TUGAS 4

ALGORITMA STRUKTUR DATA

Nama	**: Daffa Cahyo Alghifari**
NIM	**: 1941720104**
Kelas	**: TI – 1D**

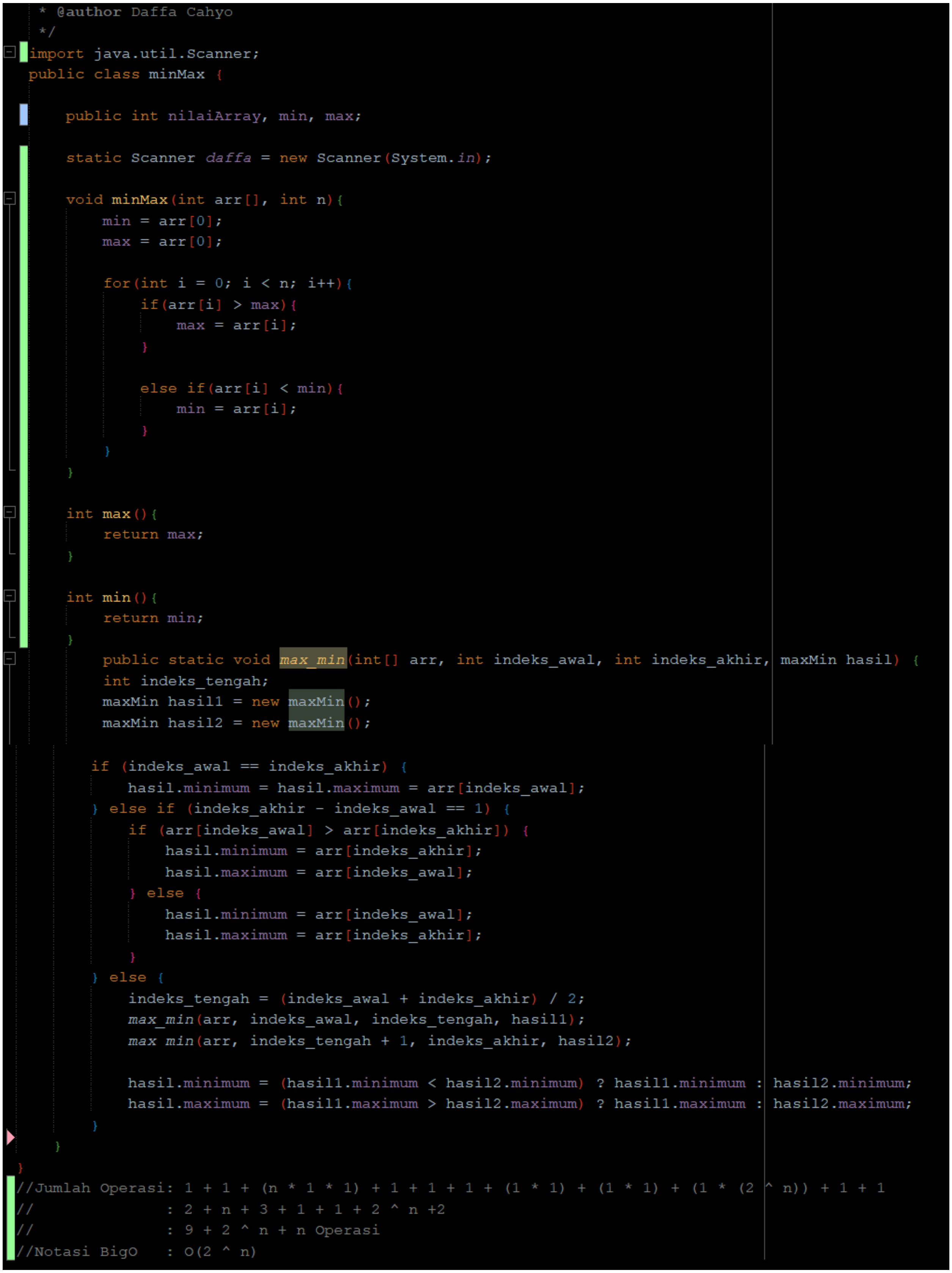
PROGRAM STUDI D4 TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

