

**LAPORAN PRAKTIKUM MINGGU KE-11**  
**“[Single Linked List]”**



Disusun oleh:

**[Daffa Aqila Rahmatullah]**  
**[2041720098]**

**D4 TEKNIK INFORMATIKA**  
**TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2021**

**LAPORAN**  
**“PRAKTIKUM”**

**A. KODE PROGRAM**

**9.2.1 Langkah-langkah Percobaan**

```
package minggull;  
public class Node {  
    int data;  
    Node next;  
    public Node(int nilai, Node berikutnya){  
        this.data = nilai;  
        this.next=berikutnya;  
    }  
}  
package minggull;  
public class SingleLinkedList {  
    Node head; // posisi awal linked list  
    Node tail; //posisi akgir linked list  
  
    public boolean isEmpty(){  
        return head == null;  
    }  
  
    public void print(){  
        if(!isEmpty())  
        {  
            Node tmp = head;  
            System.out.print("Isi Linked List:\t");  
            while (tmp != null)  
            {  
                System.out.print(tmp.data + "\t");  
                tmp = tmp.next;  
            }  
            System.out.println("");  
        }  
    }  
}
```

```
        }

    else
    {
        System.out.println("Linked List Kosong");
    }
}

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

public void addLast(int input){
    Node ndInput = new Node(input, null);
    if (isEmpty())
    {
        head = ndInput;
        tail = ndInput;
    }
    else
    {
        tail.next=ndInput;
        tail = ndInput;
    }
}
```

```
public void insertAfter(int key, int input) {  
    Node ndInput = new Node(input, null);  
    Node 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);  
}
```

```
public void insertAt(int index, int input){  
    if (index < 0)  
    {  
        System.out.println("indeks salah");  
    }  
    else if (index == 0)  
    {  
        addFirst(input);  
    }  
    else  
    {  
        Node temp = head;  
        for(int i = 0; i < index - 1; i++)  
        {  
            temp = temp.next;  
        }  
        temp.next = new Node(input, temp.next);  
    }  
}
```

```

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

package minggull;
public class SLLMain {
    public static void main(String[] args) {
        SingleLinkedList singLL = new SingleLinkedList();
        singLL.print();
        singLL.addFirst(890);
        singLL.print();
        singLL.addLast(760);
        singLL.print();
        singLL.addFirst(700);
        singLL.print();
        singLL.insertAfter(700, 999);
        singLL.print();
        singLL.insertAt(3, 833);
        singLL.print();
    }
}

```

### **9.3.1 Langkah-langkah Percobaan**

```

package minggull;
public class Node {
    int data;
    Node next;
    public Node(int nilai, Node berikutnya){
        this.data = nilai;
        this.next=berikutnya;
    }
}
package minggull;
public class SingleLinkedList {
    Node head; // posisi awal linked list

```

```
Node tail; //posisi akgir linked list

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

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

public void addFirst(int input){
    Node ndInput = new Node(input, null);
    if (isEmpty())
    {
        head = ndInput;
        tail = ndInput;
    }
    else
    {
        ndInput.next = head;
        head = ndInput;
    }
}
```

```

        head = ndInput;
    }
}

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

public void insertAfter(int key, int input) {
    Node ndInput = new Node(input,null);
    Node 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);
}

```

```
public void insertAt(int index, int input) {
    if (index < 0)
    {
        System.out.println("indeks salah");
    }
    else if (index == 0)
    {
        addFirst(input);
    }
    else
    {
        Node temp = head;
        for(int i = 0; i < index - 1; i++)
        {
            temp = temp.next;
        }
        temp.next = new Node(input, temp.next);
        if(temp.next.next == null) tail = temp.next;
    }
}

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

public int indexOf(int key) {
    Node tmp = head;
    int index = 0;
    while (tmp != null && tmp.data != key) {
        tmp = tmp.next;
    }
}
```

```
        index++;
    }

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

public 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 void removeLast(){
    if (isEmpty())
    {
        System.out.println("Linked List Masih Kosong,
tidak dapat dihapus");
    }
}
```

```

        }

    else if (head == tail)
    {
        head = tail = null;
    }

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

}

public void remove (int key){
    if (isEmpty())
    {
        System.out.println("Linked List Masih Kosong,
tidak dapat dihapus");
    }
    else
    {
        Node temp = head;
        while (temp != null)
        {
            if((temp.data == key) && (temp == head))
            {
                this.removeFirst();
                break;
            }
            else if (temp.next.data == 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
    {
        Node temp = head;
        for (int i = 0; i < index; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null)
        {
            tail = temp;
        }
    }
}
package minggu11;
```

```

public class SLLMain {
    public static void main(String[] args) {
        SingleLinkedList singLL = new SingleLinkedList();
        singLL.print();
        singLL.addFirst(890);
        singLL.print();
        singLL.addLast(760);
        singLL.print();
        singLL.addFirst(700);
        singLL.print();
        singLL.insertAfter(700, 999);
        singLL.print();
        singLL.insertAt(3, 833);
        singLL.print();

        System.out.println("Data pada indeks ke-
1="+singLL.getData(1));
        System.out.println("Data 3 berada pada indeks ke-
"+singLL.indexOf(760));

        singLL.remove(999);
        singLL.print();
        singLL.removeAt(0);
        singLL.print();
        singLL.removeFirst();
        singLL.print();
        singLL.removeLast();
        singLL.print();
    }
}

```

## B. OUTPUT PROGRAM

Percobaan 9.2.1

```

Output - StrukturData (run-single) ×
[ ant -f D:\Algostrkd\StrukturData -Dnb.internal.action.name=r
ggui\SLLMain.java -Drun.class=minggui.SLLMain run-single
init:
Deleting: D:\Algostrkd\StrukturData\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrkd\StrukturData\build\built-jar.properties
Compiling 1 source file to D:\Algostrkd\StrukturData\build\classes
compile-single:
run-single:
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
BUILD SUCCESSFUL (total time: 0 seconds)

```

### Percobaan 9.3.1

```

Output - StrukturData (run-single) ×
[ ant -f D:\Algostrkd\StrukturData -Dnb.internal.action.name=run.single -Djav
.java -Drun.class=minggui.SLLMain run-single
init:
Deleting: D:\Algostrkd\StrukturData\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrkd\StrukturData\build\built-jar.properties
Compiling 1 source file to D:\Algostrkd\StrukturData\build\classes
compile-single:
run-single:
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 indeks ke-1=999
Data 3 berada pada indeks ke-4
Isi Linked List:     700    890    833    760
Isi Linked List:     890    833    760
Isi Linked List:     833    760
Isi Linked List:     833
BUILD SUCCESSFUL (total time: 0 seconds)

```

## C. PENJELASAN

### 9.2.3 Pertanyaan

- Mengapa hasil compile kode program di baris pertama menghasilkan "Linked List Kosong"?

