



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

11.2 Pembuatan Single Linked List

```
package P11.SingleLinkedList;

public class Node25 {
    int data;
    Node25 next;

    Node25(int nilai, Node25 berikutnya){
        this.data = nilai;
        this.next = berikutnya;
    }
}

package P11.SingleLinkedList;

public class SingleLinkedList25 {
    Node25 head, tail;

    boolean isEmpty(){
        return head == null;
    }

    void print(){
        if(!isEmpty()){
            Node25 tmp = head;
            System.out.print(s:"Isi Linked List: ");
            while(tmp != null){
                System.out.print(tmp.data + "\t");
                tmp = tmp.next;
            }
            System.out.println(x:"");
        }else{
            System.out.println(x:"Linked List Kosong");
        }
    }

    void addFirst(int input){
        Node25 ndInput = new Node25(input, berikutnya:null);
        if(isEmpty()){
            head = ndInput;
            tail = ndInput;
        }else{
            ndInput.next = head;
            head = ndInput;
        }
    }

    void addLast(int input){
        Node25 ndInput = new Node25(input, berikutnya:null);
        if(isEmpty()){
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    void insertAfter(int key, int input){
        Node25 ndInput = new Node25(input, berikutnya:null);
        Node25 temp = head;
        do {
            if (temp.data == key) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if(ndInput.next != null) tail = ndInput;
                break;
            }
            temp = temp.next;
        }while (temp != null);
    }

    void insertAt(int index, int input){
        if(index < 0){
            System.out.println("perbaiki logikanya!"
                + "kalau indeksnya -1 bagaimana???");
        } else if(index == 0){
            addFirst(input);
        } else {
            Node25 temp = head;
            for(int i=0; i<index-1; i++){
                temp = temp.next;
            }
            temp.next = new Node25(input, temp.next);
            if(temp.next.next == null) tail=temp.next;
        }
    }
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

```
package P11.SingleLinkedList;

public class SLL25Main {
    Run | Debug
    public static void main(String[] args) {
        SingleLinkedList25 singLL=new SingleLinkedList25();

        singLL.print();
        singLL.addFirst(input:890);
        singLL.print();
        singLL.addLast(input:760);
        singLL.print();
        singLL.addFirst(input:700);
        singLL.print();
        singLL.insertAfter(key:700, input:999);
        singLL.print();
        singLL.insertAt(index:3, input:833);
        singLL.print();
    }
}
```

11.2.2 Verifikasi Hasil Percobaan

```
Linked List Kosong
Isi Linked List:      890
Isi Linked List:      890      760
Isi Linked List:      700      890      760
Isi Linked List:      700      999      890      760
Isi Linked List:      700      999      890      833      760
PS D:\PrakASD_1F_25>
```

11.2.3 Question

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?

Answer:

Karena linked list belum memiliki data, sedangkan di method main diawal memanggil method print, yang mana linked list nya masih kosong, sehingga akan menampilkan bagian else, yaitu “Linked List Kosong”

2. Jelaskan kegunaan variable temp secara umum pada setiap method!

Answer:

Variabel "temp" sering digunakan untuk menyimpan referensi sementara ke node saat melakukan iterasi atau operasi tertentu, seperti pencarian, penyisipan, atau penghapusan.

3. Perhatikan class SingleLinkedList, pada method insertAt Jelaskan kegunaan kode berikut

```
if(temp.next.next==null) tail=temp.next;
```

Answer:

Kode tersebut digunakan untuk memeriksa apakah node yang ditunjuk oleh `temp.next` adalah node terakhir dalam linked list atau tidak. Jika `temp.next.next` adalah `null`, ini menunjukkan bahwa `temp.next` adalah node terakhir dalam linked list, karena tidak ada node setelahnya. Dalam konteks ini, kode `tail = temp.next;` mengatur `tail` ke node terakhir dalam linked list, karena kita telah menyisipkan node baru setelah node `temp`, sehingga node baru tersebut sekarang menjadi node terakhir dalam linked list.



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

Dengan cara ini, kita memastikan bahwa `tail` selalu menunjuk ke node terakhir yang valid dalam linked list setelah operasi penyisipan.

11.3 Praktikum 2

