

LAPORAN PRAKTIKUM STRUKTUR DATA

PERTEMUAN KE-12



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya

NIM : 195410257

JURUSAN : Teknik Informatika

JENJANG : S1

Laboratorium Terpadu

Sekolah Tinggi Management Informatika Komputer

AKAKOM

YOGYAKARTA

2020

PERTEMUAN KE-12 (COLLECTION)

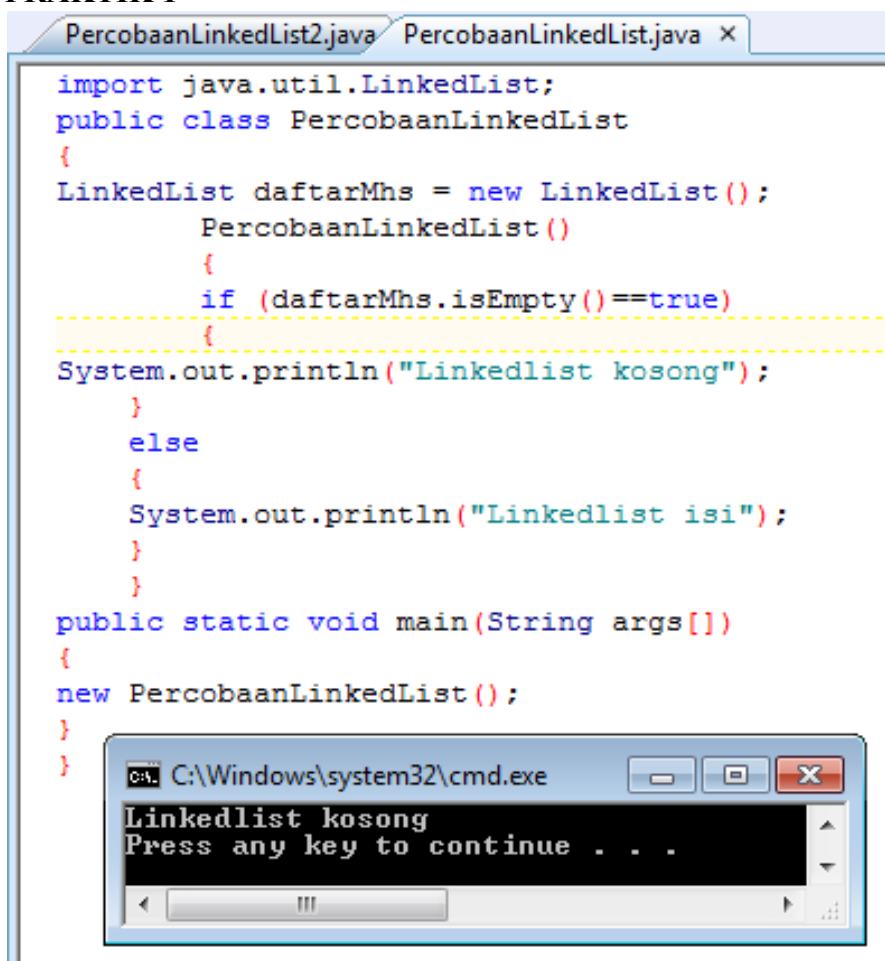
TUJUAN

Mahasiswa dapat mengimplementasikan penggunaan kelas collection

DASAR TEORI

Collections secara umum memiliki makna adalah sebuah kumpulan. Dalam OOP collection dikenal sebagai suatu tempat atau wadah atau object yang dapat menyimpan object lainnya baik yang memiliki tipe data yang sama maupun tidak. Adapun beberapa contoh implementasi dari collection adalah Vector, ArrayList dan List.

PRAKTIK 1



The screenshot shows a Java IDE with two tabs: 'PercobaanLinkedList2.java' and 'PercobaanLinkedList.java'. The active tab is 'PercobaanLinkedList.java', which contains the following code:

```
import java.util.LinkedList;
public class PercobaanLinkedList
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList()
    {
        if (daftarMhs.isEmpty() == true)
        {
            System.out.println("Linkedlist kosong");
        }
        else
        {
            System.out.println("Linkedlist isi");
        }
    }
}

public static void main(String args[])
{
    new PercobaanLinkedList();
}
```

Below the code editor, a command prompt window is open, showing the output of the program:

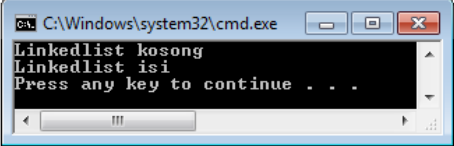
```
C:\Windows\system32\cmd.exe
Linkedlist kosong
Press any key to continue . . .
```

penjelasan:

isEmpty(), berguna untuk mengecek variable String. isEmpty mengecek apakah variable kosong atau tidak. Jika kosong, maka nilai akan muncul sebagai true, dan jika variable String mempunyai nilai, maka akan muncul sebagai false. Pada program diatas muncul output "Linkedlist kosong" karena method isEmpty() menghasilkan nilai true.

