

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022

Daftar Isi

PRAKTIKUM PEMBUATAN SINGLE LINKED LIST	4
LANGKAH 1.....	4
LANGKAH 2.....	4
LANGKAH 3.....	4
LANGKAH 4.....	5
LANGKAH 5.....	5
LANGKAH 6.....	5
LANGKAH 7.....	5
LANGKAH 8.....	6
LANGKAH 9.....	6
LANGKAH 10.....	6
LANGKAH 11.....	7
LANGKAH 12.....	7
LANGKAH 13.....	7
VERIFIKASI HASIL PERCOBAAN	8
<i>Pertanyaan.....</i>	<i>8</i>
<i>Jawaban.....</i>	<i>8</i>
MODIFIKASI ELEMEN PADA SINGLE LINKED LIST	10
LANGKAH 1.....	10
LANGKAH 2.....	10
LANGKAH 3.....	10
LANGKAH 4.....	10
LANGKAH 5.....	11
LANGKAH 6.....	11
LANGKAH 7.....	12
LANGKAH 8.....	12
LANGKAH 9.....	13
VERIFIKASI HASIL PERCOBAAN	13
<i>Pertanyaan.....</i>	<i>14</i>
<i>Jawaban.....</i>	<i>14</i>
TUGAS	15
SOAL NOMOR 1.....	15
JAWABAN	15



PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

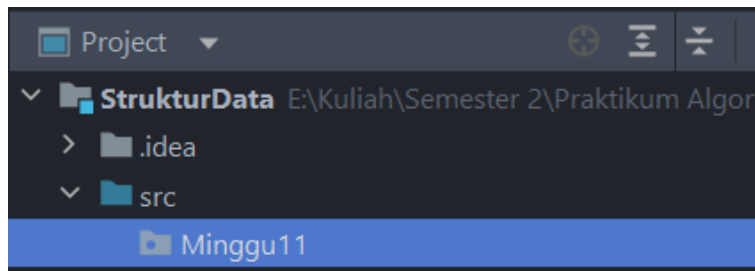
<i>Modifikasi source code SingleLinkedList</i>	15
<i>Modifikasi source code SLLMain</i>	15
<i>Ouput</i>	16
SOAL NOMOR 2	16
JAWABAN	16
<i>Source code class NodeRidwanSoal2</i>	16
<i>Source code class SingleLinkedListRidwanSoal2</i>	17
<i>Source code class SLLMainRidwanSoal2</i>	18
<i>Output</i>	19
SOAL NOMOR 3	19
JAWABAN	19
<i>Modifikasi Source code Main</i>	20
<i>Output</i>	20
SOAL NOMOR 4	20
JAWABAN	21
<i>Source Code Class LinkedListMainMahasiswa</i>	21
<i>Source Code Class MahasiswaRidwan</i>	22
<i>Source Code Class NodeRidwanSoal4</i>	22
<i>Source Code Class SingleLinkedListMahasiswa</i>	23
<i>Output Antrian Baru (Dari Belakang)</i>	25
<i>Output Antrian Keluar (Dari Depan)</i>	25
<i>Output Cek Semua Antrian</i>	26
<i>Output Cek Antrian Terdepan</i>	27
<i>Output Cek Antrian Paling Belakang</i>	27
<i>Output Mencari Posisi Mahasiswa Dalam Antrian</i>	28
<i>Output Cek Mahasiswa Dalam Antrian Tertentu</i>	28

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

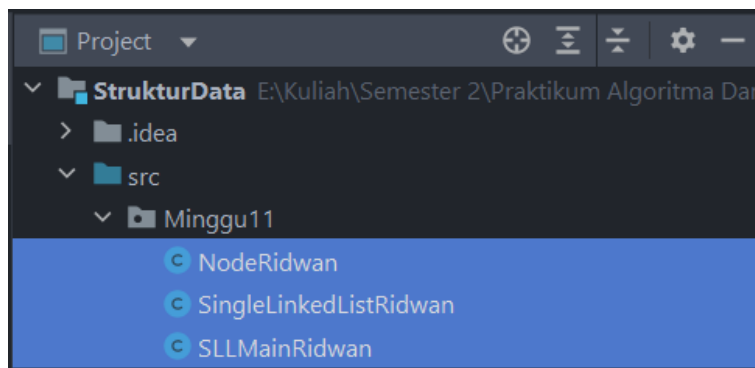
Pada Project StrukturData yang sudah dibuat pada Minggu sebelumnya, buat package dengan nama minggu11