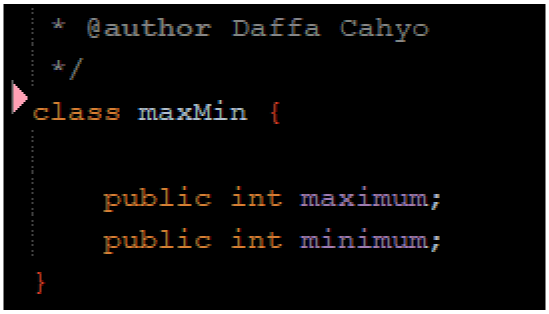
POLITEKNIK NEGERI MALANG

2020

Tugas Nomor 1

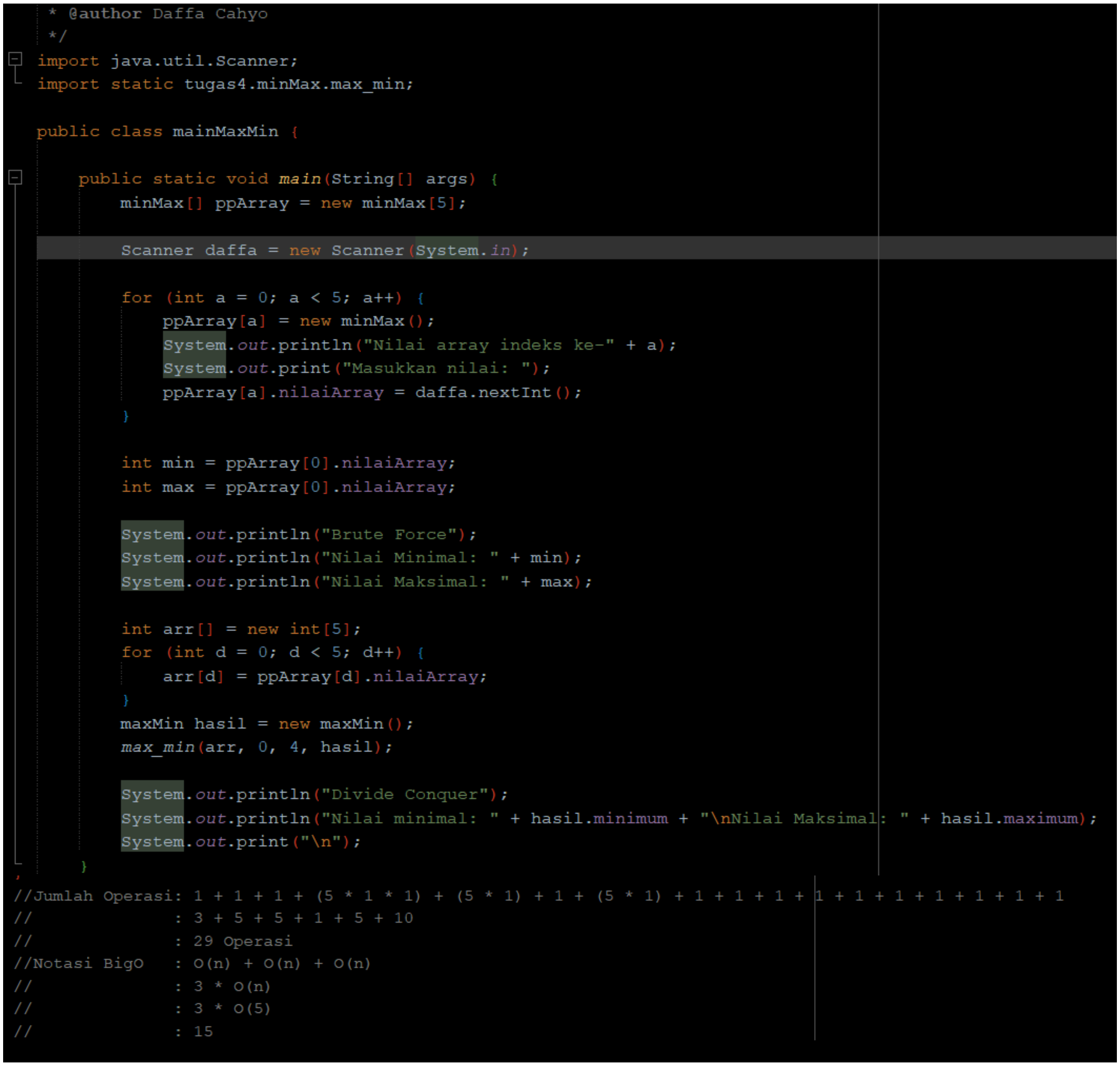


```
 * @author Daffa Cahyo
 */
import java.util.Scanner;
public class minMax {

    public int nilaiArray, min, max;

    static Scanner daffa = new Scanner(System.in);

    void minMax(int arr[], int n){
        min = arr[0];
        max = arr[0];

        for(int i = 0; i < n; i++){
            if(arr[i] > max){
                max = arr[i];
            }

            else if(arr[i] < min){
                min = arr[i];
            }
        }
    }

    int max(){
        return max;
    }

    int min(){
        return min;
    }
        public static void max_min(int[] arr, int indeks_awal, int indeks_akhir, maxMin hasil) {
        int indeks_tengah;
        maxMin hasil1 = new maxMin();
        maxMin hasil2 = new maxMin();

        if (indeks_awal == indeks_akhir) {
            hasil.minimum = hasil.maximum = arr[indeks_awal];
        } else if (indeks_akhir - indeks_awal == 1) {
            if (arr[indeks_awal] > arr[indeks_akhir]) {
                hasil.minimum = arr[indeks_akhir];
                hasil.maximum = arr[indeks_awal];
            } else {
                hasil.minimum = arr[indeks_awal];
                hasil.maximum = arr[indeks_akhir];
            }
        } else {
            indeks_tengah = (indeks_awal + indeks_akhir) / 2;
            max_min(arr, indeks_awal, indeks_tengah, hasil1);
            max_min(arr, indeks_tengah + 1, indeks_akhir, hasil2);

            hasil.minimum = (hasil1.minimum < hasil2.minimum) ? hasil1.minimum : hasil2.minimum;
            hasil.maximum = (hasil1.maximum > hasil2.maximum) ? hasil1.maximum : hasil2.maximum;
        }
    }
}
//Jumlah Operasi: 1 + 1 + (n * 1 * 1) + 1 + 1 + 1 + (1 * 1) + (1 * 1) + (1 * (2 ^ n)) + 1 + 1
//                : 2 + n + 3 + 1 + 1 + 2 ^ n +2
//                : 9 + 2 ^ n + n Operasi
//Notasi BigO     : O(2 ^ n)
```



