

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOB SHEET PERTEMUAN KE-11



RIO TRI PRAYOGO

TI 1A

26

2341720236

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

Linked List

Praktikum 1 : Pembuatan Single Linked List

Percobaan :

Node26

```
package minggull1;

public class Node26 {

    int data;

    Node26 next;

    Node26(int nilai, Node26 berikutnya) {

        data = nilai;

        next = berikutnya;

    }

}
```

SingleLinkedList26

```
package minggull1;

public class SingleLinkedList26 {

    Node26 head, tail;

    boolean isEmpty() {

        return head != null;

    }

    void print() {

        if (isEmpty()) {

            Node26 tmp = head;

            System.out.print("Isi Linked List:\t");

            while (tmp != null) {

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

                tmp = tmp.next;

            }

            System.out.println("");

        } else {

            System.out.println("Linked List kosong");

        }

    }

    void addFirst(int input) {
```

```

Node26 ndInput = new Node26(input, null);
if (isEmpty()) {
    ndInput.next = head;
    head = ndInput;
} else {
    head = ndInput;
    tail = ndInput;
}
}

void addLast(int input) {
    Node26 ndInput = new Node26(input, null);
    if (isEmpty()) {
        tail.next = ndInput;
        tail = ndInput;
    } else {
        head = ndInput;
        tail = ndInput;
    }
}

void insertAfter(int key, int input) {
    Node26 ndInput = new Node26(input, null);
    Node26 temp = head;
    while (temp != null) {
        if (temp.data == key) {
            ndInput.next = temp.next;
            temp.next = ndInput;
            if (ndInput.next != null) {
                tail = ndInput;
            }
            break;
        }
        temp = temp.next;
    }
}

void insertAt(int index, int input) {
    if (index < 0) {

```

```

        System.out.println("Perbaiki logikanya!" + "Kalau indeksnya -1
bagaimana???");
    } else if (index == 0) {
        addFirst(input);
    } else {
        Node26 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = new Node26(input, temp.next);
        if (temp.next.next == null) {
            tail = temp.next;
        }
    }
}
}

```

SLLMain26

```

package minggull1;

public class SLLMain26 {
    public static void main(String[] args) {
        SingleLinkedList26 singLL = new SingleLinkedList26();

        singLL.print();
        singLL.addFirst(890);
        singLL.print();
        singLL.addLast(760);
        singLL.print();
        singLL.addFirst(700);
        singLL.print();
        singLL.insertAfter(700, 999);
        singLL.print();
        singLL.insertAt(3, 833);
        singLL.print();
    }
}

```

Output :

```
Linked List kosong
Isi Linked List:      890
Isi Linked List:      890      760
Isi Linked List:      700      890      760
Isi Linked List:      700      999      890      760
Isi Linked List:      700      999      890      833      760
```

Pertanyaan :

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
 - Karena pada class `SLLMain26.java` program memanggil method `print()` terlebih dahulu sebelum diisi dan *linked list* tidak diisi terlebih dahulu sehingga *linked list* yang diprint di awal tidak ada isinya/kosong dan memunculkan output “Linked List Kosong”.
2. Jelaskan kegunaan variable `temp` secara umum pada setiap method!
 - Variabel `temp` secara umum memiliki kegunaan untuk menyimpan data sementara. Hal tersebut berguna pada saat melakukan perubahan data. Pada saat perubahan data, data yang diubah dipindahkan terlebih dahulu ke dalam variabel `temp` yang membuat data yang telah diubah tidak hilang.
3. Perhatikan class **SingleLinkedList**, pada method **insertAt** Jelaskan kegunaan kode berikut

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

 - Baris kode diatas berguna untuk mengecek apakah data setelah data input kosong/atau tidak. `temp.next` merupakan data yang diinput, `temp.next.next` merupakan data dibelakang/setelah data input. Jika data setelah data yang diinputkan kosong, maka data yang diinputkan berubah menjadi `tail` atau ekor yang merupakan data yang paling belakang.