Jawab :

Karena pada SLLMain belum terdapat inputan dan akan menghasilkan "Linked List Kosong"

- Pada step 10, jelaskan kegunaan kode berikut

```
ndInput.next = temp.next;
temp.next = ndInput;
```

Jawab :

berguna untuk merubah atau memindahkan data yang awal yang berada setelah key atau temp menjadi setelah inputan keyboard. Kemudian kode temp.next = ndInput berguna untuk menempatkan inputan sehingga berada setelah key atau temp.

- Perhatikan class SingleLinkedList, pada method insertAt Jelaskan kegunaan kode berikut

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

Jawab:

kondisi yang akan dilakukan saat pengecekan apakah 2 baris dibelakang temp adalah bernilai null atau tidak semisal ya, maka pernyataan setelah if akan dijalankan, sehingga posisi tail akan berpindah ke temp.next.

### **9.3.3 Pertanyaan**

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

Jawab :

Break pada fungsi remove berfungsi untuk membuat perulangan dari while berhenti jika kondisi terpenuhi

2. Jelaskan kegunaan kode dibawah pada method remove

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

Jawab:

untuk mengecek kondisi temp.next.data==key yang artinya data setelah temp adalah key inputan, maka temp.next = temp.next.next akan dijalankan dan temp.next akan dilewati, diubah menjadi temp.next.next, maka dengan begitu data temp.next otomatis telah dihilangkan

3. Apa saja nilai kembalian yang dapat dikembalikan pada method indexOf? Jelaskan maksud masing-masing kembalian tersebut!

Jawab :

Return -1 digunakan apabila tmp bernilai null, yaitu Node yang menyimpan data bernilai null, maka akan direturn -1 karena -1 tidak melambangkan index manapun pada Linked list sedangkan return index digunakan apabila tmp tidak bernilai null dan tmp.data sama dengan key maka hal tersebut menunjukkan data yang dicari telah ditemukan, maka return dengan index bisa dijalankan.

## **D. KESIMPULAN**

### **9.2.3 Langkah Percobaan**

pada dasarnya struktur dari single linked list merupakan data yang begitu kompleks dan fungsi nya hampir mirip dengan array. Beda nya jika linked list yaitu berisi data dan pointer atau node untuk menghubungkan ke data berikutnya. dan masing-masing elemen array hanya berisi data saja. Didalam percobaan diatas terdapat beberapa method yang seperti :

- isEmpty : digunakan untuk mengecek apakah linked list ada yang kosong.

- print : digunakan untuk menampilkan isi linked list
- addFirst : digunakan untuk menambah elemen index atau data di paling depan.
- addLast : digunakan untuk menambahkan elemen index atau data di paling belakang.
- insertAfter : digunakan untuk menambahkan elemen index atau data setelah key atau kunci dari index yang dimasukkan .
- insertAt : digunakan untuk menambah elemen index.

### **9.3.3 Langkah Percobaan**

pada dasarnya struktur dari single linked list merupakan data yang begitu kompleks dan fungsi nya hampir mirip dengan array. Beda nya jika linked list yaitu berisi data dan pointer atau node untuk menghubungkan ke data berikutnya.dan masing-masing elemen array hanya berisi data saja. Didalam percobaan diatas terdapat beberapa method yang seperti :

- isEmpty : digunakan untuk mengecek apakah linked list ada yang kosong.
- print : digunakan untuk menampilkan isi linked list
- addFirst : digunakan untuk menambah elemen index atau data di paling depan.
- addLast : digunakan untuk menambahkan elemen index atau data di paling belakang.
- insertAfter : digunakan untuk menambahkan elemen index atau data setelah key atau kunci dari index yang dimasukkan .
- insertAt : digunakan untuk menambah elemen index.
- getData : digunakan untuk mengetahui posisi data index pada linked list yang sudah diisi.
- indexOf : digunakan untuk mengetahui posisi dari data berdasarkan index yang telah diisi.
- removeFirst : digunakan untuk menghapus elemen paling depan linked list
- removeLast : digunakan untuk menghapus elemen paling belakang linked list
- remove : digunakan untuk menghapus elemen index dengan mencari key.
- removeAt : digunakan untuk menghapus elemen yang terdapat pada index.

**“TUGAS”**

1. Buat method insertBefore untuk menambahkan node sebelum keyword yang diinginkan
2. Implementasikan ilustrasi Linked List Berikut. Gunakan 4 macam penambahan data yang telah dipelajari sebelumnya untuk menginputkan data
3. Buatlah Implementasi Stack berikut menggunakan Linked List
4. Buatlah implementasi program Nasabah Bank menggunakan LinkedList dengan data berupa nama, alamat dan nomor rekening nasabah
5. Implementasikan Queue pada antrian Nasabah pada nomor 4 dengan menggunakan konsep LinkedList!

**~~JAWAB~~**

### **1. Tugas1**

#### **A. Kode Program**

```

public void insertBefore(int key , int input){
    Node ndInput = new Node(input, null);
    Node temp = head;
    while (temp != null) {
        if ((temp.data == key)&&(temp == head)) {
            addFirst(input);
            break;
        } else if (temp.next.data == key) {
            ndInput.next = temp.next;
            temp.next = ndInput;
            break;
        }
        temp = temp.next;
    }
}
package minggull;
public class SLLMain {
    public static void main(String[] args) {
        SingleLinkedList singLL = new
        SingleLinkedList();
        singLL.print();
        singLL.addFirst(890);
        singLL.print();
        singLL.addLast(760);
        singLL.print();
        singLL.addFirst(700);
        singLL.print();
        singLL.insertBefore(700, 100);
        singLL.print();
        singLL.insertAfter(700, 999);
        singLL.print();
        singLL.insertAt(3, 833);
        singLL.print();
    }
}

```

#### **B. Output Program**

**Running :**

The screenshot shows the Apache NetBeans IDE interface. The left pane displays the 'Projects' tree, which includes several Java packages such as 'tugas2', 'Barang2.java', 'BarangMain2.java', 'PacMan.java', 'PacManMain.java', 'Test Packages', 'Libraries', 'Test Libraries', and 'StrukturData'. Under 'StrukturData', there are 'Source Packages' containing 'Tugas' (Node.java, SLIMain1.java, SingleLinkedList.java), 'Tugas3' (Node.java, Tugas3.java, Tugas3Main.java), 'Tugas4' (Nasabah.java, Node.java, Tugas4.java, Tugas4Main.java), and 'Tugas5' (Nasabah.java, Node.java, Tugas.java, TugasMain.java). The central pane shows the code for `SLIMain.java`, specifically the `main` method of the `minggu11` package. The right pane shows the 'Output - StrukturData (run-single)' window, which displays the command-line output of the Java application. The output includes compilation logs, a list of nodes ('Isi Linked List'), and a final message indicating a successful build.

```

public static void main(String[] args) {
    SinglyLinkedList singLL;
    singLL.print();
    singLL.addFirst(890);
    singLL.print();
    singLL.addLast(760);
    singLL.print();
    singLL.addFirst(700);
    singLL.print();
    singLL.insertBefore(700, 890);
    singLL.print();
    singLL.insertAfter(700, 760);
    singLL.print();
    singLL.insertAt(3, 833);
    singLL.print();
    System.out.println("D");
    System.out.println("I");
    singLL.remove(999);
    singLL.print();
    singLL.removeAt(0);
    singLL.print();
    singLL.removeFirst();
    singLL.print();
    singLL.removeLast();
    singLL.print();
}

