

## Jobsheet 11: Linkedlist– Bisma Adhiaksa

### Praktikum 1

1. Buat folder baru dalam folder Praktikum ASD dengan nama Jobsheet11.
2. Dalam folder Jobsheet11 buat beberapa class yaitu Mahasiswa07, NodeMahasiswa07, SingleLinkedList07, dan SLLMain07.
3. Dalam class Mahasiswa07 tambahkan atribut, konstruktor, dan method tampilInformasi

```
1  package Jobsheet11;
2
3
4  public class Mahasiswa07 {
5      String nim, nama, kelas;
6      double ipk;
7
8      public Mahasiswa07() {
9      }
10
11     public Mahasiswa07(String nim, String nama, String kelas, double ipk) {
12         this.nim = nim;
13         this.nama = nama;
14         this.kelas = kelas;
15         this.ipk = ipk;
16     }
17
18     public void tampilInformasi(){
19         System.out.println(nama + "\t\t" + nim + "\t\t" + kelas + "\t" + ipk);
20     }
21
22
23 }
24
```

4. Dalam class NodeMahasiswa07 tambahkan atribut data dengan tipe class Mahasiswa07 dan next dengan tipe NodeMahasiswa07. Lalu buat konstruktornya dengan parameter

```
1  package Jobsheet11;
2
3  public class NodeMahasiswa07 {
4      Mahasiswa07 data;
5      NodeMahasiswa07 next;
6
7      public NodeMahasiswa07(Mahasiswa07 data, NodeMahasiswa07 next){
8          this.data = data;
9          this.next = next;
10     }
11 }
```

5. Dalam class SingleLinkedList07 tambahkan atribut head dan tail dengan tipe class NodeMahasiswa07. Pada class ini tambahkan method-method seperti isEmpty, print, addFirst, addLast, insertAfter, insertAt

```

1 package Jobsheet11;
2
3 public class SingleLinkedList07 {
4
5     NodeMahasiswa07 head;
6     NodeMahasiswa07 tail;
7
8     boolean isEmpty() {
9         return (head == null);
10    }
11
12    public void print() {
13        if (!isEmpty()) {
14            NodeMahasiswa07 tmp = head;
15            System.out.println(x:"Isi Linked List");
16            while (tmp != null) {
17                tmp.data.tampilIformasi();
18                tmp = tmp.next;
19            }
20            System.out.println(x:"");
21        } else {
22            System.out.println(x:"Linked List kosong");
23        }
24    }
25
26    public void addFirst(Mahasiswa07 input){
27        NodeMahasiswa07 ndInput = new NodeMahasiswa07(input, next:null);
28        if(isEmpty()){
29            head = ndInput;
30            tail = ndInput;
31        } else {
32            ndInput.next = head;
33            head = ndInput;
34        }
35    }
36
37    public void addLast(Mahasiswa07 input){

```

```

public void insertAfter(String key, Mahasiswa07 input){
    NodeMahasiswa07 ndInput = new NodeMahasiswa07(input, next:null);
    NodeMahasiswa07 temp = head;
    do {
        if(temp.data.nama.equalsIgnoreCase(key)){
            ndInput.next = temp.next;
            temp.next = ndInput;
            if(ndInput.next == null){
                tail = ndInput;
            }
            break;
        }
        temp = temp.next;
    } while (temp != null);
}

public void insertAt(int index, Mahasiswa07 input){
    if(index < 0){
        System.out.println(x:"Indeks salah");
    } else if(index == 0){
        addFirst(input);
    }else{
        NodeMahasiswa07 temp = head;
        for (int i = 0; i < index; i++) {
            temp = temp.next;
        }
        temp.next = new NodeMahasiswa07(input, temp.next);
        if(temp.next.next == null){
            tail = temp.next;
        }
    }
}
}