Praktikum 2 : Modifikasi Elemen pada Single Linked List

Percobaan :

“Baris kode di bawah merupakan tambahan yang dimasukkan ke dalam kode Praktikum 1”

SingleLinkedList26

```
...
int getData(int index) {
    Node26 tmp = head;
    for (int i = 0; i < index - 1; i++) {
        tmp = tmp.next;
    }
    return tmp.next.data;
}

int indexOf(int key) {
```

```

Node26 tmp = head;
int index = 0;
while (tmp != null && tmp.data != key) {
    tmp = tmp.next;
    index++;
}
if (tmp == null) {
    return 1;
} else {
    return index;
}
}

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

void removeLast() {
    if (!isEmpty()) {
        System.out.println("Linked List masih kosong," + "tidak dapat
dihapus");
    } else if (head == tail) {
        head = tail = null;
    } else {
        Node26 temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}

```

```

    }
}

void remove(int key) {
    if (!isEmpty()) {
        System.out.println("Linked List masih kosong," + "tidak dapat
dihapus");
    } else {
        Node26 temp = head;
        while (temp != null) {
            if (temp.data == key && temp == head) {
                removeFirst();
                break;
            } else if (temp.next.data == key) {
                temp.next = temp.next.next;
                if (temp.next == null) {
                    tail = temp;
                }
                break;
            }
            temp = temp.next;
        }
    }
}

public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        Node26 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}

```

```

    }
}
...

```

Output :

```

Linked List kosong
Isi Linked List:      890
Isi Linked List:      890      760
Isi Linked List:      700      890      760
Isi Linked List:      700      999      890      760
Isi Linked List:      700      999      890      833      760
Data pada indeks ke-1=999
Data 3 berada pada indeks ke-4
Isi Linked List:      700      890      833      760
Isi Linked List:      890      833      760
Isi Linked List:      833      760
Isi Linked List:      833

```

Pertanyaan :

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
 - Keyword `break` pada fungsi `remove()` berfungsi untuk menghentikan loop `while` pada saat kondisinya terpenuhi. Kondisi yang dimaksud adalah `temp != null` atau `temp` tidak kosong. Jika kondisi tersebut sudah terpenuhi maka elemen yang ingin dihapus sudah ditemukan dan tidak diperlukan lagi loop tersebut. Jika tidak ada `break` maka output akan error seperti gambar dibawah ini:

```

Exception in thread "main" java.lang.NullPointerException: Cannot read field "data" because "temp.next" is null
    at minggu11.SingleLinkedList26.remove(SingleLinkedList26.java:135)
    at minggu11.SLLMain26.main(SLLMain26.java:22)

```

2. Jelaskan kegunaan kode dibawah pada method remove

```

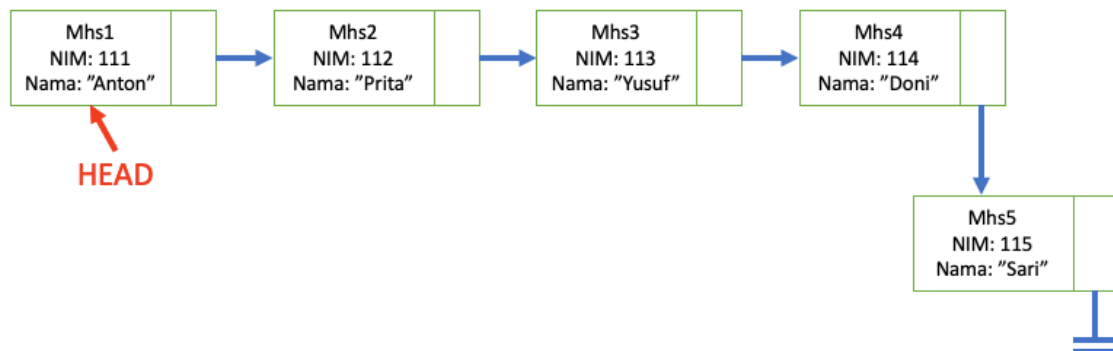
else if (temp.next.data == key) {
    temp.next = temp.next.next;

```

- Baris kode diatas berfungsi untuk untuk mengecek apakah data setelah data `temp` merupakan data yang dicari. `temp` disini dideklarasikan sebagai `head` atau data awal. Jika kondisi tersebut terpenuhi maka data setelah data yang dicari akan diubah tempatnya (`temp.next = temp.next.next`) dan data yang dicari akan terhapus/hilang.