```

### C. Penjelasan

merupakan elemen atau data yang ingin diselanjutnya dan inputan sendiri berfungsi sebagai data inputan yang disebelumkan dari key. Terdapat perulangan mencari data satu per satu yang sesuai dengan key yang dimaksud. Method ini akan berfungsi jika key sama dengan data selanjutnya, maka temp inputan akan menempatkan Node sebelum dari key yang ditentukan

## 2. Tugas no 2

### a. Kode Program

```

package Tugas;
public class Node {
    char data;
    Node next;
    public Node(char nilai, Node berikutnya) {
        this.data = nilai;
        this.next=berikutnya;
    }
}
package Tugas;
public class SinglyLinkedList {
    Node head; // posisi awal linked list
    Node tail; //posisi akhir linked list

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

    public void print() {
        if(!isEmpty())
    }
}

```

```

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

public void addFirst(char input) {
    Node ndInput = new Node(input, null);
    if (isEmpty())
    {
        head = ndInput;
        tail = ndInput;
    }
    else
    {
        ndInput.next = head;
        head = ndInput;
    }
}

public void addLast(char input) {
    Node ndInput = new Node(input, null);
    if (isEmpty())
    {
        head = ndInput;
        tail = ndInput;
    }
    else
    {
        tail.next=ndInput;
        tail = ndInput;
    }
}

public void insertAfter(char key, char input) {
    Node ndInput = new Node(input, null);
    Node 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);
}

public void insertBefore(char key , char input){
    Node ndInput = new Node(input,null);
    Node temp = tail;
    do{
        if (temp.data != key){
            ndInput.next = temp.next;
            temp.next=ndInput;
            if(ndInput.next != null) head =
ndInput;
            break;
        }
        temp = temp.next;
    }while (temp == null);
}

public void insertAt(int index, char input){
    if (index < 0)
    {
        System.out.println("indeks salah");
    }
    else if (index == 0)
    {
        addFirst(input);
    }
    else
    {
        Node temp = head;
        for(int i = 0; i < index - 1; i++)
        {
            temp = temp.next;
        }
        temp.next = new Node(input, temp.next);
        if(temp.next.next == null) tail =
temp.next;
    }
}
package Tugas;
public class SLLMain1 {
    public static void main(String[] args) {

```

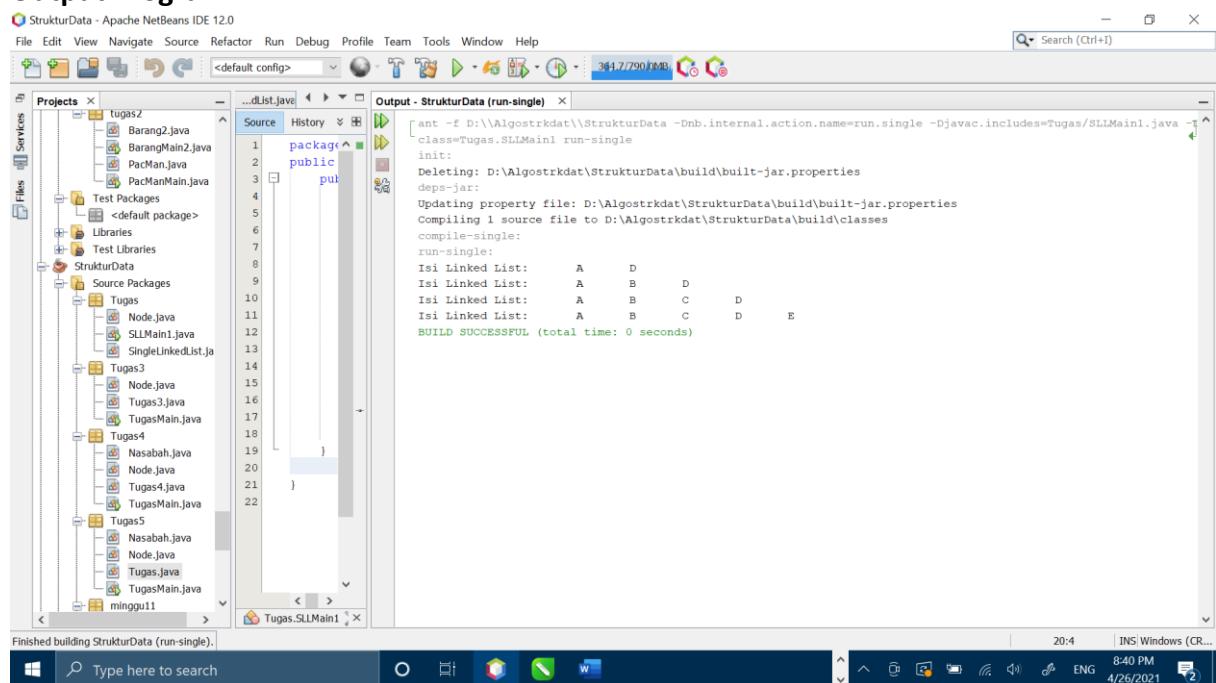
```

        SingleLinkedList singLL = new
SingleLinkedList();
        char A = 'A';
        char B = 'B';
        char C = 'C';
        char D = 'D';
        char E = 'E';
        singLL.addFirst(A);
        singLL.addLast(D);
        singLL.print();
        singLL.insertAfter(A, B);
        singLL.print();
        singLL.insertAfter(B, C);
        singLL.print();
        singLL.insertAt(4, E);
        singLL.print();
    }

}

```

## b. Output Program



## c. Penjelasan

Pada program ini saya menggunakan addFirst, addLast, insertAfter dan insertAt. method addFirst menggunakan inputan tipe data String jika kondisi awal Node kosong maka head dan tail akan sama-sama menunjuk ke node input, jika linked list telah ada Node maka next pada node input menunjuk node yang ditunjuk oleh head kemudian head akan menunjuk ke node input. method addLast menggunakan inputan tipe data String digunakan untuk operasi untuk menambah Node di akhir elemen Linked List. insertAfter memnambahkan data setelah kata kunci. insertAt menambahkan data sesuai indeks yang ditentukan.

