

LAPORAN PRAKTIKUM MINGGU KE-12
“[Double Linked Lists]”



Disusun oleh:

[Daffa Aqila Rahmatullah]
[2041720098]

D4 TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2021

LAPORAN
“PRAKTIKUM”

1. KODE PROGRAM

12.2.1 Percobaan 1

```
package doublelinkedlists;  
/**...4 lines */  
public class Node {  
    int data;  
    Node prev, next;  
    Node(Node prev, int data, Node next){  
        this.prev = prev;  
        this.data = data;  
        this.next = next;  
    }  
}  
package doublelinkedlists;  
/**...4 lines */  
public class Doublelinkedlist {  
    Node head;  
    int size;  
  
    Doublelinkedlist(){  
        head = null;  
        size = 0;  
    }  
  
    boolean isEmpty(){  
        return head == null;  
    }  
  
    void addFirst(int item){  
        if(isEmpty()){
```

```

        head = new Node(null, item, null);
    }else{
        Node newNode = new Node(null, item, head);
        head.prev = newNode;
        head = newNode;
    }
    size++;
}

void addLast(int item) {
    if(isEmpty()){
        addFirst(item);
    }else{
        Node current = head;
        while(current.next != null){
            current = current.next;
        }
        Node newNode = new Node(current, item, null);
        current.next = newNode;
        size++;
    }
}

void add(int item, int index) throws Exception{
    if(isEmpty()){
        addFirst(item);
    }else if(index < 0 || index > size){
        throw new Exception("Nilai index diluar batas");
    }else{
        Node current = head;
        int i = 0;
        while(i<index){
            current = current.next;
            i++;
        }

```

```

        }if(current.prev == null){

            Node newNode = new Node(null, item,
current);

            current.prev = newNode;

            head = newNode;

        }else{

            Node newNode = new Node(current.prev, item,
current);

            newNode.prev = current.prev;

            newNode.next = current;

            newNode.prev.next = newNode;

            current.prev = newNode;

        }

    }

    size++;

}

int size(){

    return size;

}

void clear(){

    head = null;

    size = 0;

}

void print(){

    if(!isEmpty()){

        Node tmp = head;

        while(tmp != null){

            System.out.print(tmp.data+"\t");

            tmp = tmp.next;

        }

        System.out.println("\nBerhasil diisi");

    }

}

```

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

package doublelinkedlists;
import java.util.*;
public class DoubleLinkedListMain {

    public static void main(String[] args) throws Exception {
        Doublelinkedlist dll = new Doublelinkedlist();
        dll.print();
        System.out.println("size : "+dll.size());

        System.out.println("===== =====");
        dll.addFirst(3);
        dll.addLast(4);
        dll.addFirst(7);
        dll.print();
        System.out.println("Size : "+dll.size());

        System.out.println("===== =====");
        dll.add(40, 1);
        dll.print();
        System.out.println("Size : "+dll.size());

        System.out.println("===== =====");
        dll.clear();
        dll.print();
        System.out.println("Size : "+dll.size());
    }
}
```

```
        System.out.println("=====");  
    }  
}
```

12.3.1 Tahapan Percobaan

```
package Praktikum2;  
  
public class Node {  
  
    int data;  
  
    Node prev, next;  
  
    Node(Node prev, int data, Node next){  
        this.prev = prev;  
        this.data = data;  
        this.next = next;  
    }  
}  
  
package Praktikum2;  
  
public class DoubleLinkedList {  
  
    Node head;  
  
    int size;  
  
    DoubleLinkedList(){  
        head = null;  
        size = 0;  
    }  
  
    boolean isEmpty(){  
        return head == null;  
    }  
  
    void addFirst(int item){  
        if(isEmpty()){  
            head = new Node(null, item, null);  
        }  
    }  
}
```

```

    }else{
        Node newNode = new Node(null, item, head);
        head.prev = newNode;
        head = newNode;
    }
    size++;
}

void addLast(int item) {
    if(isEmpty()){
        addFirst(item);
    }else{
        Node current = head;
        while(current.next != null){
            current = current.next;
        }
        Node newNode = new Node(current, item, null);
        current.next = newNode;
        size++;
    }
}

void add(int item, int index) throws Exception{
    if(isEmpty()){
        addFirst(item);
    }else if(index < 0 || index > size){
        throw new Exception("Nilai index diluar batas");
    }else{
        Node current = head;
        int i = 0;
        while(i<index){
            current = current.next;
            i++;
        }
        if(current.prev == null){

```

```

        Node newNode = new Node(null, item,
current);

        current.prev = newNode;
        head = newNode;
    }else{
        Node newNode = new Node(current.prev, item,
current);

        newNode.prev = current.prev;
        newNode.next = current;
        newNode.prev.next = newNode;
        current.prev = newNode;
    }
    size++;
}

int size(){
    return size;
}

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

void print(){
    if(!isEmpty()){
        Node tmp = head;
        while(tmp != null){
            System.out.print(tmp.data+"\t");
            tmp = tmp.next;
        }
        System.out.println("\nBerhasil diisi");
    }else{

```

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

}

void removeFirst() throws Exception{
    if(isEmpty()) {
        throw new Exception("Linked List masih kosong,
Tidak dapat dihapus");
    }else if(size == 1) {
        removeLast();
    }else{
        head = head.next;
        head.prev = null;
        size--;
    }
}

void removeLast() throws Exception{
    if(isEmpty()) {
        throw new Exception("Nilai Index diluar batas");
    }else if(head.next == null) {
        head = null;
        size--;
        return;
    }
    Node current = head;
    while(current.next.next != null){
        current = current.next;
    }
    current.next = null;
    size--;
}

void remove(int index) throws Exception{
```

```
        if(isEmpty() || index >= size) {
            throw new Exception("Nilai indeks diluar
batas");
        }else if(index == 0) {
            removeFirst();
        }else{
            Node current = head;
            int i = 0;
            while(i<index) {
                current = current.next;
                i++;
            }
            if(current.next == null){
                current.prev.next = null;
            }else if(current.prev == null){
                current = current.next;
                current.prev = null;
                head = current;
            }else{
                current.prev.next = current.next;
                current.next.prev = current.prev;
            }
            size--;
        }
    }

package Praktikum2;
import java.util.*;
public class DoubleLinkedListMain {
    public static void main(String[] args) throws Exception
    {
        DoubleLinkedList dll = new DoubleLinkedList();
        dll.print();

System.out.println("=====
=====");
```

```

        dll.addLast(50);
        dll.addLast(40);
        dll.addLast(10);
        dll.addLast(20);
        dll.print();
        System.out.println("Size : "+dll.size());

System.out.println("=====");
=====
        dll.removeFirst();
        dll.print();
        System.out.println("Size : "+dll.size());

System.out.println("=====");
=====
        dll.removeLast();
        dll.print();
        System.out.println("Size : "+dll.size());

System.out.println("=====");
=====
        dll.remove(1);
        dll.print();
        System.out.println("Size : "+dll.size());

System.out.println("=====");
=====
    }
}

```

12.4.1 Tahapan Percobaan

```

package Praktikum3;
public class Node {
    int data;
    Node prev, next;

```

```
Node(Node prev, int data, Node next) {
    this.prev = prev;
    this.data = data;
    this.next = next;
}
}

package Praktikum3;
public class DoubleLinkedList {
    Node head;
    int size;

    DoubleLinkedList() {
        head = null;
        size = 0;
    }

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

    void addFirst(int item) {
        if(isEmpty()) {
            head = new Node(null, item, null);
        }else{
            Node newNode = new Node(null, item, head);
            head.prev = newNode;
            head = newNode;
        }
        size++;
    }

    void addLast(int item) {
        if(isEmpty()) {
```

```

        addFirst(item);

    }else{
        Node current = head;
        while(current.next != null) {
            current = current.next;
        }
        Node newNode = new Node(current, item, null);
        current.next = newNode;
        size++;
    }
}

void add(int item, int index) throws Exception{
    if(isEmpty()){
        addFirst(item);
    }else if(index < 0 || index > size){
        throw new Exception("Nilai index diluar batas");
    }else{
        Node current = head;
        int i = 0;
        while(i<index) {
            current = current.next;
            i++;
        }
        if(current.prev == null){
            Node newNode = new Node(null, item,
current);
            current.prev = newNode;
            head = newNode;
        }else{
            Node newNode = new Node(current.prev, item,
current);
            newNode.prev = current.prev;
            newNode.next = current;
            newNode.prev.next = newNode;
        }
    }
}

```

```

        current.prev = newNode;
    }
}

size++;

}

int size() {
    return size;
}

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

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

void removeFirst()throws Exception{
    if(isEmpty()){
        throw new Exception("Linked List masih kosong,
Tidak dapat dihapus");
    }else if(size == 1){
        removeLast();
    }
}

```

```

    }else{
        head = head.next;
        head.prev = null;
        size--;
    }
}

void removeLast() throws Exception{
    if(isEmpty()){
        throw new Exception("Nilai Index diluar batas");
    }else if(head.next == null){
        head = null;
        size--;
        return;
    }
    Node current = head;
    while(current.next.next != null){
        current = current.next;
    }
    current.next = null;
    size--;
}

void remove(int index) throws Exception{
    if(isEmpty() || index >= size){
        throw new Exception("Nilai indeks diluar batas");
    }else if(index == 0){
        removeFirst();
    }else{
        Node current = head;
        int i = 0;
        while(i<index){
            current = current.next;
        }
    }
}

```

```
        i++;

    }if(current.next == null){

        current.prev.next = null;

    }else if(current.prev == null){

        current = current.next;

        current.prev = null;

        head = current;

    }else{

        current.prev.next = current.next;

        current.next.prev = current.prev;

    }

}

}

int getFirst()throws Exception{

if(isEmpty()){

    throw new Exception("Linked List Kosong");

}

return head.data;

}

int getLast()throws Exception{

if(isEmpty()){

    throw new Exception("Linked List Kosong");

}

Node tmp = head;

while(tmp.next != null){

    tmp = tmp.next;

}

return tmp.data;

}

int get(int index)throws Exception{

if(isEmpty()){


```

```
        throw new Exception("Nilai indeks di luar
batas");
    }

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

package Praktikum3;

public class DoubleLinkedListMain {
    public static void main(String[] args) throws Exception
{
    DoubleLinkedList dll = new DoubleLinkedList();
    dll.print();

    System.out.println("===== =====");
    dll.addFirst(3);
    dll.addLast(4);
    dll.addFirst(7);
    dll.print();
    System.out.println("Size : "+dll.size());

    System.out.println("===== =====");
    dll.add(40, 1);
    dll.print();
    System.out.println("Size : "+dll.size());

    System.out.println("===== =====");
}
```

```

        System.out.println("Data awal dari linked list
adalah : "+dll.getFirst());
        System.out.println("Data akhir dari linked list
adalah : "+dll.getLast());
        System.out.println("Data indeks ke-1 pada linked
list adalah : "+dll.get(1));

System.out.println("=====
=====");
}

}