PRAKTIK 2

```
PercobaanLinkedList2.java x PercobaanLinkedList.java
import java.util.LinkedList;
public class PercobaanLinkedList2
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList2 ()
    {
        if (daftarMhs.isEmpty()==true)
        {
            System.out.println("Linkedlist kosong");
        }
        else
        {
            System.out.println("Linkedlist isi");
        }
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        if (daftarMhs.isEmpty()==true)
        {
            System.out.println("Linkedlist kosong");
        }
        else
        {
            System.out.println("Linkedlist isi");
        }
        public static void main(String args[])
        {
            new PercobaanLinkedList2 ();
        }
    }
}
```

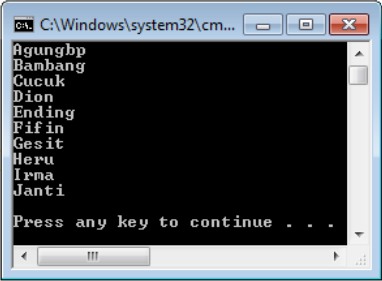


penjelasan:

add("Agungbp") berfungsi untuk menambahkan objek string ke linkedlist daftarMhs. Program mengeluarkan output seperti itu karena pada seleksi pertama linkedlist tidak bernilai atau kosong sehingga menampilkan output "Linkedlist kosong". Pada seleksi kedua, keadaan linked list sudah terisi dengan "Agungbp" dan "Bambang" sehingga program menampilkan output "Linkedlist isi"

PRAKTIK 3

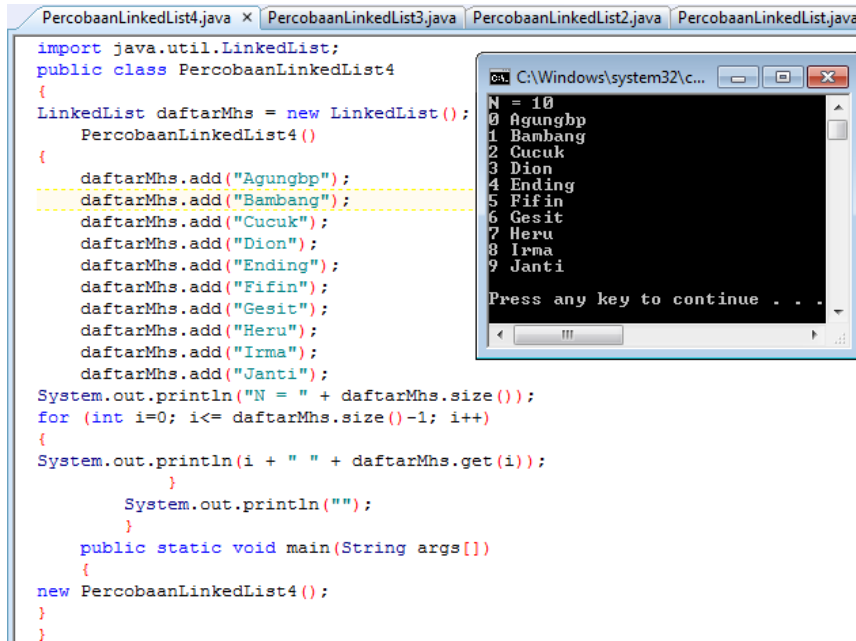
```
PercobaanLinkedList3.java x PercobaanLinkedList2.java x PercobaanLinkedList.java
import java.util.LinkedList;
import java.util.LinkedList;
public class PercobaanLinkedList3
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList3 ()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifin");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
        System.out.println(daftarMhs.get(0));
        System.out.println(daftarMhs.get(1));
        System.out.println(daftarMhs.get(2));
        System.out.println(daftarMhs.get(3));
        System.out.println(daftarMhs.get(4));
        System.out.println(daftarMhs.get(5));
        System.out.println(daftarMhs.get(6));
        System.out.println(daftarMhs.get(7));
        System.out.println(daftarMhs.get(8));
        System.out.println(daftarMhs.get(9));
        System.out.println("");
    }
    public static void main(String args[])
    {
        new PercobaanLinkedList3 ();
    }
}
```