```

6. Lalu pada Main buat object baru dengan tipe class Mahasiswa07, lalu panggil method methodnya

```

1  package Jobsheet11;
2
3  public class SLLMain07 {
4      public static void main(String[] args) {
5          SingleLinkedList07 sll = new SingleLinkedList07();
6          Mahasiswa07 mhs1 = new Mahasiswa07(nim:"244107020216", nama:"Bisma", kelas:"1B", ipk:3.7);
7          Mahasiswa07 mhs2 = new Mahasiswa07(nim:"234107020216", nama:"Agatha", kelas:"2B", ipk:3.1);
8          Mahasiswa07 mhs3 = new Mahasiswa07(nim:"224107020216", nama:"Aksa", kelas:"3B", ipk:2.7);
9          Mahasiswa07 mhs4 = new Mahasiswa07(nim:"214107020216", nama:"Samudra", kelas:"4B", ipk:4);
10
11          sll.print();
12          sll.addFirst(mhs4);
13          sll.print();
14          sll.addLast(mhs1);
15          sll.print();
16          sll.insertAfter(key:"Samudra", mhs3);
17          sll.insertAt(index:2, mhs2);
18          sll.print();
19
20      }
21  }
22

```

```

Linked List kosong
Isi Linked List
Samudra      214107020216      4B      4.0

Isi Linked List
Samudra      214107020216      4B      4.0
Bisma       244107020216      1B      3.7

Isi Linked List
Samudra      214107020216      4B      4.0
Aksa        224107020216      3B      2.7
Bisma       244107020216      1B      3.7
Agatha      234107020216      2B      3.1

PS D:\Kuliah\Semester2\PRAKTIKUM-ASD>

```

7.

### 2.1.2 Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan "Linked List Kosong"?
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Jawaban

1. Karena linkedlist belum ada isinya
2. Secara umum variabel temp digunakan sebagai penunjuk dan posisi sekarang linkedlist yang nantinya bisa digunakan untuk menambah dan mengurangi linkedlist

```

1  package Jobsheet11;
2
3  import java.util.Scanner;
4
5  public class SLLMain07 {
    Run | Debug | Run main | Debug main
6      public static void main(String[] args) {
7          SingleLinkedList07 sll = new SingleLinkedList07();
8          Mahasiswa07 mhs[] = new Mahasiswa07[4];
9          Scanner sc = new Scanner(System.in);
10
11         for (int i = 0; i < mhs.length; i++) {
12             mhs[i] = new Mahasiswa07();
13             System.out.println("\nData Mahasiswa ke-" + (i+1));
14             System.out.print("Masukkan Nama Mahasiswa ke-" + (i+1) + ": ");
15             mhs[i].nama = sc.nextLine();
16             System.out.print("Masukkan NIM Mahasiswa ke-" + (i+1) + ": ");
17             mhs[i].nim = sc.nextLine();
18             System.out.print("Masukkan Kelas Mahasiswa ke-" + (i+1) + ": ");
19             mhs[i].kelas = sc.nextLine();
20             System.out.print("Masukkan IPK Mahasiswa ke-" + (i+1) + ": ");
21             mhs[i].ipk = sc.nextDouble();
22             sc.nextLine();
23         }
24
25
26         sll.print();
27         sll.addFirst(mhs[3]);
28         sll.print();
29         sll.addLast(mhs[0]);
30         sll.print();
31         sll.insertAfter(mhs[3].nama, mhs[2]);
32         sll.insertAt(index:2, mhs[1]);
33         sll.print();
34
35     }
36 }
37

```

3.

```

Masukkan IPK Mahasiswa ke-1: 3.7
Data Mahasiswa ke-2
Masukkan Nama Mahasiswa ke-2: Aksa
Masukkan NIM Mahasiswa ke-2: 234
Masukkan Kelas Mahasiswa ke-2: 1C
Masukkan IPK Mahasiswa ke-2: 4
Data Mahasiswa ke-3
Masukkan Nama Mahasiswa ke-3: Adhi
Masukkan NIM Mahasiswa ke-3: 345
Masukkan Kelas Mahasiswa ke-3: 1D
Masukkan IPK Mahasiswa ke-3: 3.2
Data Mahasiswa ke-4
Masukkan Nama Mahasiswa ke-4: yaha
Masukkan NIM Mahasiswa ke-4: 567
Masukkan Kelas Mahasiswa ke-4: 1A
Masukkan IPK Mahasiswa ke-4: 2.3
Linked List kosong
Isi Linked List
yaha          567          1A          2.3

Isi Linked List
yaha          567          1A          2.3
Bisma         123          1B          3.7

Isi Linked List
yaha          567          1A          2.3
Adhi          345          1D          3.2
Bisma         123          1B          3.7
Aksa          234          1C          4.0