## 3. Tugas no 3

#### A. Kode Program

```
package Tugas3;
public class Node {
    String data;
    Node next;
    public Node(String nilai, Node berikutnya) {
        this.data = nilai;
        this.next = berikutnya;
    }
}
package Tugas3;
public class Tugas3 {
    Node head;
    Node tail;
    public boolean isEmpty() {
        return head == null;
    }

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

    public void addFirst (String input){
        Node ndInput = new Node(input, null);
        if(isEmpty())
        {
            head = ndInput;
            tail = ndInput;
        }
        else
        {
            ndInput.next = head;
            head = ndInput;
        }
    }
}
```

```

public void addLast(String input){
    Node ndInput = new Node(input, null);
    if(isEmpty()){//jika Linked List Kosong
        head = ndInput;// head dan tail sama
        dengan node input
        tail = ndInput;
    } else {
        tail.next = ndInput;
        tail = ndInput;
    }
}

public void insertAfter(String key, String
input) {
    Node ndInput = new Node(input, null);
    Node 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 insertBefore(String key, String input){
    Node ndInput = new Node(input, null);
    Node temp = head;
    while (temp != null)
    {
        if((temp.data == key)&&(temp == head))
        {
            addFirst(input);
            break;
        }
        else if(temp.next.data == key)
        {
            ndInput.next = temp.next;
            temp.next = ndInput;
            break;
        }
        temp = temp.next;
    }
}

```

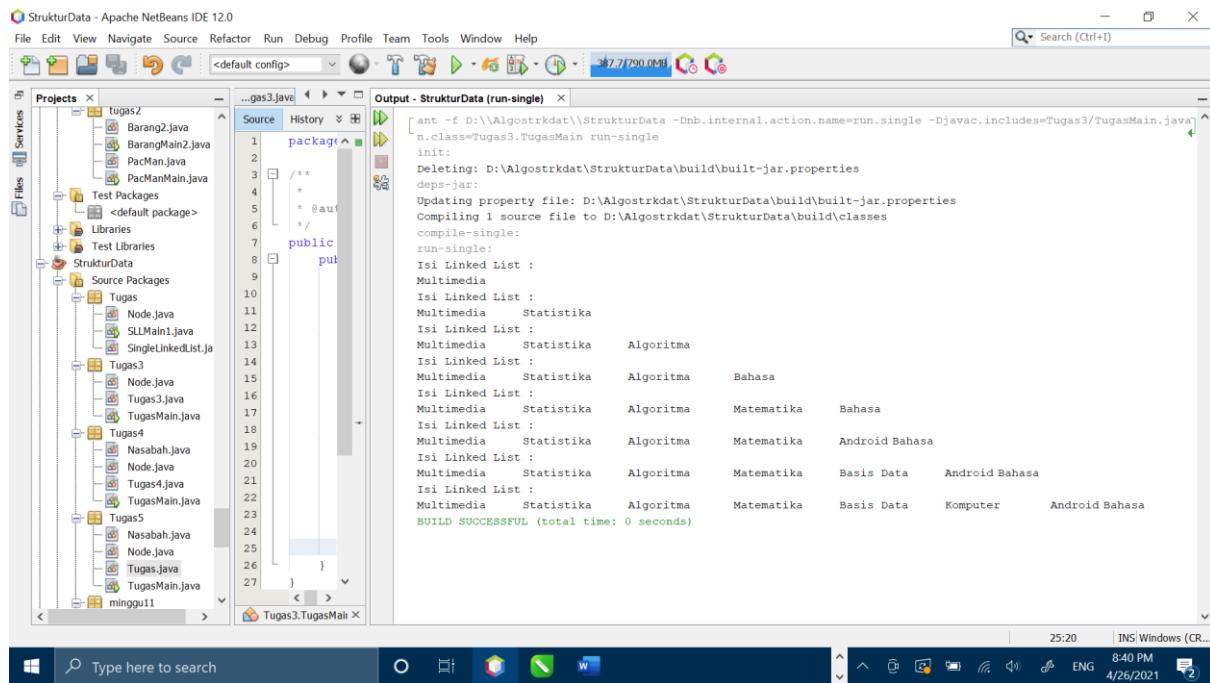
```

        public void insertAt(int index, String input) {
            if (index < 0)
            {
                System.out.println("indeks salah");
            } else if (index == 0)
            {
                addFirst(input);
            } else
            {
                Node temp = head;
                for(int i =0;i < index-1;i++)
                {
                    temp = temp.next;
                }
                temp.next = new Node(input, temp.next);
                if(temp.next.next == null)
                    tail=temp.next;
                }
            }
        package Tugas3;

    /**
     *
     * @author ACHE
     */
    public class TugasMain {
        public static void main(String[] args) {
            Tugas3 T3 = new Tugas3();
            T3.addFirst("Multimedia");
            T3.print();
            T3.insertAfter("Multimedia", "Statistika");
            T3.print();
            T3.insertAfter("Statistika", "Algoritma");
            T3.print();
            T3.addLast("Bahasa");
            T3.print();
            T3.insertAt(3, "Matematika");
            T3.print();
            T3.insertBefore("Bahasa", "Android");
            T3.print();
            T3.insertAfter("Matematika", "Basis Data");
            T3.print();
            T3.insertAt(5, "Komputer");
            T3.print();
        }
    }
}

```

## B. Output Program



```
StrukturData - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects > Services > Files > Output - StrukturData (run-single) > ...gas3.java > Source History > Package > ...
1 /**
2  * 
3  */
4 package ...
5 /**
6  * @author ACHE
7  */
8 public class ...
9 {
10 }
11 }
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
27 }

Isi Linked List :
Multimedia
Isi Linked List :
Multimedia Statistika
Isi Linked List :
Multimedia Statistika Algoritma
Isi Linked List :
Multimedia Statistika Algoritma Bahasa
Isi Linked List :
Multimedia Statistika Algoritma Matematika Bahasa
Isi Linked List :
Multimedia Statistika Algoritma Matematika Android Bahasa
Isi Linked List :
Multimedia Statistika Algoritma Matematika Basis Data Android Bahasa
Isi Linked List :
Multimedia Statistika Algoritma Matematika Komputer Android Bahasa
BUILD SUCCESSFUL (total time: 0 seconds)
```

### C. Penjelasan

Pada program ini saya menggunakan addFirst, addLast, insertAfter, insertBefore dan insertAt. method addFirst menggunakan inputan tipe data String jika kondisi awal Node kosong maka head dan tail akan sama-sama menunjuk ke node input, jika linked list telah ada Node maka next pada node input menunjuk node yang ditunjuk oleh head kemudian head akan menunjuk ke node input. method addLast menggunakan inputan tipe data String digunakan untuk operasi untuk menambah Node di akhir elemen Linked List. insertAfter memambahkan data setelah kata kunci. insertAt menambahkan data sesuai indeks yang ditentukan.insertBefore untuk menambahkan data sebelum key.