penjelasan:

makna dari `daftarMhs.get(0)`; adalah memanggil data ke-0 (data pertama) di linkedlist. Hasil output menampilkan semua data yang dipanggil dari data 0 sampai data 9 yaitu Agungbp, Bambang, Cucuk, Dion, Ending, Fifi, Gesit, Heru, Irma, Janti.

PRAKTIK 4



```
import java.util.LinkedList;
public class PercobaanLinkedList4
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList4()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifi");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
    }
    System.out.println("N = " + daftarMhs.size());
    for (int i=0; i<= daftarMhs.size()-1; i++)
    {
        System.out.println(i + " " + daftarMhs.get(i));
    }
    System.out.println("");
    public static void main(String args[])
    {
        new PercobaanLinkedList4();
    }
}
```

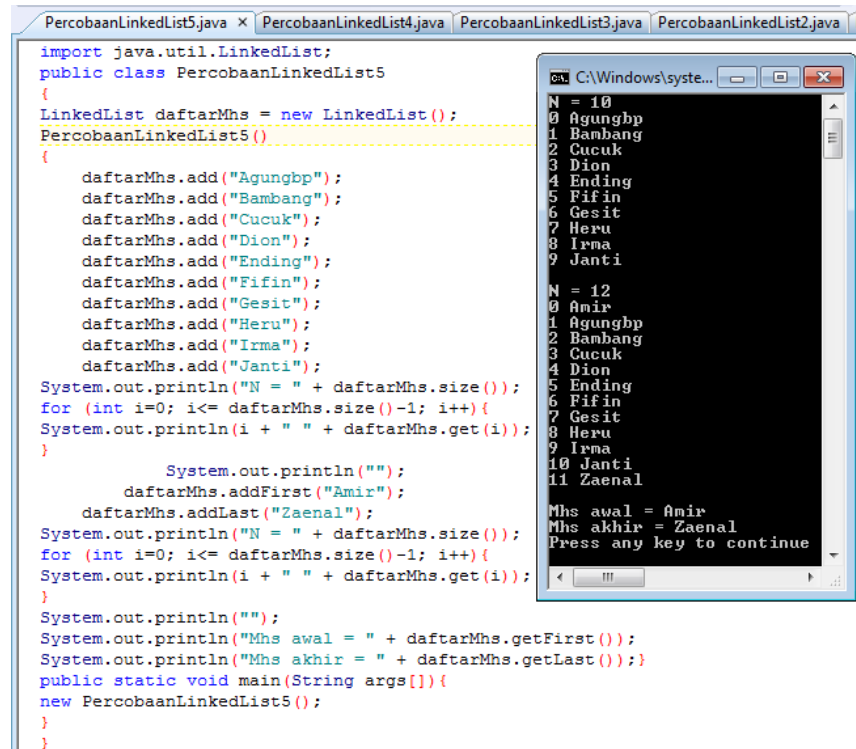
Output:

```
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifi
6 Gesit
7 Heru
8 Irma
9 Janti
Press any key to continue . . .
```

penjelasan:

`daftarMhs.size()` berfungsi untuk menghitung banyak jumlah data yang dimasukkan pada linkedlist `daftarMhs`. Pada output program diatas, `N=10` menunjukkan banyak data berjumlah 10 yang dimulai dari data ke-0 sampai data ke-9.

PRAKTIK 5



```
import java.util.LinkedList;
public class PercobaanLinkedList5
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList5()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifi");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
    }
    System.out.println("N = " + daftarMhs.size());
    for (int i=0; i<= daftarMhs.size()-1; i++){
        System.out.println(i + " " + daftarMhs.get(i));
    }
    System.out.println("");
    daftarMhs.addFirst("Amir");
    daftarMhs.addLast("Zaenal");
    System.out.println("N = " + daftarMhs.size());
    for (int i=0; i<= daftarMhs.size()-1; i++){
        System.out.println(i + " " + daftarMhs.get(i));
    }
    System.out.println("");
    System.out.println("Mhs awal = " + daftarMhs.getFirst());
    System.out.println("Mhs akhir = " + daftarMhs.getLast());
    public static void main(String args[]){
        new PercobaanLinkedList5();
    }
}
```