Langkah 2

Tambahkan class-class berikut:

- a) Node.java
- b) SingleLinkedList.java
- c) SLLMain.java



Langkah 3

Implementasi class Node

```
NodeRidwan.java x SingleLinkedListRidwan.java x SLLMainRidwan.java x
1 package Minggu11;
2
3 public class NodeRidwan {
4     int data;
5     NodeRidwan next;
6
7     public NodeRidwan(int nilai, NodeRidwan berikutnya) {
8         this.data = nilai;
9         this.next = berikutnya;
10    }
11 }
```

Langkah 4

Tambahkan atribut pada class SingleLinkedList

```
NodeRidwan.java x SingleLinkedListRidwan.java x SLLMainRidwan.java x
1 package Minggu11;
2
3 public class SingleLinkedListRidwan {
4     NodeRidwan head; // posisi awal linked list
5     NodeRidwan tail; // posisi akhir linked list
6 }
7
```

Langkah 5

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

Langkah 6

Tambahkan method isEmpty().

```
7     public boolean isEmpty() {
8         return head == null;
9     }
```

Langkah 7

Implementasi method untuk mencetak dengan menggunakan proses traverse.

```
11 public void print() {  
12     if (!isEmpty()) {  
13         NodeRidwan tmp = head;  
14         System.out.print("Isi Linked List:\t");  
15         while (tmp != null) {  
16             System.out.print(tmp.data + "\t");  
17             tmp = tmp.next;  
18         }  
19         System.out.println("");  
20     } else {  
21         System.out.println("Linked list kosong");  
22     }  
23 }
```

Langkah 8

Implementasikan method addFirst().

```
25 public void addFirst(int input) {  
26     NodeRidwan ndinput = new NodeRidwan(input, berikutnya: null);  
27     if (isEmpty()) {  
28         head = ndinput;  
29         tail = ndinput;  
30     } else {  
31         ndinput.next = head;  
32         head = ndinput;  
33     }  
34 }
```

Langkah 9

Implementasikan method addLast().

```
36 public void addLast (int input) {  
37     NodeRidwan ndinput = new NodeRidwan(input, berikutnya: null);  
38     if (isEmpty()) {  
39         head = ndinput;  
40         tail = ndinput;  
41     } else {  
42         tail.next = ndinput;  
43         tail = ndinput;  
44     }  
45 }
```

Langkah 10

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

```

47 public void insertAfter (int key, int input) {
48     NodeRidwan ndinput = new NodeRidwan(input, berikutnya: null);
49     NodeRidwan temp = head;
50     do {
51         if (temp.data == key) {
52             ndinput.next = temp.next;
53             temp.next = ndinput;
54             if (ndinput.next == null) tail = ndinput;
55             break;
56         }
57         temp = temp.next;
58     } while (temp != null);
59 }

```

Langkah 11

Tambahkan method penambahan node pada indeks tertentu.

```

61 public void insertAt(int index, int input) {
62     if (index < 0) {
63         System.out.println("indeks salah");
64     } else if (index == 0) {
65         addFirst(input);
66     } else {
67         NodeRidwan temp = head;
68         for (int i = 0; i < index - 1; i++) {
69             temp = temp.next;
70         }
71         temp.next = new NodeRidwan(input, temp.next);
72         if (temp.next.next == null) tail = temp.next;
73     }
74 }

```

Langkah 12

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

```

3 public class SLLMainRidwan {
4     public static void main(String[] args) {
5         SingleLinkedListRidwan singLL = new SingleLinkedListRidwan();
6     }
7 }