Tugas

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



➤ Code

Node26

```
package minggul1.tugas1;

public class Node26 {
    Mahasiswa26 data;
    Node26 next;

    Node26(Mahasiswa26 nilai, Node26 berikutnya) {
        data = nilai;
        next = berikutnya;
    }
}
```

Mahasiswa26

```
package minggul1.tugas1;

public class Mahasiswa26 {
    int nim;
    String nama;

    Mahasiswa26(int nim, String nama) {
        this.nim = nim;
        this.nama = nama;
    }
}
```

LinkedList26

```
package minggull.tugas1;

public class LinkedList26 {
    Node26 head, tail;

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

    void print() {
        if (isEmpty()) {
            Node26 tmp = head;
            int index = 1;
            System.out.println("Isi Linked List:\t");
            while (tmp != null) {
                Mahasiswa26 mhs = tmp.data;
                System.out.println("Mahasiswa ke-" + index);
                System.out.println("NIM: " + mhs.nim + ", Nama: " +
mhs.nama);
                tmp = tmp.next;
                index++;
            }
            System.out.println("");
        } else {
            System.out.println("Linked List kosong\n");
        }
    }

    void addFirst(Mahasiswa26 input) {
        Node26 ndInput = new Node26(input, null);
        if (isEmpty()) {
            ndInput.next = head;
            head = ndInput;
        } else {
            head = ndInput;
            tail = ndInput;
        }
    }
}
```

```

    }
}

void addLast(Mahasiswa26 input) {
    Node26 ndInput = new Node26(input, null);
    if (isEmpty()) {
        tail.next = ndInput;
        tail = ndInput;
    } else {
        head = ndInput;
        tail = ndInput;
    }
}

void insertAfter(Mahasiswa26 key, Mahasiswa26 input) {
    Node26 temp = head;
    while (temp != null) {
        if (temp.data == key) {
            Node26 ndInput = new Node26(input, temp.next);
            ndInput.next = temp.next;
            temp.next = ndInput;
            if (ndInput.next != null) {
                tail = ndInput;
            }
            break;
        }
        temp = temp.next;
    }
}

void insertAt(int index, Mahasiswa26 input) {
    if (index < 0) {
        System.out.println("Perbaiki logikanya!" + "Kalau indeksnya -1 bagaimana???");
    } else if (index == 0) {
        addFirst(input);
    }
}

```

```

    } else {
        Node26 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = new Node26(input, temp.next);
        if (temp.next.next == null) {
            tail = temp.next;
        }
    }
}
}

```

Main26

```

package minggull1.tugas1;

public class Main26 {

    public static void main(String[] args) {
        LinkedList26 list = new LinkedList26();
        Mahasiswa26 mhs1 = new Mahasiswa26(113, "Yusuf") ;
        Mahasiswa26 mhs2 = new Mahasiswa26(115, "Sari") ;
        Mahasiswa26 mhs3 = new Mahasiswa26(111, "Anton") ;
        Mahasiswa26 mhs4 = new Mahasiswa26(112, "Prita") ;
        Mahasiswa26 mhs5 = new Mahasiswa26(114, "Doni") ;

        list.print();
        list.addFirst(mhs1);
        list.print();
        list.addLast(mhs2);
        list.print();
        list.addFirst(mhs3);
        list.print();
        list.insertAfter(mhs3, mhs4);
        list.print();
        list.insertAt(3, mhs5);
        list.print();
    }
}

```

➤ **Output**

Linked List kosong	Isi Linked List: Mahasiswa ke-1 NIM: 111, Nama: Anton Mahasiswa ke-2 NIM: 112, Nama: Prita Mahasiswa ke-3 NIM: 113, Nama: Yusuf Mahasiswa ke-4 NIM: 115, Nama: Sari
Isi Linked List: Mahasiswa ke-1 NIM: 113, Nama: Yusuf Mahasiswa ke-2 NIM: 115, Nama: Sari	Isi Linked List: Mahasiswa ke-1 NIM: 111, Nama: Anton Mahasiswa ke-2 NIM: 112, Nama: Prita Mahasiswa ke-3 NIM: 113, Nama: Yusuf Mahasiswa ke-4 NIM: 114, Nama: Doni Mahasiswa ke-5 NIM: 115, Nama: Sari
Isi Linked List: Mahasiswa ke-1 NIM: 111, Nama: Anton Mahasiswa ke-2 NIM: 113, Nama: Yusuf Mahasiswa ke-3 NIM: 115, Nama: Sari	