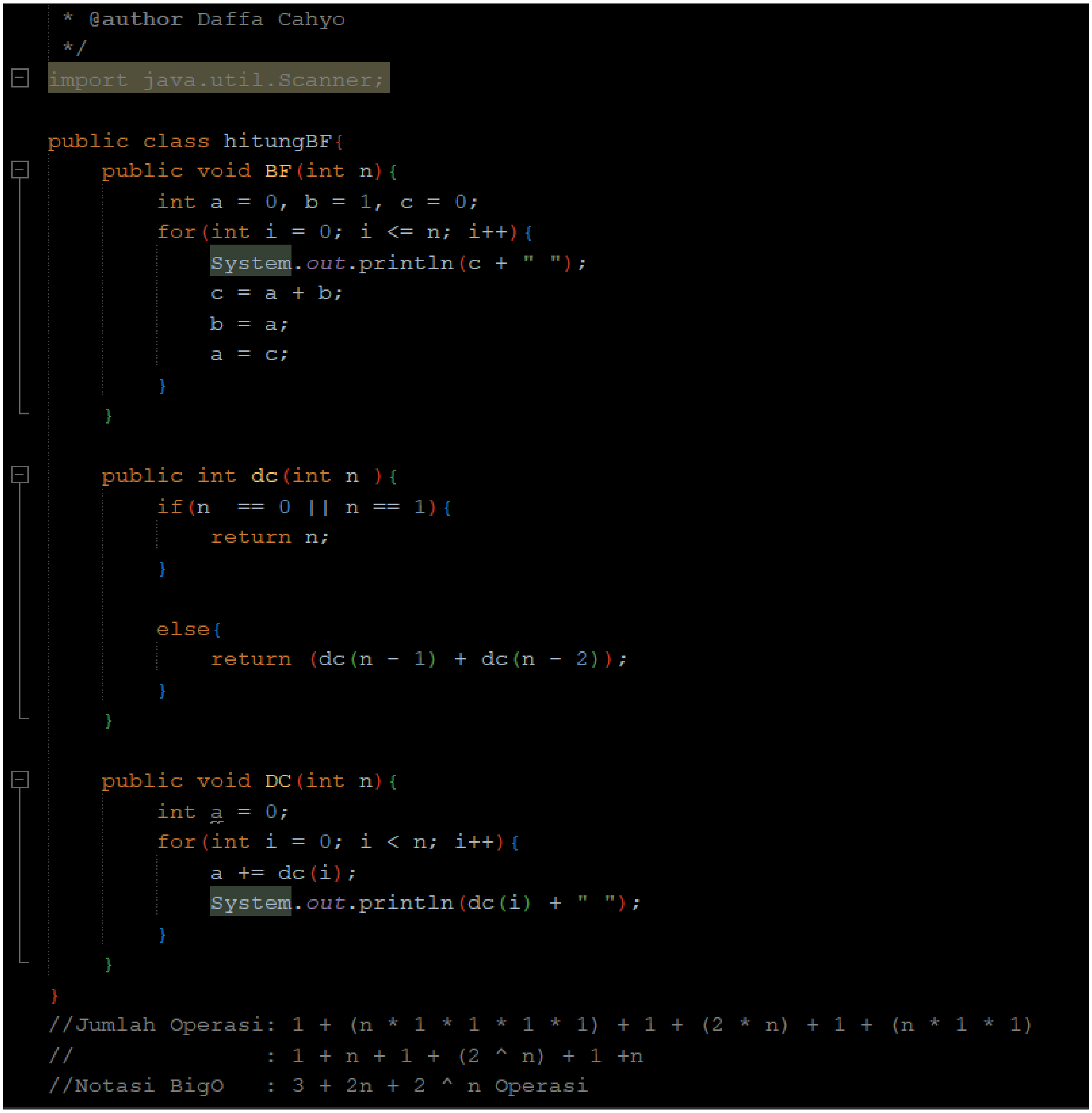
```
 * @author Daffa Cahyo
 */
class maxMin {

    public int maximum;
    public int minimum;
}
```