Output:

```
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifi
6 Gesit
7 Heru
8 Irma
9 Janti

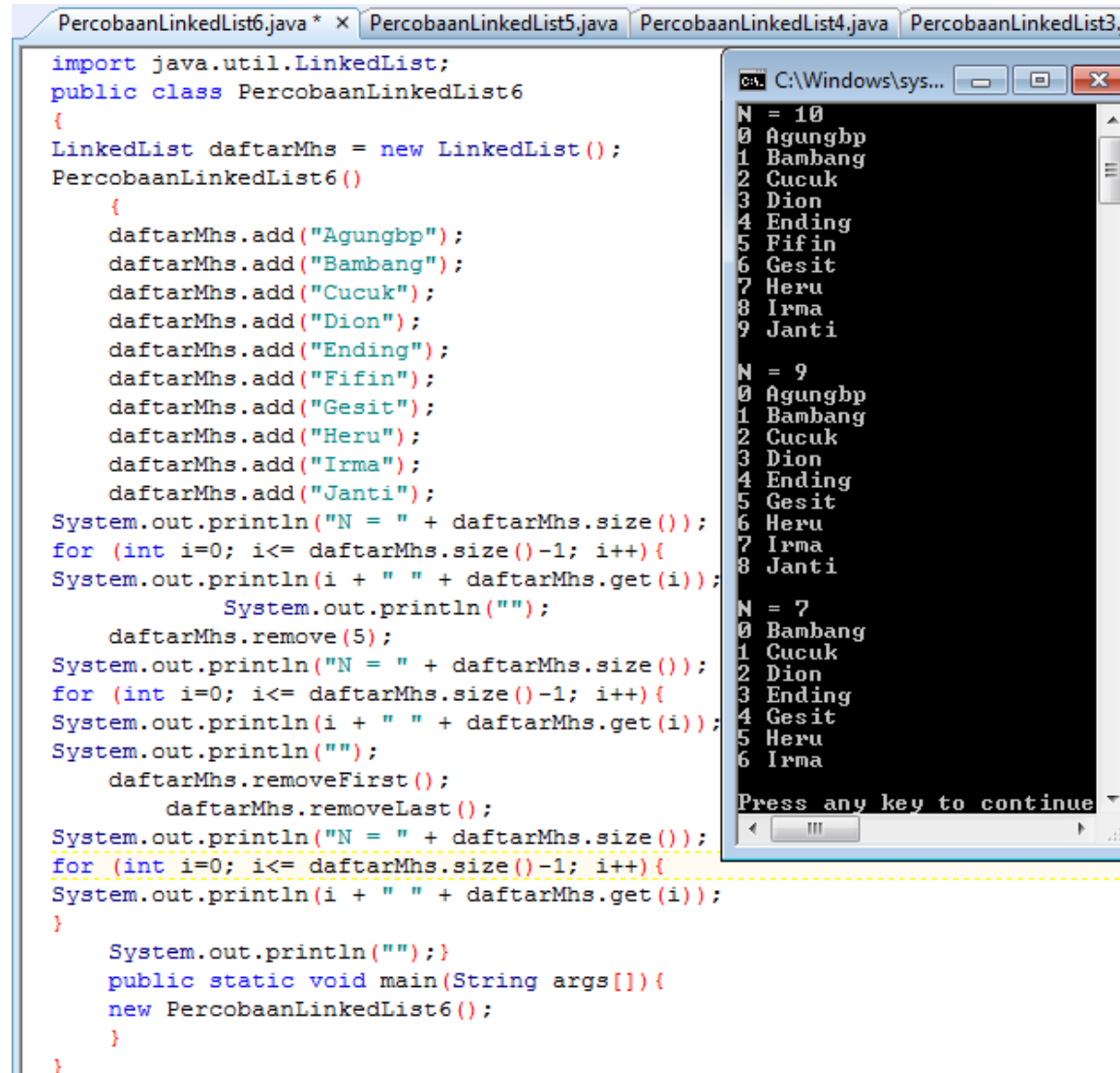
N = 12
0 Amir
1 Agungbp
2 Bambang
3 Cucuk
4 Dion
5 Ending
6 Fifi
7 Gesit
8 Heru
9 Irma
10 Janti
11 Zaenal

Mhs awal = Amir
Mhs akhir = Zaenal
Press any key to continue
```

penjelasan:

addFirst("Amir") menambahkan objek "Amir" menjadi data pertama di linkedlist daftarMhs.
addLast("Zaenal") menambahkan objek "Zaenal" menjadi data terakhir di linkedlist daftarMhs.
getFirst() memanggil data pertama pada linkedlist daftarMhs.
getLast() memanggil data terakhir pada linkedlist daftarMhs.

