**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi

[Praktikum Pembuatan Single Linked List 4](#_Toc102034983)

[Langkah 1 4](#_Toc102034984)

[Langkah 2 4](#_Toc102034985)

[Langkah 3 4](#_Toc102034986)

[Langkah 4 5](#_Toc102034987)

[Langkah 5 5](#_Toc102034988)

[Langkah 6 5](#_Toc102034989)

[Langkah 7 5](#_Toc102034990)

[Langkah 8 6](#_Toc102034991)

[Langkah 9 6](#_Toc102034992)

[Langkah 10 6](#_Toc102034993)

[Langkah 11 7](#_Toc102034994)

[Langkah 12 7](#_Toc102034995)

[Langkah 13 7](#_Toc102034996)

[Verifikasi Hasil Percobaan 8](#_Toc102034997)

[Pertanyaan 8](#_Toc102034998)

[Jawaban 8](#_Toc102034999)

[Modifikasi Elemen pada Single Linked List 10](#_Toc102035000)

[Langkah 1 10](#_Toc102035001)

[Langkah 2 10](#_Toc102035002)

[Langkah 3 10](#_Toc102035003)

[Langkah 4 10](#_Toc102035004)

[Langkah 5 11](#_Toc102035005)

[Langkah 6 11](#_Toc102035006)

[Langkah 7 12](#_Toc102035007)

[Langkah 8 12](#_Toc102035008)

[Langkah 9 13](#_Toc102035009)

[Verifikasi Hasil Percobaan 13](#_Toc102035010)

[Pertanyaan 14](#_Toc102035011)

[Jawaban 14](#_Toc102035012)

[Tugas 15](#_Toc102035013)

[Soal Nomor 1 15](#_Toc102035014)

[Jawaban 15](#_Toc102035015)

[Modifikasi source code SingleLinkedList 15](#_Toc102035016)

[Modifikasi source code SLLMain 15](#_Toc102035017)

[Ouput 16](#_Toc102035018)

[Soal Nomor 2 16](#_Toc102035019)

[Jawaban 16](#_Toc102035020)

[Source code class NodeRidwanSoal2 16](#_Toc102035021)

[Source code class SingleLinkedListRidwanSoal2 17](#_Toc102035022)

[Source code class SLLMainRidwanSoal2 18](#_Toc102035023)

[Output 19](#_Toc102035024)

[Soal Nomor 3 19](#_Toc102035025)

[Jawaban 19](#_Toc102035026)

[Modifikasi Source code Main 20](#_Toc102035027)

[Output 20](#_Toc102035028)

[Soal Nomor 4 20](#_Toc102035029)

[Jawaban 21](#_Toc102035030)

[Source Code Class LinkedListMainMahasiswa 21](#_Toc102035031)

[Source Code Class MahasiswaRidwan 22](#_Toc102035032)

[Source Code Class NodeRidwanSoal4 22](#_Toc102035033)

[Source Code Class SingleLinkedListMahasiswa 23](#_Toc102035034)

[Output Antrian Baru (Dari Belakang) 25](#_Toc102035035)

[Output Antrian Keluar (Dari Depan) 25](#_Toc102035036)

[Output Cek Semua Antrian 26](#_Toc102035037)

[Output Cek Antrian Terdepan 27](#_Toc102035038)

[Output Cek Antrian Paling Belakang 27](#_Toc102035039)

[Output Mencari Posisi Mahasiswa Dalam Antrian 28](#_Toc102035040)

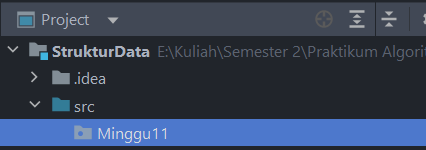
[Output Cek Mahasiswa Dalam Antrian Tertentu 28](#_Toc102035041)

# Praktikum Pembuatan Single Linked List

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat Single Linked List dengan representasi data berupa Node, pengaksesan linked list dan metode penambahan data.

## Langkah 1

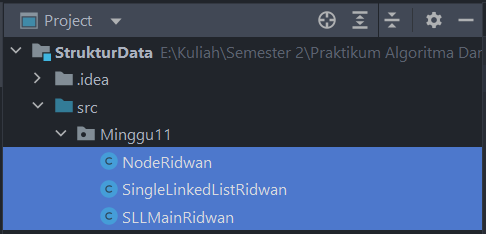
Pada Project StrukturData yang sudah dibuat pada Minggu sebelumnya, buat package dengan nama minggu11



## Langkah 2

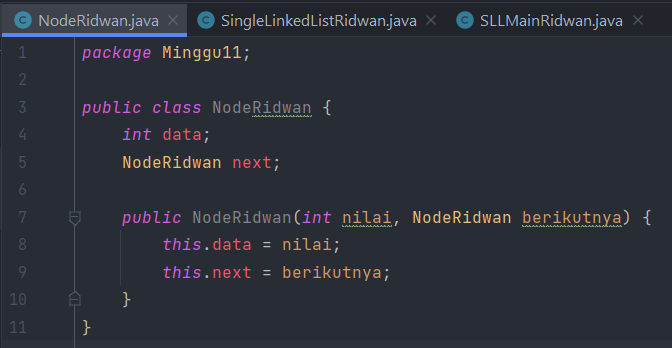
Tambahkan class-class berikut:

1. Node.java
2. SingleLinkedList.java
3. SLLMain.java



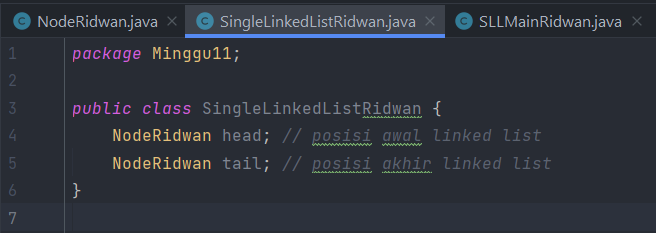
## Langkah 3

Implementasi class Node



## Langkah 4

Tambahkan atribut pada class SingleLinkedList

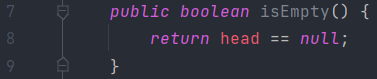


## Langkah 5

Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada SingleLinkedList.