```

## Praktikum 2

1. Tambahkan method method baru. getData untuk menampilkan data pada index tertentu, indexOf untuk mencari data pada indeks berapa sesuai dengan key

```

public void getData(int index) {
    NodeMahasiswa07 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}

public int indexOf(String key) {
    NodeMahasiswa07 tmp = head;
    int index = 0;
    while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
        tmp = tmp.next;
        index++;
    }

    if (tmp == null) {
        return -1;
    } else {
        return index;
    }
}

```

2. removeFirst untuk menghapus node di awal, removeLast untuk menghapus node terakhir

```
public void removeFirst() {
    if (isEmpty()) {
        System.out.println(x:" Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        head = head.next;
    }
}

public void removeLast() {
    if (isEmpty()) {
        System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        NodeMahasiswa07 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}
```

3. remove untuk menghapus node pada indeks tertentu berdasarkan key, removeAt untuk menghapus node pada indeks tertentu berdasarkan indeks ke berapa

```
public void remove(String key) {
    if (isEmpty()) {
        System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
    } else {
        NodeMahasiswa07 temp = head;
        while (temp != null) {
            if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
                this.removeFirst();
                break;
            } else if (temp.data.nama.equalsIgnoreCase(key)) {
                temp.next = temp.next.next;
                if (temp.next == null) {
                    tail = temp;
                }
                break;
            }
            temp = temp.next;
        }
    }
}

public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        NodeMahasiswa07 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if(temp.next == null){
            tail = temp;
        }
    }
}
}
```



```

5  public class SLLMain07 {
6
7      Run main | Debug main | Run | Debug
8      public static void main(String[] args) {
9          SingleLinkedList07 sll = new SingleLinkedList07();
10         Mahasiswa07 mhs[] = {
11             new Mahasiswa07(nim:"24212201", nama:"Bisma", kelas:"1C", ipk:4),
12             new Mahasiswa07(nim:"22212202", nama:"Cintia", kelas:"3C", ipk:3.5),
13             new Mahasiswa07(nim:"23212201", nama:"Bimon", kelas:"2B", ipk:3.8),
14             new Mahasiswa07(nim:"24212202", nama:"Aksa", kelas:"1B", ipk:3.6)};
15         Scanner sc = new Scanner(System.in);
16
17         // for (int i = 0; i < mhs.length; i++) {
18         //     mhs[i] = new Mahasiswa07();
19         //     System.out.println("\nData Mahasiswa ke-" + (i+1));
20         //     System.out.print("Masukkan Nama Mahasiswa ke-" + (i+1) + ": ");
21         //     mhs[i].nama = sc.nextLine();
22         //     System.out.print("Masukkan NIM Mahasiswa ke-" + (i+1) + ": ");
23         //     mhs[i].nim = sc.nextLine();
24         //     System.out.print("Masukkan Kelas Mahasiswa ke-" + (i+1) + ": ");
25         //     mhs[i].kelas = sc.nextLine();
26         //     System.out.print("Masukkan IPK Mahasiswa ke-" + (i+1) + ": ");
27         //     mhs[i].ipk = sc.nextDouble();
28         //     sc.nextLine();
29         // }
30         // sll.print();
31         sll.addFirst(mhs[0]);
32         System.out.println(sll.head.data.nama);
33         // sll.print();
34         sll.addLast(mhs[2]);
35         // sll.print();
36         sll.insertAfter(mhs[0].nama, mhs[1]);
37         sll.insertAt(index:2, mhs[3]);
38         // sll.print();
39
40         System.out.println(x:"data index 1 :");
41         sll.getData(index:1);
42
43         System.out.println("data mahasiswa an Bimon berada pada index : " + sll.indexOf(k
44         System.out.println();
45
46         sll.removeFirst();
47         sll.removeLast();
48         sll.print();
49         sll.removeAt(index:0);
50         sll.print();
51     }
52 }

