

Nama : Desy Dwi Puspita

NIM : 244107060145

Kelas : SIB1A

Algoritma dan Struktur Data - Jobsheet XI LINKED LIST

Praktikum Pembuatan Single Linked List

- Kode program

- Class Mahasiswa03.java

```
> OPEN EDITORS
Pertemuan12 > J Mahasiswa03.java > ...
JOBSHEET11-LINKEDLIST
  Pertemuan12
    J Mahasiswa03.java
    J Node03.java
    J SingleLinkedList03.java...
    J SLLMain03.java

1 package Pertemuan12;
2
3 public class Mahasiswa03 {
4     String nim, nama, kelas;
5     double ipk;
6
7     public Mahasiswa03() {}
8
9     public Mahasiswa03(String nim, String nama, String kelas, double ipk) {
10         this.nim = nim;
11         this.nama = nama;
12         this.kelas = kelas;
13         this.ipk = ipk;
14     }
15
16     public void tampilInformasi() {
17         System.out.printf(format: "%-12s %-20s %-6s %.2f\\n", nim, nama, kelas, ipk);
18     }
19 }
```

- Class Node03.java

```
> OPEN EDITORS
Pertemuan12 > J Node03.java > ...
JOBSHEET11-LINKEDLIST
  Pertemuan12
    J Mahasiswa03.java
    J Node03.java
    J SingleLinkedList03.java...
    J SLLMain03.java

1 package Pertemuan12;
2 public class Node03 {
3     Mahasiswa03 data;
4     Node03 next;
5
6     public Node03(Mahasiswa03 data, Node03 next) {
7         this.data = data;
8         this.next = next;
9     }
10 }
```

- Class SingleLinkedList03.java

```
> OPEN EDITORS      Pertemuan12 > J SingleLinkedList03.java > SingleLinkedList03
v JOBSHEET11-LINKEDLIST
  v Pertemuan12
    J Mahasiswa03.java
    J Node03.java
    J SingleLinkedList03.java
    J SLLMain03.java

1 package Pertemuan12;
2
3 public class SingleLinkedList03 {
4     Node03 head;
5     Node03 tail;
6
7     boolean isEmpty() {
8         return (head == null);
9     }
10
11 public void print() {
12     if (isEmpty()) {
13         System.out.println("Linked list kosong");
14     } else {
15         System.out.println("Isi Linked List:");
16         System.out.printf(format:"%-12s %-20s %-6s %s\n", ...args:"NIM", "Nama", "Kelas", "IPK");
17         Node03 tmp = head;
18         while (tmp != null) {
19             tmp.data.tampilInformasi();
20             tmp = tmp.next;
21         }
22     }
23     System.out.println();
24 }
25
26 public void addFirst(Mahasiswa03 input) {
27     Node03 ndInput = new Node03(input, next:null);
28     if (isEmpty()) {
29         head = ndInput;
30         tail = ndInput;
31     } else {
32         ndInput.next = head;
33         head = ndInput;
34     }
35 }
```

```
> OPEN EDITORS      Pertemuan12 > J SingleLinkedList03.java > SingleLinkedList03
v JOBSH... [F] [E] [U] [D]
  v Pertemuan12
    J Mahasiswa03.java
    J Node03.java
    J SingleLinkedList03.java
    J SLLMain03.java

3 public class SingleLinkedList03 {
37 public void addLast(Mahasiswa03 input) {
38     Node03 ndInput = new Node03(input, next:null);
39     if (isEmpty()) {
40         head = ndInput;
41         tail = ndInput;
42     } else {
43         tail.next = ndInput;
44         tail = ndInput;
45     }
46 }
47
48
49 public void insertAfter(String key, Mahasiswa03 input) {
50     Node03 ndInput = new Node03(input, next:null);
51     Node03 temp = head;
52     do {
53         if (temp.data.nama.equalsIgnoreCase(key)) {
54             ndInput.next = temp.next;
55             temp.next = ndInput;
56             if (ndInput.next == null) {
57                 tail = ndInput;
58             }
59             break;
60         }
61         temp = temp.next;
62     } while (temp != null);
63 }
```

```

> OPEN EDITORS
JOBSHEET11-LINKEDLIST
  Pertemuan12
    Mahasiswa03.java
    Node03.java
    SingleLinkedList03.java
    SLLMain03.java

Pertemuan12 > J SingleLinkedList03.java > ...
3 public class SingleLinkedList03 {
65     public void insertAt(int index, Mahasiswa03 input) {
66         if (index < 0) {
67             System.out.println(x:"Indeks salah");
68         } else if (index == 0) {
69             addFirst(input);
70         } else {
71             Node03 temp = head;
72             for (int i = 0; i < index - 1; i++) {
73                 temp = temp.next;
74             }
75             temp.next = new Node03(input, temp.next);
76             if (temp.next.next == null) {
77                 tail = temp.next;
78             }
79         }
80     }
81 }

