arr[mid];
        }
        return 0;

    }
}
```

Output :

```
============================================================
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
```

Pertanyaan :

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

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

   = Formulasi *return value* diatas bermaksud untuk melakukan *combine* atau menggabungkan hasil *sum* dari array yang telah dilakukan secara terpisah menggunakan algoritma Divide Conquer. `lsum` merupakan hasil *sum* dari array bagian kiri, `rsum` merupakan hasil *sum* dari array bagian kanan, sementara `arr[mid]` array bagian tengah. Ketiga bagian array tersebut digabungkan untuk mendapatkan output total dari *sum*.
2. Kenapa dibutuhkan variable `mid` pada method `TotalDC()`?
   = Karena algoritma `TotalDC()` membagi array menjadi 2 bagian maka dibutuhkan nilai tengah untuk menandai bagian mana yang menjadi tengah-tengah dari hasil pembagi 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!

```
package minggu5;


import java.util.Scanner;


public class MainSum26 {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("================================");
        System.out.println("Program menghitung Keuntungan Total");
        System.out.print("Masukkan jumlah bulan :");
        int elm = scan.nextInt();
```

```
        Sum sm = new Sum(elm);
        System.out.println("=========================================="
);
        for (int i = 0; i < sm.elemen; i++) {
            System.out.print("Masukkan keuntungan bulan ke - " + (i + 1)
+ " = ");
            sm.keuntungan[i] = scan.nextDouble();
        }

        System.out.println("=========================================
==================");
        System.out.println("ALGORITMA BRUCE FORCE");
        System.out.println(
                "Total Keuntungan perusahaan selama " + sm.elemen + "
bulan adalah " + sm.totalBF(sm.keuntungan));
        System.out.println("=========================================
==================");
        System.out.println("Algoritma Divide Conquer");
        System.out.println("Total keuntungan perusahaan selama " +
sm.elemen + " bulan adalah "
                + sm.totalDC(sm.keuntungan, 0, sm.elemen - 1));
        System.out.println("=========================================
==================");
    }

}
```

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!

Main

```
package minggu5;


public class LatihanMain26 {

    public static void main(String[] args) {

        int[] Accel = { 1111, 2222, 3333, 4444, 5555 };

        int[] topPower = { 555, 444, 333, 222, 111 };

        Latihan lat = new Latihan(Accel, topPower);


        System.out.println("_____________________________________________________
_____________________");

        System.out.println("|Merk        | Tipe            | Tahun |
top_acceleration | top_power |");

        System.out.println("_____________________________________________________
_____________________");

        System.out.println("|Mazda       | RX-7            | 2016  |
1111             | 555       |");

        System.out.println("|BMW         | GTR 3           | 2009  |
2222             | 444       |");

        System.out.println("|Subaru      | Impreza WRX STI | 2013  |
3333             | 333       |");

        System.out.println("|Toyota      | AE86 Trueno     | 1986  |
4444             | 222       |");
```

```
        System.out.println("|Volkswagen  | Golf GTI        | 2014  |
5555             | 111       |");
        System.out.println("==========================================
========================");


        System.out.println();


        System.out.println("Top Acceleration adalah " +
lat.highestAccel());
        System.out.println("Low Acceletaion adalah " + lat.lowAccel());
        System.out.println("Rata Rata power adalah " + lat.avgPower());
    }


}
```

Object

```
package minggu5;
public class Latihan {
    int[] Accel;
    int[] topPower;

    Latihan(int[] Accel, int[] topPower) {
        this.Accel = Accel;
        this.topPower = topPower;
    }

    int highestAccel() {
        if (Accel.length == 1) {
            return Accel[0];
        }
        return highestAccel(Accel, 0, Accel.length - 1);
    }

    int highestAccel(int[] data, int l, int r) {
        if (l == r) {
            return data[l];
        }
```

```
        int mid = (r + l) / 2;
        int lAccel = highestAccel(data, l, mid);
        int rAccel = highestAccel(data, mid + 1, r);

        if (lAccel > rAccel) {
            return lAccel;
        } else {
            return rAccel;
        }
    }

    int lowAccel() {
        if (Accel.length == 1) {
            return Accel[0];
        }
        return lowAccel(Accel, 0, Accel.length - 1);
    }

    int lowAccel(int[] data, int l, int r) {
        if (l == r) {
            return data[l];
        }

        int mid = (r + l) / 2;
        int lAccel = lowAccel(data, l, mid);
        int rAccel = lowAccel(data, mid + 1, r);

        if (lAccel > rAccel) {
            return rAccel;
        } else {
            return lAccel;
        }
    }

    int avgPower() {
        int sum = 0;
```

```
        for (int i : topPower) {
            sum += i;
        }
        sum /= topPower.length;
        return sum;
    }
}
```

Output

```
_____________________________________________________________________
|Merk        | Tipe            | Tahun | top_acceleration | top_power |
_____________________________________________________________________
|Mazda       | RX-7            | 2016  | 1111             | 555       |
|BMW         | GTR 3           | 2009  | 2222             | 444       |
|Subaru      | Impreza WRX STI | 2013  | 3333             | 333       |
|Toyota      | AE86 Trueno     | 1986  | 4444             | 222       |
|Volkswagen  | Golf GTI        | 2014  | 5555             | 111       |
=====================================================================

Top Acceleration adalah 5555
Low Acceletaion adalah 1111
Rata Rata power adalah 333
```