## 4. Tugas No 4

### A. Kode Program

```
package Tugas4;
public class Nasabah {
    String nama, alamat, nomor;
    Nasabah(String nama, String alamat, String nomor) {
        this.nama = nama;
        this.alamat = alamat;
        this.nomor = nomor;
    }
}
package Tugas4;

/**
 *
 * @author ACHE
 */
public class Node {
    Nasabah data;
```

```

        Node next;

        public Node(Nasabah data, Node berikutnya) {
            this.data = data;
            this.next = berikutnya;
        }

        public void tampilNasabah(){
            System.out.println("Nama \t\t:"+data.nama);
            System.out.println("Alamat \t\t :" +data.alamat);
            System.out.println("No Rekening\t:" +data.nomor);
        }

    }
}

package Tugas4;
public class Tugas4 {
    Node head;
    Node tail;

    public boolean isEmpty(){
        return head == null;
    }
    public void print(){
        if (!isEmpty())
        {
            Node tmp = head;
            System.out.println("Isi dari linked list
:");
            while (tmp != null)
            {
                tmp.tampilNasabah();
                tmp = tmp.next;
                System.out.println("-----");
            }
        } else
        {
            System.out.println("Linked List
Kosong");
        }
    }
    public void addFirst(Nasabah input){
        Node ndInput = new Node(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head; //head ditaruh
setelah inputan
            head = ndInput; // head diubah mjd
inputan
        }
    }
}

```

```

        }
    }
    public void addLast(Nasabah input){
        Node ndInput = new Node(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput; //inputan ditaruh
setelah tail
            tail = ndInput; //tail dipindah ke
inputan
        }
    }
    public void insertAt(int index, Nasabah input){
        if(index <0){
            System.out.println("Index salah");
        } else if (index==0){
            addFirst(input);
        } else {
            Node temp = head; //traverse dimulai
dari head
            for (int i = 0; i < index-1; i++) {
//mencari next dari input, karena itu index -1
                temp = temp.next; //mengubah temp
supaya berjalan
            }
            temp.next = new Node(input, temp.next);
            if (temp.next.next==null) { tail =
temp.next;
            }
            }
        }
    }
    public void removeFirst(){
        if (isEmpty()) {
            System.out.println("Linked List masih,
kosong tidak dapat dihapus!");
        } else if(head == tail){
            head = tail = null;
        } else {
            System.out.println("Antrian yang keluar
adalah ");
            head.tampilNasabah();
            head = head.next;
        }
    }
    public void removeLast(){
        if (isEmpty()) {
            System.out.println("Linked List masih,
kosong tidak dapat dihapus!");
        } else if (head == tail){

```

```

        head = tail = null;
    } else{
        Node temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        System.out.println("Antrian yang keluar
adalah ");
        temp.next.tampilNasabah();
        temp.next = null;
        tail = temp;
    }
}
public void removeAt(int index){
    if (index == 0) {
        removeFirst(); }
    else {
        Node temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        System.out.println("Antrian yang keluar
adalah ");
        temp.tampilNasabah();
        temp.next = temp.next.next;
        if (temp.next== null) {
            tail = temp;
        }
    }
}
package Tugas4;

import java.util.Scanner;

public class TugasMain {
    static void menu(){
        System.out.println("====Program Nasabah
Bank====");

        System.out.println("=====Menu=====");
        System.out.println("1. Tambahkan antrian
dari depan");
        System.out.println("2. Tambahkan antrian
daribelakang");
        System.out.println("3. Tambahkan antrian
pada urutan tertentu");
        System.out.println("4. Keluarkan antrian
dari depan");
        System.out.println("5. Keluarkan antrian
dari belakang");
    }
}

```

```

        System.out.println("6. Keluarkan antrian
pada urutan tertentu");
        System.out.println("7. Tampilkan seluruh
antrian nasabah");
        System.out.println("8. Keluar");
        System.out.print("pilih menu : ");
    }
    public static void main(String[] args) throws
Exception{
    Scanner sc = new Scanner(System.in);
    Scanner sc1 = new Scanner(System.in);
    Tugas4 t4 = new Tugas4();
    Nasabah n1 = new Nasabah("Wawan",
"Semarang", "123456");
    t4.addFirst(n1);
    Nasabah n2 = new Nasabah("Wiwin", "Malang",
"123654");
    t4.addLast(n2);
    Nasabah n3 = new Nasabah("Wuwun",
"Tulungagung", "321456");
    t4.addLast(n3);
    Nasabah n4 = new Nasabah("Wewen", "Bali",
"321654");
    t4.addLast(n4);
    Nasabah n5 = new Nasabah("Wowon",
"Surabaya", "654321");
    t4.addLast(n5);
    int pilih = 0;
    do {
        menu();
        pilih = sc.nextInt();
        switch (pilih){
            case 1 :
System.out.println("=====");
System.out.println("Masukkan
data nasabah :");
                System.out.print("Nama : ");
                String nama = sc1.nextLine();
                System.out.print("Alamat : ");
                String alamat = sc1.nextLine();
                System.out.print("No Rekening :
");
                String noRek = sc1.nextLine();
                Nasabah nsatu = new
Nasabah(nama, alamat, noRek);
                t4.addFirst(nsatu);
                System.out.println("");
                break;
            case 2 :