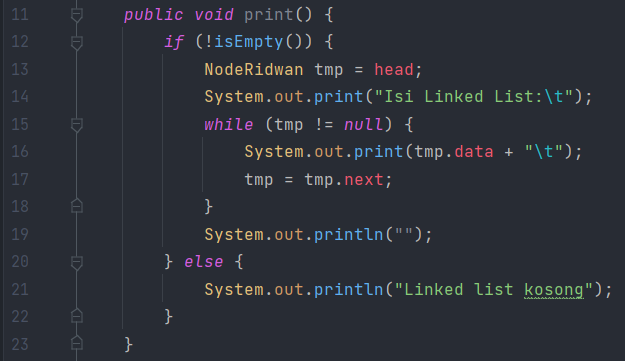
## Langkah 6

Tambahkan method isEmpty().



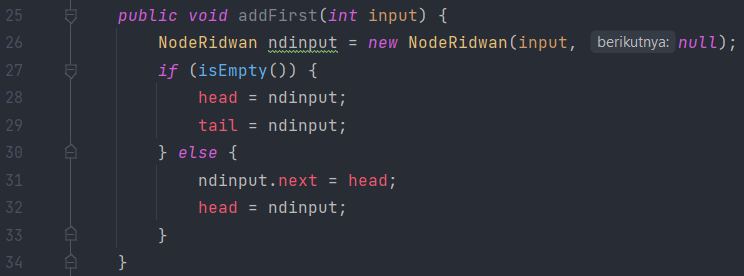
## Langkah 7

Implementasi method untuk mencetak dengan menggunakan proses traverse.



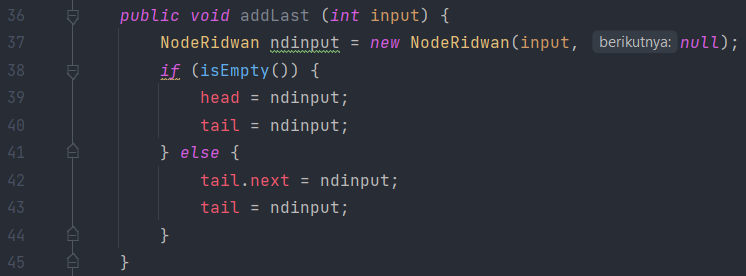
## Langkah 8

Implementasikan method addFirst().



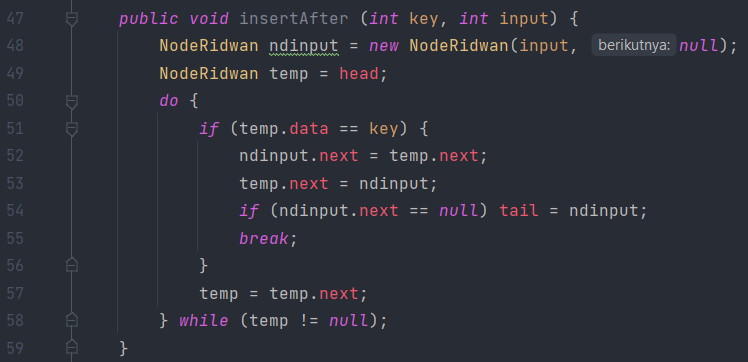
## Langkah 9

Implementasikan method addLast().



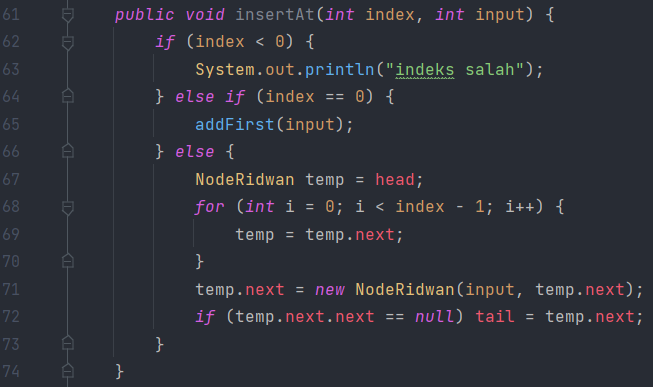
## Langkah 10

Implementasikan method insertAfter, untuk memasukkan node yang memiliki data input setelah node yang memiliki data key



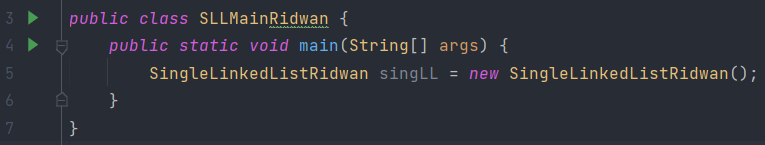
## Langkah 11

Tambahkan method penambahan node pada indeks tertentu.