```

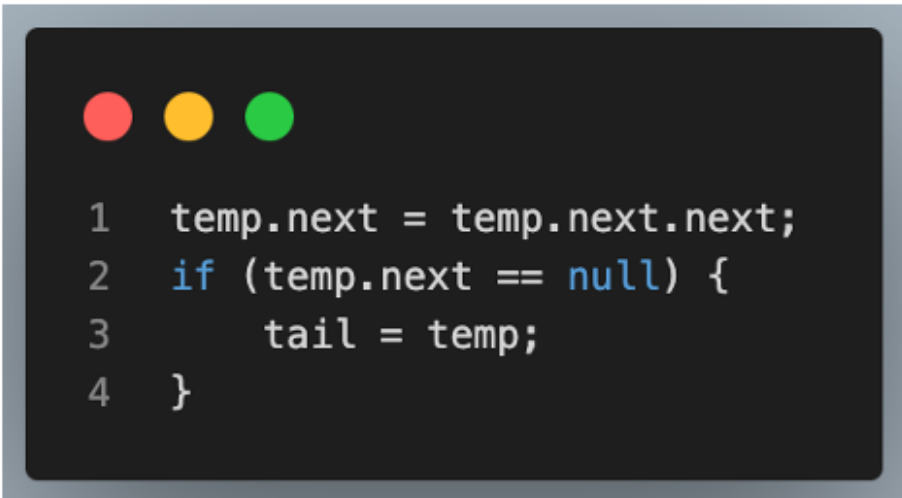
4.

5. Hasil run

```
data index 1 :  
Cintia      22212202      3C      3.5  
data mahasiswa an Bimon berada pada index : 2  
  
Isi Linked List  
Cintia      22212202      3C      3.5  
Bimon      23212201      2B      3.8  
  
Isi Linked List  
Bimon      23212201      2B      3.8  
PS D:\Kuliah\Semester2\PRAKTIKUM-ASD> []
```

### 2.2.3 Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
2. Jelaskan kegunaan kode dibawah pada method remove



```
1 temp.next = temp.next.next;  
2 if (temp.next == null) {  
3     tail = temp;  
4 }
```

Jawaban

1. Untuk keluar dari perulangan while, break ini untuk menghentikan perulangan while karena data yang ingin dicari sudah ketemu
2. Kode yang temp.next = temp.next.next berarti data setelah temp dihapus dan temp akan menunjuk setelah data yang dihapus tersebut. Lalu jika data yang ditunjuk temp itu adalah null maka temp akan menjadi tail.

### 3. Tugas

**Waktu pengerjaan : 50 menit**

Buatlah implementasi program antrian layanan unit kemahasiswaan sesuai dengan berikut ini