```

```

System.out.println("=====");
");
System.out.println("Masukkan
data nasabah :");
System.out.print("Nama : ");
String namal = sc1.nextLine();
System.out.print("Alamat : ");
String alamat1 = sc1.nextLine();
System.out.print("No Rekening :
");
String noRek1 = sc1.nextLine();
Nasabah ndua = new
Nasabah(namal, alamat1, noRek1);
t4.addLast(ndua);
System.out.println("");
break;
case 3 :

System.out.println("=====");
");
System.out.println("Masukkan
data nasabah :");
System.out.print("Nama : ");
String nama2 = sc1.nextLine();
System.out.print("Alamat : ");
String alamat2 = sc1.nextLine();
System.out.print("No Rekening :
");
String noRek2 = sc1.nextLine();
Nasabah ntiga = new
Nasabah(nama2, alamat2, noRek2);
System.out.println("Masukkan
baris yang ingin ditambahkan : ");
int posisi = sc.nextInt();
t4.insertAt(posisi, ntiga);
System.out.println("");
break;
case 4 :
t4.removeFirst();
System.out.println("Berhasil
Mengeluarkan Nasabah");
System.out.println("");
break;
case 5 :
t4.removeLast();
System.out.println("Berhasil
Mengeluarkan Nasabah");
System.out.println("");
break;
case 6 :

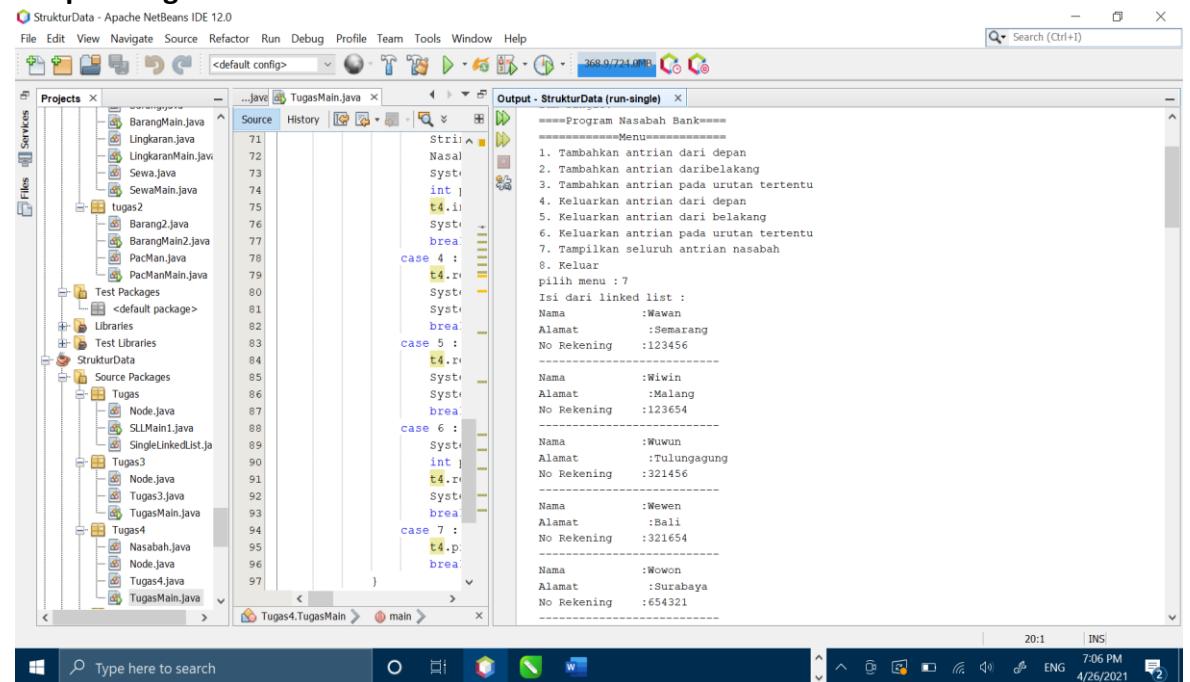
```

```

        System.out.println("Masukkan
baris yang ingin dikeluarkan : ");
        int pss = sc.nextInt();
        t4.removeAt(pss);
        System.out.println("Berhasil
Mengeluarkan Nasabah");
        break;
    case 7 :
        t4.print();
        break;
    }
} while (pilih != 8);
}
}

```

## B. Output Program



The screenshot shows the Apache NetBeans IDE interface. The left pane displays the project structure under 'Projects' and files under 'Files'. The central pane shows the code editor for 'TugasMain.java'. The right pane shows the 'Output' window for the run configuration 'run-single'. The output text is as follows:

```

=====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian dari belakang
3. Tambahkan antrian pada urutan tertentu
4. Keluaran antrian dari depan
5. Keluaran antrian dari belakang
6. Keluaran antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 7
Isi dari linked list :
Nama :Wawan
Alamat :Semarang
No Rekening :123456
-----
Nama :Wiwin
Alamat :Malang
No Rekening :123654
-----
Nama :Wuwun
Alamat :Tulungagung
No Rekening :321456
-----
Nama :Wewen
Alamat :Bali
No Rekening :321654
-----
Nama :Wowon
Alamat :Surabaya
No Rekening :654321
-----
```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```
====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 4
Antrian yang keluar adalah
Nama :Wawan
Alamat :Semarang
No Rekening :123456
Berhasil Mengeluarkan Nasabah

====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 5
Antrian yang keluar adalah
Nama :Wowon
Alamat :Surabaya
Alamat :Surabaya

20:1 INS
7:06 PM 4/26/2021
```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```
====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 5
Antrian yang keluar adalah
Nama :Wowon
Alamat :Surabaya
Alamat :Surabaya

====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 1
Masukkan data nasabah :
Nama :Bima

20:1 INS
7:06 PM 4/26/2021
```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Output - StrukturData (run-single) :

```
pilih menu : 5
Antrian yang keluar adalah
Nama :Wowon
Alamat :Surabaya
No Rekening :654321
Berhasil Mengeluarkan Nasabah

====Program Nasabah Bank====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar

pilih menu : 1
=====
Masukkan data nasabah :
Nama : Bima
Alamat : Blitar
No Rekening : 234598

====Program Nasabah Bank====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Output - StrukturData (run-single) :

```
=====
Masukkan data nasabah :
Nama : Refy
Alamat : kediri
No Rekening : 109898

====Program Nasabah Bank====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar

pilih menu : 3
=====
Masukkan data nasabah :
Nama : Pratama
Alamat : Malang
```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```

====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian dari belakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 3

Masukkan data nasabah :
Nama : Pratama
Alamat : Malang
No Rekening : 567890
Masukkan baris yang ingin ditambahkan :

4

====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian dari belakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 6

Masukkan baris yang ingin dikeluarkan :

```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```

====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian dari belakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 6

Masukkan baris yang ingin dikeluarkan :
3
Antrian yang keluar adalah
Nama :Wuwun
Alamat :Tulungagung
No Rekening :321456
Berhasil Mengeluarkan Nasabah
====Program Nasabah Bank=====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian dari belakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu : 7
Isi dari linked list :
Nama :Bima

```

```

StrukturData - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
...Java TugasMain.java <default config> S76.2/724.0MB Search (Ctrl+I)
Projects Services Files
Output - StrukturData (run-single) x
=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu :7
Isi dari linked list :
Nama :Bima
Alamat :Blitar
No Rekening :234598
-----
Nama :Wiwin
Alamat :Malang
No Rekening :123654
-----
Nama :Wuwun
Alamat :Tulungagung
No Rekening :321456
-----
Nama :Pratama
Alamat :Malang
No Rekening :567890
-----
Nama :Refy
Alamat :kediri
No Rekening :109898
=====
====Program Nasabah Bank====

```