```

Langkah 13

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

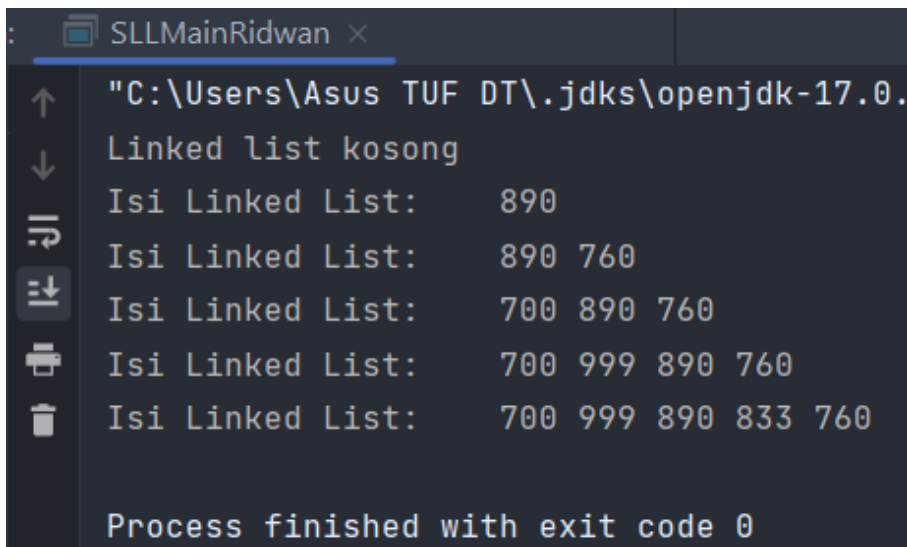
```

5      SingleLinkedListRidwan singLL = new SingleLinkedListRidwan();
6      singLL.print();
7      singLL.addFirst( input: 890);
8      singLL.print();
9      singLL.addLast( input: 760);
10     singLL.print();
11     singLL.addFirst( input: 700);
12     singLL.print();
13     singLL.insertAfter( key: 700, input: 999);
14     singLL.print();
15     singLL.insertAt( index: 3, input: 833);
16     singLL.print();

```

Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini



```

SLLMainRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.
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
Process finished with exit code 0

```

Pertanyaan

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

```

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

```

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

```

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

```

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

```
76      public int getData(int index) {  
77          NodeRidwan tmp = head;  
78          for (int i = 0; i < index; i++) {  
79              tmp = tmp.next;  
80          }  
81          return tmp.data;  
82      }
```

Langkah 3

Implementasikan method indexOf.

```
84      public int indexOf(int key) {  
85          NodeRidwan tmp = head;  
86          int index = 0;  
87          while (tmp != null && tmp.data != key) {  
88              tmp = tmp.next;  
89              index++;  
90          }  
91          if (tmp == null) {  
92              return -1;  
93          } else {  
94              return index;  
95          }  
96      }
```

Langkah 4

Tambahkan method removeFirst pada class SingleLinkedList.

```
198     public void removeFirst() {
199         if (isEmpty()) {
200             System.out.println("Linked List masih kosong, tidak dapat dihapus!");
201         } else if (head == tail) {
202             head = tail = null;
203         } else {
204             head = head.next;
205         }
206     }
```

Langkah 5

Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinkedList

```
1108     public void removeLast() {
1109         if (isEmpty()) {
1110             System.out.println("Linked List masih kosong, tidak dapat dihapus1");
1111         } else if (head == tail) {
1112             head = tail = null;
1113         } else {
1114             NodeRidwan temp = head;
1115             while (temp.next != tail) {
1116                 temp = temp.next;
1117             }
1118             temp.next = null;
1119             tail = temp;
1120         }
1121     }
```

Langkah 6

Sebagai langkah berikutnya, akan diimplementasikan method remove

```

123     public void remove(int key) {
124         if (isEmpty()) {
125             System.out.println("Linked List masih Kosong, tidak dapat dihapus1");
126         } else {
127             NodeRidwan temp = head;
128             while (temp != null) {
129                 if ((temp.data == key) && (temp == head)) {
130                     this.removeFirst();
131                     break;
132                 } else if (temp.next.data == key) {
133                     temp.next = temp.next.next;
134                     if (temp.next == null) {
135                         tail = temp;
136                     }
137                     break;
138                 }
139                 temp = temp.next;
140             }
141         }
142     }