PRAKTIK 6



```
import java.util.LinkedList;
public class PercobaanLinkedList6
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList6()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifin");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++){
            System.out.println(i + " " + daftarMhs.get(i));
            System.out.println("");
            daftarMhs.remove(5);
        }
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++){
            System.out.println(i + " " + daftarMhs.get(i));
            System.out.println("");
            daftarMhs.removeFirst();
        }
        daftarMhs.removeLast();
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++){
            System.out.println(i + " " + daftarMhs.get(i));
        }
        System.out.println("");
        public static void main(String args[]){
            new PercobaanLinkedList6();
        }
    }
}
```

Output Console:

```
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifin
6 Gesit
7 Heru
8 Irma
9 Janti

N = 9
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Gesit
6 Heru
7 Irma
8 Janti

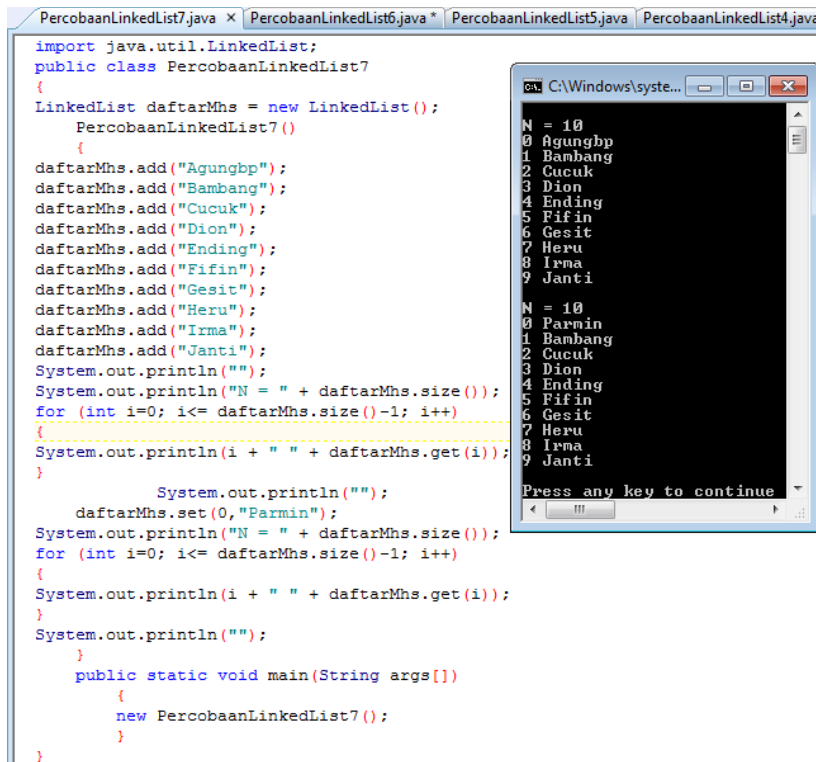
N = 7
0 Bambang
1 Cucuk
2 Dion
3 Ending
4 Gesit
5 Heru
6 Irma

Press any key to continue
```

penjelasan:

Remove(5) menghapus data ke-5 yang ada pada linkedlist daftarMhs.
Mhs. RemoveFirst() menghapus data pertama yang ada pada linkedlist daftarMhs.
removeLast() menghapus data terakhir pada linkedlist daftarMhs.