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

➤ **Code**

Node26

```
package minggul1.tugas2;

public class Node26 {
    Mahasiswa26 data;
    Node26 next;

    Node26(Mahasiswa26 nilai, Node26 berikutnya) {
        data = nilai;
        next = berikutnya;
    }
}
```

Mahasiswa26

```
package minggul1.tugas2;

public class Mahasiswa26 {
    String nim;
    String nama;

    Mahasiswa26(String nim, String nama) {
        this.nim = nim;
        this.nama = nama;
    }
}
```

LinkedList26

```
package minggull1.tugas2;

public class LinkedList26 {
    Node26 head, tail;

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

    void print() {
        if (isEmpty()) {
            Node26 tmp = head;
            int index = 1;
            System.out.println("Isi Antrian:");
            System.out.println("-----");
            while (tmp != null) {
                Mahasiswa26 mhs = tmp.data;
                System.out.println("Antrian Mahasiswa ke-" + index);
                System.out.println("NIM: " + mhs.nim + ", Nama: " +
mhs.nama);
                tmp = tmp.next;
                index++;
            }
        } else {
            System.out.println("Antrian kosong");
        }
    }

    int idx() {
        int count = 0;
        Node26 temp = head;
        while (temp != null) {
            count++;
            temp = temp.next;
        }
        return count;
    }
}
```

```

    }

    void addLast(Mahasiswa26 input) {
        Node26 ndInput = new Node26(input, null);
        if (isEmpty()) {
            tail.next = ndInput;
            tail = ndInput;
            System.out.println("-----");
            System.out.println("Antrian berhasil ditambahkan!\nNomor antrian: " + idx());
        } else {
            head = ndInput;
            tail = ndInput;
            System.out.println("-----");
            System.out.println("Antrian berhasil ditambahkan!\nNomor antrian: 1");
        }
    }

    Mahasiswa26 getData(int index) {
        Node26 tmp = head;
        for (int i = 0; i < index; i++) {
            if (tmp == null) {
                return null;
            }
            tmp = tmp.next;
        }
        return tmp != null ? tmp.data : null;
    }

    int indexOf(String key) {
        Node26 tmp = head;
        int index = 0;
        while (tmp != null) {
            if (tmp.data.nim.equals(key) || tmp.data.nama.equals(key)) {
                return index;
            }
        }
    }

```

```

        }

        tmp = tmp.next;

        index++;

    }

    return -1;

}

void removeFirst() {
    if (!isEmpty()) {
        System.out.println("Antrian masih kosong," + "tidak dapat
dihapus");
    } else if (head == tail) {
        Mahasiswa26 mhs = head.data;

        System.out.println("Antrian dengan NIM: " + mhs.nim + "
dan Nama: " + mhs.nama + " berhasil dihapus!");

        head = tail = null;
    } else {
        Mahasiswa26 mhs = head.data;

        System.out.println("Antrian dengan NIM: " + mhs.nim + "
dan Nama: " + mhs.nama + " berhasil dihapus!");

        head = head.next;
    }
}

void removeLast() {
    if (!isEmpty()) {
        System.out.println("Antrian masih kosong," + "tidak dapat
dihapus");
    } else if (head == tail) {
        Mahasiswa26 mhs = tail.data;

        System.out.println("Antrian dengan NIM: " + mhs.nim + "
dan Nama: " + mhs.nama + " berhasil dihapus!");

        head = tail = null;
    } else {
        Mahasiswa26 mhs = tail.data;

        System.out.println("Antrian dengan NIM: " + mhs.nim + "
dan Nama: " + mhs.nama + " berhasil dihapus!");

        Node26 temp = head;

        while (temp.next != tail) {

```



```

        }

        temp.next = null;
        tail = temp;
    }
}

void remove(Mahasiswa26 mhs) {
    if (!isEmpty()) {
        System.out.println("Antrian masih kosong, tidak dapat
dihapus");

        System.out.println("-----");
    } else {
        Node26 temp = head;
        Node26 prev = null;
        while (temp != null) {
            if (temp.data.nim.equals(mhs.nim) ||
temp.data.nama.equals(mhs.nama)) {
                if (temp == head) {
                    removeFirst();
                } else {
                    prev.next = temp.next;
                    if (temp.next == null) {
                        tail = prev;
                    }
                }
                break;
            }
            prev = temp;
            temp = temp.next;
        }
    }
}

void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    }
}