Main class.



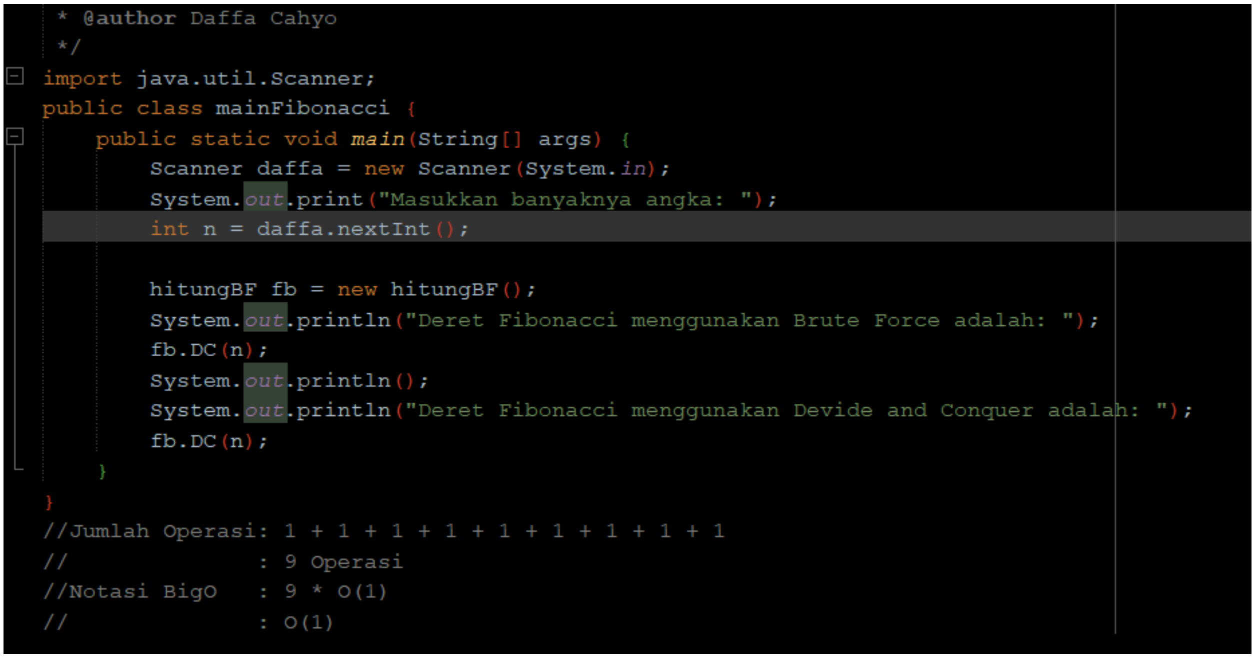
```
 * @author Daffa Cahyo
 */
import java.util.Scanner;
import static tugas4.minMax.max_min;

public class mainMaxMin {

    public static void main(String[] args) {
        minMax[] ppArray = new minMax[5];

        Scanner daffa = new Scanner(System.in);

        for (int a = 0; a < 5; a++) {
            ppArray[a] = new minMax();
            System.out.println("Nilai array indeks ke-" + a);
            System.out.print("Masukkan nilai: ");
            ppArray[a].nilaiArray = daffa.nextInt();
        }

        int min = ppArray[0].nilaiArray;
        int max = ppArray[0].nilaiArray;

        System.out.println("Brute Force");
        System.out.println("Nilai Minimal: " + min);
        System.out.println("Nilai Maksimal: " + max);

        int arr[] = new int[5];
        for (int d = 0; d < 5; d++) {
            arr[d] = ppArray[d].nilaiArray;
        }
        maxMin hasil = new maxMin();
        max_min(arr, 0, 4, hasil);

        System.out.println("Divide Conquer");
        System.out.println("Nilai minimal: " + hasil.minimum + "\nNilai Maksimal: " + hasil.maximum);
        System.out.print("\n");
    }
//Jumlah Operasi: 1 + 1 + 1 + (5 * 1 * 1) + (5 * 1) + 1 + (5 * 1) + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
//              : 3 + 5 + 5 + 1 + 5 + 10
//              : 29 Operasi
//Notasi BigO   : O(n) + O(n) + O(n)
//              : 3 * O(n)
//              : 3 * O(5)
//              : 15
```

