

LAPORAN PRAKTIKUM
STRUKTUR DATA
PERTEMUAN KE – 7



Disusun Oleh :

NAMA : TARISA DWI SEPTIA
NIM : 205410126
JURUSAN : TEKNIK INFORMATIKA
JENJANG : S1

Sekolah Tinggi Management Informatika Komputer

AKAKOM

YOGYAKARTA

2020

MODUL 7

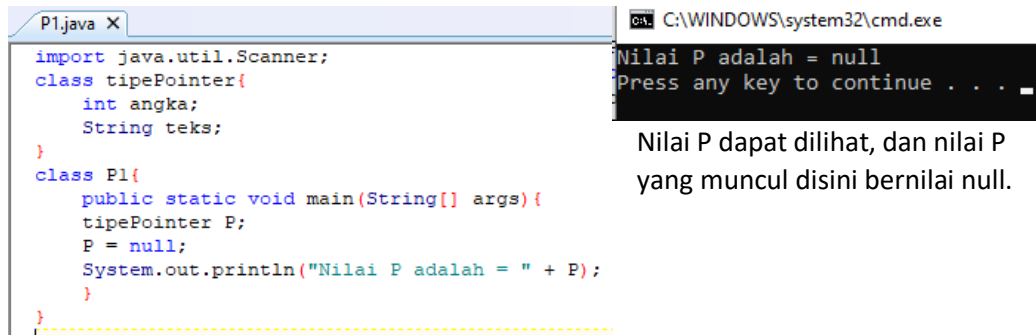
POINTER DALAM JAVA

A. Tujuan

- Mahasiswa dapat mengeksploitasi variable pointer dalam java

B. Praktik

a. Praktik 1

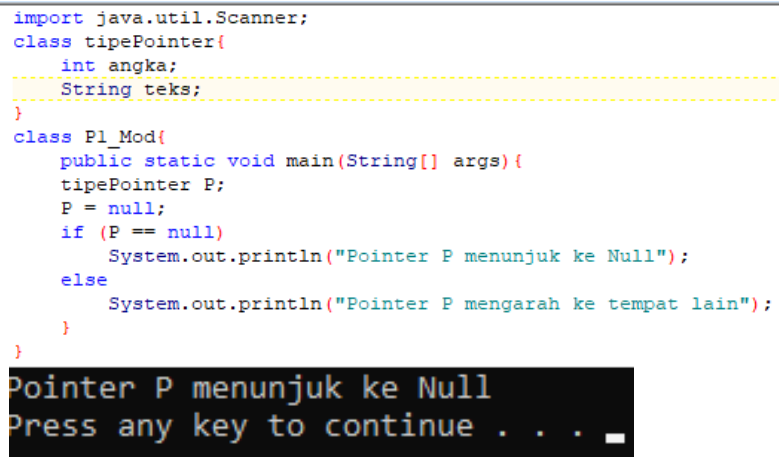


The screenshot shows a Java IDE with a file named P1.java. The code defines a class `tiPePointer` with attributes `int angka;` and `String teks;`. A `main` method in class `P1` creates a `tiPePointer` object `P`, sets `P = null;`, and prints `"Nilai P adalah = " + P;`. The output window shows the command prompt with the text `Nilai P adalah = null` and `Press any key to continue . . .`.

```
import java.util.Scanner;
class tiPePointer{
    int angka;
    String teks;
}
class P1{
    public static void main(String[] args){
        tiPePointer P;
        P = null;
        System.out.println("Nilai P adalah = " + P);
    }
}
```

Nilai P dapat dilihat, dan nilai P yang muncul disini bernilai null.

Setelah dimodifikasi :



The screenshot shows the modified Java code. The class `tiPePointer` is unchanged. The `main` method in class `P1_Mod` creates a `tiPePointer` object `P`, sets `P = null;`, and uses an `if` statement to check `P == null`. If true, it prints `"Pointer P menunjuk ke Null"`; otherwise, it prints `"Pointer P mengarah ke tempat lain"`. The output window shows `Pointer P menunjuk ke Null` and `Press any key to continue . . .`.

```
import java.util.Scanner;
class tiPePointer{
    int angka;
    String teks;
}
class P1_Mod{
    public static void main(String[] args){
        tiPePointer P;
        P = null;
        if (P == null)
            System.out.println("Pointer P menunjuk ke Null");
        else
            System.out.println("Pointer P mengarah ke tempat lain");
    }
}
```