```

2. OUTPUT PROGRAM

12.2.1 Percobaan 1

```

Output - Minggu12 (run-single) ×
ant -f D:\Algostrkdat\Minggu12 -Dnb.internal.action.name=run.single -Djavac
    -blelinedListMain.java -Drun.class=doublelinkedlists.DoubleLinkedListMain run
init:
Deleting: D:\Algostrkdat\Minggu12\build\built-Jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\Minggu12\build\built-jar.properties
compiling 1 source file to D:\Algostrkdat\Minggu12\build\classes
compile-single:
run-single:
Linked List Kosong
size : 0
=====
7      3      4
Berhasil diisi
Size : 3
=====
7      40     3      4
Berhasil diisi
Size : 4
=====
Linked List Kosong
size : 0
=====
BUILD SUCCESSFUL (total time: 2 seconds)

```

12.3.1 Tahapan Percobaan

```

Output - Minggu12 (run-single) ×
ant -f D:\Algostrkdat\Minggu12 -Dnb.internal.action.name=run.single -Djavac
    .inc
stMain.java -Drun.class=Praktikum2.DoubleLinkedListMain run-single
init:
Deleting: D:\Algostrkdat\Minggu12\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\Minggu12\build\built-jar.properties
Compiling 1 source file to D:\Algostrkdat\Minggu12\build\classes
compile-single:
run-single:
Linked List Kosong
=====
50      40      10      20
Berhasil diisi
Size : 4
=====
40      10      20
Berhasil diisi
Size : 3
=====
40      10
Berhasil diisi
Size : 2
=====
40
Berhasil diisi
Size : 1
=====
BUILD SUCCESSFUL (total time: 1 second)

```

12.4.1 Tahapan Percobaan

```

Output - Minggu12 (run-single) x
ant -f D:\\Algostrkdat\\Minggu12 -Dnb.internal.action.name=run_single -Djavac
eLinkedListMain.java -Drun.class=Praktikum3.DoubleLinkedListMain run-single
init:
Deleting: D:\\Algostrkdat\\Minggu12\\build\\built-jar.properties
deps-jar:
Updating property file: D:\\Algostrkdat\\Minggu12\\build\\built-jar.properties
Compiling 1 source file to D:\\Algostrkdat\\Minggu12\\Build\\classes
compile-single:
run-single:
Linked List Kosong
=====
7      3      4
Berhasil diisi
Size : 3
=====
7      40     3      4
Berhasil diisi
Size : 4
=====
Data awal dari linked list adalah : 7
Data akhir dari linked list adalah : 4
Data indeks ke-1 pada linked list adalah : 40
=====
BUILD SUCCESSFUL (total time: 1 second)

```

3. PENJELASAN

9.2.3 Pertanyaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!

Jawab :

Pada Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya, biasanya field pada tail menunjuk ke null. Sedangkan pada Double Linked List merupakan suatu linked list yang memiliki dua variabel pointer yaitu pointer yang menunjuk ke node selanjutnya dan pointer yang menunjuk ke node sebelumnya. Setiap head dan tailnya juga menunjuk ke null.

2. Perhatikan class Node, didalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?

Jawab :

Next adalah variabel yang digunakan untuk menandai node sesudahnya . Sedangkan Prev adalah variabel yang digunakan untuk menandai node sebelumnya.

3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan inisialisasi atribut head dan size seperti pada gambar berikut ini?

```

public DoubleLinkedLists() {
    head = null;
    size = 0;
}

```

Jawab:

head=null berarti list tersebut masih kosong , sedangkan size =0 berarti list tersebut tidak mempunyai elemen.

4. Pada method addFirst(), kenapa dalam pembuatan object dari konstruktor class Node prev dianggap sama dengan null?

```
Node newNode = new Node(null, item, head);
```

Jawab:

Pada method tersebut elemen ditambahkan pada urutan paling depan karena sehingga didepannya tidak ada elemen lagi maka diarahkan di null.

5. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?

Jawab:

berarti bahwa head.prev akan diisi dengan newNode.

6. Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?

```
Node newNode = new Node(current, item, null);
```

Jawab:

Item prev dihubungkan dengan current berfungsi agar menghubungkan elemen yang ditambahkan di baris terakhir dengan elemen sebelumnya, yang awalnya current tersebut berada di baris terakhir. Sedangkan item dihubungkan dengan null(next) berarti bahwa setelah item tidak ada elemen lagi(item terakhir).

9.3.3 Pertanyaan

1. Apakah maksud statement berikut pada method removeFirst()?

```
head = head.next;  
head.prev = null;
```

Jawab :

head=head.next; berfungsi untuk menghubungkan posisi head dengan elemen yang ada dibelakangnya. Sedangkan head.prev= null berfungsi untuk menghubungkan head yang saat ini dengan null didepannya.

2. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method removeLast()?

Jawab:

dengan cara metode traverse atau perulangan while menggunakan variabel current yang dibandingkan apakah memenuhi kondisi dalam perulangan. Jika tidak maka akan keluar dari perulangan.

3. Jelaskan alasan potongan kode program di bawah ini tidak cocok untuk perintah remove!

```
Node tmp = head.next;  
head.next=tmp.next;  
tmp.next.prev=head;
```

Jawab :

Karena pada method remove harus terdapat index data yang akan dihapus, sedangkan script tersebut tidak terdapat index dari manapun.

- 4 . Jelaskan fungsi kode program berikut ini pada fungsi remove!

```
current.prev.next = current.next;  
current.next.prev = current.prev;
```

Jawab:

current.prev.next= current.next; berfungsi sebagai penghubung langsung dari elemen sebelum current dengan setelahnya dengan tidak menyertakan current itu sendiri. Sedangkan current.next.prev = current.prev; berfungsi untuk menghubungkan langsung dari elemen setelah setelah current dengan sebelumnya dengan tidak menyertakan current itu sendiri.

12.4.3 Pertanyaan Percobaan

- 1 . Jelaskan method size() pada class DoubleLinkedLists!

Jawab:

size() berfungsi untuk mengetahui panjang atau jumlah elemen dalam list.

- 2 . Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke-1!

Jawab:

cara mengatur double linked list agar dapat dimulai dari indeks ke-1 adalah dengan mengganti nilai awal dari perulangan pada method remove dan add dengan angka

1. contoh:

```
For(int i=1 ; i< index ;i++ )
```

- 3 . Jelaskan perbedaan karakteristik fungsi Add pada Double Linked Lists dan Single Linked Lists!

Jawab:

Pada single linked list, add dilakukan hanya di pointer next untuk mengarahkan ke elemen yang diinginkan. Sedangkan pada double linked list, add dilakukan pada pointer next dan prev untuk mengarahkan ke elemen yang diinginkan.

4 . Jelaskan perbedaan logika dari kedua kode program di bawah ini!

```
public boolean isEmpty(){
    if(size == 0){
        return true;
    } else{
        return false;
    }
}
```

(a)

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

(b)

Jawab:

a.) Menggunakan logika percabangan(if else) dengan acuan variabel size, jika size==0 maka list tersebut kosong dan akan memberi return true. Sedangkan jika size tidak 0 maka akan memberi return false atau tidak kosong.

b.) Menggunakan logika langsung kepada return dengan pengecekan head == null. Jika iya maka akan menghasilkan return true dan sebaliknya.

7. KESIMPULAN

Double Linked Lists memiliki dua buah pointer yaitu pointer next dan prev. Pointer next menunjuk pada node setelahnya dan pointer prev menunjuk pada node sebelumnya. Double memilki arti field pointer-nya dua buah dan dua arah, ke node sebelum dan sesudahnya. Berbeda dengan single linked lists yang hanya memiliki satu pointer next saja, double linked lists memiliki dua pointer (prev dan next) pada setiap Node. Oleh karena itu, atribut yang dimiliki terdiri dari tiga yaitu Data, next node, dan previous node.

“TUGAS”

- Buatlah sebuah program menggunakan double linked lists dengan pilihan menu sesuai dengan ilustrasi di bawah ini! Fitur pencarian harus menggunakan sequential search dan fitur pengurutan secara Descending mengimplementasikan (silakan pilih satu): bubble sort, selection sort, insertion sort, atau merge sort.

Contoh ilustrasi:

Menu awal dan tambah data

<pre>===== PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLY LINKED LIST ===== 1. Tambah head 2. Tambah tail 3. Tambah Data 4. Hapus Data Pertama 5. Hapus Data Terakhir 6. Hapus Data Tertentu 7. Cetak 8. Cari 9. Urut Data 10. Keluar =====</pre>	<pre>===== PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLY LINKED LIST ===== 1. Tambah head 2. Tambah tail 3. Tambah Data 4. Hapus Data Pertama 5. Hapus Data Terakhir 6. Hapus Data Tertentu 7. Cetak 8. Cari 9. Urut Data 10. Keluar ===== 1 -Masukkan Data Posisi Head 34</pre>
---	---

```

Tambah data di node posisi tertentu dan cetak data
=====
PROGRAM PENDALAMAN ANGKA DENGAN DOUBLY LINKED LIST
=====
1. Tambah Head
2. Tambah Tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak Data
8. Cari Data
9. Simpan Data
10. Keluar
=====
3
Masukkan Data
Data node:
10
Alamat pointer:
1
=====
Pencarian data
=====
PROGRAM PENDALAMAN ANGKA DENGAN DOUBLY LINKED LIST
=====
1. Tambah Head
2. Tambah Tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak Data
8. Cari Data
9. Simpan Data
10. Keluar
=====
8
Cari Data
Masukkan nilai yang dicari:
10
Data di depan di node ber:
1
=====
Pengurutan data
=====
PROGRAM PENDALAMAN ANGKA DENGAN DOUBLY LINKED LIST
=====
1. Tambah Head
2. Tambah Tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak Data
8. Cari Data
9. Simpan Data
10. Keluar
=====
9

```

2. Implementasikan stack menggunakan double linked list dengan contoh kasus tumpukan buku perpustakaan sesuai dengan fitur-fitur yang ditunjukkan pada gambar di bawah ini!

