

Nama: Wahda Adella Putri Febriana

Kelas: 1B / 24

NIM: 2441070202156

Praktikum 1

1. Buat file Mahasiswa24.java beserta isinya

```
public class Mahasiswa24 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa24(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
    }

    void tampilkanInformasi() {
        System.out.println(nim + " - " + nama + " - " + kelas + " - " + ipk);
    }
}
```

2. Buat File Node24.java dan isinya

```
public class Node24 {
    Mahasiswa24 data;
    Node24 next;

    Node24(Mahasiswa24 data, Node24 next) {
        this.data = data;
        this.next = next;
    }
}
```

3. Buat file SingleLinkedList24.java dan isinya

```
public class SingleLinkedList24 {
    boolean isEmpty() {
        return (head == null);
    }

    void print() {
        if (!isEmpty()) {
            Node24 tmp = head;
            System.out.println("Isi linked list : \t");
            while (tmp != null) {
                tmp.data.tampilkanInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        } else {
            System.out.println("Linked list kosong");
        }
    }

    void addFirst(Mahasiswa24 input) {
        Node24 ndInput = new Node24(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    void addLast(Mahasiswa24 input) {
        Node24 ndInput = new Node24(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    void insertAfter(String key, Mahasiswa24 input) {
        Node24 ndInput = new Node24(input, null);
        Node24 temp = head;
        do {
            if (temp.data.nama.equalsIgnoreCase(key)) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if (ndInput.next == null) {
                    tail = ndInput;
                }
                break;
            }
            temp = temp.next;
        } while (temp != null);
    }

    void insertAt(int index, Mahasiswa24 input) {
        if (index < 0) {
            System.out.println("Index Salah");
        } else if (index == 0) {
            addFirst(input);
        } else {
            Node24 temp = head;
            for (int i = 0; i < index - 1; i++) {
                temp = temp.next;
            }
            temp.next = new Node24(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
}
```

4. Buat file SLLMain24.java dan isinya

```
Main24.java > SLLMain24
public class SLLMain24 {
    public static void main(String[] args) {
        SingleLinkedList24 s1 = new SingleLinkedList24();
        s1.print();
        Mahasiswa24 mhs1 = new Mahasiswa24(nim:"12345", nama:"Wahda", kelas:"1B");
        s1.addFirst(mhs1);
        s1.print();
        Mahasiswa24 mhs2 = new Mahasiswa24(nim:"22345", nama:"Galih", kelas:"1C");
        s1.addLast(mhs2);
        s1.print();
        Mahasiswa24 mhs3 = new Mahasiswa24(nim:"32345", nama:"Ega", kelas:"1D");
        s1.insertAfter(key:"Galih", mhs3);
        s1.print();
        Mahasiswa24 mhs4 = new Mahasiswa24(nim:"42345", nama:"Sandy", kelas:"1E");
        s1.insertAt(index:2, mhs4);
        s1.print();
    }
}
```

5. Hasil Run

```
java (jre-7s (3083ncc11_3083201
Linked list kosong
Isi linked list :
12345 - Wahda - 1B - 0.0

Isi linked list :
12345 - Wahda - 1B - 0.0
22345 - Galih - 1C - 0.0

Isi linked list :
12345 - Wahda - 1B - 0.0
32345 - Ega - 1D - 0.0
22345 - Galih - 1C - 0.0

Isi linked list :
12345 - Wahda - 1B - 0.0
32345 - Ega - 1D - 0.0
42345 - Sandy - 1E - 0.0
22345 - Galih - 1C - 0.0
```