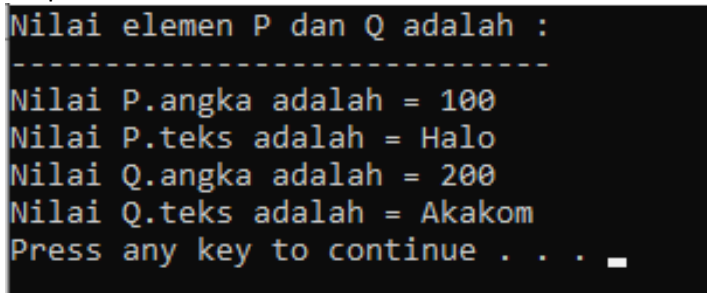
Pointer P menunjuk ke Null
Press any key to continue . . .

Pionter P tetap menunjuk ke null dikarenakan pointer sendiri bernilai null kemudian di seleksi. Jika P bernilai null maka output yang keluar adalah "Pointer P menunjuk ne null"

b. Praktik 2

```
import java.util.Scanner;
class tipePointer{
    int angka;
    String teks;
}
class P2{
    public static void main(String[] args){
        tipePointer P;
        P = new tipePointer();
        P.angka = 100;
        P.teks = "Halo";
        tipePointer Q = new tipePointer();
        Q.angka = 200;
        Q.teks = "Akakom";
        System.out.println("Nilai elemen P dan Q adalah :");
        System.out.println("-----");
        System.out.println("Nilai P.angka adalah = " + P.angka);
        System.out.println("Nilai P.teks adalah = " + P.teks);
        System.out.println("Nilai Q.angka adalah = " + Q.angka);
        System.out.println("Nilai Q.teks adalah = " + Q.teks);
    }
}
```

Output :



```
Nilai elemen P dan Q adalah :
-----
Nilai P.angka adalah = 100
Nilai P.teks adalah = Halo
Nilai Q.angka adalah = 200
Nilai Q.teks adalah = Akakom
Press any key to continue . . .
```

Untuk P.angka bernilai 100 dan P.teks bernilai Halo, sedangkan untuk Q.angka bernilai 200 dan Q.teks bernilai Akakom

- Setelah dimodifikasi :

```
import java.util.Scanner;
class tipePointer{
    int angka;
    String teks;
}
class P2_Mod{
    public static void main(String[] args){
        tipePointer P;
        P = new tipePointer();
        P.angka = 100;
        P.teks = "Halo";
        tipePointer Q = new tipePointer();
        Q.angka = 200;
        Q.teks = "Akakom";
        System.out.println("Nilai elemen P dan Q adalah :");
        System.out.println("-----");
        System.out.println("Nilai P.angka adalah = " + P.angka);
        System.out.println("Nilai P.teks adalah = " + P.teks);
        System.out.println("Nilai Q.angka adalah = " + Q.angka);
        System.out.println("Nilai Q.teks adalah = " + Q.teks);
    }
}
```

```

        tipePointer R;
        R = P;
        System.out.println("Nilai elemen R adalah :");
        System.out.println("-----");
        System.out.println("Nilai R.angka adalah = " + R.angka);
        System.out.println("Nilai R.teks adalah = " + R.teks);
        R = Q;
        System.out.println("Nilai elemen R saat ini adalah :");
        System.out.println("-----");
        System.out.println("Nilai R.angka adalah = " + R.angka);
        System.out.println("Nilai R.teks adalah = " + R.teks);
    }
}

```

Output :

```

Nilai elemen P dan Q adalah :
-----
Nilai P.angka adalah = 100
Nilai P.teks adalah = Halo
Nilai Q.angka adalah = 200
Nilai Q.teks adalah = Akakom
Nilai elemen R adalah :
-----
Nilai R.angka adalah = 100
Nilai R.teks adalah = Halo
Nilai elemen R saat ini adalah :
-----
Nilai R.angka adalah = 200
Nilai R.teks adalah = Akakom
Press any key to continue . . .

```

Dikarenakan tipe pointer R tidak membuat pointer baru dan hanya mengandalkan tipe pointer dari P dan Q, sehingga jika di deklarasikan R = P maka nilai yang ada di tipe pointer R akan sama dengan nilai yang ada di tipe pointer P. Begitu juga dengan tipe pointer Q.