Ilustrasi Program

Menu Awal dan Tambah Data (Push)

```

*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Hapus Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
*****
1
Masukkan Judul Buku
Practical Digital Forensics

```

Cetak Semua Data

```

*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Hapus Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
*****
4
Cetak Semua Judul Buku
3D Computer Vision
Understanding Software Underlying Notes For Professionals Getting Started with C/C++ And Programming for Game Development
Practical Digital Forensics
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Hapus Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
*****
3
Cetak Semua Judul Buku
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Hapus Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
*****
2
Buku pada tumpukan teratas telah diambil

```

3. Buat program antrian vaksinasi menggunakan queue berbasis double linked list sesuai ilustrasi dan menu di bawah ini! (counter jumlah antrian tersisa di menu cetak(3) dan data orang yang telah divaksinasi di menu Hapus Data(2) harus ada).

Ilustrasi Program

Menu Antri dan Penambahan Data

```

*****
PERAWATAN VAKSIN COVID19
*****
1. Tambah Data Penambahan Vaksin
2. Hapus Data Penambahan Vaksin
3. Daftar Penambahan Vaksin
4. Keluar
*****
3
*****
Data Penambahan Vaksin
*****
1. Tambah Data Penambahan Vaksin
2. Hapus Data Penambahan Vaksin
3. Daftar Penambahan Vaksin
4. Keluar
*****
2
*****
Hapus Data (Komponen di area merah harus ada)
*****
1. Hapus Data Penambahan Vaksin
2. Keluar
*****
1
*****
Data Penambahan Vaksin
*****
1. Tambah Data Penambahan Vaksin
2. Hapus Data Penambahan Vaksin
3. Daftar Penambahan Vaksin
4. Keluar
*****
3
*****
Hapus Data (Komponen di area merah harus ada)
*****
1. Hapus Data Penambahan Vaksin
2. Keluar
*****
1

```

4. Buatlah implementasi program daftar nilai mahasiswa yang terdiri dari nim, nama dan ipk menggunakan double linked lists, bentuk program memiliki fitur seperti

halnya nomor 1 dengan fitur pencarian melalui NIM dan pengurutan IPK secara descending. Class Mahasiswa wajib diimplementasikan dalam soal ini.



~~JAWAB~~

1. Tugas1

A. Kode Program

```
package Tugas1;
public class Node {
    int data;
    Node prev;
    Node next;

    public Node(Node prev, int data, Node next) {
        this.prev = prev;
        this.data = data;
        this.next = next;
    }
}
package Tugas1;
public class DataServices {
    Node head;
    int size;

    DataServices() {
        head = null;
        size = 0;
    }

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

```

void addFirst(int item) {
    if(isEmpty()) {
        head = new Node(null, item, null);
    }else{
        Node newNode = new Node(null, item, head);
        head.prev = newNode;
        head = newNode;
    }
    size++;
}

void addLast(int item) {
    if(isEmpty()) {
        addFirst(item);
    }else{
        Node current = head;
        while(current.next != null){
            current = current.next;
        }
        Node newNode = new Node(current, item,
null);
        current.next = newNode;
        size++;
    }
}

void add(int item, int index) throws Exception{
    if(isEmpty()){
        addFirst(item);
    }else if(index < 0 || index > size){
        throw new Exception("Nilai index diluar
batas");
    }else{
        Node current = head;
        int i = 0;
        while(i<index) {
            current = current.next;
            i++;
        }
        if(current.prev == null){
            Node newNode = new Node(null, item,
current);
            current.prev = newNode;
            head = newNode;
        }else{
            Node newNode = new Node(current.prev,
item, current);
            newNode.prev = current.prev;
            newNode.next = current;
            newNode.prev.next = newNode;
            current.prev = newNode;
        }
    }
}

```

```

        }
        size++;
    }

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

void removeFirst()throws Exception{
    if(isEmpty()){
        throw new Exception("Linked List masih kosong, Tidak dapat dihapus");
    }else if(size == 1){
        removeLast();
    }else{
        head = head.next;
        head.prev = null;
        size--;
    }
}

void removeLast()throws Exception{
    if(isEmpty()){
        throw new Exception("Nilai Index diluar batas");
    }else if(head.next == null){
        head = null;
        size--;
        return;
    }
    Node current = head;
    while(current.next.next != null){
        current = current.next;
    }
    current.next = null;
    size--;
}

void remove(int index)throws Exception{
    if(isEmpty() || index >= size){
        throw new Exception("Nilai indeks diluar batas");
    }
}

```

```

        }else if(index == 0) {
            removeFirst();
        }else{
            Node current = head;
            int i = 0;
            while(i<index) {
                current = current.next;
                i++;
            }
            if(current.next == null){
                current.prev.next = null;
            }else if(current.prev == null){
                current = current.next;
                current.prev = null;
                head = current;
            }else{
                current.prev.next = current.next;
                current.next.prev = current.prev;
            }
        }
    }

    public void searchNode(int value) throws Exception
    {
        int i = 0;
        boolean search = false;
        Node current = head;

        if(head == null) {
            System.out.println("List Kosong");
            return;
        }
        while(current != null) {
            if(current.data == value) {
                search = true;
                break;
            }
            current = current.next;
            i++;
        }
        if(search)
            System.out.println("Data terletak di
indeks ke : " + i);
        else
            System.out.println("Data tidak ada di
list");
    }

    public void sortList() {
        Node current = null, index = null;
        int temp;
        if(head == null) {
            return;
        }
        ...
    }
}

```

```

        }
    else {
        for(current = head; current.next != null;
current = current.next) {
            for(index = current.next; index !=
null; index = index.next) {
                if(current.data > index.data) {
                    temp = current.data;
                    current.data = index.data;
                    index.data = temp;
                }
            }
        }
    }
}
package Tugas1;

import java.util.Scanner;

public class DataMain {
    static void menu(){

System.out.println("=====
=====");
        System.out.println(" PROGRAM PENGOLAHAN ANGKA
DENGAN DOUBLE LINKED LIST");

System.out.println("=====
=====");
        System.out.println("1. Tambah head");
        System.out.println("2. Tambah tail");
        System.out.println("3. Tambah Data");
        System.out.println("4. Hapus Data Pertama");
        System.out.println("5. Hapus Data Terakhir");
        System.out.println("6. Hapus Data Tertentu");
        System.out.println("7. Cetak");
        System.out.println("8. Cari");
        System.out.println("9. Urut Data");
        System.out.println("10. Keluar");

System.out.println("=====");
    }
    public static void main(String[] args) throws
Exception {
        DataServices Ds = new DataServices();
        Scanner sc = new Scanner(System.in);
        Scanner scl = new Scanner(System.in);

```

```

int ulang = 0;
do{
    menu();
    int pilih = sc.nextInt();
    switch(pilih) {
        case 1:
            System.out.println("Masukkan Data
pada Posisi Head");
            int head = sc.nextInt();
            Ds.addFirst(head);
            break;
        case 2:
            System.out.println("Masukkan Data
pada posisi tail");
            int tail = sc.nextInt();
            Ds.addLast(tail);
            break;
        case 3:
            System.out.println("Masukkan
Data");
            int a = sc.nextInt();
            System.out.println("Pointer : ");
            int pointer = sc.nextInt();
            Ds.add(a, pointer);
            break;
        case 4:
            Ds.removeFirst();
            break;
        case 5:
            Ds.removeLast();
            break;
        case 6:
            System.out.println("Pointer");
            int indeks = sc.nextInt();
            Ds.remove(indeks);
            break;
        case 7:
            MasukData ad;
            Ds.print();
            break;
        case 8:
            System.out.println("Cari Data: ");
            int cariData = sc.nextInt();
            if(cariData !=0){
                Ds.searchNode(cariData);
            }
            break;
        case 9:
    }
}

```

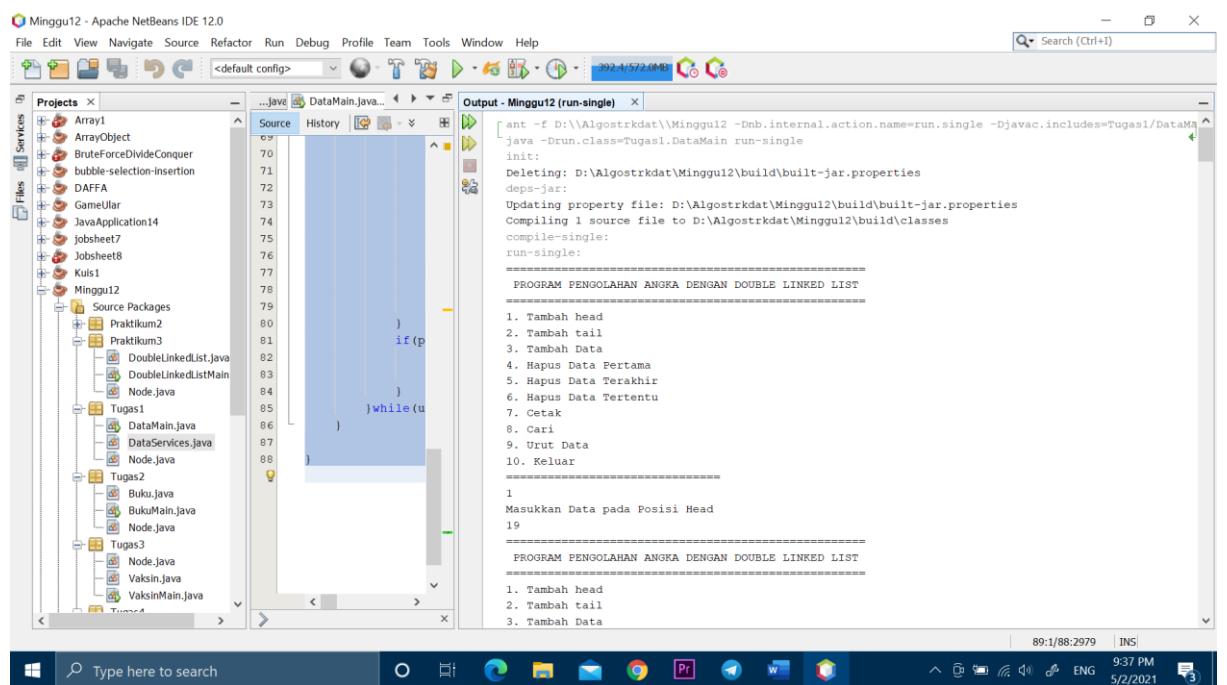
```

        Ds.sortList();
        break;
    }
    if(pilih == 10) {
        System.out.println("Anda telah keluar
dari program");
        break;
    }
}while(ulang!=0 || ulang<3);
}

```

B. Output Program

Running :



Minggu12 - Apache NetBeans IDE 12.0

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

259.4/572.0MB 259.4/572.0MB

Output - Minggu12 (run-single) x

```
Masukkan Data pada Posisi Head
19
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
2
Masukkan Data pada posisi tail
7
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
```

Type here to search

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

360.4/572.0MB 360.4/572.0MB

Output - Minggu12 (run-single) x