```

```

        System.out.println("Antrian ke-" + (index + 1) + "
berhasil dihapus!");
    } else {
        Node26 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
        System.out.println("Antrian ke-" + (index + 1) + "
berhasil dihapus!");
    }
}
}
}

```

Main26

```

package minggull1.tugas2;
import java.util.Scanner;

public class Main26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        LinkedList26 list = new LinkedList26();
        String nim, nama;
        boolean run = true;
        do {
            System.out.println("\n-----");
            System.out.println("Selamat datang di program antrian
Layanan Unit Kemahasiswaan!");
            System.out.println("-----");
            System.out.println("Pilih layanan: ");
            System.out.println("1. Input antrian");
            System.out.println("2. Hapus antrian");
            System.out.println("3. Lihat antrian");
            System.out.println("4. Cari antrian");
            System.out.println("0. Keluar");

```

```

        System.out.print("(1/2/3/0): ");
        int input = scan.nextInt();
        System.out.println("-----");
        switch (input) {
            case 1:
                System.out.println("Masukkan antrian: ");
                System.out.println("-----");

                System.out.print("Input NIM: ");
                scan.nextLine();
                nim = scan.nextLine();
                System.out.print("Input Nama: ");
                nama = scan.nextLine();
                Mahasiswa26 mhs = new Mahasiswa26(nim, nama);
                list.addLast(mhs);
                System.out.println("-----");

                break;
            case 2:
                boolean kembali = true;
                do {
                    System.out.println("Hapus antrian
berdasarkan: ");
                    System.out.println("-----");

                    System.out.println("1. Antrian terdepan");
                    System.out.println("2. Antrian paling
belakang");

                    System.out.println("3. Nomor antrian");
                    System.out.println("4. NIM>Nama");
                    System.out.println("0. Kembali");
                    System.out.print("(1/2/3/4/0): ");
                    int hapusAntri = scan.nextInt();
                    System.out.println("-----");

                    switch (hapusAntri) {
                        case 1:
                            System.out.println("Hapus antrian
paling depan: ");

```

```

-----");
                                System.out.println("-----");
                                list.removeFirst();
                                System.out.println("-----");
-----");
                                kembali = false;
                                break;
                                case 2:
                                System.out.println("Hapus antrian
paling belakang: ");
                                System.out.println("-----");
                                list.removeLast();
                                System.out.println("-----");
-----");
                                kembali = false;
                                break;
                                case 3:
                                System.out.println("Hapus antrian
berdasarkan nomor antrian: ");
                                System.out.println("-----");
                                list.print();
                                System.out.println("-----");
-----");
                                System.out.print("Masukkan nomor
antrian: ");
                                int inputNo = scan.nextInt();
                                list.removeAt((inputNo - 1));
                                System.out.println("-----");
-----");
                                kembali = false;
                                break;
                                case 4:
                                System.out.println("Hapus antrian
berdasarkan NIM>Nama: ");
                                System.out.println("-----");
-----");
                                list.print();
                                System.out.println("-----");
-----");

```

```

        System.out.print("Masukkan NIM>Nama:
");

        scan.nextLine();
        String inputNm = scan.nextLine();
        Mahasiswa26 hpsMhs = new
Mahasiswa26(inputNm, inputNm);

        list.remove(hpsMhs);
        System.out.println("-----
-----");

        kembali = false;
        break;

        case 0:
            System.out.println("Kembali ke menu
awal!");

            System.out.println("-----
-----");

            kembali = false;
            break;

            default:
                System.out.println("Pilihan tidak
valid!");

                System.out.println("-----
-----");

                }
            } while (kembali);
            break;

        case 3:
            list.print();
            System.out.println("-----
");

            break;

        case 4:
            boolean back = true;
            do {
                System.out.println("Cari antrian berdasarkan:
");

                System.out.println("-----
----");

                System.out.println("1. Nomor antrian");
                System.out.println("2. NIM>Nama");