- Setelah dimodifikasi lagi :

```

import java.util.Scanner;
class tipePointer{
    int angka;
    String teks;
}
class P2_Mod1{
    public static void main(String[] args){
        tipePointer P = new tipePointer();
        P.angka = 100;
        P.teks = "Halo";
        tipePointer Q = new tipePointer();
        Q.angka = 200;
        Q.teks = "Akakom";
        System.out.println("Nilai P dan Q sebelum pointer dimanipulasi :");
        System.out.println("-----");
        System.out.println("Nilai P.angka adalah = " + P.angka);
        System.out.println("Nilai P.teks adalah = " + P.teks);
        System.out.println("Nilai Q.angka adalah = " + Q.angka);
        System.out.println("Nilai Q.teks adalah = " + Q.teks);
        tipePointer R;
        R = P;
    }
}

```

```

        P = Q;
        Q = R;
        System.out.println("Nilai P dan Q setelah pointer dimanipulasi :");
        System.out.println("-----");
        System.out.println("Nilai P.angka adalah = " + P.angka);
        System.out.println("Nilai P.teks adalah = " + P.teks);
        System.out.println("Nilai Q.angka adalah = " + Q.angka);
        System.out.println("Nilai Q.teks adalah = " + Q.teks);
    }
}

```

Output :

```

Nilai P dan Q sebelum pointer dimanipulasi :
-----
Nilai P.angka adalah = 100
Nilai P.teks adalah = Halo
Nilai Q.angka adalah = 200
Nilai Q.teks adalah = Akakom
Nilai P dan Q setelah pointer dimanipulasi :
-----
Nilai P.angka adalah = 200
Nilai P.teks adalah = Akakom
Nilai Q.angka adalah = 100
Nilai Q.teks adalah = Halo
Press any key to continue . . .

```

Setelah dimanipulasi nilai di pointer P.angka menjadi 200 dan di P.teks menjadi Akakom begitu pula dengan nilai Q.angka menjadi 100 dan Q.teks menjadi Halo. Pointer tersebut bisa berubah nilai dikarenakan sudah dimanipulasi dengan deklarasi :

```

        R = P;
        P = Q;
        Q = R;

```

c. Praktik 3

- Tuliskan program berikut ini.

```

import java.util.Scanner;
class tipePointer{
    String namaKota;
    tipePointer kanan;
}
class P3{
    public static void main(String[] args){
        tipePointer P;
        P = new tipePointer();
        P.namaKota = "Yogyakarta";
        tipePointer Q;
        Q = new tipePointer();
        Q.namaKota = "Klaten";
        tipePointer R;
        R = new tipePointer();
        R.namaKota = "Solo";
        tipePointer S;
        S = new tipePointer();
        S.namaKota = "Sragen";
        tipePointer T;
        T = new tipePointer();
    }
}

```

```

T.namaKota = "Ngawi";
System.out.println("Nilai P,Q,R,S,T adalah :");
System.out.println("-----");
System.out.println("Nilai P.namaKota adalah = " + P.namaKota);
System.out.println("Nilai Q.namaKota adalah = " + Q.namaKota);
System.out.println("Nilai R.namaKota adalah = " + R.namaKota);
System.out.println("Nilai S.namaKota adalah = " + S.namaKota);
System.out.println("Nilai T.namaKota adalah = " + T.namaKota);
P.kanan = Q;
Q.kanan = R;
R.kanan = S;
S.kanan = T;
T.kanan = null;
System.out.println("Nilai-nilai yang dapat diakses dari pointer P adalah :");
System.out.println("-----");
System.out.println(P.namaKota);
System.out.println(P.kanan.namaKota);
System.out.println(P.kanan.kanan.namaKota);
System.out.println(P.kanan.kanan.kanan.namaKota);
System.out.println(P.kanan.kanan.kanan.kanan.namaKota);
}
}

```

Output :

```

Nilai P,Q,R,S,T adalah :
-----
Nilai P.namaKota adalah = Yogyakarta
Nilai Q.namaKota adalah = Klaten
Nilai R.namaKota adalah = Solo
Nilai S.namaKota adalah = Sragen
Nilai T.namaKota adalah = Ngawi
Nilai-nilai yang dapat diakses dari pointer P adalah :
-----
Yogyakarta
Klaten
Solo
Sragen
Ngawi
Press any key to continue . . .