```

Langkah 7

Implementasi method untuk menghapus node dengan menggunakan index.

```

144     public void removeAt(int index) {
145         if (index == 0) {
146             removeFirst();
147         } else {
148             NodeRidwan temp = head;
149             for (int i = 0; i < index - 1; i++) {
150                 temp = temp.next;
151             }
152             temp.next = temp.next.next;
153             if (temp.next == null) {
154                 tail = temp;
155             }
156         }
157     }

```

Langkah 8

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

```
18      System.out.println("Data pada indeks ke - 1" + singLL.getData( index: 1));
19      System.out.println("Data 3 berada pada indeks ke-" + singLL.indexOf(760));
20
21      singLL.remove( key: 999);
22      singLL.print();
23      singLL.removeAt( index: 0);
24      singLL.print();
25      singLL.removeFirst();
26      singLL.print();
27      singLL.removeLast();
28      singLL.print();
```

Langkah 9

Method SLLMain menjadi:

```
NodeRidwan.java x SingleLinkedListRidwan.java x SLLMainRidwan.java x
1  package Minggu11;
2
3  public class SLLMainRidwan {
4  public static void main(String[] args) {
5      SingleLinkedListRidwan singLL = new SingleLinkedListRidwan();
6      singLL.print();
7      singLL.addFirst( input: 890);
8      singLL.print();
9      singLL.addLast( input: 760);
10     singLL.print();
11     singLL.addFirst( input: 700);
12     singLL.print();
13     singLL.insertAfter( key: 700, input: 999);
14     singLL.print();
15     singLL.insertAt( index: 3, input: 833);
16     singLL.print();
17
18     System.out.println("Data pada indeks ke - 1" + singLL.getData( index: 1));
19     System.out.println("Data 3 berada pada indeks ke-" + singLL.indexOf(760));
20
21     singLL.remove( key: 999);
22     singLL.print();
23     singLL.removeAt( index: 0);
24     singLL.print();
25     singLL.removeFirst();
26     singLL.print();
27     singLL.removeLast();
28     singLL.print();
29 }
30 }
```

Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Run: SLLMainRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.
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
Process finished with exit code 0
```

Pertanyaan

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

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

3. 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 menemukan 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

```
47     public void insertBefore(int key, int input) {  
48         NodeRidwan ndinput = new NodeRidwan(input, berikutnya: null);  
49         NodeRidwan temp = head;  
50         do {  
51             if (temp.data == key) {  
52                 ndinput.next = temp;  
53                 head = ndinput;  
54             }  
55             else if (temp.next.data == key) {  
56                 ndinput.next = temp.next;  
57                 temp.next = ndinput;  
58                 break;  
59             }  
60             temp = temp.next;  
61         } while (temp.next != null);  
62     }
```

Modifikasi source code SLLMain

```
13     singLL.insertBefore( key: 760, input: 600);  
14     singLL.print();  
15     singLL.insertAfter( key: 700, input: 999);  
16     singLL.print();
```

Output

```
"C:\Users\Asus TUF DT\.jdkcs\openjdk-17.0.2\b  
Linked list kosong  
Isi Linked List:      890  
Isi Linked List:      890 760  
Isi Linked List:      700 890 760  
Isi Linked List:      700 890 600 760  
Isi Linked List:      700 999 890 600 760  
Isi Linked List:      700 999 890 833 600 760  
Data pada indeks ke - 1= 999  
Data 3 berada pada indeks ke-5  
Isi Linked List:      700 890 833 600 760  
Isi Linked List:      890 833 600 760  
Isi Linked List:      833 600 760  
Isi Linked List:      833 600  
  
Process finished with exit code 0
```


Source code class SingleLinkedListRidwanSoal2

