POLITEINIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



#### Nama : Muhammad Nuril Huda

#### Kelas : TI-1A

#### No : 19

#### Mata Kuliah : Algoritma dan Struktur Data

## 5.2 Menghitung Nilai Faktorial dengan Algoritma Brute Force dan DivideandConquer

##### 5.2.1 Kode Program

package Pertemuan5;

public class Faktorial {

    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;

        }

    }

}

package Pertemuan5;

import java.util.Scanner;

public class MainFaktorial {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Masukkan nilai: ");

        int nilai = input.nextInt();

        Faktorial fk = new Faktorial();

        System.out.println("Nilai faktorial "+nilai+" menggunakan BF: "+fk.faktorialBF(nilai));

        System.out.println("Nilai faktorial "+nilai+" menggunakan DC: "+fk.faktorialDC(nilai));

    }

}

##### 5.2.2 Hasil Percobaan

##### 5.2.3 Pertanyaan

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

* If adalah base case dari rekursi. Ketika n sudah mencapai 1, proses rekursi akan berhenti dan mengembalikan nilai 1, karena factorial 1 adalah 1. Sedangkan else adalah recursive case, dimana akan terus mengalikan n dengan hasil pemanggilan rekursif faktorialDC(n-1), yang berarti akan menjadi lebih kecil hingga mendekati base case.

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

* Iya memungkinkan. Bisa menggunakan While.
* Kode Program

package Pertemuan5;

public class Faktorial {

    int faktorialBF(int n) {

        int fakto = 1;

        int i = 1;

        while (i <= n) {

            fakto \*= i;

            i++;

        }

        return fakto;

    }

    int faktorialDC(int n){

        if(n==1){

            return 1;

        } else {

            int fakto = n\*faktorialDC(n-1);

            return fakto;

        }

    }

}

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

* fakto \*= i;

Ini adalah operasi dalam perulangan yang akan terus mengalikan nilai fakto dengan I di setiap iterasi.

* int fakto = n \* faktorialDC(n-1);

Ini adala pemanggilan rekursif, dimana n dikalikan dengan hasil pemanggilan faktorialDC(n-1). Proses ini akan terus berjalan sampai mencapai base case.

1. Buat Kesimpulan tentang perbedaan cara kerja method faktorialBF() dan faktorialDC()!

* FaktorialBF() menggunakan perulangan for dan while, sedangkan faktorialDC() menggunakan pemanggilan rekursif.
* Penggunaan faktorialBF() lebih simple dibandingan dengan faktorialDC().
* FaktorialBF lebih cepat dan hemat memori dibandikan faktorialDC yang banyak melakukan pemanggilan rekursi.

## 5.3 Menghitung Hasil Pangkat dengan Algoritma Brute Force dan DivideandConquer

##### 5.3.1 Kode Program

package Pertemuan5;

public class Pangkat {

    int nilai, pangkat;

    Pangkat (int n, int p){

        nilai = n;

        pangkat = p;

    }

    int pangkatBF(int a, int n){

        int hasil = 1;

        for (int i=0; i<n; i++){

            hasil = 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));

            }

        }

    }

}

package Pertemuan5;

import java.util.Scanner;

public class MainPangkat {

    public static void main(String[] args) {

        Scanner sc = new Scanner (System.in);

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

        int elemen = sc.nextInt();

        Pangkat[] png = new Pangkat[elemen];

        for (int i=0; i<elemen; i++){

            System.out.print("Masukkan nilai basis elemen ke-"+(i+1)+": ");

            int basis = sc.nextInt();

            System.out.print("Masukkan nilai pangkat elemen ke-"+(i+1)+": ");

            int pangkat = sc.nextInt();

            png [i] = new Pangkat(basis, pangkat);

        }

        System.out.println("HASIL PANGKAT BRUTEFORCE: ");

        for(Pangkat p : png){

            System.out.println(p.nilai+"^"+p.pangkat+": "+p.pangkatBF(p.nilai, p.pangkat));

        }

        System.out.println("HASIL PANGKAT DIVIDE AND CONQUER: ");

        for(Pangkat p : png){

            System.out.println(p.nilai+"^"+p.pangkat+": "+p.pangkatDC(p.nilai, p.pangkat));

        }

    }

}

##### 5.3.2 Hasil Percobaan

##### 5.3.3 Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu pangkatBF() dan pangkatDC()!

* pangkatBF mengunakan perulangan for yaitu dengan cara mengkalikan a sebanyak n.
* pangkatDC menggunakan perulangan rekursif dan menggunakan Teknik divide and conquer yaitu membagi pangkat menjadi dua.

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

* Iya sudah

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

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

1. Pada method pangkatBF()terdapat parameter untuk melewatkan nilai yangakandipangkatkan dan pangkat berapa, padahal di sisi lain di class Pangkat telah adaatribut nilai dan pangkat, apakah menurut Anda method tersebut tetap relevan untuk memiliki parameter? Apakah bisa jika method tersebut dibuat dengan tanpa parameter? Jikabisa, seperti apa method pangkatBF() yang tanpa parameter?
2. Tarik tentang cara kerja method pangkatBF() dan pangkatDC()!