```

Nilai dari masing masing nama kota sesuai dengan output yang ada, sedangkan output yang dihasilkan dari deret perintah adalah nama kota yaitu Yogyakarta, Klaten, Solo, Sragen dan Ngawi.

- Sekarang tambahkan perintah berikut ini di bagian terakhir dari program anda.

```

import java.util.Scanner;
class tipePointer{
    String namaKota;
    tipePointer kanan;
}
class P3{
    public static void main(String[] args){
        tipePointer P;
        P = new tipePointer();
        P.namaKota = "Yogyakarta";
        tipePointer Q;
    }
}

```

```

Q = new tipePointer();
Q.namaKota = "Klaten";
tipePointer R;
R = new tipePointer();
R.namaKota = "Solo";
tipePointer S;
S = new tipePointer();
S.namaKota = "Sragen";
tipePointer T;
T = new tipePointer();
T.namaKota = "Ngawi";
System.out.println("Nilai P,Q,R,S,T adalah :");
System.out.println("-----");
System.out.println("Nilai P.namaKota adalah = " + P.namaKota);
System.out.println("Nilai Q.namaKota adalah = " + Q.namaKota);
System.out.println("Nilai R.namaKota adalah = " + R.namaKota);
System.out.println("Nilai S.namaKota adalah = " + S.namaKota);
System.out.println("Nilai T.namaKota adalah = " + T.namaKota);
P.kanan = Q;
Q.kanan = R;
R.kanan = S;
S.kanan = T;
T.kanan = null;
System.out.println("Nilai-nilai yang dapat diakses dari pointer P adalah :");
System.out.println("-----");
System.out.println(P.namaKota);
System.out.println(P.kanan.namaKota);
System.out.println(P.kanan.kanan.namaKota);
System.out.println(P.kanan.kanan.kanan.namaKota);
System.out.println(P.kanan.kanan.kanan.kanan.namaKota);
System.out.println(R.namaKota);
P.kanan.kanan.namaKota = "Surakarta";
System.out.println(R.namaKota);

```

```

}
}

```

Output :

```
Nilai P,Q,R,S,T adalah :
-----
Nilai P.namaKota adalah = Yogyakarta
Nilai Q.namaKota adalah = Klaten
Nilai R.namaKota adalah = Solo
Nilai S.namaKota adalah = Sragen
Nilai T.namaKota adalah = Ngawi
Nilai-nilai yang dapat diakses dari pointer P adalah :
-----
Yogyakarta
Klaten
Solo
Sragen
Ngawi
Solo
Surakarta
Press any key to continue . . . _
```

Setelah ditambahkan perintah dibagian terakhir program, nilai R.namaKota berubah menjadi “Surakarta”.

- Modifikasi

```
import java.util.Scanner;
class tipePointer{
    String namaKota;
    tipePointer kanan;
}
class P3_Mod{
    public static void main(String[] args){
        tipePointer P;
        P = new tipePointer();
        P.namaKota = "Yogyakarta";
        tipePointer Q;
        Q = new tipePointer();
        Q.namaKota = "Klaten";
        tipePointer R;
        R = new tipePointer();
        R.namaKota = "Solo";
        tipePointer S;
        S = new tipePointer();
        S.namaKota = "Sragen";
        tipePointer T;
        T = new tipePointer();
        T.namaKota = "Ngawi";
        System.out.println("Elemen namaKota untuk pointer P,Q,R,S,T adalah
:");

        System.out.println("-----");
        System.out.println("Nilai P.namaKota adalah = " + P.namaKota);
        System.out.println("Nilai Q.namaKota adalah = " + Q.namaKota);
        System.out.println("Nilai R.namaKota adalah = " + R.namaKota);
        System.out.println("Nilai S.namaKota adalah = " + S.namaKota);
        System.out.println("Nilai T.namaKota adalah = " + T.namaKota);
        P.kanan = Q;
        Q.kanan = R;
        R.kanan = S;
        S.kanan = T;
```



```

        T.kanan = null;
        System.out.println("Elemen namaKota untuk pointer P,Q,R,S,T adalah
        :");

        System.out.println("-----");
        System.out.println("Nilai P.namaKota adalah = " + P.namaKota);
        System.out.println("Nilai Q.namaKota adalah = " +
        P.kanan.namaKota);
        System.out.println("Nilai R.namaKota adalah = " +
        P.kanan.kanan.namaKota);
        System.out.println("Nilai S.namaKota adalah = " +
        P.kanan.kanan.kanan.namaKota);
        System.out.print("Nilai T.namaKota adalah = " +
        P.kanan.kanan.kanan.kanan.namaKota);
        tipePointer BANTU;
        BANTU = P;
        while (BANTU!=null){
            System.out.println("Nilai BANTU.namaKota adalah = " +
            BANTU.namaKota);

            BANTU = BANTU.kanan;
        }
    }
}