```
in.java x SLLMainRidwan.java x NodeRidwanSoal2.java x SingleLinkedListRidwanSoal2.java x
1 package Tugas;
2
3 public class SingleLinkedListRidwanSoal2 {
4     NodeRidwanSoal2 head, tail;
5
6     public boolean isEmpty() {
7         return head == null;
8     }
9
10    public void print() {
11        if (isEmpty()) {
12            System.out.println("Linked List Kosong");
13        } else {
14            NodeRidwanSoal2 temp = head;
15            System.out.print("Isi Linked List:\t");
16            while (temp != null) {
17                System.out.print(temp.data + "\t");
18                temp = temp.next;
19            }
20            System.out.println();
21        }
22    }
23
24    public void addFirst(char input) {
25        NodeRidwanSoal2 ndinput = new NodeRidwanSoal2(input, next: null);
26        if (isEmpty()) {
27            head = tail = ndinput;
28        } else {
29            ndinput.next = head;
30            head = ndinput;
31        }
32    }
33
34    public void addLast (char input) {
35        NodeRidwanSoal2 ndinput = new NodeRidwanSoal2(input, next: null);
36        if (isEmpty()) {
37            head = tail = ndinput;
38        } else {
39            tail.next = ndinput;
40            tail = ndinput;
41        }
42    }
43
44    public void insertAfter (char key, char input) {
45        NodeRidwanSoal2 ndinput = new NodeRidwanSoal2(input, next: null);
46        NodeRidwanSoal2 temp = head;
47        do {
48            if (temp.data == key) {
49                ndinput.next = temp.next;
50                temp.next = ndinput;
51                if (ndinput.next == null) tail = ndinput;
52                break;
53            }
54            temp = temp.next;
55        } while (temp != null);
56    }
57 }
```

```

58     public void insertAt(int index, char input) {
59         if (index < 0) {
60             System.out.println("indeks salah");
61         } else if (index == 0) {
62             addFirst(input);
63         } else {
64             NodeRidwanSoal2 temp = head;
65             for (int i = 0; i < index - 1; i++) {
66                 temp = temp.next;
67             }
68             temp.next = new NodeRidwanSoal2(input, temp.next);
69             if (temp.next.next == null) tail = temp.next;
70         }
71     }
72
73     public void insertBefore(char key, char input) {
74         NodeRidwanSoal2 ndinput = new NodeRidwanSoal2(input, next null);
75         NodeRidwanSoal2 temp = head;
76         do {
77             if (temp.data == key) {
78                 ndinput.next = temp;
79                 head = ndinput;
80             }
81             else if (temp.next.data == key) {
82                 ndinput.next = temp.next;
83                 temp.next = ndinput;
84                 break;
85             }
86             temp = temp.next;
87         } while (temp.next != null);
88     }
89 }

```

Source code class SLLMainRidwanSoal2

```

SLLMainRidwan.java x NodeRidwanSoal2.java x SingleLinkedListRidwanSoal2.java x SLLMainRidwan
1     package Tugas;
2
3     public class SLLMainRidwanSoal2 {
4         public static void main(String[] args) {
5             SingleLinkedListRidwanSoal2 singLL = new SingleLinkedListRidwanSoal2();
6             singLL.print();
7             singLL.addLast(input: 'e');
8             singLL.print();
9             singLL.addFirst(input: 'a');
10            singLL.print();
11            singLL.insertAfter(key: 'a', input: 'b');
12            singLL.print();
13            singLL.insertAt(index: 2, input: 'd');
14            singLL.print();
15            singLL.insertBefore(key: 'd', input: 'c');
16            singLL.print();
17        }
18    }

```

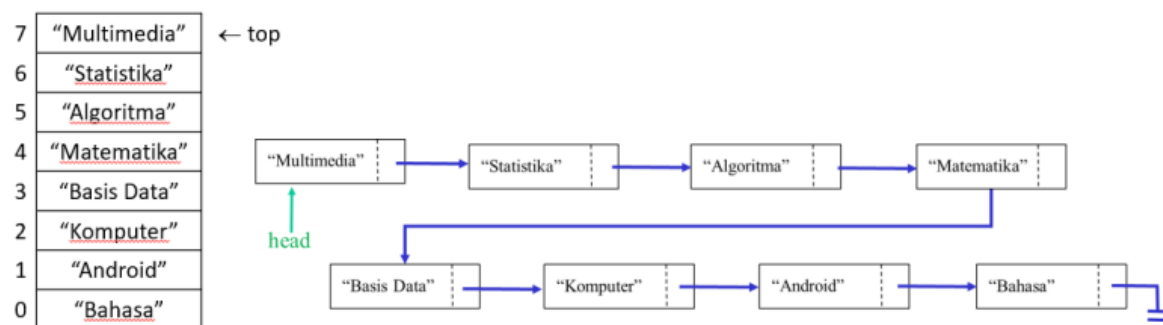
Output