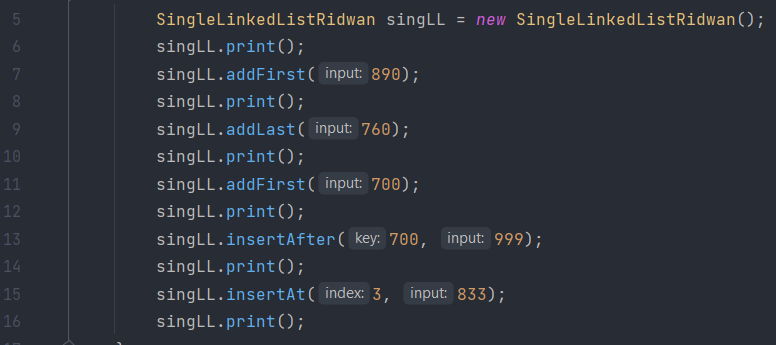
## Langkah 12

Pada class SLLMain, buatlah fungsi main, kemudian buat object dari class SingleLinkedList



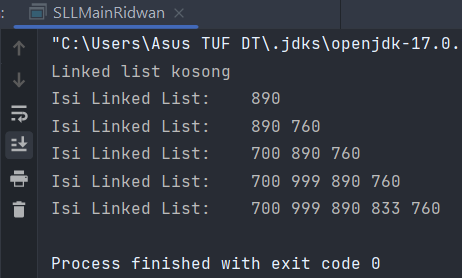
## Langkah 13

Tambahkan Method penambahan data dan pencetakan data di setiap penambahannya agar terlihat perubahannya.



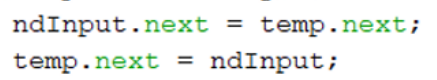
## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini



### Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
2. Pada step 10, jelaskan kegunaan kode berikut



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



### Jawaban

1. Karena linked list masih belum di inputkan nilainya maka pada program method print akan ditampilkan bahwa linked list kosong.
2. Kode tersebut berguna untuk jika temp.data sudah sesuai dengan key yang dicari maka pointer nodeinput akan menunjuk ke yang ditunjuk oleh variable temp, dan pointer temp akan menunjuk ke node input.
3. Kode tersebut berguna untuk jika node yang ditunjuk oleh temp menunjuk pada null maka node yang ditunjuk oleh temp akan menjadi tail.

# Modifikasi Elemen pada Single Linked List

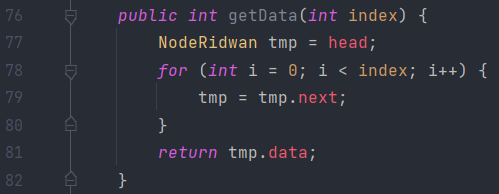
Didalam praktikum ini, kita akan mempraktekkan bagaimana mengakses elemen, mendapatkan indeks dan melakukan penghapusan data pada Single Linked List.:

## Langkah 1

Implementasikan method untuk mengakses data dan indeks pada linked list

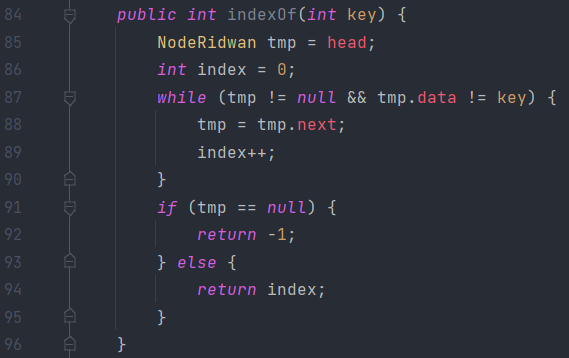
## Langkah 2

Tambahkan method untuk mendapatkan data pada indeks tertentu pada class Single Linked List



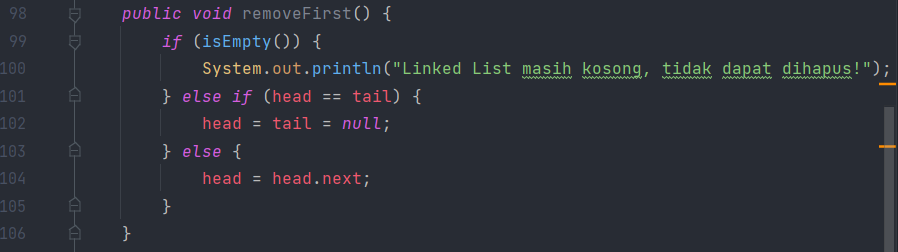
## Langkah 3

Implementasikan method indexOf.



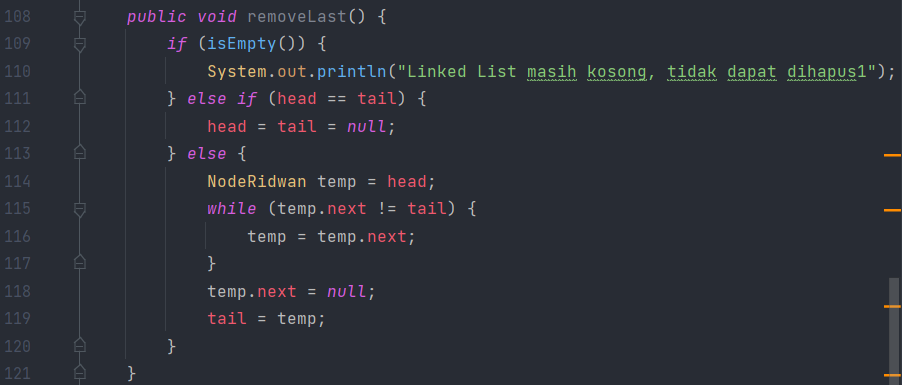
## Langkah 4

Tambahkan method removeFirst pada class SingleLinkedList.



