

FINAL PROJECT 2014

No	Tugas	Kelompok
1	Graph, melengkapi method-method yang terdapat pada interface Graph	Elham Budianto Taufik Arifuddin
2	Algoritma Traversal Graph, 5 studi kasus	Khofi Annisa Arif Akhmad Fajri Yudiharto
3	Graph : Dijkstra's Minimum-Path Algorithma (ch25 hal 33)	Abimanyu Bagarela Anjaya Putra Kaharudin Triambudi
4	Graph : Minimum Spanning Tree(Prim's Algorithm) – (ch25 hal 60)	Uswatun Hasanah Raizal Islami Nursyah Pregnanta
5	Membaca Tree dengan menggunakan, Inorder rekursif, Inorder Non Rekursif dan menerapkan Iterator Inorder Node pada Tree adalah data mahasiswa berupa nama dan nrp. Aturan dalam membangun tree menggunakan metode Binary Search Tree	Ana Faradina Rizqi Amalia
6	Membaca Tree dengan menggunakan Preorder rekursif, Preorder Non Rekursif dan menerapkan Iterator preorder Node pada Tree adalah data mahasiswa berupa nama dan nrp. Aturan dalam membangun tree menggunakan metode Binary Search Tree	Rendy Tri Swasono Muhamma d Anim Falahuddin
7	Membaca Tree dengan menggunakan Postorder rekursif, PostOrder Non Rekursif dan menerapkan Iterator postorder Node pada Tree adalah data mahasiswa berupa nama dan nrp. Aturan dalam membangun tree menggunakan metode Binary Search Tree	Muhamma d Masruhin Ina Ratudduja
8	Bangunlah Tree dari notasi postfix menjadi Tree. Tambahkan pada class PostfixEval method <pre>public static TNode<Character> buildExpTree(String postfixExp)</pre> Algoritma mengubah notasi Postfix menjadi Tree Input dari algoritma ini adalah String postfix. Panjang dari string postfix adalah N karakter. Siapkan Stack kosong S, untuk menyimpan TNode yang berisi data operand. Dimulai dari i = 0 sampai N-1 kerjakan langkah-langkah sebagai berikut: <ul style="list-style-type: none"> • R = postfix[i] • Test nilai R. Jika R adalah: 	Rizalsyah Wahyu Harida Muhamma d Anang Maruf


```

1 public class TestPolynomial {
2     public static void main(String[] args) {
3         Polynomial p = new Polynomial(3, -8, 0, 0, 2, 1);
4         Polynomial q = new Polynomial(0, 5, 6, 9);
5         System.out.println("p: " + p);
6         System.out.println("p.degree(): " + p.degree());
7         System.out.println("q: " + q);
8         System.out.println("q.degree(): " + q.degree());
9         System.out.println("p.plus(q): " + p.plus(q));
10        System.out.println("q.plus(p): " + q.plus(p));
11        System.out.println("p.plus(q).degree(): " + p.plus(q).degree());
12        Polynomial z = new Polynomial(0);
13        System.out.println("z: " + z);
14        System.out.println("z.degree(): " + z.degree());
15        System.out.println("p.plus(z): " + p.plus(z));
16        System.out.println("z.plus(p): " + z.plus(p));
17        System.out.println("p: " + p);
18        Polynomial t = new Polynomial(8.88, 44);
19        System.out.println("t: " + t);
20        System.out.println("t.degree(): " + t.degree());
21    }
22 }

```

Output :

```

p: 3.00 - 8.00x + 2.00x^4 + 1.00x^5
p.degree(): 5
q: 5.00x + 6.00x^2 + 9.00x^3
q.degree(): 3
p.plus(q): 3.00 - 3.00x + 6.00x^2 + 9.00x^3 + 2.00x^4 + 1.00x^5
q.plus(p): 3.00 - 3.00x + 6.00x^2 + 9.00x^3 + 2.00x^4 + 1.00x^5
p.plus(q).degree(): 5
z: 0
z.degree(): -1
p.plus(z): 3.00 - 8.00x + 2.00x^4 + 1.00x^5
z.plus(p): 3.00 - 8.00x + 2.00x^4 + 1.00x^5
p: 3.00 - 8.00x + 2.00x^4 + 1.00x^5
t: 8.88x^44
t.degree(): 44

```

Final Project :

- **Presentasi sesuai jadwal**
- **Softcopy berupa**
- **Laporan resmi**
 - **Program, penjelasan program dan output program**
 - **file program**