```
Run: SLLMainRidwanSoal2 x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.
LinkedList Kosong
Isi LinkedList:      e
Isi LinkedList:      a      e
Isi LinkedList:      a      b      e
Isi LinkedList:      a      b      d      e
Isi LinkedList:      a      b      c      d      e

Process finished with exit code 0
```

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

```
inRidwan.java x  NodeRidwanSoal2.java x  SingleLinkedListRidwanSoal2.java
1      package Tugas;
2
3      public class NodeRidwanSoal2<T> {
4          T data;
5          NodeRidwanSoal2<T> next;
6
7          NodeRidwanSoal2(T data, NodeRidwanSoal2<T> next) {
8              this.data = data;
9              this.next = next;
10         }
11     }
```

Modifikasi Source code Main

```
inRidwan.java x NodeRidwanSoal2.java x SingleLinkedListRidwanSoal2.java x SLLMainRidwanSoal2.java x
1 package Tugas;
2
3 public class SLLMainRidwanSoal2 {
4     public static void main(String[] args) {
5         SingleLinkedListRidwanSoal2 singLL = new SingleLinkedListRidwanSoal2();
6         singLL.addFirst(input: "Bahasa");
7         singLL.addFirst(input: "Android");
8         singLL.addFirst(input: "Komputer");
9         singLL.addFirst(input: "Basis Data");
10        singLL.addFirst(input: "Matematika");
11        singLL.addFirst(input: "Algoritma");
12        singLL.addFirst(input: "Statistika");
13        singLL.addFirst(input: "Multimedia");
14        singLL.print();
15    }
16 }
```

Output

```
Run: SLLMainRidwanSoal2 x
"C:\Users\Asus TUF DT\.jdk\openjdk-17
Isi Linked List:
Multimedia
Statistika
Algoritma
Matematika
Basis Data
Komputer
Android
Bahasa

Process finished with exit code 0
```

Soal Nomor 4

Buatlah implementasi program antrian untuk mengilustrasikan 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

```
LinkedListMainMahasiswa.java x MahasiswaRidwan.java x NodeRidwanSoal4.java x SingleLinked
1 package Tugas4;
2
3 import java.util.Scanner;
4
5 public class LinkedListMainMahasiswa {
6     public static void menu() {
7         System.out.println("Masukkan menu yang anda pilih: ");
8         System.out.println("| [1] Antrian Baru (Dari Belakang)");
9         System.out.println("| [2] Antrian Keluar (Dari Depan)");
10        System.out.println("| [3] Cek Semua Antrian");
11        System.out.println("| [4] Cek Antrian Terdepan");
12        System.out.println("| [5] Cek Antrian Paling Belakang");
13        System.out.println("| [6] Mencari Posisi Mahasiswa Pada Antrian");
14        System.out.println("| [7] Cek Mahasiswa Pada Antrian Tertentu");
15        System.out.println("-----");
16    }
17
18    public static void main(String[] args) {
19        Scanner input = new Scanner(System.in);
20        SingleLinkedListMahasiswa antri = new SingleLinkedListMahasiswa();
21        int pilih;
22        do {
23            menu();
24            pilih = input.nextInt();
25            input.nextLine();
26            switch (pilih) {
27                case 1 -> {
28                    System.out.print("NIM\t\t\t : ");
29                    String nim = input.nextLine();
30                    System.out.print("Nama\t\t : ");
31                    String nama = input.nextLine();
32                    System.out.print("Absen\t\t : ");
33                    int absen = input.nextInt();
34
35                    System.out.print("IPK\t\t\t : ");
36                    double ipk = input.nextDouble();
37                    input.nextLine();
38                    MahasiswaRidwan mhs = new MahasiswaRidwan(nim, nama, absen, ipk);
39                    antri.addLast(mhs);
40                }
41                case 2 -> antri.removeFirst();
42                case 3 -> antri.print();
43                case 4 -> antri.getDataFirst();
44                case 5 -> antri.getDataLast();
45                case 6 -> {
46                    System.out.print("Masukkan NIM mahasiswa yang anda cari : ");
47                    String search = input.next();
48                    antri.indexOf(search);
49                }
50                case 7 -> {
51                    System.out.println("Masukkan posisi antrian yang anda cari : ");
52                    int nomor = input.nextInt();
53                    antri.getData(nomor);
54                }
55            }
56        } while (pilih >= 1 && pilih <= 7);
57    }
58 }
```

Source Code Class MahasiswaRidwan