## Langkah 5

Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinkedList



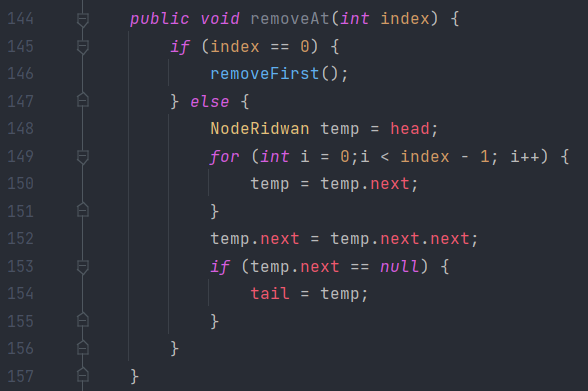
## Langkah 6

Sebagai langkah berikutnya, akan diimplementasikan method remove



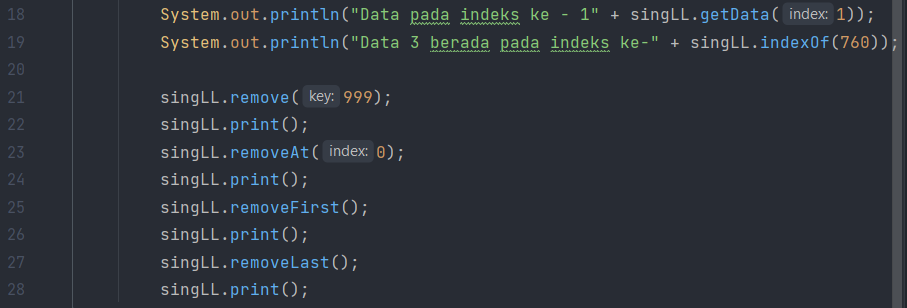
## Langkah 7

Implementasi method untuk menghapus node dengan menggunakan index.



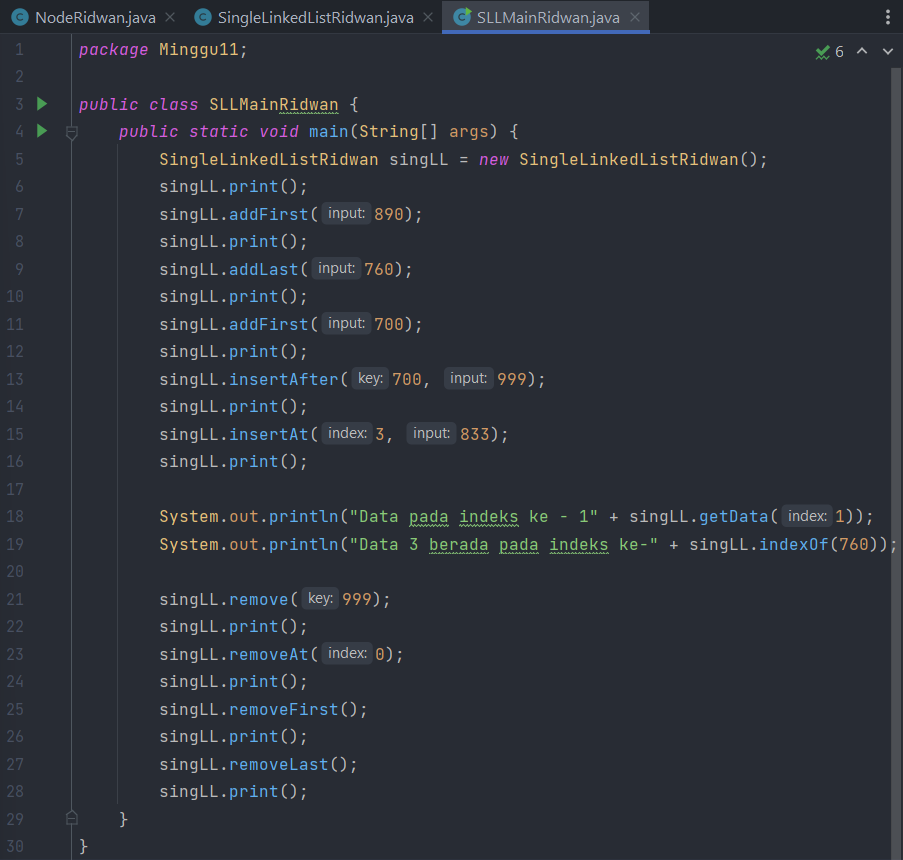
## Langkah 8

Kemudian, coba lakukan pengaksesan dan penghapusan data di method main pada class SLLMain dengan menambahkan kode berikut



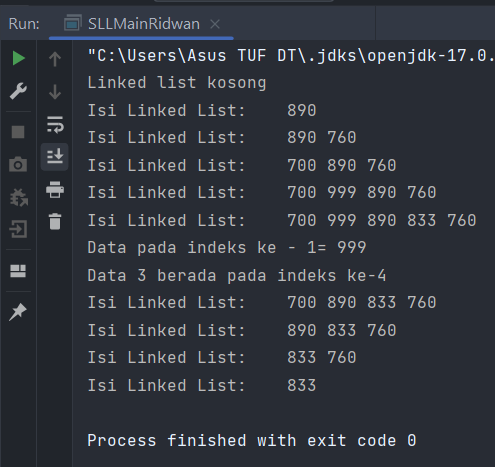
## Langkah 9

Method SLLMain menjadi:



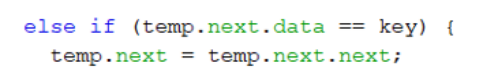
## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



### Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
2. Jelaskan kegunaan kode dibawah pada method remove



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

### Jawaban

1. Jika salah satu kondisi terpenuhi baik if maupun else if dan menemuka break maka akan keluar dari perulangan while, agar tidak banyak membuang waktu.
2. Kode tersebut berfungsi apabila data pada node yang ditunjuk temp bernilai sama dengan key maka nilai node yang ditunjuk temp akan diubah menjadi nilai yang ditunjuk oleh node yang ditunjuk oleh temp.
3. Kembalian pada method indexOf yaitu -1 yang berarti data tidak ditemukan dan akan mengembalikan indeks jika data ditemukan.

