

Nama : Chandra Harkat Raharja
NPM : 23304089
Kelas : B
Dosen : ADE SUKENDAR, ST.,MT,
-SANDRA ISLAMA PUTRA., S.Si., M.Kom
Link Repository : [ComradeChandra/PP12025_B_233040089](https://github.com/ComradeChandra/PP12025_B_233040089)

-Latihan-1

Latihan-1

Latihan ini akan memberikan implementasi pembuatan elemen list bertipe bilangan bulat (*integer*). Elemen list direpresentasikan dengan Node. Sebuah node terdiri dari atribut nilai dan atribut *next*. Atribut *next* akan menunjuk ke node yang lain.

Kode Program Kelas Node

```
public class Node {  
    private int nilai;  
    private Node next;  
  
    // inisialisasi Node  
    public Node(int nilai) {  
        this.nilai = nilai;  
    }  
  
    // Setter & Getter  
}
```

- a. Contoh **setter**: `public void setNama(String nama) { this.nama = nama; }`
b. Contoh **getter**: `public String getNama() { return nama; }`

```
TugasDanPertemuanCoding > TugasPertemuan2 > node.java > node  
1 public class node {  
2     private int nilai;  
3     private node next;  
4  
5  
6     //inisialisasi node  
7     public node(int nilai) {  
8         this.nilai = nilai;  
9     }  
10  
11     //setter & getter  
12  
13     public int getNilai() {  
14         return nilai;  
15     }  
16  
17  
18     public void setNilai(int nilai) {  
19         this.nilai = nilai;  
20     }  
21  
22  
23     public node getNext() {  
24         return next;  
25     }  
26  
27  
28     public void setNext(node next) {  
29         this.next = next;  
30     }  
31  
32  
33 }
```

Jawaban dan Penjelasan:

Deklarasi Kelas dan Atribut:

Kode dimulai dengan "public class node {" yang berarti kita membuat kelas bernama node dengan akses publik, jadi bisa diakses dari kelas lain. Ada dua variabel privat:

- "private int nilai;" menyimpan data berupa angka.
- "private node next;" menyimpan referensi ke node selanjutnya, yang memungkinkan kita membuat linked list.

Konstruktor:

Baris "public node(int nilai)" adalah konstruktor, yakni method yang dipanggil saat membuat objek baru.

Di dalam konstruktor, "this.nilai = nilai;" menetapkan nilai yang diberikan saat objek dibuat ke variabel "nilai". Perlu diketahui, variabel "next" tidak diinisialisasi di sini, jadi secara default nilainya null (belum terhubung ke node lain).

Getter dan Setter untuk Atribut "nilai"

- "public int getNilai()" adalah method yang mengembalikan nilai dari variabel "nilai".
- "public void setNilai(int nilai)" memungkinkan kita untuk mengubah nilai yang disimpan dengan nilai baru. Di sini, "this.nilai = nilai;" berfungsi untuk mengganti nilai yang lama dengan nilai baru.

Getter dan Setter untuk Atribut "next"

- "public node getNext()" berfungsi untuk mengambil referensi ke node berikutnya.
- "public void setNext(node next)" berfungsi untuk menetapkan node berikutnya dengan memasukkan objek node yang diinginkan.

Latihan-2

Latihan-2

Latihan ini akan memberikan contoh pemanggilan Node dan menambahkan Node berikutnya ke atribut next. Fungsi untuk menampilkan nilai

Kode Program Kelas NodeMain:

```
public class NodeMain {  
    public static void main(String[] args) {  
  
        // Membuat 2 buah node n1 & n2  
        Node n1 = new Node(5);  
        Node n2 = new Node(7);  
  
        // Membuat relasi Node n1 & n2  
        n1.setNext(n2);  
  
        // Menampilkan Node n1 & n2 dengan Pointer p  
        Node p = n1;  
        while(p != null)  
        {  
            System.out.printf("%d ", p.getNilai());  
            p = p.getNext();  
        }  
    }  
}
```

```
public class nodemain {  
    Run | Debug  
    public static void main(String[] args) {  
        //membuat 2 buatnode n1 & n2  
        node n1 = new node(nilai:5);  
        node n2 = new node(nilai:7);  
  
        //membuat relasinode n1 & n2  
        n1.setNext(n2);  
  
        //menampilkannode n1 & n2 dengan pointer  
        node p = n1;  
        while (p != null) {  
            System.out.printf(" %d ", p.getNilai());  
            p = p.getNext();  
        }  
    }  
}
```

Penjelasan:

A. Deklarasi Kelas dan Method Utama

Kelas "nodemain" dideklarasikan dengan method main sebagai titik awal eksekusi program. Di sini, program dijalankan dari method main.