PRAKTIK 7



```
import java.util.LinkedList;
public class PercobaanLinkedList7
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList7()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifin");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
        System.out.println("");
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++)
        {
            System.out.println(i + " " + daftarMhs.get(i));
        }
        System.out.println("");
        daftarMhs.set(0,"Parmin");
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++)
        {
            System.out.println(i + " " + daftarMhs.get(i));
        }
        System.out.println("");
    }
    public static void main(String args[])
    {
        new PercobaanLinkedList7();
    }
}
```

```
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifin
6 Gesit
7 Heru
8 Irma
9 Janti

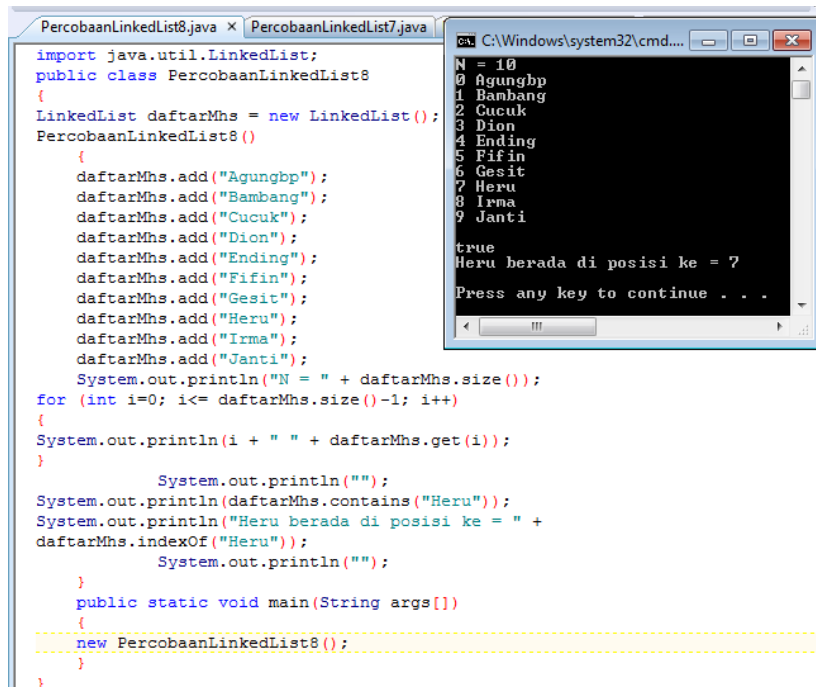
N = 10
0 Parmin
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifin
6 Gesit
7 Heru
8 Irma
9 Janti

Press any key to continue
```

penjelasan:

daftarMhs. set(0,"Parmin") ; mengubah data ke-0 menjadi "Parmin". Bisa dilihat di N = 10 pertama, larik ke-0 diisi oleh Agungbp. Di N = 10 ke-dua, Agungbp telah digantikan oleh Parmin

PRAKTIK 8



```
import java.util.LinkedList;
public class PercobaanLinkedList8
{
    LinkedList daftarMhs = new LinkedList();
    PercobaanLinkedList8()
    {
        daftarMhs.add("Agungbp");
        daftarMhs.add("Bambang");
        daftarMhs.add("Cucuk");
        daftarMhs.add("Dion");
        daftarMhs.add("Ending");
        daftarMhs.add("Fifin");
        daftarMhs.add("Gesit");
        daftarMhs.add("Heru");
        daftarMhs.add("Irma");
        daftarMhs.add("Janti");
        System.out.println("N = " + daftarMhs.size());
        for (int i=0; i<= daftarMhs.size()-1; i++)
        {
            System.out.println(i + " " + daftarMhs.get(i));
        }
        System.out.println("");
        System.out.println(daftarMhs.contains("Heru"));
        System.out.println("Heru berada di posisi ke = " +
            daftarMhs.indexOf("Heru"));
        System.out.println("");
    }
    public static void main(String args[])
    {
        new PercobaanLinkedList8();
    }
}
```

```
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifin
6 Gesit
7 Heru
8 Irma
9 Janti

true
Heru berada di posisi ke = 7

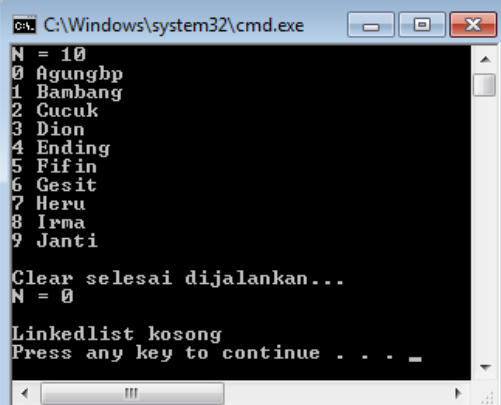
Press any key to continue . . .
```

penjelasan:

daftarMhs. contains("Heru") mengecek apakah data "Heru" ada didalam linkedlist daftarMhs. daftarMhs. indexOf("Heru") memanggil urutan index dari "Heru" berada pada index ke berapa. Dapat dilihat output memunculkan true dan pada index ke-berapa Heru berada, yaitu pada index ke-7.