```

- SLLMain03.java

```

> OPEN EDITORS
JOBSHEET11-LINKEDLIST
  Pertemuan12
    Mahasiswa03.java
    Node03.java
    SingleLinkedList03.java
    SLLMain03.java

Pertemuan12 > J SLLMain03.java > ...
1 package Pertemuan12;
2
3 public class SLLMain03 {
4     public static void main(String[] args) {
5         SingleLinkedList03 sll = new SingleLinkedList03();
6
7         Mahasiswa03 mhs1 = new Mahasiswa03(nim:"12345", nama:"Desy Dwi Puspita", kelas:"1A", ipk:4.0);
8         Mahasiswa03 mhs2 = new Mahasiswa03(nim:"12346", nama:"Dany Setya Pradana", kelas:"3A", ipk:3.8);
9         Mahasiswa03 mhs3 = new Mahasiswa03(nim:"12347", nama:"Kiara Gayatri", kelas:"1A", ipk:3.9);
10        Mahasiswa03 mhs4 = new Mahasiswa03(nim:"12348", nama:"Arshaka Virendra", kelas:"2A", ipk:3.7);
11
12        sll.print();
13        sll.addFirst(mhs4);
14        sll.print();
15        sll.addLast(mhs1);
16        sll.print();
17        sll.insertAfter(key:"Arshaka Virendra", mhs3);
18        sll.insertAt(index:2, mhs2);
19        sll.print();
20    }
21 }

```

- Output

```

Linked list kosong

Isi Linked List:
NIM      Nama      Kelas  IPK
12348    Arshaka Virendra  2A     3.70

Isi Linked List:
NIM      Nama      Kelas  IPK
12348    Arshaka Virendra  2A     3.70
12345    Desy Dwi Puspita  1A     4.00

Isi Linked List:
NIM      Nama      Kelas  IPK
12348    Arshaka Virendra  2A     3.70
12347    Kiara Gayatri    1A     3.90
12346    Dany Setya Pradana 3A     3.80
12345    Desy Dwi Puspita  1A     4.00

```

- Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
Karena pada class main, program menjalankan method print(); di mana saat itu belum dijalankan method penambahan data, sehingga Linked List data masih kosong.
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
variabel temp digunakan untuk menelusuri (traverse) node-node dalam linked list mulai dari head. Ini mempermudah pencarian posisi atau node tertentu untuk penambahan, penghapusan, atau pencetakan data.
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Kode program :

```
5  import java.util.Scanner;
6
7  public class SLLMain03 {
8      public static void main(String[] args) {
9          Scanner sc = new Scanner(System.in);
10         SingleLinkedList03 sll = new SingleLinkedList03();
11
12         System.out.print(s:"Masukkan jumlah mahasiswa yang ingin ditambahkan: ");
13         int jumlah = sc.nextInt();
14         sc.nextLine();
15
16         for (int i = 0; i < jumlah; i++) {
17             System.out.println("Mahasiswa ke-" + (i + 1));
18             System.out.print(s:"NIM: ");
19             String nim = sc.nextLine();
20             System.out.print(s:"Nama: ");
21             String nama = sc.nextLine();
22             System.out.print(s:"Kelas: ");
23             String kelas = sc.nextLine();
24             System.out.print(s:"IPK: ");
25             double ipk = sc.nextDouble();
26             sc.nextLine();
27
28             Mahasiswa03 mhs = new Mahasiswa03(nim, nama, kelas, ipk);
29             sll.addLast(mhs);
30         }
31
32         System.out.println();
33         sll.print();
34
35         sc.close();
36     }
37 }
```

Output :

```
Masukkan jumlah mahasiswa yang ingin ditambahkan: 1
Mahasiswa ke-1
NIM: 1234
Nama: desy
Kelas: 1a
IPK: 4.0

Isi Linked List:
NIM      Nama      Kelas  IPK
1234     desy      1a     4.00
```

Praktikum Modifikasi Elemen pada Single Linked List

- Kode program Class SingleLinkedList :

```
public void getData(int index) {
    Node03 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}

public int indexOf(String key) {
    Node03 temp = head;
    int index = 0;
    while (temp != null && !temp.data.nama.equalsIgnoreCase(key)) {
        temp = temp.next;
        index++;
    }

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

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 {
        Node03 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}

public void remove(String key) {
    if (isEmpty()) {
        System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
    } else {
        Node03 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 {
        Node03 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}

```

- Output

```

data index 1 :
12347      Kiara Gayatri      1A      3.90
data mahasiswa an Bimon berada pada index : 2

Isi Linked List:
NIM      Nama      Kelas  IPK
12347      Kiara Gayatri      1A      3.90
12346      Bimon      3A      3.80

Isi Linked List:
NIM      Nama      Kelas  IPK
12346      Bimon      3A      3.80

```

- Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
Keyword break digunakan agar perulangan berhenti setelah data yang dicari ditemukan dan dihapus, sehingga proses lebih efisien dan hanya menghapus satu node yang sesuai
2. Jelaskan kegunaan kode dibawah pada method remove

```

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

```

Kode tersebut digunakan untuk menghapus node setelah temp, lalu mengecek jika node tersebut adalah node terakhir. Jika iya, maka tail diperbarui ke temp

Tugas

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.