```

StrukturData - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
...Java TugasMain.java <default config> 620.6/724.0MB Search (Ctrl+I)
Projects Services Files
Output - StrukturData (run-single) x
=====
Alamat :Blitar
No Rekening :234598
-----
Nama :Wiwin
Alamat :Malang
No Rekening :123654
-----
Nama :Wuwun
Alamat :Tulungagung
No Rekening :321456
-----
Nama :Pratama
Alamat :Malang
No Rekening :567890
-----
Nama :Refy
Alamat :kediri
No Rekening :109898
=====
====Program Nasabah Bank====
=====Menu=====
1. Tambahkan antrian dari depan
2. Tambahkan antrian daribelakang
3. Tambahkan antrian pada urutan tertentu
4. Keluarkan antrian dari depan
5. Keluarkan antrian dari belakang
6. Keluarkan antrian pada urutan tertentu
7. Tampilkan seluruh antrian nasabah
8. Keluar
pilih menu :8
BUILD SUCCESSFUL (total time: 2 minutes 10 seconds)

```

### C. Penjelasan

Pada class Nasabah terdapat 3 atribut yang bertipe data String, pada class ini juga terdapat konstruktor berparameter Nasabah. Pada class Node terdapat 2 atribut dan terdapat Node sendiri yang berisi data dan pointer, disini terdapat juga method tampilNasabah yang digunakan untuk menampilkan data nasabah itu sendiri. Pada class ini memiliki tipe data dari data adalah String yang nantinya inputan atau data pada Linked List bertipe data String, selebihnya mengenai next dan parameter Node tetap sama seperti pada soal sebelumnya. Pada class ini terdapat Node head dan tail yang mana digunakan untuk menentukan Node yang paling depan dan yang paling belakang. Terdapat method isEmpty yang bertipe data Boolean yang mana jika nilai

paling depan bernilai null maka Boolean bernilai true. Fungsi return disini digunakan untuk mengembalikan nilai ke class main. Pada method ini digunakan untuk mencetak isi Node dari head sampai tail dimana jika Linked tidak kosong maka operasi akan dijalankan. Terdapat perulangan yang digunakan untuk menginputkan data nasabah yang dimasukan melalui class main. Pada method addFirst menggunakan inputan dari data nasabah jika kondisi awal Node kosong maka head dan tail akan sama-sama menunjuk ke node input, jika linked list telah ada Node maka next pada node input menunjuk node yang ditunjuk oleh head kemudian head akan menunjuk ke node input. Pada method addLast menggunakan inputan tipe data String digunakan untuk operasi untuk menambah Node di akhir elemen Linked List. Jika kondisi awal Node kosong maka head dan tail akan sama-sama menunjuk ke Node input. Jika pada Linked List telah ada Node, maka next pada node yang ditunjuk oleh tail menunjuk ke node input kemudian tail akan menunjuk ke Node input Menginisialisasikan Linked List serta mengisikan nilai nasabah. Pada case ini terdapat penambahan data nasabah di posisi terakhir antrian menggunakan method addLast. Pada case ini menggunakan method insertAt yang digunakan untuk menempatkan data nasabah yang diinputkan, method removeFirst berguna untuk menghapuskan data nasabah pada antrian pertama, dan method removeLast digunakan untuk menghapuskan data nasabah yang berada pada antrian terakhir. Pada case 6 digunakan untuk menghapuskan data nasabah dari antrian dengan menggunakan method removeAt dan posisi yang akan dihapus melalui inputan. Case terakhir digunakan untuk menampilkan semua data nasabah.

## 5. Tugas no 5

### A. Kode Program

```
package Tugas5;

/**
 *
 * @author ACHE
 */
public class Node {
    Nasabah data;
    Node next;

    public Node(Nasabah data, Node berikutnya) {
        this.data = data;
        this.next = berikutnya;
    }

    public void tampilNasabah(){
        System.out.println("Nama \t\t:" + data.nama);
        System.out.println("Alamat \t\t :" + data.alamat);
        System.out.println("No Rekening\t:" + data.nomor);
    }
}

package Tugas5;
```

```

/**
 *
 * @author ACHE
 */
public class Nasabah {
    String nama, alamat,nomor;
    Nasabah(String nama, String alamat, String nomor) {
        this.nama = nama;
        this.alamat = alamat;
        this.nomor = nomor;
    }

}

package Tugas5;

/**
 *
 * @author ACHE
 */
public class Tugas {
    Node head;
    Node tail;
    public boolean isEmpty() {
        return head == null;
    }
    public void print(){
        if (!isEmpty()) {
            Node tmp = head;
            System.out.println("Isi dari linked list
:");
            while (tmp != null) {
                tmp.tampilNasabah();
                tmp = tmp.next; System.out.println("-----
-----");
            } else {
                System.out.println("Linked List Kosong");
            }
        }
    }
    public void addLast(Nasabah input){
        Node ndInput = new Node(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
    public void removeFirst(){
        if (isEmpty()) {

```

```

        System.out.println("Linked List
masih kosong tidak dapat dihapus!");
    } else if(head == tail){
        head = tail = null;
    } else {
        System.out.println("Antrian yang keluar
adalah ");
        head.tampilNasabah();
        head = head.next;
    }
}
public void peek(){
    System.out.println("Antrian yang berada pada
posisi terdepan adalah :");
    head.tampilNasabah(); }
public void clear(){
head = tail = null;
System.out.println("Linked List telah
dikosongkan");
}

}
package Tugas5;

import java.util.Scanner;

/**
 *
 * @author ACHE
 */
public class TugasMain {

    static void menu(){
        System.out.println("=====Menu=====");
        System.out.println("1. Tambahkan antrian
Nasabah");
        System.out.println("2. Tampilkan antrian
terdepan");
        System.out.println("3. Tampilkan seluruh data
nasabah");
        System.out.println("4. Keluarkan antrian dari
nasabah");
        System.out.println("5. Hapus seluruh data
nasabah");
        System.out.println("6. Keluar");
        System.out.print("pilih menu : ");
    }
    public static void main(String[] args) throws
Exception{
    Scanner sc = new Scanner(System.in);
    Scanner sc1 = new Scanner(System.in);

```

```

        Tugas t4 = new Tugas();
        Nasabah n1 = new Nasabah("Siti", "Semarang",
"123456");
        t4.addLast(n1);
        Nasabah n2 = new Nasabah("Pratama", "Malang",
"123654");
        t4.addLast(n2);
        Nasabah n3 = new Nasabah("Viko", "Tulungagung",
"321456");
        t4.addLast(n3);
        Nasabah n4 = new Nasabah("Made", "Bali",
"321654");
        t4.addLast(n4);
        Nasabah n5 = new Nasabah("Ujang", "Bandung",
"654321");
        t4.addLast(n5);
        int pilih = 0;
        do {
            menu();
            pilih = sc.nextInt();
            switch (pilih) {
                case 1 :
                    System.out.println("=====");
                    System.out.println("Masukkan data
nasabah :");
                    System.out.print("Nama : ");
                    String nama = sc1.nextLine();
                    System.out.print("Alamat : ");
                    String alamat = sc1.nextLine();
                    System.out.print("No Rekening : ");
                    String noRek = sc1.nextLine();
                    Nasabah nsatu = new Nasabah(nama,
alamat, noRek);
                    t4.addLast(nsatu);
                    System.out.println("");
                    break;

                case 2 :
                    t4.peek(); System.out.println("");
                    break;
                case 3 :
                    t4.print();
                    System.out.println("");
                    break;
                case 4 :
                    t4.removeFirst();
                    System.out.println("");
                    break;
                case 5 :
                    t4.clear();

```