Pertanyaan

1. Karena LinkedList belum diisi dengan data apapun
2. Untuk mengolah data tanpa mengubah data aslinya

3. Hasil Run

```
Linked list kosong
Isi linked list :
12345 - Wahda - 1B - 4.0

Isi linked list :
12345 - Wahda - 1B - 4.0
22345 - Galih - 1C - 3.9

Isi linked list :
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Isi linked list :
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Apakah ingin Menambah data baru? (ya / tidak): ya
Masukkan nama mahasiswa :
eka
Masukkan nim mahasiswa :
11
Masukkan kelas mahasiswa :
1
Masukkan ipk mahasiswa :
1

Silahkan pilih menu:
1. Menambahkan data Diawal
2. Menambahkan data Setelah Nama Tertentu
3. Menambahkan data di index tertentu
4. Menambahkan data di Akhir
Masukkan Pilihan mu : 1

Isi linked list :
11 - eka - 1 - 1.0
12 - nikma - 2 - 2.0
12345 - Wahda - 1B - 4.0
13 - adel - 3 - 3.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9
14 - fika - 4 - 4.0

Isi linked list :
11 - eka - 1 - 1.0
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Isi linked list :
11 - eka - 1 - 1.0
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Apakah ingin Menambah data baru? (ya / tidak): ya
Masukkan nama mahasiswa :
nikma
Masukkan nim mahasiswa :
12
Masukkan kelas mahasiswa :
2
Masukkan ipk mahasiswa :
2

Silahkan pilih menu:
1. Menambahkan data Diawal
2. Menambahkan data Setelah Nama Tertentu
3. Menambahkan data di index tertentu
4. Menambahkan data di Akhir
Masukkan Pilihan mu : 2
Masukkan Nama yang ingin di cari : eka
Isi linked list :
11 - eka - 1 - 1.0
12 - nikma - 2 - 2.0
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Apakah ingin Menambah data baru? (ya / tidak): ya
Masukkan nama mahasiswa :
adel
Masukkan nim mahasiswa :
13

Isi linked list :
11 - eka - 1 - 1.0
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Masukkan kelas mahasiswa :
3
Masukkan ipk mahasiswa :
3

Silahkan pilih menu:
1. Menambahkan data Diawal
2. Menambahkan data Setelah Nama Tertentu
3. Menambahkan data di index tertentu
4. Menambahkan data di Akhir
Masukkan Pilihan mu : 3
Masukkan index tempat : 3
Isi linked list :
11 - eka - 1 - 1.0
12 - nikma - 2 - 2.0
12345 - Wahda - 1B - 4.0
13 - adel - 3 - 3.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Apakah ingin Menambah data baru? (ya / tidak): ya
Masukkan nama mahasiswa :
fika
Masukkan nim mahasiswa :
14
Masukkan kelas mahasiswa :
4
Masukkan ipk mahasiswa :
4

Silahkan pilih menu:
1. Menambahkan data Diawal
2. Menambahkan data Setelah Nama Tertentu
3. Menambahkan data di index tertentu
4. Menambahkan data di Akhir
Masukkan Pilihan mu : 4
```

Code

```
String jawab = "";
do {
    System.out.print(s:"Apakah ingin Menambah data baru? (ya / tidak): ");
    jawab = input.nextLine();

    System.out.println(s:"Masukkan nama mahasiswa : ");
    String nama = input.nextLine();
    System.out.println(s:"Masukkan nim mahasiswa : ");
    String nim = input.nextLine();
    System.out.println(s:"Masukkan kelas mahasiswa : ");
    String kelas = input.nextLine();
    System.out.println(s:"Masukkan ipk mahasiswa : ");
    double ipk = input.nextDouble();

    Mahasiswa mhs = new Mahasiswa24(nim, nama, kelas, ipk);

    System.out.println();

    System.out.println(s:"Silahkan pilih menu: ");
    System.out.println(s:"1. Menambahkan data Diawal");
    System.out.println(s:"2. Menambahkan data Setelah Nama Tertentu");
    System.out.println(s:"3. Menambahkan data di index tertentu");
    System.out.println(s:"4. Menambahkan data di Akhir");
    System.out.print(s:"Masukkan Pilihan mu : ");
    int inputJawaban = input.nextInt();
    input.nextLine();
    switch (inputJawaban) {
        case 1:
            sl.addFirst(mhs);
            sl.print();
            break;
        case 2:
            System.out.print(s:"Masukkan Nama yang ingin di cari : ");
            String dicari = input.nextLine();
            sl.insertAfter(dicari, mhs);
            sl.print();
            break;
        case 3:
            System.out.print(s:"Masukkan index tempat : ");
            int index = input.nextInt();
            input.nextLine();
            sl.insertAt(index, mhs);
            sl.print();
            break;
        case 4:
            sl.addLast(mhs);
            sl.print();
            break;
        default:
            break;
    }
} while (jawab.equalsIgnoreCase("ya"));
```