Tugas Nomor 2



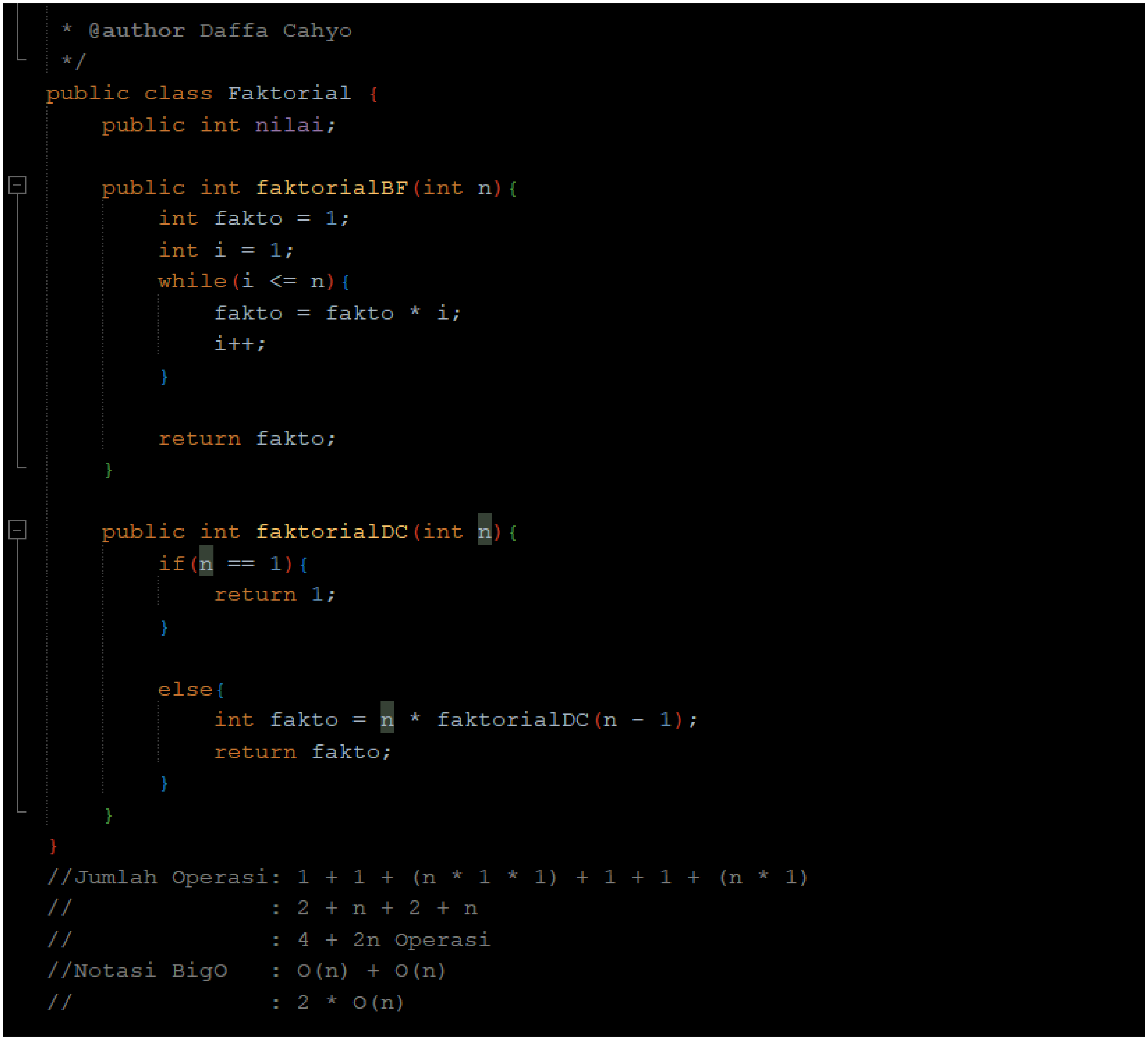
```
 * @author Daffa Cahyo
 */
import java.util.Scanner;

public class hitungBF{
    public void BF(int n){
        int a = 0, b = 1, c = 0;
        for(int i = 0; i <= n; i++){
            System.out.println(c + " ");
            c = a + b;
            b = a;
            a = c;
        }
    }

    public int dc(int n ){
        if(n  == 0 || n == 1){
            return n;
        }

        else{
            return (dc(n - 1) + dc(n - 2));
        }
    }

    public void DC(int n){
        int a = 0;
        for(int i = 0; i < n; i++){
            a += dc(i);
            System.out.println(dc(i) + " ");
        }
    }
}
//Jumlah Operasi: 1 + (n * 1 * 1 * 1 * 1) + 1 + (2 * n) + 1 + (n * 1 * 1)
//               : 1 + n + 1 + (2 ^ n) + 1 +n
//Notasi BigO    : 3 + 2n + 2 ^ n Operasi
```

Main class.