```
Masukkan Data pada posisi tail
7
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
3
Masukkan Data
14
Pointer :
1
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
```

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) | 89:1/88:2979 | INS

19 14 7
Berhasil diisi

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

```
3
Masukkan Data
14
Pointer :
1
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
7
19 14 7
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
```

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) | 89:1/88:2979 | INS

19 14 7
Berhasil diisi

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

```
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
1
Masukkan Data pada Posisi Head
50
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
```

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects x ...Java DataMain.java... <default config> T T G G 219.6/652.0M

Services

Files

Source History

Output - Minggu12 (run-single) x

```
1 Masukkan Data pada Posisi Head
50
=====
PROGRAM PENGOLOHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
2
Masukkan Data pada posisi tail
17
=====
PROGRAM PENGOLOHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
```

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects x ...Java DataMain.java... <default config> T T G G 323.2/652.0M

Services

Files

Source History

Output - Minggu12 (run-single) x

```
Masukkan Data pada posisi tail
17
=====
PROGRAM PENGOLOHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
7
50 19 14 7 17
Berhasil diisi
=====
PROGRAM PENGOLOHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
```

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
3
Masukkan Data
49
Pointer :
2
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
```

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```
Masukkan Data
49
Pointer :
2
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
7
50 19 49 14 7 17
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
```

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects

- ...Java DataMain.java...
- Services
- Files
- Minggu12
- Source Packages
- Praktikum2
- DoubleLinkedList.java
- DoubleLinkedListMain.java
- Node.java
- Tugas1
- DataMain.java
- DataServices.java
- Node.java
- Tugas2
- Buku.java
- BukuMain.java
- Node.java
- Tugas3
- Node.java
- Vaksin.java
- VaksinMain.java
- Tugas4

Output - Minggu12 (run-single) x

```
BERANDA SISI GURU  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
1  
Masukkan Data pada Posisi Head  
38  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
2
```

89:1/88:2979 | INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects

- ...Java DataMain.java...
- Services
- Files
- Minggu12
- Source Packages
- Praktikum2
- DoubleLinkedList.java
- DoubleLinkedListMain.java
- Node.java
- Tugas1
- DataMain.java
- DataServices.java
- Node.java
- Tugas2
- Buku.java
- BukuMain.java
- Node.java
- Tugas3
- Node.java
- Vaksin.java
- VaksinMain.java
- Tugas4

Output - Minggu12 (run-single) x

```
38  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
2  
Masukkan Data pada posisi tail  
16  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====
```

89:1/88:2979 | INS

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects

- Array1
- ArrayObject
- BruteForceDivideConquer
- bubble-selection-insertion
- DAFFA
- GameUlar
- JavaApplication14
- jobsheet7
- Jobsheet8
- Kuis1
- Minggu12
- Source Packages
 - Praktikum2
 - Praktikum3
 - DoubleLinkedList.java
 - DoubleLinkedListMain.java
 - Node.java
 - Tugas1
 - DataMain.java
 - DataServices.java
 - Node.java
 - Tugas2
 - Buku.java
 - BukuMain.java
 - Node.java
 - Tugas3
 - Node.java
 - Vaksin.java
 - VaksinMain.java
 - Tugas4

Files

Output - Minggu12 (run-single) BUILD SUCCESSFUL

```
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
7
38 50 19 49 14 7 17 16
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
4
```

89:1/88:2979 | INS

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects

- Array1
- ArrayObject
- BruteForceDivideConquer
- bubble-selection-insertion
- DAFFA
- GameUlar
- JavaApplication14
- jobsheet7
- Jobsheet8
- Kuis1
- Minggu12
- Source Packages
 - Praktikum2
 - Praktikum3
 - DoubleLinkedList.java
 - DoubleLinkedListMain.java
 - Node.java
 - Tugas1
 - DataMain.java
 - DataServices.java
 - Node.java
 - Tugas2
 - Buku.java
 - BukuMain.java
 - Node.java
 - Tugas3
 - Node.java
 - Vaksin.java
 - VaksinMain.java
 - Tugas4

Files

Output - Minggu12 (run-single) BUILD SUCCESSFUL

```
4
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
=====
7
50 19 49 14 7 17 16
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar
```

89:1/88:2979 | INS

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects x ...Java DataMain.java... <default config> T T G G 780.8/652.0MB Q Q

Services

Files

Source History

Source Packages

Praktikum2

Praktikum3

DoubleLinkedList.java

DoubleLinkedListMain.java

Node.java

Tugas1

DataMain.java

DataServices.java

Node.java

Tugas2

Buku.java

BukuMain.java

Node.java

Tugas3

Node.java

Vaksin.java

VaksinMain.java

Tugas4

Output - Minggu12 (run-single) x

Berhasil diisi

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar

=====

5

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar

=====

7

50 19 49 14 7 17

89:1/88:2979 | INS

Type here to search

9:41 PM 5/2/2021

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects x ...Java DataMain.java... <default config> T T G G 259.3/652.0MB Q Q

Services

Files

Source History

Source Packages

Praktikum2

Praktikum3

DoubleLinkedList.java

DoubleLinkedListMain.java

Node.java

Tugas1

DataMain.java

DataServices.java

Node.java

Tugas2

Buku.java

BukuMain.java

Node.java

Tugas3

Node.java

Vaksin.java

VaksinMain.java

Tugas4

Output - Minggu12 (run-single) x

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar

=====

6

Pointer

3

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah head
2. Tambah tail
3. Tambah Data
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari
9. Urut Data
10. Keluar

=====

7

50 19 49 14 7 17

89:1/88:2979 | INS

Type here to search

9:41 PM 5/2/2021

Minggu12 - Apache NetBeans IDE 12.0

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

361.9/652.0MB Search (Ctrl+F)

Projects Services Files <default config> T T G G D D S S O O P P M M C C R R

Source History DataMain.java...

Output - Minggu12 (run-single) x

```
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
1  
Masukkan Data pada Posisi Head  
34  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
2  
Masukkan Data pada posisi tail  
18
```

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

306.0/652.0MB Search (Ctrl+F)

Projects Services Files <default config> T T G G D D S S O O P P M M C C R R

Source History DataMain.java...

Output - Minggu12 (run-single) x

```
Masukkan Data pada posisi tail  
18  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
3  
Masukkan Data  
24  
Pointer :  
3  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data
```

Minggu12 - Apache NetBeans IDE 12.0

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

296.3/652.0MB Search (Ctrl+F)

Projects Services Files <default config> 296.3/652.0MB

Output - Minggu12 (run-single) x

```
Pointer :  
3  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
7  
34 50 19 24 49 7 17 18  
Berhasil diisi  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar
```

89:1/88:2979 INS

Type here to search

9:41 PM 5/2/2021

Minggu12 - Apache NetBeans IDE 12.0

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

253.2/652.0MB Search (Ctrl+F)

Projects Services Files <default config> 253.2/652.0MB

Output - Minggu12 (run-single) x

```
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar  
=====  
8  
Cari Data:  
24  
Data terletak di indeks ke : 3  
=====  
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST  
=====  
1. Tambah head  
2. Tambah tail  
3. Tambah Data  
4. Hapus Data Pertama  
5. Hapus Data Terakhir  
6. Hapus Data Tertentu  
7. Cetak  
8. Cari  
9. Urut Data  
10. Keluar
```

89:1/88:2979 INS

Type here to search

9:42 PM 5/2/2021

The screenshot shows the Apache NetBeans IDE 12.0 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar reads "Minggu12 - Apache NetBeans IDE 12.0". The left sidebar displays the "Projects" and "Files" panes, listing various Java projects and files. The main workspace shows a Java code editor with the file "DataMain.java" open. The code implements a double linked list with methods for adding, removing, and printing nodes. The right pane shows the "Output" window for the run configuration "Minggu12 (run-single)". The output displays the program's menu options and the results of inserting the numbers 17, 19, 24, 34, 49, and 50 into the list. The status bar at the bottom indicates the build was successful.

C. Penjelasan

Pada program diatas terdapat 3 class yaitu :

- Class Node. Yang terdapat 3 atribut yaitu int data, Node prev, dan Node next. Dan di konstruktor dengan this.prev = prev; this.data = data; dan this.next = next;
- Class NodeService. Didalam class ini terdapat 2 atribut, 1 konstruktor, dan 10 method.

2 atribut diatas diantaranya adalah : Node head yang digunakan untuk menentukan awal pada node atau null, dan int size yang digunakan untuk menentukan Panjang dari data double linked list.

Terdapat 1 konstruktor yang berfungsi untuk menginisialisasikan head adalah null dan size = 0.

Dan ada 9 macam method, diantaranya yaitu :

- isEmpty() : digunakan untuk mengecek apakah ada data yang kosong.
 - addFirst() : digunakan untuk menambah data sebelum node head (prev).
 - addLast() : digunakan untuk menambah data sesudah node head (next).
 - add() : digunakan untuk menambah data di bagian node atau di indeks tertentu.
 - print() : menampilkan seluruh data pada double linked list.
 - removeFirst() : digunakan untuk menghapus atau mengeluarkan pada awal data pada double linked list.
 - removeLast() : digunakan untuk menghapus atau mengeluarkan pada akhir data pada double linked list.
 - sortList() : digunakan untuk mengurutkan data double linked list menggunakan bubble sort.
 - searchNode(int value) : digunakan untuk mencari node dengan menggunakan indeks node.
- c. Dan yang terakhir class Main yang digunakan untuk mengeluarkan script data dengan menggunakan menu. Dan didalam class Main terdapat 3 method fungsi yaitu :
- Method menu(), yang digunakan untuk list data menu apa saja yang akan ditampilkan di method data().
 - Method data(), yang digunakan untuk menginput data dari class lain untuk menghasilkan suatu data output an disetiap pilihan pada menu dan akan ditampilkan pada method main.
 - Method main, yang digunakan untuk mengeluarkan data dari method data().

2. Tugas no 2

a. Kode Program

```
package Tugas2;
public class Node {
    String data;
    Node prev, next;

    Node(Node prev, String data, Node next) {
        this.prev = prev;
        this.data = data;
        this.next = next;
    }
}
package Tugas2;
public class Buku {
    Node head;
    int size;