```

```

        System.out.println("0. Kembali");
        System.out.print("(1/2/0): ");
        int cariAntri = scan.nextInt();
        System.out.println("-----
----");

        switch (cariAntri) {
            case 1:
                System.out.println("Cari antrian
berdasarkan nomor antrian: ");
                System.out.println("-----
-----");
                list.print();
                System.out.println("-----
-----");
                System.out.print("Masukkan nomor
antrian: ");
                int cariNo = scan.nextInt();
                System.out.println("-----
-----");
                Mahasiswa26 noMhs =
list.getData(cariNo - 1);
                if (noMhs != null) {
                    System.out.println("Antrian ke-"
+ cariNo + " ditemukan!");
                    System.out.println("NIM: " +
noMhs.nim + ", Nama: " + noMhs.nama);
                } else {
                    System.out.println("Antrian ke-"
+ cariNo + " tidak ditemukan!");
                }
                System.out.println("-----
-----");
                back = false;
                break;
            case 2:
                System.out.println("Cari antrian
berdasarkan NIM>Nama: ");
                System.out.println("-----
-----");
                list.print();
                System.out.println("-----
-----");

```

```

        System.out.print("Masukkan NIM>Nama:
");

        scan.nextLine();
        String cariNm = scan.nextLine();
        int posisi = list.indexOf(cariNm);
        if (posisi != -1) {
            Mahasiswa26 dataMhs =
list.getData(posisi);

            System.out.println("Antrian
dengan NIM>Nama tersebut ditemukan pada posisi ke-"
+ (posisi + 1));

            System.out.println("Pemilik
antrian NIM: " + dataMhs.nim + " dan Nama: "
+ dataMhs.nama + "
ditemukan!");
        } else {
            System.out.println("Antrian
dengan NIM>Nama tersebut tidak ditemukan!");
        }

        System.out.println("-----
-----");

        back = false;
        break;

        case 0:
            System.out.println("Kembali ke menu
awal!");

            System.out.println("-----
-----");

            back = false;
            break;

            default:
                System.out.println("Pilihan tidak
valid!");

                System.out.println("-----
-----");

        }

        } while (back);

        break;

        case 0:
            System.out.println("Keluar
program\nTerimakasih!");

```

```

        System.out.println("-----");
    );

    run = false;

    break;

default:

    System.out.println("Pilihan tidak valid!");

    System.out.println("-----");

);

    }

    } while (run);

}

}

```

➤ Output

- Input antrian

<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 1 ----- Masukkan antrian: ----- Input NIM: 111 Input Nama: Rio ----- Antrian berhasil ditambahkan! Nomor antrian: 1 </pre>	<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 1 ----- Masukkan antrian: ----- Input NIM: 222 Input Nama: Ali ----- Antrian berhasil ditambahkan! Nomor antrian: 2 </pre>	<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 1 ----- Masukkan antrian: ----- Input NIM: 333 Input Nama: Dina ----- Antrian berhasil ditambahkan! Nomor antrian: 3 </pre>
---	---	--

- Lihat antrian

```

Selamat datang di program antrian Layanan Unit Kemahasiswaan!
-----
Pilih layanan:
1. Input antrian
2. Hapus antrian
3. Lihat antrian
4. Cari antrian
0. Keluar
(1/2/3/0): 3
-----
Isi Antrian:
-----
Antrian Mahasiswa ke-1
NIM: 111, Nama: Rio
Antrian Mahasiswa ke-2
NIM: 222, Nama: Ali
Antrian Mahasiswa ke-3
NIM: 333, Nama: Dina
-----

```

- Hapus antrian

```

Selamat datang di program antrian Layanan Unit Kemahasiswaan!
-----
Pilih layanan:
1. Input antrian
2. Hapus antrian
3. Lihat antrian
4. Cari antrian
0. Keluar
(1/2/3/0): 2
-----
Hapus antrian berdasarkan:
-----
1. Antrian terdepan
2. Antrian paling belakang
3. Nomor antrian
4. NIM/Nama
0. Kembali
(1/2/3/4/0): 1
-----

```

a. Paling depan

<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3 ----- Isi Antrian: ----- Antrian Mahasiswa ke-1 NIM: 111, Nama: Rio Antrian Mahasiswa ke-2 NIM: 222, Nama: Ali Antrian Mahasiswa ke-3 NIM: 333, Nama: Dina ----- </pre>	<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 2 ----- Hapus antrian berdasarkan: ----- 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM/Nama 0. Kembali (1/2/3/4/0): 1 ----- Hapus antrian paling depan: ----- Antrian dengan NIM: 111 dan Nama: Rio berhasil dihapus! </pre>	<pre> Selamat datang di program antrian Layanan Unit Kemahasiswaan! ----- Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3 ----- Isi Antrian: ----- Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali Antrian Mahasiswa ke-2 NIM: 333, Nama: Dina ----- </pre>
--	--	---

b. Paling belakang

Selamat datang di program antrian Layanan Unit Kemahasiswaan!	Selamat datang di program antrian Layanan Unit Kemahasiswaan!	Selamat datang di program antrian Layanan Unit Kemahasiswaan!
Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3	Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 2	Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3
Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali Antrian Mahasiswa ke-2 NIM: 333, Nama: Dina	Hapus antrian berdasarkan: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 2	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali
	Hapus antrian paling belakang: Antrian dengan NIM: 333 dan Nama: Dina berhasil dihapus!	