Kode program :

```
Pertemuan12 > Tugas > J Mahasiswa03.java > ...
1  package Pertemuan12.Tugas;
2  public class Mahasiswa03 {
3      String nim, nama, keperluan;
4
5      public Mahasiswa03(String nim, String nama, String keperluan) {
6          this.nim = nim;
7          this.nama = nama;
8          this.keperluan = keperluan;
9      }
10     public void tampilInformasi() {
11         System.out.println("NIM: " + nim + ", Nama: " + nama + ", Keperluan: " + keperluan);
12     }
13 }
```

```
Pertemuan12 > Tugas > J Node03.java > ...
1  package Pertemuan12.Tugas;
2  public class Node03 {
3      Mahasiswa03 data;
4      Node03 next;
5
6      public Node03(Mahasiswa03 data) {
7          this.data = data;
8          this.next = null;
9      }
10 }
```

Pertemuan12 > Tugas > J Layanan03.java > Layanan03 > tampilDepanBelakang()

```
1  package Pertemuan12.Tugas;
2
3  public class Layanan03{
4      Node03 front, rear;
5      int size = 0;
6      int kapasitas;
7
8      public Layanan03(int kapasitas) {
9          this.front = this.rear = null;
10         this.kapasitas = kapasitas;
11     }
12
13     public boolean isEmpty() {
14         return front == null;
15     }
16
17     public boolean isFull() {
18         return size >= kapasitas;
19     }
20
21     public void enqueue(Mahasiswa03 mhs) {
22         if (isFull()) {
23             System.out.println(x:"Antrian penuh, tidak dapat menambah mahasiswa.");
24             return;
25         }
26
27         Node03 newNode03 = new Node03(mhs);
28         if (isEmpty()) {
29             front = rear = newNode03;
30         } else {
31             rear.next = newNode03;
32             rear = newNode03;
33         }
34         size++;
35         System.out.println(x:"Mahasiswa berhasil ditambahkan ke antrian.");
36     }
```



```

38     public void dequeue() {
39         if (isEmpty()) {
40             System.out.println(x:"Antrian kosong, tidak ada yang dipanggil.");
41         } else {
42             System.out.print(s:"Memanggil mahasiswa: ");
43             front.data.tampilInformasi();
44             front = front.next;
45             size--;
46             if (front == null) {
47                 rear = null;
48             }
49         }
50     }

51     public void clear() {
52         front = rear = null;
53         size = 0;
54         System.out.println(x:"Antrian berhasil dikosongkan.");
55     }

56     public void tampilDepanBelakang() {
57         if (isEmpty()) {
58             System.out.println(x:"Antrian kosong.");
59         } else {
60             System.out.print(s:"Antrian terdepan: ");
61             front.data.tampilInformasi();

62             if (front != rear) {
63                 System.out.print(s:"Antrian paling belakang: ");
64                 rear.data.tampilInformasi();
65             } else {
66                 System.out.println(x:"Hanya ada satu orang dalam antrian.");
67             }
68         }
69     }
70 }
71 }
72 }

```

```

74  ✓ public void printAntrian() {
75  ✓     if (isEmpty()) {
76         System.out.println(x:"Antrian kosong.");
77         return;
78     }

79     System.out.println(x:"Daftar antrian mahasiswa:");
80     Node03 current = front;
81     int no = 1;
82     while (current != null) {
83     ✓         System.out.print(no + ". ");
84             current.data.tampilInformasi();
85             current = current.next;
86             no++;
87         }
88     }
89 }
90 }

```

Output :

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM: 789
Nama: desta
Keperluan: bayar ukt
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 2
Memanggil mahasiswa: NIM: 123, Nama: desy, Keperluan: bayar ukt
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Antrian tidak kosong.

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 4
Antrian belum penuh.

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Antrian terdepan: NIM: 124, Nama: dany, Keperluan: bayar ukt
Antrian paling belakang: NIM: 789, Nama: desta, Keperluan: bayar ukt
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 6
Daftar antrian mahasiswa:
1. NIM: 124, Nama: dany, Keperluan: bayar ukt
2. NIM: 156, Nama: dany, Keperluan: bayar ukt
3. NIM: 789, Nama: desta, Keperluan: bayar ukt

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Tampilkan Antrian Terdepan & Paling Belakang
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 7
Antrian berhasil dikosongkan.
```