```
public class SingleLinkedList {  
    //MODIFIKASI ELEMEN  
    int getData(int index){  
        Node25 tmp = head;  
        for (int i=0; i<index; i++){  
            tmp = tmp.next;  
        }  
        return tmp.data;  
    }  
  
    int indexOf(int key){  
        Node25 tmp = head;  
        int index = 0;  
        while (tmp != null && tmp.data != key) {  
            tmp = tmp.next;  
            index++;  
        }  
        if (tmp == null){  
            return -1;  
        }else{  
            return index;  
        }  
    }  
  
    void removeFirst(){  
        if(isEmpty()){  
            System.out.println("Linked List Masih Kosong,"  
                                + "tidak dapat dihapus!");  
        }else if (head == tail){  
            head = tail = null;  
        }else{  
            head = head.next;  
        }  
    }  
}
```

```
public class SingleLinkedList {  
    void removeLast() {  
        if (isEmpty()){  
            System.out.println("Linked List Masih kosong,"  
                                + "tidak dapat dihapus!");  
        }else if (head == tail){  
            head = tail = null;  
        }else{  
            Node25 temp = head;  
            while (temp.next.next != null){  
                temp = temp.next;  
            }  
            temp.next = null;  
            tail = temp.next;  
        }  
    }  
  
    void remove(int key){  
        if(isEmpty()){  
            System.out.println("Linked List Masih kosong,"  
                                + "tidak dapat dihapus!");  
        }else {  
            Node25 temp = head;  
            while (temp != null){  
                if((temp.data == key) && (temp == head)){  
                    removeFirst();  
                    break;  
                }else if (temp.next.data == key){  
                    temp.next = temp.next.next;  
                    if(temp.next == null){  
                        tail = temp;  
                    }  
                    break;  
                }  
                temp = temp.next;  
            }  
        }  
    }  
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

```
void removeAt(int index){  
    if(index == 0){  
        removeFirst();  
    }else{  
        Node25 temp = head;  
        for(int i=0; i < index-1; i++){  
            temp = temp.next;  
        }  
        temp.next = temp.next.next;  
        if (temp.next == null){  
            tail = temp;  
        }  
    }  
}
```

11.3.2 Verifikasi Hasil Percobaan

```
Linked List Kosong  
Isi Linked List:      890  
Isi Linked List:      890      760  
Isi Linked List:      700      890      760  
Isi Linked List:      700      999      890      760  
Isi Linked List:      700      999      890      833      760  
Data pada index ke - 1 = 999  
Data 3 berada pada index ke - 4  
Isi Linked List:      700      890      833      760  
Isi Linked List:      890      833      760  
Isi Linked List:      833      760  
Isi Linked List:      833  
PS D:\PrakASD_1F_25>
```

11.3.3 Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!

Answer:

Karena keyword break tersebut berfungsi untuk menghentikan perulangan. Keyword break yang pertama berfungsi untuk menghentikan program jika data yang dihapus ada pada head dan ditemukan. Keyword kedua untuk menghentikan program jika data yang dihapus berada ditengah dan ditemukan.

2. Jelaskan kegunaan kode dibawah pada method remove

```
else if (temp.next.data == key) {  
    temp.next = temp.next.next;
```

Answer:

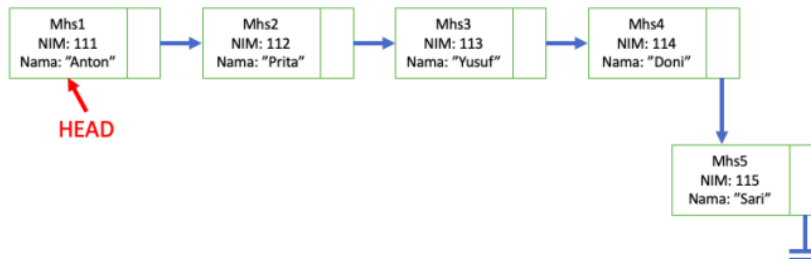
Kode tersebut berfungsi jika node yang dipilih memiliki data yang sama dengan key, maka akan terjadi perubahan posisi dimana node tersebut akan digantikan oleh node yang selanjutnya.



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

11.4 Tugas

1. Implementasikan ilustrasi Linked List Berikut. Gunakan 4 macam penambahan data yang telah dipelajari sebelumnya untuk menginputkan data.