```

        System.out.println("");
        break;
    case 6 :
        break;
    default :
        System.out.println("Kode yang anda
masukkan salah");
        break;
    }
} while (pilih != 6);
}
}

```

## B. Output Program

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```

ant -f D:\Algostrkd\StrukturData -Dnb.internal.action.name=run.single -Djavac.inch
seTugas/TugasMain.java -Drun.class=Tugas5.TugasMain run-single
init:
Deleting: D:\Algostrkd\StrukturData\build\built-jar.properties
Updating property file: D:\Algostrkd\StrukturData\build\built-jar.properties
Compiling 1 source file to D:\Algostrkd\StrukturData\build\classes
compile-single:
run-single:
=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 1
=====
Masukkan data nasabah :
Nama : Daffa
Alamat : Nganjuk
No Rekening : 999811

=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 2
Antrian yang berada pada posisi terdepan adalah :

```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```

Type here to search
O W
32:42 | INS
8:20 PM
4/26/2021
| Search (Ctrl+I)

Output - StrukturData (run-single) x
=====
Masukkan data nasabah :
Nama : Daffa
Alamat : Nganjuk
No Rekening : 999811

=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 2
Antrian yang berada pada posisi terdepan adalah :
Nama : Siti
Alamat : Semarang
No Rekening : 123456

=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 3
Isi dari linked list :
Nama : Siti
Alamat : Semarang
No Rekening : 123456

```

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```
=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 3
Isi dari linked list :
Nama :Siti
Alamat :Semarang
No Rekening :123456
-----
Nama :Pratama
Alamat :Malang
No Rekening :123654
-----
Nama :Viko
Alamat :Tulungagung
No Rekening :321456
-----
Nama :Made
Alamat :Bali
No Rekening :321654
-----
Nama :Ujang
Alamat :Bandung
No Rekening :654321
-----
Nama :Daffa
Alamat :Nganjuk
No Rekening :999811
```

Type here to search

StrukturData - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - StrukturData (run-single) x

```
=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 4
Antrian yang keluar adalah
Nama :Siti
Alamat :Semarang
No Rekening :123456
-----
=====Menu=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 5
Linked List telah dikosongkan
=====
1. Tambahkan antrian Nasabah
2. Tampilkan antrian terdepan
3. Tampilkan seluruh data nasabah
4. Keluarkan antrian dari nasabah
5. Hapus seluruh data nasabah
6. Keluar
pilih menu : 6
BUILD SUCCESSFUL (total time: 1 minute 2 seconds)
```

```

StrukturData - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
...Java TugasMain.java <default config>
Projects Services Files <default package> StrukturData Source Packages Tugas Main.java
Barang2.java BarangMain2.java PacMan.java PacManMain.java
Test Packages <default package> Libraries Test Libraries
18. System.out.println("1. ^");
19. System.out.println("2.");
20. System.out.println("3.");
21. System.out.println("4.");
22. System.out.println("5.");
23. System.out.println("6.");
24. System.out.print("pilih : ");
25. }
26. public static void main(String[] args) {
27. Scanner sc = new Scanner(System.in);
28. Scanner scl = new Scanner(System.in);
29. Tugas t4 = new Tugas();
30. Nasabah n1 = new Nasabah();
31. t4.addlast(n1);
32. Nasabah n2 = new Nasabah();
33. t4.addlast(n2);
34. Nasabah n3 = new Nasabah();
35. t4.addlast(n3);
36. Nasabah n4 = new Nasabah();
37. t4.addlast(n4);
38. Nasabah n5 = new Nasabah();
39. t4.addlast(n5);
40. int pilih = 0;
41. do {
42.     menu();
43.     pilih = sc.nextInt();
        switch (pilih) {
    }
}
44.     pilih menu : 4
Antrian yang keluar adalah
Nama :Siti
Alamat :Semarang
No Rekening :123456
45. =====Menu=====
46. 1. Tambahkan antrian Nasabah
47. 2. Tampilkan antrian terdepan
48. 3. Tampilkan seluruh data nasabah
49. 4. Keluarkan antrian dari nasabah
50. 5. Hapus seluruh data nasabah
51. 6. Keluar
52. pilih menu : 5
Linked List telah dikosongkan
53. =====Menu=====
54. 1. Tambahkan antrian Nasabah
55. 2. Tampilkan antrian terdepan
56. 3. Tampilkan seluruh data nasabah
57. 4. Keluarkan antrian dari nasabah
58. 5. Hapus seluruh data nasabah
59. 6. Keluar
60. pilih menu : 6
BUILD SUCCESSFUL (total time: 1 minute 2 seconds)

```

### C. Penjelasan

Pada class ini memiliki tipe data dari data adalah String yang nantinya inputan atau data pada Linked List bertipe data String, selebihnya mengenai next dan parameter Node tetap sama seperti pada soal sebelumnya. Pada class ini terdapat Node head dan tail yang mana digunakan untuk menentukan Node yang paling depan dan yang paling belakang. Terdapat method isEmpty yang bertipe data Boolean yang mana jika nilai paling depan bernilai null maka Boolean bernilai true. Fungsi return disini digunakan untuk mengembalikan nilai ke class main. Pada method ini digunakan untuk mencetak isi Node dari head sampai tail dimana jika Linked tidak kosong maka operasi akan dijalankan. Terdapat perulangan yang digunakan untuk menginputkan data nasabah yang dimasukan melalui class main. Pada method addLast digunakan untuk menambahkan data ke barisan paling belakang antrian dengan cara memindahkan data dari temp ke inputan. Pada method removeFirst digunakan untuk menghapuskan data antrian paling depan. Pada method peek digunakan untuk menampilkan data barisan paling depan. Pada method clear digunakan untuk menghapuskan semua data antrian nasabah dengan cara membuat head dan tail bernilai null. Terdapat perulangan do-while yang digunakan untuk mengembalikan menu, method-method tadi akan ditempatkan pada case case tertentu, pada case pertama terdapat inputan yang digunakan untuk data nasabah, terdapat metho addLast yang digunakan untuk menambahkan data. Pada case kedua digunakan untuk menampilkan antrian paling depan dengan menggunakan method peek. Pada case ketiga digunakan untuk menampilkan seluruh data nasabah dengan menggunakan method print. Pada case keempat digunakan untuk menghapuskan antrian paling depan dengan method removeFirst. Pada case kelima digunakan untuk menghapus selurut data nasabah pada antrian dengan menggunakan method clear. Pada method 6 digunakan untuk keluar dari perulangan menu