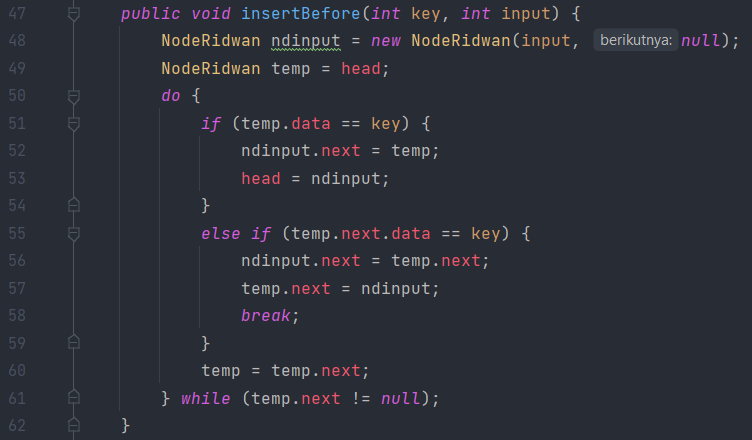
# Tugas

## Soal Nomor 1

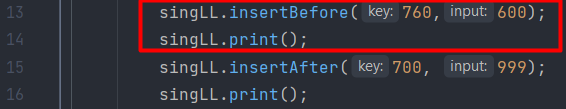
Buat method insertBefore untuk menambahkan node sebelum keyword yang diinginkan

## Jawaban

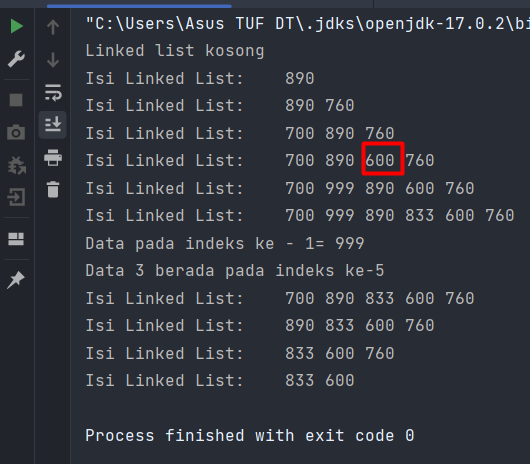
### Modifikasi source code SingleLinkedList



### Modifikasi source code SLLMain

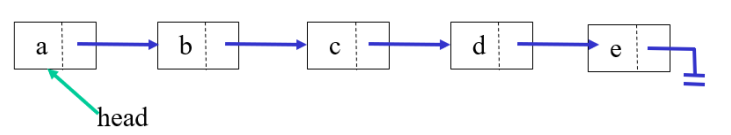


### Ouput



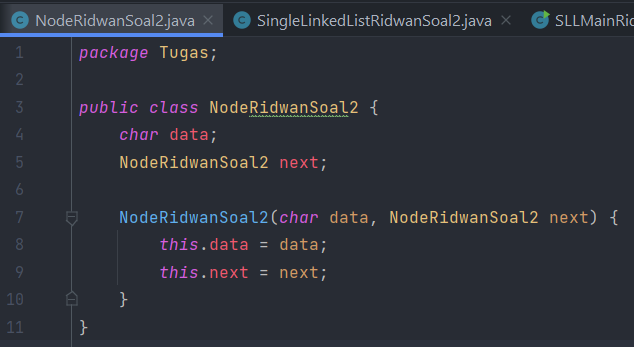
## Soal Nomor 2

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



## Jawaban

### Source code class NodeRidwanSoal2



### Source code class SingleLinkedListRidwanSoal2



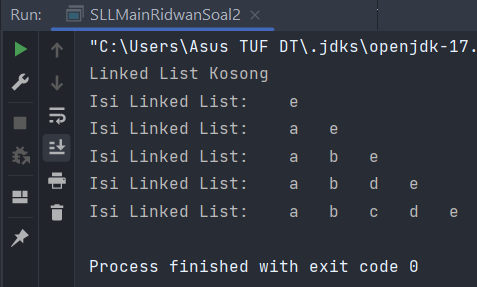




### Source code class SLLMainRidwanSoal2

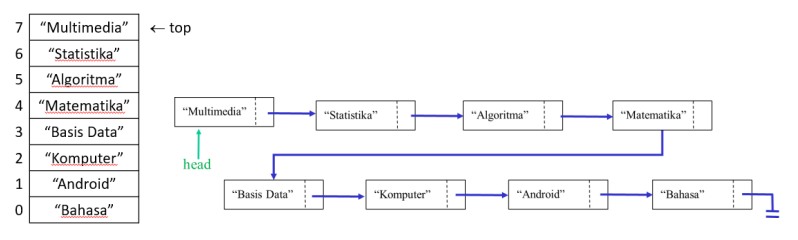


### Output



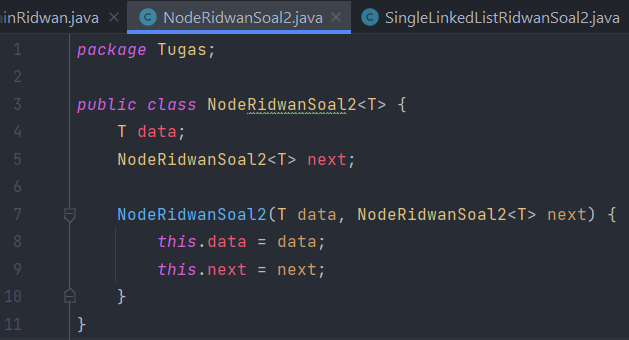
## Soal Nomor 3

Buatlah Implementasi Stack berikut menggunakan Single Linked List



## Jawaban

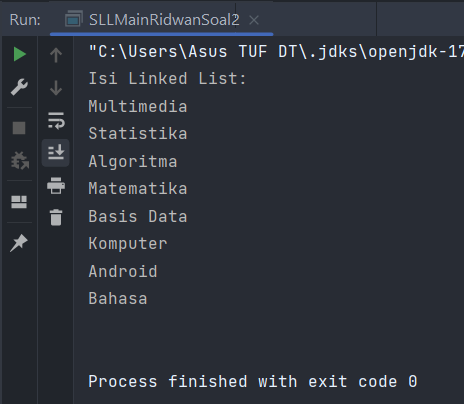
Disini saya menggunakan source code sama seperti pada soal nomor 2 namun saya mengubahnya menggunakan Tipe Data Generic dan menggunakan String