    Buku() {
        head = null;
        size = 0;
    }
}
```

```

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

        void addFirst(String item){
            if(isEmpty()){
                head = new Node(null, item, null);
            }else{
                Node newNode = new Node(null, item,
head);
                head.prev = newNode;
                head = newNode;
            }
            size++;
        }

        void addLast(String item){
            if(isEmpty()){
                addFirst(item);
            }else{
                Node current = head;
                while(current.next != null){
                    current = current.next;
                }
                Node newNode1 = new Node(current, item,
null);
                current.next = newNode1;
                size++;
            }
        }

        void removeFirst()throws Exception{
            if(isEmpty()){
                throw new Exception("Linked List masih
kosong, Tidak dapat dihapus");
            }else if(size == 1){
                removeLast();
            }else{
                head = head.next;
                head.prev = null;
                size--;
            }
        }

        void removeLast()throws Exception{
            if(isEmpty()){
                throw new Exception("Nilai Index diluar
batas");
            }else if(head.next == null){
                head = null;
                size--;
            }
        }
    }
}

```

```

        return;
    }
    Node current = head;
    while(current.next.next != null){
        current = current.next;
    }
    current.next = null;
    size--;
}

void peek(){
    Node tmp = head;
    System.out.println("Data yang berada pada
posisi teratas adalah "+head.data);
}

void print2(){
    if(!isEmpty()){
        Node tmp = head;
        int a = 0;
        while(tmp != null){
            System.out.println((a+1)+ ".
"+tmp.data);
            tmp = tmp.next;
            a++;
        }
        size++;
        System.out.println("\nBerhasil diisi");
    }else{
        System.out.println("Linked List
Kosong");
    }
}
package Tugas2;

import java.util.Scanner;

public class BukuMain {
    static void menu(){

System.out.println("*****");
        System.out.println(" Data Buku
Perpustakaan");

System.out.println("*****");
        System.out.println("1. Entry Judul Buku");
        System.out.println("2. Ambil Buku Teratas");
        System.out.println("3. Cek Judul Buku
Teratas");
    }
}
```

```

        System.out.println("4. Info Semua Judul
Buku");
        System.out.println("5. Keluar");
        System.out.println("-----");
    }
    public static void main(String[] args) throws
Exception {
    Scanner sc = new Scanner(System.in);
    Scanner sc1 = new Scanner(System.in);

    Buku ds = new Buku();

    int ulang = 0;
    do{
        menu();
        int pilih = sc.nextInt();
        System.out.println("-----");
-----");
        switch(pilih){
            case 1:
                System.out.println("Masukkan
judul buku : ");
                String jb = sc1.nextLine();
                ds.addLast(jb);
                break;
            case 2:
                System.out.println("Buku pada
tumpukan teratas telah diambil");
                ds.removeFirst();
                break;
            case 3:
                ds.peek();
                break;
            case 4:
                ds.print2();
                break;
        }
        if(pilih == 5){
            System.out.println("Anda telah
keluar!");
            break;
        }
    }while(ulang!=0 || ulang<3);
}
}

```

b. Output Program

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Tugas1 > Node.java

```
34 } if(p!=null){  
35     p.print();  
36 }  
37 }  
38  
39 }  
40  
41 }  
42 }  
43 }  
44 } if(p!=null){  
45     p.print();  
46 }  
47 } while(u!=null);  
48 }  
49 }  
50 }  
51 }
```

Output - Minggu12 (run-single) >

```
[ ant -f D:\\Algostrkdat\\Minggu12 -Dnb.internal.action.name=run.single -Djavac.includes=Tugas2/BukuMain  
java -Drun.class=Tugas2.BukuMain run-single  
init:  
Deleting: D:\\Algostrkdat\\Minggu12\\build\\built-jar.properties  
deps-jar:  
Updating property file: D:\\Algostrkdat\\Minggu12\\build\\built-jar.properties  
Compiling 1 source file to D:\\Algostrkdat\\Minggu12\\build\\classes  
compile-single:  
run-single:  
*****  
Data Buku Perpustakaan  
*****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :  
Matematika Diskrit  
*****  
Data Buku Perpustakaan  
*****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :  
Story My life  
*****  
Data Buku Perpustakaan  
*****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :
```

52:1/52:1682 INS 9:46 PM 5/2/2021

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Tugas1 > Node.java

```
34 } if(p!=null){  
35     p.print();  
36 }  
37 }  
38  
39 }  
40  
41 }  
42 }  
43 }  
44 } if(p!=null){  
45     p.print();  
46 }  
47 } while(u!=null);  
48 }  
49 }  
50 }  
51 }
```

Output - Minggu12 (run-single) >

```
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :  
Matematika Diskrit  
*****  
Data Buku Perpustakaan  
*****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :  
Story My life  
*****  
Data Buku Perpustakaan  
*****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
1  
  
Masukkan judul buku :
```

52:1/52:1682 INS 9:46 PM 5/2/2021

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects > Minggu12 > Source Packages > DoubleLinkedList > DoubleLinkedListMain.java

```
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

Output - Minggu12 (run-single) >

```
4. Info Semua Judul Buku
5. Keluar
-----
1
-----
Masukkan judul buku :
Matematika Diskrit
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Ambil Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
-----
1
-----
Masukkan judul buku :
Story My life
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Ambil Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
-----
1
-----
Masukkan judul buku :
```

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Search (Ctrl+F)

Projects > Minggu12 > Source Packages > DoubleLinkedList > DoubleLinkedListMain.java

```
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

Output - Minggu12 (run-single) >

```
1
-----
Masukkan judul buku :
Negeri 5 Menara
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Ambil Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
-----
1
-----
Masukkan judul buku :
Dasar Pemrograman
*****
Data Buku Perpustakaan
*****
1. Entry Judul Buku
2. Ambil Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar
-----
1
-----
Masukkan judul buku :
Aljabar Linier
*****
Data Buku Perpustakaan
```

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Praktikum3 > DoubleLinkedList.java

```
34 } if(p != null) { 35     p = p.next; 36 } 37 } 38 } 39 } 40 } 41 } 42 } 43 } 44 } if(p != null) { 45     p = p.next; 46 } 47 } while(u != null); 48 } 49 } 50 } 51 }
```

Output - Minggu12 (run-single) >

```
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
4  
-----  
1. Matematika Diskrit  
2. Story My life  
3. Negeri 5 Menara  
4. Dasar Pemrograman  
5. Aljabar Linier  
Berhasil diisi  
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
3  
-----  
Data yang berada pada posisi teratas adalah Matematika Diskrit  
***** Data Buku Perpustakan *****
```

52:1/52:1682 INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Praktikum3 > DoubleLinkedList.java

```
34 } if(p != null) { 35     p = p.next; 36 } 37 } 38 } 39 } 40 } 41 } 42 } 43 } 44 } if(p != null) { 45     p = p.next; 46 } 47 } while(u != null); 48 } 49 } 50 } 51 }
```

Output - Minggu12 (run-single) >

```
Data yang berada pada posisi teratas adalah Matematika Diskrit  
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
2  
-----  
Buku pada tumpukan teratas telah diambil  
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
4  
-----  
1. Story My life  
2. Negeri 5 Menara  
3. Dasar Pemrograman  
4. Aljabar Linier  
Berhasil diisi  
***** Data Buku Perpustakan *****
```

52:1/52:1682 INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Praktikum3 > DoubleLinkedList.java

```
34 } if(p != null) { 35     p = p.next; 36 } 37 } 38 } 39 } 40 } 41 } 42 } 43 } 44 } if(p != null) { 45     p = p.next; 46 } 47 } while(u != null); 48 } 49 } 50 } 51 }
```

Output - Minggu12 (run-single) >

```
Data yang berada pada posisi teratas adalah Matematika Diskrit  
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
2  
-----  
Buku pada tumpukan teratas telah diambil  
***** Data Buku Perpustakan *****  
1. Entry Judul Buku  
2. Ambil Buku Teratas  
3. Cek Judul Buku Teratas  
4. Info Semua Judul Buku  
5. Keluar  
-----  
4  
-----  
1. Story My life  
2. Negeri 5 Menara  
3. Dasar Pemrograman  
4. Aljabar Linier  
Berhasil diisi  
***** Data Buku Perpustakan *****
```

52:1/52:1682 INS

The screenshot shows the Apache NetBeans IDE 12.0 interface. The left pane displays a project structure with several Java files under the 'Mingu12' project. The right pane shows the 'Output' window for the 'Mingu12 (run-single)' configuration. The output text is as follows:

```

Berhasil diisi
*****
Data Buku Perpustakan
*****
1. Story My life
2. Negeri 5 Menara
3. Dasar Pemrograman
4. Aljabar Linier

Data yang berada pada posisi teratas adalah Story My life
*****
Data Buku Perpustakan
*****
1. Entry Judul Buku
2. Ambil Buku Teratas
3. Cek Judul Buku Teratas
4. Info Semua Judul Buku
5. Keluar

Anda telah keluar!
BUILD SUCCESSFUL (total time: 1 minute 29 seconds)

```

c. Penjelasan

terdapat method addFirst, addLast, print, removeFirst, removeLast, peek dan print2 pada method peek untuk menampilkan tumpukan buku paling atas dengan Node tmp = head; karena yang akan ditampilkan bagian head saja

3. Tugas no 3

A. Kode Program

```

package Tugas3;
public class Node {
    int data;
    String nama;
    Node prev, next;
    Node(Node prev, int data, String nama, Node next) {
        this.prev = prev;
        this.data = data;
        this.nama = nama;
        this.next = next;
    }
}
package Tugas3;
public class Vaksin {
    Node head;
    int size;

    Vaksin() {
        head = null;
        size = 0;
    }

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

```

```

void addFirst(int data, String nama){
    if(isEmpty()){
        head = new Node(null, data, nama, null);
    } else{
        Node temp = new Node(null, data, nama,
head);
        head.prev = temp;
        head = temp;
    }
    size++;
}

void addLast(int data, String nama){
    if(isEmpty()){
        head = new Node(null, data, nama, null);
    } else{
        Node current = head;
        while(current.next != null){
            current = current.next;
        }
        Node nn = new Node(current, data, nama,
null);
        current.next = nn;
        size++;
    }
}

void removeFirst() throws Exception{
    if(isEmpty()){
        throw new Exception("Linked List masih
kosong, Tidak dapat dihapus");
    } else if(size == 1){
        removeLast();
    } else{
        System.out.println(head.nama+" selesai
divaksinasi");
        head = head.next;
        head.prev = null;
        size--;
    }
    //size--;
}

void removeLast() throws Exception{
    if(isEmpty()){
        throw new Exception("Nilai Index diluar
batas");
    } else if(head.next == null){
        head = null;
        size--;
    }
}

```

```

        return;
    }
    Node current = head;
    while(current.next.next != null) {
        current = current.next;
    }
    current.next = null;
    size--;
}

void print(){
    if(!isEmpty()){
        Node tmp = head;
        while(tmp != null) {