- Implementasi antrian menggunakan Queue berbasis Linked List!
- Program merupakan proyek baru bukan modifikasi dari percobaan
- Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya
- Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- Menambahkan antrian
- Memanggil antrian
- Menampilkan antrian terdepan dan antrian paling akhir
- Menampilkan jumlah mahasiswa yang masih mengantre.

```
1 package TugasJobsheet11;
2
3
4 public class Mahasiswa07 {
5     String nim, nama, kelas;
6     double ipk;
7
8
9     public Mahasiswa07() {
10    }
11
12     public Mahasiswa07(String nim, String nama, String kelas, double ipk) {
13         this.nim = nim;
14         this.nama = nama;
15         this.kelas = kelas;
16         this.ipk = ipk;
17     }
18
19     public void tampilIformasi(){
20         System.out.println(nama + "\t\t" + nim + "\t\t" + kelas + "\t" + ipk);
21     }
22
23
24 }
25
```

```
TugasJobsheet11 / NodeMahasiswa07.java / Java / NodeMahasiswa07 / next
1 package TugasJobsheet11;
2
3 public class NodeMahasiswa07 {
4     Mahasiswa07 data;
5     NodeMahasiswa07 next;
6
7     public NodeMahasiswa07(Mahasiswa07 data, NodeMahasiswa07 next){
8         this.data = data;
9         this.next = next;
10    }
11 }
12
```

```

1  package TugasJobsheet11;
2
3  public class SingleLinkedList07 {
4
5      NodeMahasiswa07 head;
6      NodeMahasiswa07 tail;
7      int size = 0, max = 10;
8
9      boolean isEmpty() {
10         return (head == null);
11     }
12
13     boolean isFull() {
14         return (size >= max);
15     }
16
17     int count(){
18         return size;
19     }
20
21     public void print() {
22         if (!isEmpty()) {
23             NodeMahasiswa07 tmp = head;
24             System.out.println(x:"\nIsi Linked List");
25             while (tmp != null) {
26                 tmp.data.tampilIformasi();
27                 tmp = tmp.next;
28             }
29             System.out.println(x:"");
30         } else {
31             System.out.println(x:"\nLinked List kosong");
32         }
33     }
34
35     public void printFirst(){
36         if (!isEmpty()) {
37             System.out.println(x:"\nIsi Linked List depan:");
38             head.data.tampilIformasi();
39         } else {
40             System.out.println(x:"\nLinked List kosong");
41         }
42     }
43
44     public void printLast(){
45         if (!isEmpty()) {
46             System.out.println(x:"\nIsi Linked List depan:");
47             tail.data.tampilIformasi();;
48         } else {
49             System.out.println(x:"\nLinked List kosong");
50         }
51     }

```

```

public void add(Mahasiswa07 input) {
    if (isFull()) {
        System.out.println("x:\nLinked List Penuh broo");
    } else {
        NodeMahasiswa07 ndInput = new NodeMahasiswa07(input, next:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
            size++;
        } else {
            tail.next = ndInput;
            tail = ndInput;
            size++;
        }
    }
}

public void dequeueFirst() {
    if (isEmpty()) {
        System.out.println("x: Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head.data.tampilIformasi();
        head = tail = null;
        size--;
    } else {
        head.data.tampilIformasi();
        head = head.next;
        size--;
    }
}

public void clear(){
    head = tail = null;
    size = 0;
}
}

```

TugasJobsheet11 > SLLMain07.java > Java > SLLMain07 > main(String[] args)

```
1 package TugasJobsheet11;
2
3 import java.util.Scanner;
4
5 public class SLLMain07 {
6
7     Run main | Debug main | Run | Debug
8     public static void main(String[] args) {
9         SingleLinkedList07 sll = new SingleLinkedList07();
10        Scanner sc = new Scanner(System.in);
11        String nama, kelas, nim;
12        double ipk;
13        int i = 1;
14        while (!sll.isFull()) {
15            System.out.println(x:"Menu");
16            System.out.println(x:"1. Tambahkan Data");
17            System.out.println(x:"2. Panggil Antrian");
18            System.out.println(x:"3. Cek Antrian Depan");
19            System.out.println(x:"4. Cek Antrian Belakang");
20            System.out.println(x:"5. Jumlah Antrian");
21            int pilihan = sc.nextInt();
22            sc.nextLine();
23            switch (pilihan) {
24                case 1:
25                    System.out.println("\nMasukkan Antrian ke-" + (i));
26                    System.out.print(s:"Nama: ");
27                    nama = sc.nextLine();
28                    System.out.print(s:"Kelas: ");
29                    kelas = sc.nextLine();
30                    System.out.print(s:"NIM: ");
31                    nim = sc.nextLine();
32                    System.out.print(s:"IPK: ");
33                    ipk = sc.nextDouble();
34                    sc.nextLine();
35                    sll.add(new Mahasiswa07(nim, nama, kelas, ipk));
36                    i++;
37                    break;
38                case 2:
39                    sll.dequeueFirst();
40                    break;
41                case 3:
42                    sll.printFirst();
43                    break;
44                case 4:
45                    sll.printLast();
46                    ;
47                    break;
48                case 5:
49                    sll.count();
50                    break;
51                default:
52                    throw new AssertionError();
53            }
54        }
55    }
```