```
LinkedListMainMahasiswa.java x MahasiswaRidwan.java x NodeRidwanSoal4.java x Sing
1 package Tugas4;
2
3 public class MahasiswaRidwan {
4     String nim;
5     String nama;
6     int absen;
7     double ipk;
8
9     MahasiswaRidwan(String nim, String nama, int absen, double ipk) {
10         this.nim = nim;
11         this.nama = nama;
12         this.absen = absen;
13         this.ipk = ipk;
14     }
15 }
```

Source Code Class NodeRidwanSoal4

```
LinkedListMainMahasiswa.java x MahasiswaRidwan.java x NodeRidwanSoal4.java x
1 package Tugas4;
2
3 public class NodeRidwanSoal4 {
4     MahasiswaRidwan data;
5     NodeRidwanSoal4 next;
6
7     NodeRidwanSoal4(MahasiswaRidwan data, NodeRidwanSoal4 next) {
8         this.data = data;
9         this.next = next;
10    }
11 }
```

Source Code Class SingleLinkedListMahasiswa

```
LinkedListMainMahasiswa.java x MahasiswaRidwan.java x NodeRidwanSoal4.java x SingleLinkedListM

1 package Tugas4;
2
3 public class SingleLinkedListMahasiswa {
4     NodeRidwanSoal4 head, tail;
5
6     public boolean isEmpty() { return head == null; }
7
8
9
10    public void print() {
11        if (!isEmpty()) {
12            NodeRidwanSoal4 tmp = head;
13            int count = 0;
14            System.out.println("Isi Linked List:\t");
15            while (tmp != null) {
16                count++;
17                System.out.println("Mahasiswa Antrian ke " + count);
18                System.out.println("Nim\t\t\t: " + tmp.data.nim);
19                System.out.println("Nama\t\t: " + tmp.data.nama);
20                System.out.println("Absen\t\t: " + tmp.data.absen);
21                System.out.println("IPK\t\t\t: " + tmp.data.ipk);
22                System.out.println("=====");
23                tmp = tmp.next;
24            }
25            System.out.println();
26        } else {
27            System.out.println("Linked list kosong");
28        }
29    }
30
31    public void addLast (MahasiswaRidwan input) {
32        NodeRidwanSoal4 ndinput = new NodeRidwanSoal4(input, next: null);
33        if (isEmpty()) {
34            head = ndinput;
35            tail = ndinput;
36        } else {
37            tail.next = ndinput;
38            tail = ndinput;
39        }
40    }
41
42    public void removeFirst() {
43        if (isEmpty()) {
44            System.out.println("Linked List masih kosong, tidak dapat dihapus!");
45        } else if (head == tail) {
46            System.out.println("Antrian yang keluar :");
47            System.out.println("Nim\t\t\t: " + head.data.nim);
48            System.out.println("Nama\t\t: " + head.data.nama);
49            System.out.println("Absen\t\t: " + head.data.absen);
50            System.out.println("IPK\t\t\t: " + head.data.ipk);
51            head = tail = null;
52        } else {
53            System.out.println("Antrian yang keluar :");
54            System.out.println("Nim\t\t\t: " + head.data.nim);
55            System.out.println("Nama\t\t: " + head.data.nama);
56            System.out.println("Absen\t\t: " + head.data.absen);
57            System.out.println("IPK\t\t\t: " + head.data.ipk);
58            head = head.next;
59        }
60    }
61 }
```