```

Output :

```

Elemen namaKota untuk pointer P,Q,R,S,T adalah :
-----
Nilai P.namaKota adalah = Yogyakarta
Nilai Q.namaKota adalah = Klaten
Nilai R.namaKota adalah = Solo
Nilai S.namaKota adalah = Sragen
Nilai T.namaKota adalah = Ngawi
Elemen namaKota untuk pointer P,Q,R,S,T adalah :
-----
Nilai P.namaKota adalah = Yogyakarta
Nilai Q.namaKota adalah = Klaten
Nilai R.namaKota adalah = Solo
Nilai S.namaKota adalah = Sragen
Nilai T.namaKota adalah= Ngawi
Nilai BANTU.namaKota adalah = Yogyakarta
Nilai BANTU.namaKota adalah = Klaten
Nilai BANTU.namaKota adalah = Solo
Nilai BANTU.namaKota adalah = Sragen
Nilai BANTU.namaKota adalah = Ngawi
Press any key to continue . . .

```

instruksi `System.out.println("Nilai BANTU.namaKota adalah = " + BANTU.namaKota);` di eksekusi oleh sistem sebanyak 5 kali, hasilnya tidak selalu sama. Tergantung oleh pointer mana yang ditunjuk.

- Modifikasi 1

```

import java.util.Scanner;
class tipePointer{
    String namaKota;
    tipePointer kiri;
    tipePointer kanan;
}
class P3_Mod1{
    public static void main(String[] args){

```

```

tipePointer P;
P = new tipePointer();
P.namaKota = "Yogyakarta";
tipePointer Q;
Q = new tipePointer();
Q.namaKota = "Klaten";
tipePointer R;
R = new tipePointer();
R.namaKota = "Solo";
tipePointer S;
S = new tipePointer();
S.namaKota = "Sragen";
tipePointer T;
T = new tipePointer();
T.namaKota = "Ngawi";
System.out.println("Elemen namaKota untuk pointer P,Q,R,S,T adalah
:");

System.out.println("-----");
System.out.println("Nilai P.namaKota adalah " + P.namaKota);
System.out.println("Nilai Q.namaKota adalah " + Q.namaKota);
System.out.println("Nilai R.namaKota adalah " + R.namaKota);
System.out.println("Nilai S.namaKota adalah " + S.namaKota);
System.out.println("Nilai T.namaKota adalah " + T.namaKota);
P.kanan = Q;
Q.kanan = R;
R.kanan = S;
S.kanan = T;
T.kanan = null;
P.kiri = null;
Q.kiri = P;
R.kiri = R;
S.kiri = R;
T.kiri = S;
System.out.println("Elemen namaKota untuk pointer P adalah :");
System.out.println("-----");
System.out.println(P.namaKota);
System.out.println(P.kanan.kiri.namaKota);
System.out.println(P.kanan.kanan.kiri.kiri.namaKota);
tipePointer BANTU;
BANTU = T;
while (BANTU!=null){
    System.out.println("Nilai BANTU.namaKota adalah " +
BANTU.namaKota);
    BANTU = BANTU.kiri;
}
}

```

Output :

[illegible]

Pada program yang ini, instruksi `System.out.println("Nilai BANTU.namaKota adalah " + BANTU.namaKota);` melakukan looping terus menerus dikarenakan pointer BANTU memanipulasi T. Sedangkan T memanipulasi S, sehingga pinter BANTU tidak berhenti melooping.

C. Kesimpulan

Setelah melakukan praktik diatas sesuai dengan tujuan dari modul praktikum 6 ini, dapat disimpulkan bahwa mahasiswa dapat mengelola variable pointer didalam java.