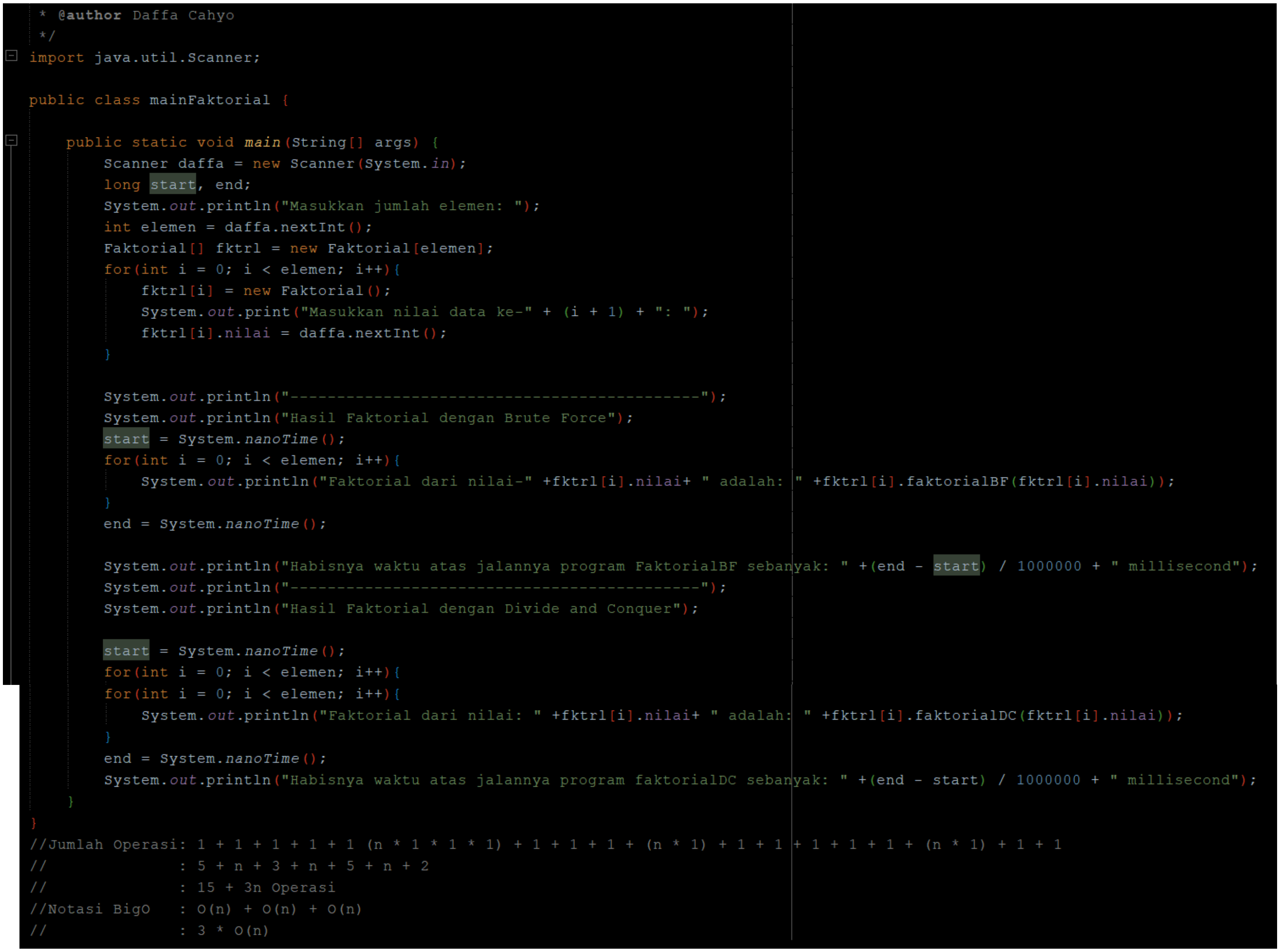
```java
 * @author Daffa Cahyo
 */
import java.util.Scanner;
public class mainFibonacci {
    public static void main(String[] args) {
        Scanner daffa = new Scanner(System.in);
        System.out.print("Masukkan banyaknya angka: ");
        int n = daffa.nextInt();

        hitungBF fb = new hitungBF();
        System.out.println("Deret Fibonacci menggunakan Brute Force adalah: ");
        fb.DC(n);
        System.out.println();
        System.out.println("Deret Fibonacci menggunakan Devide and Conquer adalah: ");
        fb.DC(n);
    }
}
//Jumlah Operasi: 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
//                : 9 Operasi
//Notasi BigO   : 9 * O(1)
//                : O(1)
```