```
62 public void getDataFirst() {
63     System.out.println("Mahasiswa yang berada di antrian terdepan :");
64     System.out.println("Nim\t\t\t : " + head.data.nim);
65     System.out.println("Nama\t\t\t : " + head.data.nama);
66     System.out.println("Absen\t\t\t : " + head.data.absen);
67     System.out.println("IPK\t\t\t\t : " + head.data.ipk);
68     System.out.println("=====");
69 }
70
71 public void getDataLast() {
72     System.out.println("Mahasiswa yang berada di antrian paling belakang :");
73     System.out.println("Nim\t\t\t\t : " + tail.data.nim);
74     System.out.println("Nama\t\t\t\t : " + tail.data.nama);
75     System.out.println("Absen\t\t\t\t : " + tail.data.absen);
76     System.out.println("IPK\t\t\t\t\t : " + tail.data.ipk);
77     System.out.println("=====");
78 }
79
80 public void indexOf(String key) {
81     NodeRidwanSoal4 tmp = head;
82     int index = 0;
83     while (tmp != null && !tmp.data.nim.equals(key)) {
84         tmp = tmp.next;
85         index++;
86     }
87     if (tmp == null) {
88         System.out.println("Data tidak ditemukan");
89     } else {
90         System.out.println("Mahasiswa dengan NIM " + key + " berada " +
91             "pada antrian ke-" + (index + 1));
92     }
93 }
94
95 public void getData(int index) {
96     NodeRidwanSoal4 tmp = head;
97     for (int i = 0; i < index - 1; i++) {
98         tmp = tmp.next;
99     }
100     System.out.println("Mahasiswa pada antrian ke - " + index +
101         " adalah :");
102     System.out.println("Nim\t\t\t\t : " + tmp.data.nim);
103     System.out.println("Nama\t\t\t\t : " + tmp.data.nama);
104     System.out.println("Absen\t\t\t\t : " + tmp.data.absen);
105     System.out.println("IPK\t\t\t\t\t : " + tmp.data.ipk);
106 }
107
108 }
```


Output Antrian Baru (Dari Belakang)

```
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
1
NIM      : 101
Nama     : Ridwan
Absen    : 1
IPK      : 3.9
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
1
NIM      : 102
Nama     : Caesar
Absen    : 2
IPK      : 3.9
```

Output Antrian Keluar (Dari Depan)

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
2
Antrian yang keluar :
Nim      : 101
Nama     : Ridwan
Absen    : 1
IPK      : 3.9
```

Output Cek Semua Antrian

Sebelumnya sudah saya tambahkan antrian baru

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
3
Isi Linked List:
Mahasiswa Antrian ke 1
Nim      : 102
Nama     : Caesar
Absen    : 2
IPK      : 3.9
=====
Mahasiswa Antrian ke 2
Nim      : 103
Nama     : Rizqi
Absen    : 3
IPK      : 3.9
=====
Mahasiswa Antrian ke 3
Nim      : 104
Nama     : Karisma
Absen    : 4
IPK      : 3.9
=====
Masukkan menu yang anda pilih:
```

Output Cek Antrian Terdepan

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
4
Mahasiswa yang berada di antrian terdepan :
Nim      : 102
Nama     : Caesar
Absen    : 2
IPK      : 3.9
=====
```

Output Cek Antrian Paling Belakang

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
5
Mahasiswa yang berada di antrian paling belakang :
Nim      : 104
Nama     : Karisma
Absen    : 4
IPK      : 3.9
=====
```

Output Mencari Posisi Mahasiswa Dalam Antrian

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
6
Masukkan NIM mahasiswa yang anda cari : 103
Mahasiswa dengan NIM 103 berada pada antrian ke-2
```

Output Cek Mahasiswa Dalam Antrian Tertentu

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru (Dari Belakang)
| [2] Antrian Keluar (Dari Depan)
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
-----
7
Masukkan posisi antrian yang anda cari :
2
Mahasiswa pada antrian ke - 2 adalah :
Nim           : 103
Nama          : Rizqi
Absen         : 3
IPK           : 3.9
```