Answer :

```
package P11.Tugas_P11.Nomor1;

public class Node25 {
    int nim;
    String nama;
    Node25 next;

    Node25(int nim, String nama, Node25 berikutnya){
        this.nim = nim;
        this.nama = nama;
        next = berikutnya;
    }
}

package P11.Tugas_P11.Nomor1;

public class Mahasiswa25 {
    Node25 head, tail;

    boolean isEmpty(){
        return head == null;
    }

    void print() {
        if (!isEmpty()) {
            Node25 tmp = head;
            System.out.print(s:"Isi linked list: \t");
            while (tmp != null) {
                System.out.print(tmp.nim + " " + tmp.nama + " ");
                tmp = tmp.next;
            }
            System.out.println(x:"");
        } else {
            System.out.println(x:"Linked list kosong");
        }
    }

    void addFirst(int nim, String nama) {
        Node25 ndInput = new Node25(nim, nama, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    void addLast(int nim, String nama) {
        Node25 ndInput = new Node25(nim, nama, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

```
public class Mahasiswa25 {  
  
    void insertAfter(int key, int nim, String nama) {  
        Node25 ndInput = new Node25(nim, nama, berikutnya:null);  
        Node25 temp = head;  
        while (temp != null) {  
            if (temp.nim == key) {  
                ndInput.next = temp.next;  
                temp.next = ndInput;  
  
                if (temp == tail) {  
                    tail = ndInput;  
                }  
                break;  
            }  
            temp = temp.next;  
        }  
    }  
  
    void insertAt(int index, int nim, String nama) {  
        if (index > 0) {  
            Node25 temp = head;  
            for (int i = 1; i < index; i++) {  
                temp = temp.next;  
            }  
            temp.next = new Node25(nim, nama, temp.next);  
            if (temp.next.next == null) {  
                tail = temp.next;  
            }  
        } else if (index == 0) {  
            addFirst(nim, nama);  
        } else {  
            System.out.println("Indeks -1. Tidak bisa melakukan operasi linked list!");  
        }  
    }  
}
```

```
package P11.Tugas_P11.Nomor1;  
  
public class Mhs25Main {  
    Run | Debug  
    public static void main(String[] args) {  
        Mahasiswa25 singLL = new Mahasiswa25();  
  
        singLL.print();  
        singLL.addFirst(nim:112, nama:"Prita");  
        singLL.print();  
        singLL.addFirst(nim:111, nama:"Anton");  
        singLL.print();  
        singLL.insertAfter(key:112, nim:114, nama:"Doni");  
        singLL.print();  
        singLL.insertAt(index:2, nim:113, nama:"Yusuf");  
        singLL.print();  
        singLL.addLast(nim:115, nama:"Sari");  
        singLL.print();  
    }  
}
```

```
Linked list kosong  
Isi linked list: [112, Prita]  
Isi linked list: [111, Anton] [112, Prita]  
Isi linked list: [111, Anton] [112, Prita] [114, Doni]  
Isi linked list: [111, Anton] [112, Prita] [113, Yusuf] [114, Doni]  
Isi linked list: [111, Anton] [112, Prita] [113, Yusuf] [114, Doni] [115, Sari]  
PS D:\PrakASD_1F_25>
```

2. Buatlah implementasi program antrian layanan unit kemahasiswaan sesuai dengan kondisi yang ditunjukkan pada soal nomor 1! Ketentuan
- Implementasi antrian menggunakan Queue berbasis Linked List!
 - Program merupakan proyek baru, bukan modifikasi dari soal nomor 1!

Answer :