Tugas Nomor 3



```
 * @author Daffa Cahyo
 */
public class Faktorial {
    public int nilai;

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

        return fakto;
    }

    public int faktorialDC(int n){
        if(n == 1){
            return 1;
        }

        else{
            int fakto = n * faktorialDC(n - 1);
            return fakto;
        }
    }
}
//Jumlah Operasi: 1 + 1 + (n * 1 * 1) + 1 + 1 + (n * 1)
//              : 2 + n + 2 + n
//              : 4 + 2n Operasi
//Notasi BigO   : O(n) + O(n)
//              : 2 * O(n)
```

Main class.



```
 * @author Daffa Cahyo
 */
import java.util.Scanner;

public class mainFaktorial {

    public static void main(String[] args) {
        Scanner daffa = new Scanner(System.in);
        long start, end;
        System.out.println("Masukkan jumlah elemen: ");
        int elemen = daffa.nextInt();
        Faktorial[] fktrl = new Faktorial[elemen];
        for(int i = 0; i < elemen; i++){
            fktrl[i] = new Faktorial();
            System.out.print("Masukkan nilai data ke-" + (i + 1) + ": ");
            fktrl[i].nilai = daffa.nextInt();
        }

        System.out.println("--------------------------------------------");
        System.out.println("Hasil Faktorial dengan Brute Force");
        start = System.nanoTime();
        for(int i = 0; i < elemen; i++){
            System.out.println("Faktorial dari nilai-" +fktrl[i].nilai+ " adalah: " +fktrl[i].faktorialBF(fktrl[i].nilai));
        }
        end = System.nanoTime();

        System.out.println("Habisnya waktu atas jalannya program FaktorialBF sebanyak: " +(end - start) / 1000000 + " millisecond");
        System.out.println("--------------------------------------------");
        System.out.println("Hasil Faktorial dengan Divide and Conquer");

        start = System.nanoTime();
        for(int i = 0; i < elemen; i++){
        for(int i = 0; i < elemen; i++){
            System.out.println("Faktorial dari nilai: " +fktrl[i].nilai+ " adalah: " +fktrl[i].faktorialDC(fktrl[i].nilai));
        }
        end = System.nanoTime();
        System.out.println("Habisnya waktu atas jalannya program faktorialDC sebanyak: " +(end - start) / 1000000 + " millisecond");
    }
}
//Jumlah Operasi: 1 + 1 + 1 + 1 + 1 (n * 1 * 1 * 1) + 1 + 1 + 1 + (n * 1) + 1 + 1 + 1 + 1 + 1 + (n * 1) + 1 + 1
//              : 5 + n + 3 + n + 5 + n + 2
//              : 15 + 3n Operasi
//Notasi BigO   : O(n) + O(n) + O(n)
//              : 3 * O(n)
```