### Modifikasi Source code Main



### Output

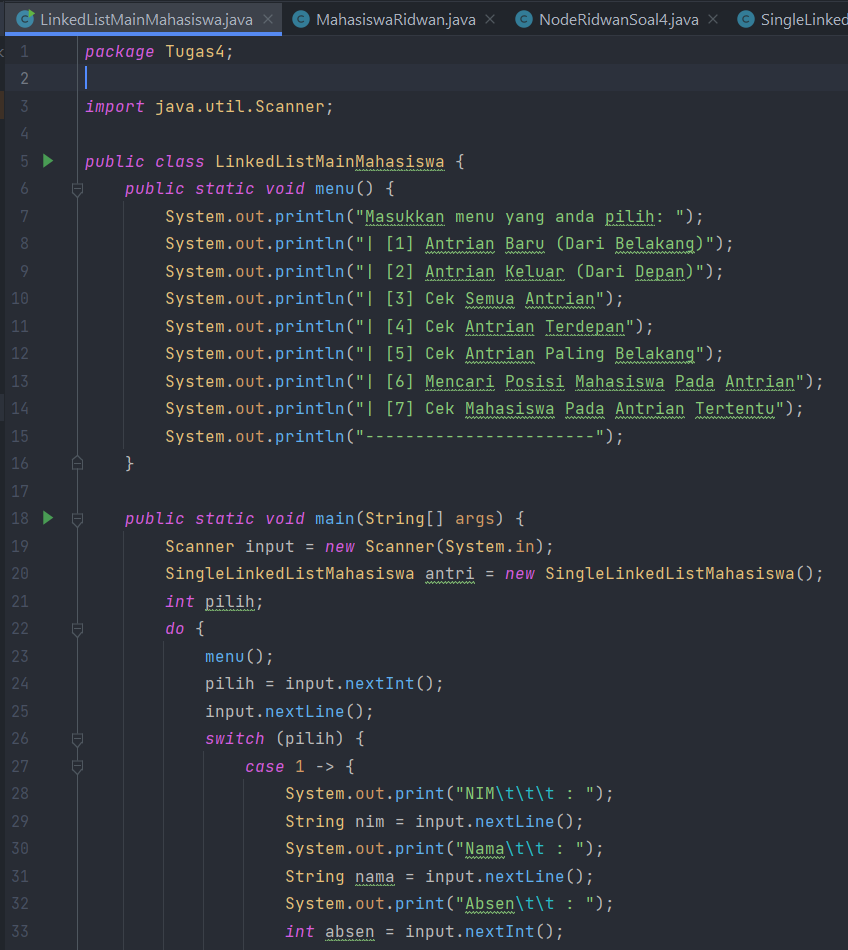


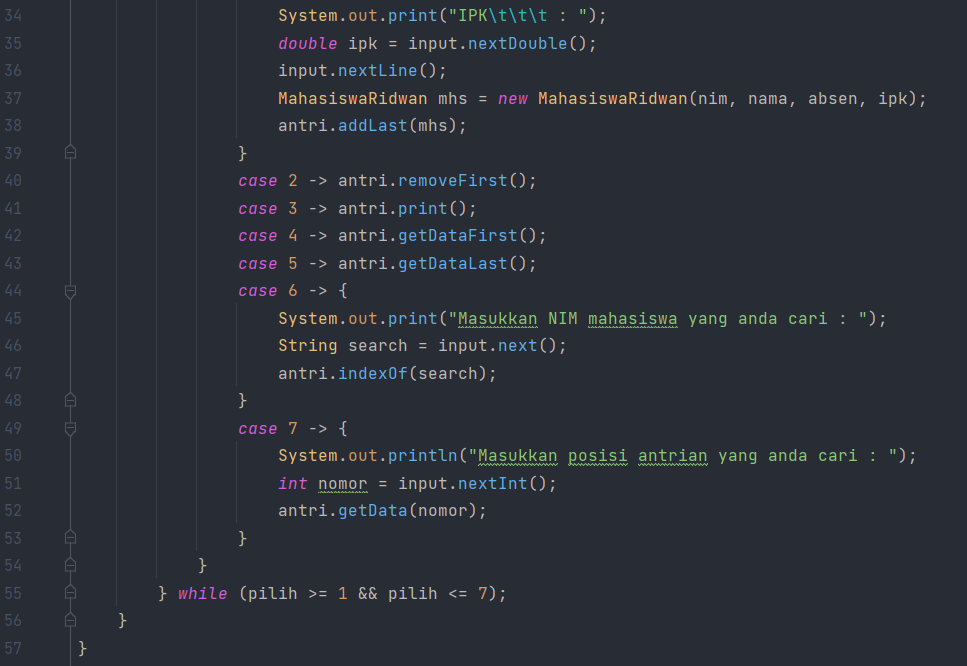
## Soal Nomor 4

Buatlah implementasi program antrian untuk mengilustasikan mahasiswa yang sedang meminta tanda tangan KRS pada dosen DPA di kampus pada tugas jobsheet 8 menggunakan LinkedList. Implementasikan Queue pada antrian mahasiswa dengan menggunakan konsep LinkedList!