Praktikum 2

1. Tambahkan kode di SingleLinkedList24.java

```
void getData(int index) {
    Node24 tmp = head;
    for(int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilkanInformasi();
}

int indexOf(String key) {
    Node24 tmp = head;
    int index = 0;
    while(tmp != null && !tmp.data.nama.equalsIgnoreCase(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 {
        Node24 tmp = head;
        while(tmp.next != tail) {
            tmp = tmp.next;
        }
        tmp.next = null;
        tail = tmp;
    }
}

void remove(String key) {
    if(isEmpty()) {
        System.out.println("Linked List masih kosong, tidak dapat dihapus");
    } else {
        Node24 tmp = head;
        while(tmp != null) {
            if((tmp.data.nama.equalsIgnoreCase(key)) && (tmp == head)) {
                tmp.next = tmp.next.next;
                if(tmp.next == null) {
                    tail = tmp;
                }
                break;
            }
            tmp = tmp.next;
        }
    }
}

void removeAt(int index) {
    if(index == 0) {
        removeFirst();
    } else {
        Node24 tmp = head;
        for(int i = 0; i < index - 1; i++) {
            tmp = tmp.next;
        }
        tmp.next = tmp.next.next;
        if(tmp.next == null) {
            tail = tmp;
        }
    }
}
```

2. Tambahkan kode di SLLMain24.java

```
System.out.println(x:"Data index 1: ");
s1.getData(index:1);

System.out.println("Data mahasiswa wahda berada pada index: " + s1.indexOf(key:"Wahda"));
System.out.println();

s1.removeFirst();
s1.removeLast();
s1.print();
s1.removeAt(index:0);
s1.print();
```

3. Hasil Run

```
Isi linked list :
12345 - Wahda - 1B - 4.0
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8
22345 - Galih - 1C - 3.9

Data index 1:
32345 - Ega - 1D - 4.0
Data mahasiswa wahda berada pada index: 0

Isi linked list :
32345 - Ega - 1D - 4.0
42345 - Sandy - 1E - 3.8

Isi linked list :
42345 - Sandy - 1E - 3.8
```

Pertanyaan

1. Agar while tidak dijalankan terus
2. Membuat tmp.next dengan data yang ada di tmp.next.next, dan jika tmp.next adalah null berarti tmp adalah data paling terakhir, makanya tmp di jadikan tail

Tugas

1. Hasil Run

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 1
Masukkan Nama Mahasiswa: a
Masukkan NIM Mahasiswa: 1
Masukkan Kelas Mahasiswa: 1
Masukkan ipk Mahasiswa: 1
Mahasiswa berhasil ditambahkan ke antrian.

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 1
Masukkan Nama Mahasiswa: b
Masukkan NIM Mahasiswa: 2
Masukkan Kelas Mahasiswa: 2
Masukkan ipk Mahasiswa: 2
Mahasiswa berhasil ditambahkan ke antrian.
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 3
Antrian tidak kosong.

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 5
Mahasiswa terdepan:
a - 1 - 1 - 1.0

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 6
Mahasiswa terakhir:
b - 2 - 2 - 2.0
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 7
Jumlah mahasiswa dalam antrian: 2

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 8
Daftar mahasiswa dalam antrian:
a - 1 - 1 - 1.0
b - 2 - 2 - 2.0

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 2
Memanggil mahasiswa:
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 5
Mahasiswa terdepan:
b - 2 - 2 - 2.0
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 3
Antrian tidak kosong.

===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 4
Antrian telah dikosongkan.
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 3
Antrian kosong.
```

```
===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====
1. Tambahkan Mahasiswa ke Antrian
2. Panggil Mahasiswa
3. Cek Antrian Kosong
4. Kosongkan Antrian
5. Tampilkan Mahasiswa Terdepan
6. Tampilkan Mahasiswa Terakhir
7. Tampilkan Jumlah Antrian
8. Tampilkan Semua Antrian
0. Keluar
Pilihan Anda: 8
Antrian kosong.
```

2. Isi file Antrian24.java

```
public class Antrian24 {
    Node24 front, rear;
    int size = 0;

    public Antrian24() {
        this.front = null;
        this.rear = null;
        this.size = 0;
    }