```
P11 > Tugas_P11 > Nomor2 > Node25.java > Node25 > Node25(int, St  
1 package P11.Tugas_P11.Nomor2;  
2  
3 public class Node25 {  
4     int nim;  
5     String nama;  
6     Node25 next;  
7  
8     Node25(int nim, String nama, Node25 berikutnya) {  
9         this.nim = nim;  
10        this.nama = nama;  
11        next = berikutnya;  
12    }  
13 }  
14
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

```
package P11.Tugas_P11.Nomer1;  
  
public class UnitLayanan25 {  
    Node25 head, tail;  
  
    boolean isEmpty() {  
        return head == null;  
    }  
  
    void print() {  
        if (!isEmpty()) {  
            Node25 tmp = head;  
            int antrian = 0;  
            while (tmp != null) {  
                System.out.println("Antrian ke - " + (antrian+1) + " = [" + tmp.nim + "] [" + tmp.nama + "]");  
                tmp = tmp.next;  
                antrian++;  
            }  
            System.out.println(x:"");  
        } else {  
            System.out.println(x:"Linked list kosong\n");  
        }  
    }  
  
    void addLast(int nim, String nama){  
        Node25 ndInput = new Node25(nim, nama, berikutnya:null);  
        if(isEmpty()){  
            head = ndInput;  
            tail = ndInput;  
        }else{  
            tail.next = ndInput;  
            tail = ndInput;  
        }  
    }  
  
    void removeFirst(){  
        if(isEmpty()){  
            System.out.println(x:"Linked List Masih Kosong");  
        }else if(head == tail){  
            head = tail = null;  
        }else{  
            head = head.next;  
        }  
    }  
}
```

```
1 package P11.Tugas_P11.Nomer1;  
2  
3 import java.util.Scanner;  
4  
5 public class UnitLayananMain {  
6     public static void menu(){  
7         System.out.println(x:"---Selamat Datang di Layanan Unit Kemahasiswaan---");  
8         System.out.println(x:"-----");  
9         System.out.println(x:"Pilih Menu : ");  
10        System.out.println(x:"1. Antrian Baru");  
11        System.out.println(x:"2. Antrian Keluar");  
12        System.out.println(x:"3. Cek Semua Antrian");  
13        System.out.println(x:"4. Keluar");  
14        System.out.println(x:"-----");  
15    }  
16  
17    public static void main(String[] args) {  
18        Scanner sc25 = new Scanner(System.in);  
19  
20        UnitLayanan25 singLL = new UnitLayanan25();  
21  
22        int pilih;  
23        do{  
24            menu();  
25            pilih = sc25.nextInt();  
26            sc25.nextLine();  
27  
28            switch (pilih) {  
29                case 1:  
30                    System.out.print(s:"Nim Mahasiswa\t: ");  
31                    int nim = sc25.nextInt();  
32                    System.out.print(s:"Nama Mahasiswa\t: ");  
33                    String nama = sc25.next();  
34                    singLL.addLast(nim, nama);  
35                    System.out.println(x:"Antrian berhasil ditambahkan!\n");  
36                    System.out.println(x:"-----");  
37                    sc25.nextLine();  
38                    break;  
39                case 2:  
40                    singLL.removeFirst();  
41                    System.out.println(x:"Mahasiswa antrian pertama sudah selesai dilayani!\n");  
42                    System.out.println(x:"Sisa Antrian : \n");  
43                    singLL.print();  
44                    break;  
45                case 3:  
46                    System.out.println(x:"Berikut adalah data semua antrian : \n");  
47                    singLL.print();  
48                    break;  
49                case 4:  
50                    System.out.println(x:"Terimakasih Telah Menggunakan Unit Layanan Kemahasiswaan ");  
51                    System.exit(status:0);  
52                    break;  
53            }  
54        }while ([pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4]);  
55  
56        sc25.close();  
57    }  
58 }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : LINKED LIST

```
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
1
Nim Mahasiswa : 111
Nama Mahasiswa : Anton
Antrian berhasil ditambahkan!

-----
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
1
Nim Mahasiswa : 112
Nama Mahasiswa : Prita
Antrian berhasil ditambahkan!

-----
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
1
Nim Mahasiswa : 113
Nama Mahasiswa : Yusuf
Antrian berhasil ditambahkan!

-----
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
3
Berikut adalah data semua antrian :
Antrian ke - 1 = [111] [Anton]
Antrian ke - 2 = [112] [Prita]
Antrian ke - 3 = [113] [Yusuf]
Antrian ke - 4 = [114] [Doni]
Antrian ke - 5 = [115] [Sari]
```

```
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
2
Mahasiswa antrian pertama sudah selesai dilayani!

Sisa Antrian :
Antrian ke - 1 = [112] [Prita]
Antrian ke - 2 = [113] [Yusuf]
Antrian ke - 3 = [114] [Doni]
Antrian ke - 4 = [115] [Sari]

-----
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
2
Mahasiswa antrian pertama sudah selesai dilayani!

Sisa Antrian :
Antrian ke - 1 = [113] [Yusuf]
Antrian ke - 2 = [114] [Doni]
Antrian ke - 3 = [115] [Sari]
```

```
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
3
Berikut adalah data semua antrian :
Antrian ke - 1 = [113] [Yusuf]
Antrian ke - 2 = [114] [Doni]
Antrian ke - 3 = [115] [Sari]

-----
===Selamat Datang di Layanan Unit Kemahasiswaan===
-----
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Semua Antrian
4. Keluar
-----
4
Terimakasih Telah Menggunakan Unit Layanan Kemahasiswaan
PS D:\PrakASD_1F_25>
```