System.out.println("| "+tmp.data+"\t| "+tmp.nama+"\t| "
);
        tmp = tmp.next;
    }
    size++;
    System.out.println("Sisa Antrian :
"+(size-1));
        System.out.println("\nBerhasil diisi");
        size--;
    }else{
        System.out.println("Linked List
Kosong");
    }
}
package Tugas3;

import java.util.Scanner;

public class VaksinMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws
Exception {
        Scanner sc = new Scanner(System.in);
        Scanner scl = new Scanner(System.in);

        Vaksin vs = new Vaksin();

        vs.addLast(123, "Daffa");
        vs.addLast(234, "Taufiq");
        vs.addLast(345, "Wisnu");
    }
}
```

```

        int ulang = 0;
        do{

System.out.println("=====");
        System.out.println(" PENGANTRI VAKSIN
EXTRAVAGANZA");

System.out.println("=====");
        System.out.println("1. Tambah Data
Penerima Vaksin");
        System.out.println("2. Hapus Data
Penerima Vaksin");
        System.out.println("3. Daftar Penerima
Vaksin");
        System.out.println("4. Keluar");
        System.out.println("-----");
-----
        int pilih = sc.nextInt();
        switch(pilih){
            case 1:
                System.out.println("-----");
                System.out.println("Masukkan
Data Penerima Vaksin");
                System.out.println("-----");
                System.out.println("Nomer
Antrian : ");
                int antri = sc.nextInt();
                System.out.println("Nama
Penerima : ");
                String nama = scl.nextLine();

System.out.println("|"+antri+"|"+nama+"\t|");
                vs.addFirst(antri, nama);
                break;
            case 2:
                vs.removeFirst();

System.out.println("=====");
                break;
            case 3:

System.out.println("|No.\t|Nama\t|");
                vs.print();
                break;
        }
    }

```

```

        if(pilih == 4) {
            System.out.println("Anda telah
keluar!");
            break;
        }
    }while(ulang!=0 || ulang<3);

}

```

B. Output Program

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single)

```

        ant -f D:\Algostrkd\ Minggu12 -Dnb.internal.action.name=run.single -Djavac.includes=Tugas3\VaksinMain.java
        -Drun.class=Tugas3\VaksinMain run-single
        init:
        Deleting: D:\Algostrkd\ Minggu12\build\built-jar.properties
        deps-jar:
        Updating property file: D:\Algostrkd\ Minggu12\build\built-jar.properties
        Compiling 1 source file to D:\Algostrkd\ Minggu12\build\classes
        compile-single:
        run-single:
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        1
        -----
        Masukkan Data Penerima Vaksin
        -----
        Nomer Antrian :
        456
        Nama Penerima :
        Pratama
        |456|Pratama |
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        1
        -----
        Masukkan Data Penerima Vaksin
        -----
        Nomer Antrian :
        567
        Nama Penerima :
        Naufal
        |567|Naufal |
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        3
        |No. |Nama |
        |567 |Naufal |
        |456 |Pratama |
        |123 |Daffa |

```

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single)

```

        ant -f D:\Algostrkd\ Minggu12 -Dnb.internal.action.name=run.single -Djavac.includes=Tugas3\VaksinMain.java
        -Drun.class=Tugas3\VaksinMain run-single
        init:
        Deleting: D:\Algostrkd\ Minggu12\build\built-jar.properties
        deps-jar:
        Updating property file: D:\Algostrkd\ Minggu12\build\built-jar.properties
        Compiling 1 source file to D:\Algostrkd\ Minggu12\build\classes
        compile-single:
        run-single:
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        1
        -----
        Masukkan Data Penerima Vaksin
        -----
        Nomer Antrian :
        456
        Nama Penerima :
        Pratama
        |456|Pratama |
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        1
        -----
        Masukkan Data Penerima Vaksin
        -----
        Nomer Antrian :
        567
        Nama Penerima :
        Naufal
        |567|Naufal |
        =====
        PENGANTRI VAKSIN EXTRAVAGANZA
        =====
        1. Tambah Data Penerima Vaksin
        2. Hapus Data Penerima Vaksin
        3. Daftar Penerima Vaksin
        4. Keluar
        -----
        3
        |No. |Nama |
        |567 |Naufal |
        |456 |Pratama |
        |123 |Daffa |

```

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```
1
-----
Masukkan Data Penerima Vaksin

Nomer Antrian :
567
Nama Penerima :
Naufal
[567]Naufal |

PENGANTRI VAKSIN EXTRAVAGANZA
-----
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

-----
3
|No. |Nama |
|567 |Naufal |
|456 |Pratama |
|123 |Daffa |
|234 |Taufiq |
|345 |Wisnu |
Sisa Antrian : 4

Berhasil diisi
-----
PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar
```

1345 |Wisnu |
Sisa Antrian : 4

Berhasil diisi

PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

2
Naufal selesai divaksinasi

PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

3
No.	Nama
456	Pratama
123	Daffa
234	Taufiq
345	Wisnu
Sisa Antrian : 3

Berhasil diisi

63:1/63:2192 INS

Type here to search

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

Output - Minggu12 (run-single) x

```
1
-----
Masukkan Data Penerima Vaksin

Nomer Antrian :
567
Nama Penerima :
Naufal
[567]Naufal |

PENGANTRI VAKSIN EXTRAVAGANZA
-----
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

-----
3
|No. |Nama |
|567 |Naufal |
|456 |Pratama |
|123 |Daffa |
|234 |Taufiq |
|345 |Wisnu |
Sisa Antrian : 4

Berhasil diisi
-----
PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar
```

1345 |Wisnu |
Sisa Antrian : 4

Berhasil diisi

PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

2
Naufal selesai divaksinasi

PENGANTRI VAKSIN EXTRAVAGANZA
1. Tambah Data Penerima Vaksin
2. Hapus Data Penerima Vaksin
3. Daftar Penerima Vaksin
4. Keluar

3
No.	Nama
456	Pratama
123	Daffa
234	Taufiq
345	Wisnu
Sisa Antrian : 3

Berhasil diisi

63:1/63:2192 INS

Type here to search

```

public class Vaksin {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner s2 = new Scanner(System.in);
        Vaksin vs = new Vaksin();
        int ulang;
        do{
            System.out.println("1. Tambah Data Penerima Vaksin");
            System.out.println("2. Hapus Data Penerima Vaksin");
            System.out.println("3. Daftar Penerima Vaksin");
            System.out.println("4. Keluar");
            System.out.print("Silahkan pilih menu : ");
            ulang = s.nextInt();
            switch(ulang){
                case 1:
                    vs.addLast(s2.nextLine());
                    break;
                case 2:
                    vs.removeFirst();
                    break;
                case 3:
                    vs.print();
                    break;
                case 4:
                    System.out.println("Anda telah keluar!");
                    break;
                default:
                    System.out.println("Maaf, inputan salah!");
            }
        }while(ulang != 4);
    }
}

```

C. Penjelasan

terdapat method addFirst, addLast, print, removeFirst, removeLast pada method print digunakan untuk menampilkan data dan sisa antrian

4. Tugas No 4

A. Kode Program

```

package Tugas4;
public class Node {
    Mahasiswa data;
    Node prev, next;
    Node(Node prev, Mahasiswa data, Node next) {
        this.prev = prev;
        this.data = data;
        this.next = next;
    }

    void printmahasiswa() {
        System.out.println("nim \t: "+data.nim);
        System.out.println("nama\t: "+data.nama);
        System.out.println("ipk \t: "+data.ipk);
    }
}
package Tugas4;
public class Mahasiswa {
    int nim;
    String nama;
    double ipk;
    Mahasiswa(int nim, String nama, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.ipk = ipk;
    }
}

```

```
package Tugas4;
public class MahasiswaServices {
    Node head;
    int size;

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

    void addFirst(Mahasiswa data) {
        if(isEmpty()){
            head = new Node(null, data,null);
        }else{
            Node temp = new Node(null, data, head);
            head.prev = temp;
            head = temp;
        }
        size++;
    }

    void addLast(Mahasiswa data) {
        if(isEmpty()){
            head = new Node(null, data, null);
        }else{
            Node current = head;
            while(current.next != null){
                current = current.next;
            }
            Node nn = new Node(current, data, null);
            current.next = nn;
            size++;
        }
    }

    void add(Mahasiswa item, int index) throws
Exception{
        if(isEmpty()){
            addFirst(item);
        }else if(index < 0 || index > size){
            throw new Exception("Nilai index diluar
batas");
        }else{
            Node current = head;
            int i = 0;
            while(i<index){
                current = current.next;
                i++;
            }
            if(current.prev == null){
                Node newNode = new Node(null, item,
current);
                current.prev = newNode;
            }
        }
    }
}
```

```

                head = newNode;
            }else{
                Node newNode = new
Node(current.prev, item, current);
                newNode.prev = current.prev;
                newNode.next = current;
                newNode.prev.next = newNode;
                current.prev = newNode;
            }
        }
        size++;
    }

    void removeFirst()throws Exception{
        if(isEmpty()){
            throw new Exception("Linked List masih
kosong, Tidak dapat dihapus");
        }else if(size == 1){
            removeLast();
        }else{
            head = head.next;
            head.prev = null;
            size--;
        }
    }

    void removeLast()throws Exception{
        if(isEmpty()){
            throw new Exception("Nilai Index diluar
batas");
        }else if(head.next == null){
            head = null;
            size--;
            return;
        }
        Node current = head;
        while(current.next.next != null){
            current = current.next;
        }
        current.next = null;
        size--;
    }

    void remove(int index)throws Exception{
        if(isEmpty() || index >= size){
            throw new Exception("Nilai indeks diluar
batas");
        }else if(index == 0){
            removeFirst();
        }else{
            Node current = head;

```

```

        int i = 0;
        while(i<index) {
            current = current.next;
            i++;
            }if(current.next == null){
                current.prev.next = null;
            }else if(current.prev == null){
                current = current.next;
                current.prev = null;
                head = current;
            }else{
                current.prev.next = current.next;
                current.next.prev = current.prev;
            }
        }
    }

    int size(){
        return size;
    }

    void print(){
        if(!isEmpty()){
            Node tmp = head;
            int a = 1;
            while(tmp != null){
//                System.out.println("nim\t|nama\t|ipk\t|");
//                System.out.println(a+".
                |"+tmp.data.nim+"\t|"+tmp.data.nama+"\t|"+tmp.data.i
                pk+"\t|");
                tmp.printmahasiswa();
                System.out.println("-----");
                tmp = tmp.next;
                a++;
            }
            size++;
            System.out.println("Sisa Antrian :
"+(size-1));
            System.out.println("\nBerhasil diisi");
            size--;
        }else{
            System.out.println("Linked List
Kosong");
        }
    }

    public void seqSearch(int nims){
        int idx = -1;
        Node current = head;

```

```

        for (int i = 0; i < size; i++) {
            if (current.data.nim == nims) {
                idx = i;
            } else {
                current = current.next;
            }
        }
        if (idx != -1) {
            System.out.println("Data "+nims+" berada
di node ke-"+idx);
            System.out.println("IDENTITAS:");
            System.out.println("NIM:
"+current.data.nim); System.out.println(
                    "Nama: "+current.data.nama);
            System.out.println("IPK:
"+current.data.ipk);
        } else {
            System.out.println("Data yang anda cari
tidak ditemukan");
        }
    }

    public Node bubbleSort() {
        int swapped;
        Node p1;
        Node p2 = null;
        if (head == null){
            return null;
        }
        do{
            swapped = 0;
            p1 = head;
            while (p1.next != p2){
                if (p1.data.ipk < p1.next.data.ipk) {
                    Mahasiswa t = p1.data;
                    p1.data = p1.next.data;
                    p1.next.data = t;
                    swapped = 1;
                }
                p1 = p1.next;
            }
            p2 = p1;
        }
        while (swapped != 0);
        return head;
    }
}
/*
 * To change this license header, choose License
Headers in Project Properties.