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

    public void enqueue(Mahasiswa24 mhs) {
        Node24 newNode = new Node24(mhs, null);
        if (rear == null) {
            front = rear = newNode;
        } else {
            rear.next = newNode;
            rear = newNode;
        }
        size++;
        System.out.println(x:"Mahasiswa berhasil ditambahkan ke antrian.");
    }

    public void dequeue() {
        if (isEmpty()) {
            System.out.println(x:"Antrian kosong. Tidak ada yang bisa dipanggil.");
            return;
        }
        System.out.println(x:"Memanggil mahasiswa: ");
        front.data.tampilkanInformasi();
        front = front.next;
        if (front == null) {
            rear = null;
        }
    }
}
```

```
public void tampilkanDepan() {
    if (isEmpty()) {
        System.out.println(x:"Antrian kosong.");
    } else {
        System.out.println(x:"Mahasiswa terdepan: ");
        front.data.tampilkanInformasi();
    }
}

public void tampilkanBelakang() {
    if (isEmpty()) {
        System.out.println(x:"Antrian kosong.");
    } else {
        System.out.println(x:"Mahasiswa terakhir: ");
        rear.data.tampilkanInformasi();
    }
}

public void clear() {
    front = rear = null;
    size = 0;
    System.out.println(x:"Antrian telah dikosongkan.");
}

public void jumlahAntrian() {
    System.out.println("Jumlah mahasiswa dalam antrian: " + size);
}

public void tampilkanSemua() {
    if (isEmpty()) {
        System.out.println(x:"Antrian kosong.");
        return;
    }
    Node24 temp = front;
    System.out.println(x:"Daftar mahasiswa dalam antrian:");
    while (temp != null) {
        temp.data.tampilkanInformasi();
        temp = temp.next;
    }
}
```

3. Isi file Mahasiswa24.java

```
package Tugas;
public class Mahasiswa24 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa24(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilkanInformasi() {
        System.out.println(nim + " - " + nama + " - " + kelas + " - " + ipk);
    }
}
```

4. Isi Main.java

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Antrian24 antrian = new Antrian24();
        int pilihan;

        do {
            System.out.println(x:"===== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====");
            System.out.println(x:"1. Tambahkan Mahasiswa ke Antrian");
            System.out.println(x:"2. Panggil Mahasiswa");
            System.out.println(x:"3. Cek Antrian Kosong");
            System.out.println(x:"4. Kosongkan Antrian");
            System.out.println(x:"5. Tampilkan Mahasiswa Terdepan");
            System.out.println(x:"6. Tampilkan Mahasiswa Terakhir");
            System.out.println(x:"7. Tampilkan Jumlah Antrian");
            System.out.println(x:"8. Tampilkan Semua Antrian");
            System.out.println(x:"9. Keluar");
            System.out.print(x:"Pilihan Anda: ");
            pilihan = scanner.nextInt();
            scanner.nextLine(); // Buang newline

            switch (pilihan) {
                case 1:
                    System.out.print(x:"Masukkan Nama Mahasiswa: ");
                    String nama = scanner.nextLine();
                    System.out.print(x:"Masukkan NIM Mahasiswa: ");
                    String nim = scanner.nextLine();
                    System.out.print(x:"Masukkan Kelas Mahasiswa: ");
                    String kelas = scanner.nextLine();
                    System.out.print(x:"Masukkan ipk Mahasiswa: ");
                    double ipk = scanner.nextDouble();
                    Mahasiswa24 mhs = new Mahasiswa24(nama, nim, kelas, ipk);
                    antrian.enqueue(mhs);
                    break;
                case 2:
                    antrian.dequeue();
                    break;
                case 3:
                    System.out.println(antrian.isEmpty() ? "Antrian kosong." : "");
                    break;
                case 4:
                    antrian.clear();
                    break;
                case 5:
                    antrian.tampilkanDepan();
                    break;
                case 6:
                    antrian.tampilkanBelakang();
                    break;
                case 7:
                    antrian.jumlahAntrian();
                    break;
                case 8:
                    antrian.tampilkanSemua();
                    break;
                case 9:
                    System.out.println(x:"Terima kasih!");
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid.");
            }
        } while (pilihan != 0);

        scanner.close();
    }
}
```

5. Isi Node24.java

```
package Tugas;
public class Node24 {
    Mahasiswa24 data;
    Node24 next;

    Node24(Mahasiswa24 data, Node24 next) {
        this.data = data;
        this.next = next;
    }
}
```