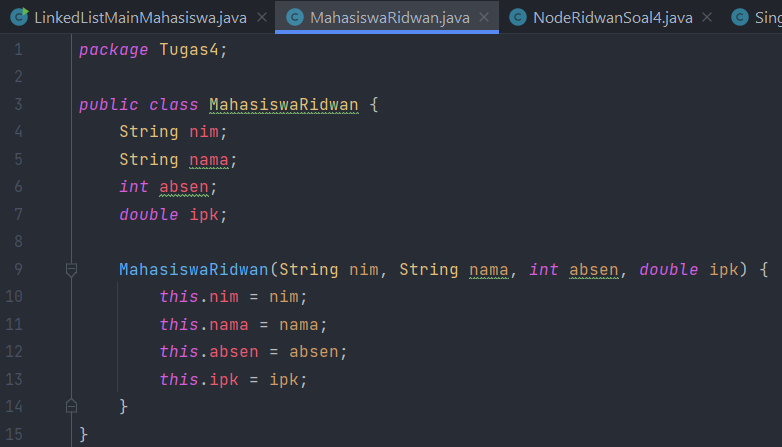
## Jawaban

### Source Code Class LinkedListMainMahasiswa

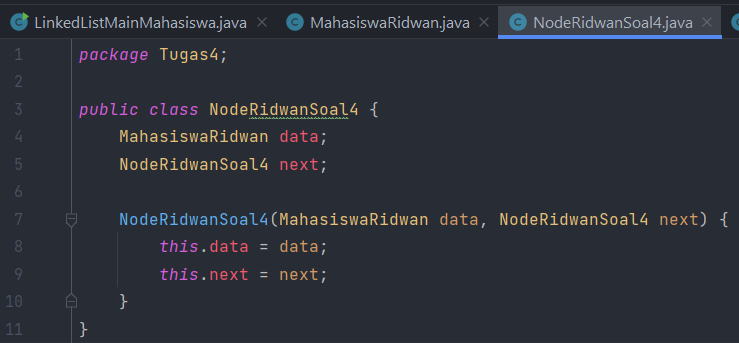




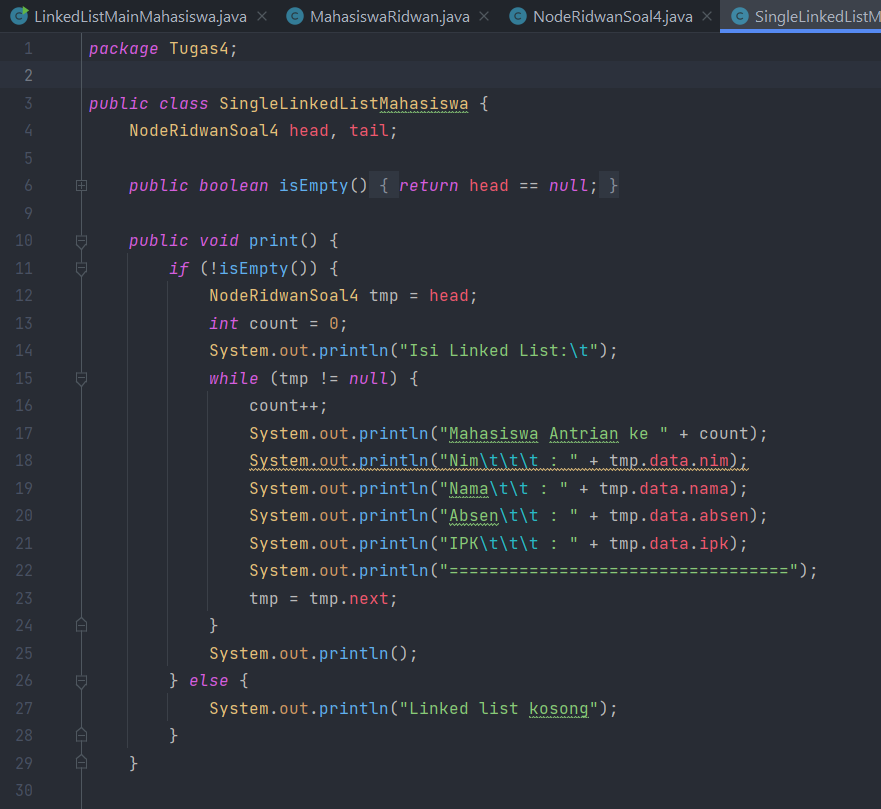
### Source Code Class MahasiswaRidwan



### Source Code Class NodeRidwanSoal4

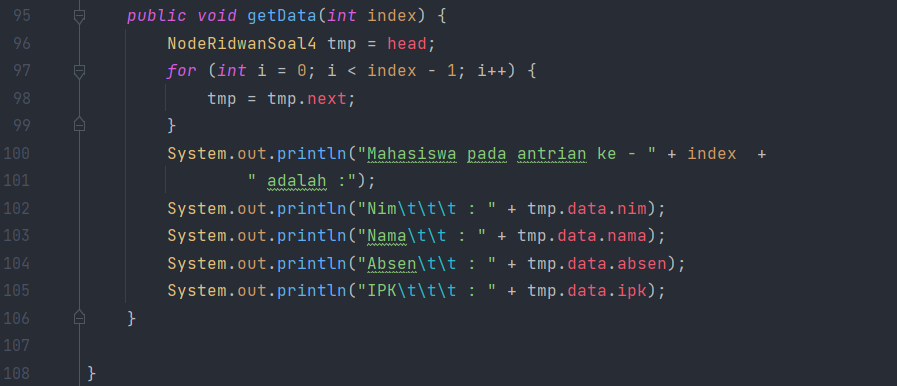


### Source Code Class SingleLinkedListMahasiswa

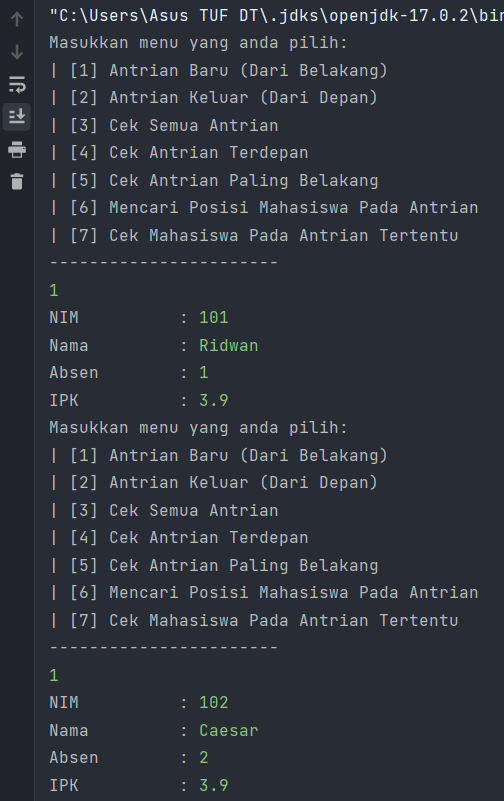




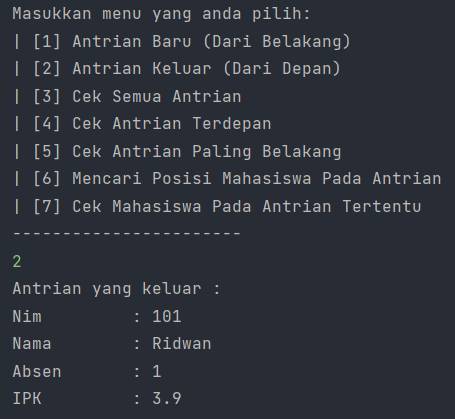