```

```

 * To change this template file, choose Tools | 
Templates
 * and open the template in the editor.
 */
package Tugas4;

import java.util.Scanner;

/**
 *
 * @author ACHE
 */
public class MahasiswaMain {
    static void menu() {

System.out.println("=====
=====");
        System.out.println(" PROGRAM PENGOLAHAN
ANGKA DENGAN DOUBLE LINKED LIST");

System.out.println("=====
=====");
        System.out.println("1. Tambah data awal");
        System.out.println("2. Tambah data akhir");
        System.out.println("3. Tambah Data
tertentu");
        System.out.println("4. Hapus Data Pertama");
        System.out.println("5. Hapus Data
Terakhir");
        System.out.println("6. Hapus Data
Tertentu");
        System.out.println("7. Cetak");
        System.out.println("8. Cari Nim");
        System.out.println("9. Urut Data IPK-DESC");
        System.out.println("10. Keluar");

System.out.println("=====
");
    }

    public static void main(String[] args) throws
Exception {
        Scanner sc = new Scanner(System.in);
        Scanner sc1 = new Scanner(System.in);
        Scanner sc2 = new Scanner(System.in);

        MahasiswaServices ds = new
MahasiswaServices();

        Mahasiswa mhs1 = new
Mahasiswa(788,"Wendy",3.87);
    }
}

```

```

        Mahasiswa mhs2 = new
Mahasiswa(234,"Rendy",3.45);
        Mahasiswa mhs3 = new
Mahasiswa(564,"Ahmad",4);

        ds.addLast(mhs1);
        ds.addLast(mhs2);
        ds.addLast(mhs3);

        int ulang = 0;
        do{
            menu();
            int pilih = sc.nextInt();
            switch(pilih){
                case 1:
                    System.out.println("Masukkan
data mahasiswa posisi awal");
                    System.out.println("nim : ");
                    int nim = sc.nextInt();
                    System.out.println("nama : ");
                    String nama = sc.next();
                    System.out.println("IPK : ");
                    double ipk = sc2.nextDouble();
                    Mahasiswa awal = new
Mahasiswa(nim, nama, ipk);
                    ds.addFirst(awal);
                    break;
                case 2:
                    System.out.println("Masukkan
data mahasiswa posisi akhir");
                    System.out.println("nim : ");
                    int nim2 = sc.nextInt();
                    System.out.println("nama : ");
                    String nama2 = sc.next();
                    System.out.println("IPK : ");
                    double ipk2 = sc2.nextDouble();
                    Mahasiswa akhir = new
Mahasiswa(nim2, nama2, ipk2);
                    ds.addLast(akhir);
                    break;
                case 3:
                    System.out.println("Masukkan
data mahasiswa posisi akhir");
                    System.out.println("nim : ");
                    int nim3 = sc.nextInt();
                    System.out.println("nama : ");
                    String nama3 = sc.next();
                    System.out.println("IPK : ");
                    double ipk3 = sc2.nextDouble();
                    System.out.println("Pointer :
");
            }
        }
    }
}

```

```

        int pointer = sc.nextInt();
        Mahasiswa add = new
Mahasiswa(nim3, nama3, ipk3);
        ds.add(add, pointer);
        break;
    case 4:
        ds.removeFirst();
        break;
    case 5:
        ds.removeLast();
        break;
    case 6:
        System.out.println("Masukkan
pointer : ");
        int pointer1 = sc.nextInt();
        ds.remove(pointer1);
        break;
    case 7:
        ds.print();
        break;
    case 8:
        System.out.println("Cari Data :
");
        int cari = sc.nextInt();
        if (cari != 0) {
            ds.seqSearch(cari);
        }
        ds.print();
        break;
    case 9:
        ds.bubbleSort();
        System.out.println("Data
berhasil diurutkan (Desc)");
        break;
    }
    if(pilih == 10){
        System.out.println("Anda telah
keluar dari program");
        break;
    }
}while(ulang!=0 || ulang<3);

}
}

```

B. Output Program

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```
ant -f D:\Algostrkdat\Minggu12 -Dnb.internal.action.name=run.single -Djavac.includes=Tugas1/MahasiswaMain.java -Drun.class=Tugas4.MahasiswaMain run-single
init:
Deleting: D:\Algostrkdat\Minggu12\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\Minggu12\build\built-jar.properties
Compiling 1 source file to D:\Algostrkdat\Minggu12\build\classes
compile-single:
run-single:
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
=====
1
Masukkan data mahasiswa posisi awal
nim :
988
nama :
Tama
Tama
IPK :
3.5
```

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```
ant -f D:\Algostrkdat\Minggu12 -Dnb.internal.action.name=run.single -Djavac.includes=Tugas1/MahasiswaMain.java -Drun.class=Tugas4.MahasiswaMain run-single
init:
Deleting: D:\Algostrkdat\Minggu12\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\Minggu12\build\built-jar.properties
Compiling 1 source file to D:\Algostrkdat\Minggu12\build\classes
compile-single:
run-single:
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
=====
2
Masukkan data mahasiswa posisi akhir
nim :
877
nama :
Naufal
IPK :
3.4
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
```

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Praktikum3 > DoubleLinkedList.java

Source History System.out.print System.out.print System.out.print

```
27 } public static void main(Scanner sc = new Scanner(scl = ne Scanner sc2 = ne
28 System.out.print("MahasiswaService")
29 Mahasiswa mhs1 = Mahasiswa mhs2 =
30 Mahasiswa mhs3 =
31 Mahasiswa mhs1 =
32 Mahasiswa mhs2 =
33 Mahasiswa mhs3 =
34 ds.addLast(mhs1);
35 ds.addLast(mhs2);
36 ds.addLast(mhs3);
37 int ulang = 0;
38 do{
39     menu();
40     int pilih =
41     switch(pilih)
42         case 1:
43             Syst
44
45
46
47
48
49
50
51
52
53 }
```

Output - Minggu12 (run-single) >

```
3.4
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
=====
3
Masukkan data mahasiswa posisi akhir
nim :
967
nama :
Yogi
IPK :
3.0
Pointer :
2
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
4. Hapus Data Pertama
```

123:1/123:4608 INS

Minggu12 - Apache NetBeans IDE 12.0

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

Projects > Minggu12 > Source Packages > Praktikum3 > DoubleLinkedList.java

Source History System.out.print System.out.print System.out.print

```
27 } public static void main(Scanner sc = new Scanner(scl = ne Scanner sc2 = ne
28 System.out.print("MahasiswaService")
29 Mahasiswa mhs1 = Mahasiswa mhs2 =
30 Mahasiswa mhs3 =
31 Mahasiswa mhs1 =
32 Mahasiswa mhs2 =
33 Mahasiswa mhs3 =
34 ds.addlast(mhs1);
35 ds.addlast(mhs2);
36 ds.addlast(mhs3);
37 int ulang = 0;
38 do{
39     menu();
40     int pilih =
41     switch(pilih)
42         case 1:
43             Syst
44
45
46
47
48
49
50
51
52
53 }
```

Output - Minggu12 (run-single) >

```
2
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
=====
7
nim : 988
nama : Tama
ipk : 3.5
-----
nim : 788
nama : Wendy
ipk : 3.87
-----
nim : 967
nama : Yogi
ipk : 3.0
-----
nim : 234
nama : Rendy
ipk : 3.45
```

123:1/123:4608 INS

Minggu12 - Apache NetBeans IDE 12.0

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

...java MahasiswaMain.java

```

27 System.out.print
28 System.out.print
29 System.out.print
30 }
31 public static void m
32 Scanner sc = new
33 Scanner scl = ne
34 Scanner sc2 = ne
35
36 MahasiswaService
37
38 Mahasiswa mhs1 =
39 Mahasiswa mhs2 =
40 Mahasiswa mhs3 =
41
42 ds.addLast(mhs1)
43 ds.addLast(mhs2)
44 ds.addLast(mhs3)
45
46 int ulang = 0;
47 do{
48     menu();
49     int pilih =
50     switch(pilih
51         case 1:
52             Syst
53

```

Output - Minggu12 (run-single)

```

7 nim : 988
nama : Tama
ipk : 3.5
-----
nim : 788
nama : Wendy
ipk : 3.87
-----
nim : 967
nama : Yogi
ipk : 3.0
-----
nim : 234
nama : Rendy
ipk : 3.45
-----
nim : 564
nama : Ahmad
ipk : 4.0
-----
nim : 877
nama : Naufal
ipk : 3.4
-----
Sisa Antrian : 5
Berhasil diisi
PROGRAM PENGSOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal

```

123:1/123-4608 | INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

...java MahasiswaMain.java

```

27 System.out.print
28 System.out.print
29 System.out.print
30 }
31 public static void m
32 Scanner sc = new
33 Scanner scl = ne
34 Scanner sc2 = ne
35
36 MahasiswaService
37
38 Mahasiswa mhs1 =
39 Mahasiswa mhs2 =
40 Mahasiswa mhs3 =
41
42 ds.addlast(mhs1)
43 ds.addlast(mhs2)
44 ds.addlast(mhs3)
45
46 int ulang = 0;
47 do{
48     menu();
49     int pilih =
50     switch(pilih
51         case 1:
52             Syst
53

```

Output - Minggu12 (run-single)

```

Berhasil diisi
PROGRAM PENGSOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
=====
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
-----
8 Cari Data :
967 Data 967 berada di node ke-4
IDENTITAS:
NIM: 967
Nama: Yogi
IPK: 3.0
nim : 988
nama : Tama
ipk : 3.5
-----
nim : 788
nama : Wendy
ipk : 3.87