c. Nomor antrian

Selamat datang di program antrian Layanan Unit Kemahasiswaan!	Selamat datang di program antrian Layanan Unit Kemahasiswaan!	Selamat datang di program antrian Layanan Unit Kemahasiswaan!
Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3	Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 2	Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3
Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali Antrian Mahasiswa ke-2 NIM: 444, Nama: Toni	Hapus antrian berdasarkan nomor antrian: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 3	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali
	Hapus antrian berdasarkan nomor antrian: Antrian ke-2 berhasil dihapus!	

d. NIM>Nama

Selamat datang di program antrian Layanan Unit Kemahasiswaan!	Hapus antrian berdasarkan: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 4	Hapus antrian berdasarkan: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 4
Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3	Hapus antrian berdasarkan NIM>Nama: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 4	Hapus antrian berdasarkan NIM>Nama: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 4
Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali
Antrian Mahasiswa ke-1 NIM: 222, Nama: Ali	Masukkan NIM>Nama: 222 Antrian dengan NIM: 222 dan Nama: Ali berhasil dihapus!	Masukkan NIM>Nama: Ali Antrian dengan NIM: 222 dan Nama: Ali berhasil dihapus!
Selamat datang di program antrian Layanan Unit Kemahasiswaan!		
Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 3		
Antrian kosong		

- Cari antrian

Selamat datang di program antrian Layanan Unit Kemahasiswaan!
Pilih layanan: 1. Input antrian 2. Hapus antrian 3. Lihat antrian 4. Cari antrian 0. Keluar (1/2/3/0): 4
Cari antrian berdasarkan: 1. Nomor antrian 2. NIM>Nama 0. Kembali (1/2/0): 1
Cari antrian berdasarkan nomor antrian: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 1
Isi Antrian: Antrian Mahasiswa ke-1 NIM: 111, Nama: Rio Antrian Mahasiswa ke-2 NIM: 222, Nama: Dina
Masukkan nomor antrian: 2 Antrian ke-2 ditemukan! NIM: 222, Nama: Dina

Nomor antrian	NIM	Nama
Cari antrian berdasarkan: 1. Nomor antrian 2. NIM>Nama 0. Kembali (1/2/0): 1	Cari antrian berdasarkan: 1. Nomor antrian 2. NIM>Nama 0. Kembali (1/2/0): 2	Cari antrian berdasarkan: 1. Nomor antrian 2. NIM>Nama 0. Kembali (1/2/0): 2
Cari antrian berdasarkan nomor antrian: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 1	Cari antrian berdasarkan NIM>Nama: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 1	Cari antrian berdasarkan NIM>Nama: 1. Antrian terdepan 2. Antrian paling belakang 3. Nomor antrian 4. NIM>Nama 0. Kembali (1/2/3/4/0): 1
Isi Antrian: Antrian Mahasiswa ke-1 NIM: 111, Nama: Rio Antrian Mahasiswa ke-2 NIM: 222, Nama: Dina	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 111, Nama: Rio Antrian Mahasiswa ke-2 NIM: 222, Nama: Dina	Isi Antrian: Antrian Mahasiswa ke-1 NIM: 111, Nama: Rio Antrian Mahasiswa ke-2 NIM: 222, Nama: Dina
Masukkan nomor antrian: 2 Antrian ke-2 ditemukan! NIM: 222, Nama: Dina	Masukkan NIM>Nama: 222 Antrian dengan NIM>Nama tersebut ditemukan pada posisi ke-2 Pemilik antrian NIM: 222 dan Nama: Dina ditemukan!	Masukkan NIM>Nama: Rio Antrian dengan NIM>Nama tersebut ditemukan pada posisi ke-1 Pemilik antrian NIM: 111 dan Nama: Rio ditemukan!