### Output Antrian Baru (Dari Belakang)

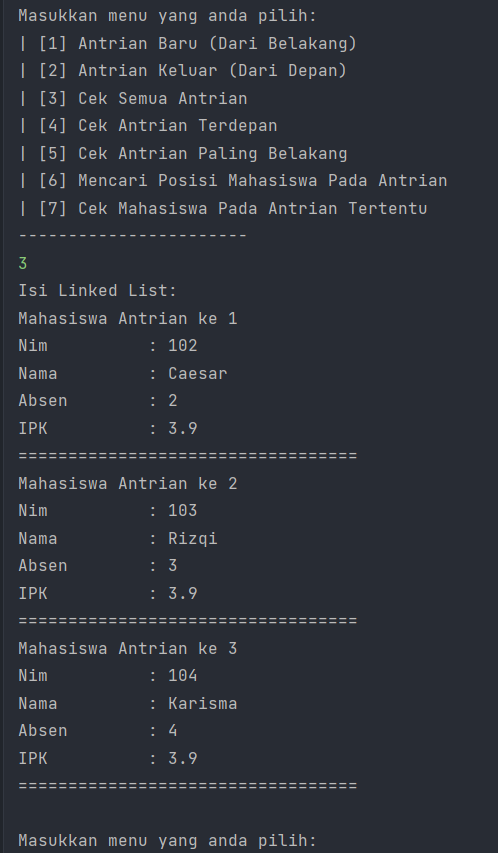


### Output Antrian Keluar (Dari Depan)

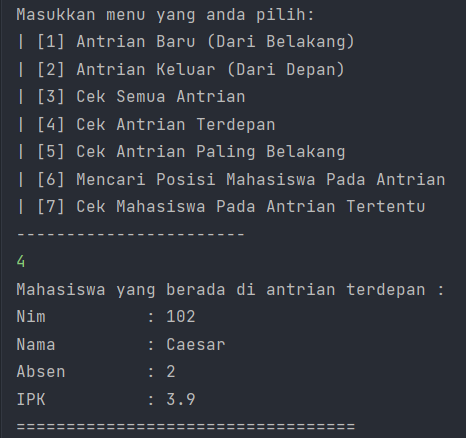


### Output Cek Semua Antrian

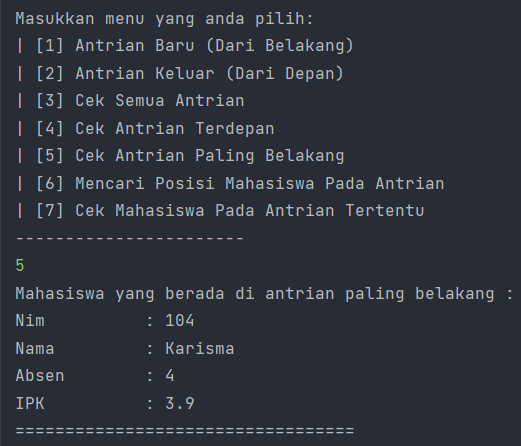
Sebelumnya sudah saya tambahkan antrian baru



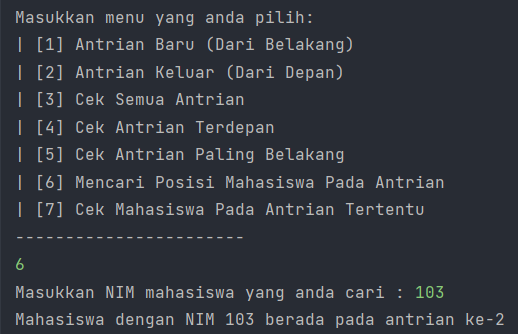
### Output Cek Antrian Terdepan



### Output Cek Antrian Paling Belakang



### Output Mencari Posisi Mahasiswa Dalam Antrian



### Output Cek Mahasiswa Dalam Antrian Tertentu