```

123:1/123-4608 | INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

...Java MahasiswaMain.java

```
27 System.out.print
28 System.out.print
29 System.out.print
30 }
31 public static void m
32 Scanner sc = new
33 Scanner scl = ne
34 Scanner sc2 = ne
35
36 MahasiswaService
37
38 Mahasiswa mhs1 =
39 Mahasiswa mhs2 =
40 Mahasiswa mhs3 =
41
42 ds.addLast(mhs1)
43 ds.addLast(mhs2)
44 ds.addLast(mhs3)
45
46 int ulang = 0;
47 do{
48     menu();
49     int pilih =
50     switch(pilih
51         case 1:
52             Syst
53 }
```

Berhasil diisi

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar

=====

9

Data berhasil diurutkan (Desc)

=====

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar

=====

123:1/123:4608 INS

Type here to search

Minggu12 - Apache NetBeans IDE 12.0

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

...Java MahasiswaMain.java

```
27 System.out.print
28 System.out.print
29 System.out.print
30 }
31 public static void m
32 Scanner sc = new
33 Scanner scl = ne
34 Scanner sc2 = ne
35
36 MahasiswaService
37
38 Mahasiswa mhs1 =
39 Mahasiswa mhs2 =
40 Mahasiswa mhs3 =
41
42 ds.addLast(mhs1)
43 ds.addLast(mhs2)
44 ds.addLast(mhs3)
45
46 int ulang = 0;
47 do{
48     menu();
49     int pilih =
50     switch(pilih
51         case 1:
52             Syst
53 }
```

PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

=====

1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar

=====

7

nim : 564
nama : Ahmad
ipk : 4.0

nim : 788
nama : Wendy
ipk : 3.87

nim : 988
nama : Tama
ipk : 3.5

nim : 234
nama : Rendy
ipk : 3.45

nim : 877

123:1/123:4608 INS

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```

9. Urut Data IPK-DESC
10. Keluar
=====
7
nim : 564
nama : Ahmad
ipk : 4.0
-----
nim : 788
nama : Wendy
ipk : 3.87
-----
nim : 988
nama : Tama
ipk : 3.5
-----
nim : 234
nama : Rendy
ipk : 3.45
-----
nim : 877
nama : Naufal
ipk : 3.4
-----
nim : 967
nama : Yogi
ipk : 3.0
-----
Sisa Antrian : 5
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST

```

123:1/123-4608 | INS

Minggu12 - Apache NetBeans IDE 12.0

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

Output - Minggu12 (run-single) x

```

nama : Rendy
ipk : 3.45
-----
nim : 877
nama : Naufal
ipk : 3.4
-----
nim : 967
nama : Yogi
ipk : 3.0
-----
Sisa Antrian : 5
Berhasil diisi
=====
PROGRAM PENGOLAHAN ANGKA DENGAN DOUBLE LINKED LIST
1. Tambah data awal
2. Tambah data akhir
3. Tambah Data tertentu
4. Hapus Data Pertama
5. Hapus Data Terakhir
6. Hapus Data Tertentu
7. Cetak
8. Cari Nim
9. Urut Data IPK-DESC
10. Keluar
=====
10
Anda telah keluar dari program
BUILD SUCCESSFUL (total time: 3 minutes 36 seconds)

```

123:1/123-4608 | INS

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 samasama 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;
    }
}

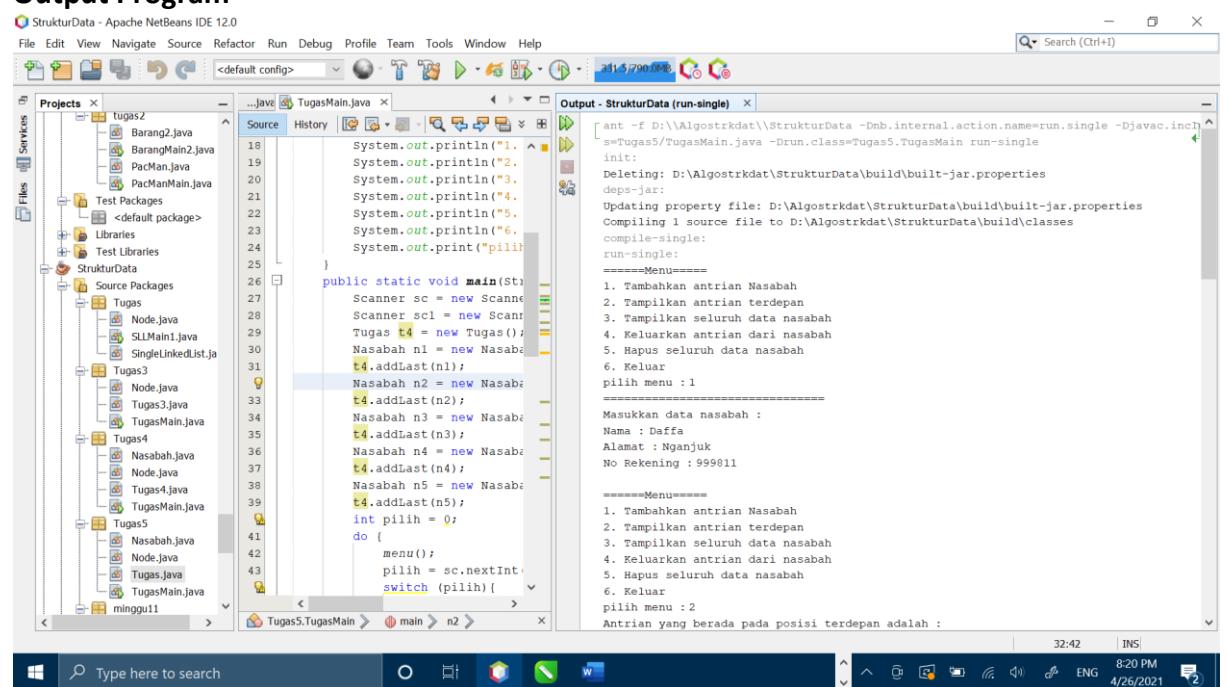
```

```

        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



The screenshot shows the Apache NetBeans IDE 12.0 interface. The left pane displays the project structure under the 'Projects' tab, showing several Java files across different packages like 'tugas2', 'StrukturData', and 'Tugas'. The right pane shows the 'Source' editor for the file `TugasMain.java`, which contains Java code for a queue system. Below the editor is the 'Output' window, which shows the command-line interface for running the program. The output includes the command used to run the program, the menu options displayed, user input for adding a customer ('Daffa'), and the final output message indicating the customer is at the front of the queue.

```

StrukturData - Apache NetBeans IDE 12.0.
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> Search (Ctrl+F)
Projects ...Java TugasMain.java ...
Source History ...
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(St
27. Scanner sc = new Scanne
28. Scanner sc1 = new Scanne
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();
44.     switch (pilih) {
45.         case 1 :
46.             t4.addlast(n1);
47.             System.out.println("Antrian
48.             break;
49.         case 2 :
50.             t4.removefront();
51.             System.out.println("Antrian
52.             break;
53.         case 3 :
54.             t4.print();
55.             break;
56.         case 4 :
57.             t4.remove();
58.             System.out.println("Antrian
59.             break;
60.         case 5 :
61.             t4.clear();
62.             System.out.println("");
63.             break;
64.         case 6 :
65.             break;
66.         default :
67.             System.out.println("Kode yang anda
68. masukkan salah");
69.             break;
70.     }
71. } while (pilih != 6);
72. }
73. }

```

Output - StrukturData (run-single) :

```

ant -f D:\Algostrikdat\StrukturData -Dnb.internal.action.name=run.single -Djavac.incl
seTugas5/TugasMain.java -Drun.class=Tugas5.TugasMain run-single
init:
Deleting: D:\Algostrikdat\StrukturData\build\built-jar.properties
deps-jar:
Updating property file: D:\Algostrikdat\StrukturData\build\built-jar.properties
Compiling 1 source file to D:\Algostrikdat\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 : 999011

=====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

```
=====
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
```

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

```
=====
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
```

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

=====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 : 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

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

=====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 : 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

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

=====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 : 6
BUILD SUCCESSFUL (total time: 1 minute 2 seconds)

```

C. Penjelasan

Pada program diatas terdapat 4 class yaitu :

- Class mahasiswa.** Terdapat 3 atribut yaitu int nim, String nama, dan double ipk. Dengan konstruktur mahasiswa berparameter (int nim, String nama, double ipk) this.nim = nim; this.nama = nama; this.ipk = ipk;. Pada class ini akan disambungkan ke class Node dengan mengganti salah satu tipe data pada atribut class Node.
- Class Node.** Yang terdapat 3 atribut yaitu mahasiswa data, Node prev, dan Node next. Dan di konstruktur dengan this.prev = prev; this.data = data; dan this.next = next;.
- Class NodeService.** Didalam class ini terdapat 2 atribut, 1 konstruktur, dan 11 method.

2 atribut diatas diantaranya adalah : Node head yang digunakan untuk menentukan awal pada node atau null, dan int size yang digunakan untuk menentukan Panjang dari data double linked list.

Terdapat 1 konstruktor yang berfungsi untuk menginisialisasikan head adalah null dan size = 0.

Dan ada 9 macam method, diantaranya yaitu :

- isEmpty() : digunakan untuk mengecek apakah ada data yang kosong.
 - addFirst() : digunakan untuk menambah data sebelum node head (prev).
 - addLast() : digunakan untuk menambah data sesudah node head (next).
 - add() : digunakan untuk menambah data di bagian node atau di indeks tertentu.
 - print() : menampilkan seluruh data pada double linked list.
 - removeFirst() : digunakan untuk menghapus atau mengeluarkan pada awal data pada double linked list.
 - removeLast() : digunakan untuk menghapus atau mengeluarkan pada akhir data pada double linked list.
 - Remove() : digunakan untuk menghapus atau mengeluarkan pada indeks data tertentu pada double linked list.
 - Size() : digunakan untuk mengetahui panjang data pada double linked list.
 - sortList() : digunakan untuk mengurutkan data double linked list menggunakan bubble sort.
 - searchNode(int nims) : digunakan untuk mencari node dengan menggunakan indeks node nims.
- d. Dan yang terakhir class Main yang digunakan untuk mengeluarkan script data dengan menggunakan menu. Dan didalam class Main terdapat 3 method fungsi yaitu :
- Method menu(), yang digunakan untuk list data menu apa saja yang akan ditampilkan di method data().
 - Method data(), yang digunakan untuk menginput data dari class lain untuk menghasilkan suatu data output an disetiap pilihan pada menu dan akan ditampilkan pada method main.
 - Method main, yang digunakan untuk mengeluarkan data dari method data().