PRAKTIK 9

```
PercobaanLinkedList9.java x PercobaanLinkedList8.java PercobaanLinkedList7.java
1 import java.util.LinkedList;
2 public class PercobaanLinkedList9
3 {
4     LinkedList daftarMhs = new LinkedList();
5     PercobaanLinkedList9()
6     {
7         daftarMhs.add("Agungbp");
8         daftarMhs.add("Bambang");
9         daftarMhs.add("Cucuk");
10        daftarMhs.add("Dion");
11        daftarMhs.add("Ending");
12        daftarMhs.add("Fifin");
13        daftarMhs.add("Gesit");
14        daftarMhs.add("Heru");
15        daftarMhs.add("Irma");
16        daftarMhs.add("Janti");
17        System.out.println("N = " + daftarMhs.size());
18        for (int i=0; i<= daftarMhs.size()-1; i++)
19        {
20            System.out.println(i + " " + daftarMhs.get(i));
21        }
22        System.out.println("");
23        daftarMhs.clear();
24        System.out.println("Clear selesai dijalankan... ");
25        System.out.println("N = " + daftarMhs.size());
26        for (int i=0; i<= daftarMhs.size()-1; i++)
27        {
28            System.out.println(i + " " + daftarMhs.get(i));
29        }
30        System.out.println("");
31        if (daftarMhs.isEmpty()==true)
32        {
33            System.out.println("Linkedlist kosong");
34        }
35        else
36        {
37            System.out.println("Linkedlist isi");
38        }
39    }
40    public static void main(String args[])
41    {
42        new PercobaanLinkedList9();
43    }
44 }
```



```
C:\Windows\system32\cmd.exe
N = 10
0 Agungbp
1 Bambang
2 Cucuk
3 Dion
4 Ending
5 Fifin
6 Gesit
7 Heru
8 Irma
9 Janti

Clear selesai dijalankan...
N = 0

Linkedlist kosong
Press any key to continue . . .
```

penjelasan:

clear() menghapus semua data yang ada pada linkedlist daftarMhs. Terlihat pada hasil output setelah dilakukan clear() jumlah data yang ada pada linkedlist daftarMhs menjadi N=0. pengecekan dengan isEmpty() menghasilkan true sehingga menciptakan output adalah "Linkedlist kosong".

TUGAS

1. Apa perbedaan Arraylist dan Vektor pada packet java.util

Arraylist	Vektor
Dasarnya tidak disinkronkan	Dasarnya disinkronkan
Kelas warisan adalah kelas koleksi standar	Kelas warisan direkayasa ulang untuk mendukung kelas koleksi
Deklarasi kelasnya adalah kelas Arraylist	Deklarasi kelasnya adalah kelas vektor
Ketika tidak ditentukan, Arraylist bertambah setengah ukurannya	Ketika tidak ditentukan, vektor bertambah dua kali lipat dari ukurannya.
Karena ArrayList tidak disinkronkan, ia beroperasi lebih cepat daripada Vector.	Ketika Vector disinkronkan, ia beroperasi lebih lambat daripada ArrayList.
ArrayList menggunakan antarmuka Iterator untuk melintasi objek yang disimpan di ArrayList.	Vektor menggunakan enumerasi serta antarmuka Iterator untuk melintasi objek yang disimpan dalam Vektor.

2. Bagaimana perintah untuk menambah, menghapus dan menampilkan data pada List?
add(), untuk menambah elemen baru
remove(), untuk menghapus nilai pada suatu indeks
PercobaanLinkedList(); memanggil dengan nama linkedlist.

KESIMPULAN

Collection adalah sebuah framework yang dibuat untuk menyimpan dan memanipulasi sebuah objek. Collection biasanya digunakan pada sebuah data seperti mencari, meng-urutkan, meng-input dan menghapus. Kerangka kerja (Framework) menyediakan dalam bentuk Interface (contoh : Set, List, Queue, Deque dll). untuk class (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet dll)

Collection sendiri merupakan wadah yang menampung value pada kontainer tunggal. Framework java collection menyediakan dengan cara extensible (secara kamus internasional : dapat dibuat panjang atau luas) dan dapat disatukan untuk menangani collections.