B. Pembuatan Objek Node

Di dalam method main, dibuat dua objek node:

- Objek "n1" dengan nilai 5.
- Objek "n2" dengan nilai 7. Keduanya merupakan instance dari kelas node yang sebelumnya telah didefinisikan.

C. Penghubungan Node

Dengan memanggil "n1.setNext(n2)", node "n1" dihubungkan ke "n2". Artinya, setelah "n1" diikuti oleh "n2" dalam struktur linked list.

D. Menampilkan Data Node

Variabel pointer "p" diinisialisasi dengan "n1" sebagai titik awal. Kemudian, dilakukan perulangan while selama "p" tidak null:

- Setiap iterasi, nilai yang tersimpan pada node yang ditunjuk oleh "p" ditampilkan menggunakan format printf.
- Pointer "p" kemudian dipindahkan ke node berikutnya dengan "p = p.getNext()".
Proses ini akan menampilkan nilai 5 (dari n1) kemudian 7 (dari n2) secara berurutan.

Tes-1

Apa fungsi atribut next pada kelas Node?

Jawab: fungsi atribut next kelas node adalah untuk menunjuk ke node berikutnya dalam struktur data linked list

-Tes-2 Perintah apa yang digunakan untuk menambahkan relasi

antar Node?

-Jawab: Perintah yang digunakan untuk menambahkan relasi antar node adalah `setNext()`.

-Tes-3

Apa fungsi atribut pointer p yang terdapat di Latihan-2?

Jawab: Atribut pointer `p` dalam file nodemain.java digunakan untuk menelusuri dan menampilkan nilai dari setiap node yang terhubung mulai dari node `n1`.

-Tes-4

Ubahlah Latihan-2 yang telah anda kerjakan sehingga menjadi urutan Node seperti dibawah ini a.

5, 7, 9, 8

b. 2, 3, 5, 7, 9

Jawab:

A.

```
public class Tes4 {  
    Run | Debug  
    public static void main(String[] args) {  
  
        node n1 = new node(nilai:5);  
        node n2 = new node(nilai:7);  
        node n3 = new node(nilai:9);  
        node n4 = new node(nilai:8);  
  
        n1.setNext(n2);  
        n2.setNext(n3);  
        n3.setNext(n4);  
  
        node p = n1;  
        while (p != null) {  
            System.out.printf(" %d ", p.getNilai());  
            p = p.getNext();  
        }  
    }  
}
```

Code ini berfungsi membuat node baru, dengan urutan 5,7,9,8, sementara untuk baris code selanjutnya(n1.setNext (n2);)) berfungsi untuk membuat relasi untuk node.

Baris code ini berfungsi untuk menampilkan node dengan pengkondisian 'while' Dimana jika 'p' belum bernilai null, akan menampilkan nilai node dan mengambil node selanjutnya yang berelasi

B.

```
System.out.println();  
  
        node n5 = new node(nilai:2);  
        node n6 = new node(nilai:3);  
        node n7 = new node(nilai:5);  
        node n8 = new node(nilai:7);  
        node n9 = new node(nilai:9);  
        n5.setNext(n6);  
        n6.setNext(n7);  
        n7.setNext(n8);  
        n8.setNext(n9);  
  
        p = n5;  
        while (p != null) {  
            System.out.printf(" %d ", p.getNilai());  
            p = p.getNext();  
        }  
        System.out.println();  
    }  
}
```

Code ini berfungsi pindah ke line/baris baru dan membuat node baru, dengan urutan 2,3,5,7,9 sementara untuk baris code selanjutnya (n5.setNext (n6);)) berfungsi untuk membuat relasi untuk node.

Sama seperti di soal/Tes-4A, Baris code ini berfungsi untuk menampilkan node dengan pengkondisian 'while' Dimana jika 'p' belum bernilai null, akan menampilkan nilai node dan mengambil node selanjutnya yang berelasi.

Untuk outputnya akan terlihat seperti ini:

```
S D:\KuliahSMT4\Praktikum Pemrograman 1> d:;  
\\ACER\AppData\Roaming\Code\User\workspaceStor  
5 7 9 8  
2 3 5 7 9  
S D:\KuliahSMT4\Praktikum Pemrograman 1>
```