



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : ADT SINGLE LINKED LIST
NAMA : CHARISMA PRASETYA PUTERA AMMAL
NIM : 185150700111019
TANGGAL : 20/09/2019
ASISTEN : AFIF MUSYAYYIDIN

A. DEFINISI MASALAH

1. Modifikasilah program di atas sehingga SLL dapat menampung sembarang object. Untuk itu anda perlu membuat class baru bernama Buku dengan data dan method sebagai berikut :

Buku
String kategori String judul Double hargaSewa double hargaDenda
Constructor Buku double getHargaSewa double getHargaDenda String getKategori String getJudul

2. Untuk penyisipan data secaraurut gunakan pengurutan berdasarkan HargaSewa apabila HargaSewa sama urutkan berdasarkan HargaDenda. Gunakan potongan kode berikut pada main

```
Buku buku1 = new Buku("Komputer", "Pemrograman Java", 2000, 3000);  
Buku buku2 = new Buku("Pendidikan", "Matematika SMA", 2500, 2500);  
Buku buku3 = new Buku("Komputer", "Belajar Microsoft Office", 2000,  
7000);  
Buku buku4 = new Buku("Novel", "Laskar Pelangi", 1000, 4000);  
Buku buku5 = new Buku("Komik", "Doraemon", 2000, 5000);  
SingleLinkedList ll = new SingleLinkedList;  
ll.sisipDataUrut(buku1);  
ll.sisipDataUrut(buku2);  
ll.sisipDataUrut(buku3);  
ll.sisipDataUrut(buku4);  
ll.sisipDataUrut(buku5);  
ll.cetak("Linked List Buku");
```

B. SOURCE CODE

SingleLinkedList.java	
1	package praktikum;
2	
3	class Node {
4	
5	Buku data;
6	Node next;
7	}
8	
9	public class SingleLinkedList {
10	
11	private Node pTail;
12	private Node pHeader;
13	int size;
14	
15	boolean isEmpty() {
16	return (size == 0);
17	}
18	
19	public void sisipDataUrut(Buku data) {
20	if (isEmpty()) {
21	Node nodeBaru = new Node();
22	nodeBaru.data = data;
23	nodeBaru.next = null;
24	pHeader = nodeBaru;
25	pTail = nodeBaru;
26	} else if (size == 1) {
27	Node n = pHeader;
28	if (data.hargaSewa > n.data.hargaSewa) {
29	Node nodeBaru = new Node();
30	nodeBaru.data = data;
31	pTail.next = nodeBaru;
32	pTail = nodeBaru;
33	nodeBaru.next = null;
34	} else {
35	Node nodeBaru = new Node();
36	nodeBaru.data = data;
37	nodeBaru.next = pHeader;
38	pHeader = nodeBaru;
39	}
40	size++;
41	} else {
42	Node n = pHeader;
43	Node angka = new Node();
44	angka.data = data;
45	Node bantuan = pHeader;
46	
47	if (angka.data.hargaSewa <
	pHeader.data.hargaSewa) {

48	angka.data = data;
49	angka.next = pHeader;
50	pHeader = angka;
51	} else {
52	int jumlahSama = 0;
53	while (n != pTail) {
54	if (angka.data.hargaSewa ==
	n.data.hargaSewa) {
55	jumlahSama++;
56	}
57	n = n.next;
58	}
59	n = pHeader;
60	if (jumlahSama >= 1) {
61	while (n != pTail) {
62	if (angka.data.hargaSewa ==
	n.data.hargaSewa) {
63	if (angka.data.hargaDenda >
	n.data.hargaDenda) {
64	bantuan = n;
65	}
66	}
67	n = n.next;
68	}
69	
70	n = bantuan.next;
71	if (bantuan == pTail) {
72	angka.data = data;
73	pTail.next = angka;
74	pTail = angka;
75	angka.next = null;
76	} else {
77	bantuan.next = angka;
78	angka.next = n;
79	}
80	} else {
81	while (n != pTail) {
82	if (angka.data.hargaSewa >
	n.data.hargaSewa) {
83	bantuan = n;
84	}
85	n = n.next;
86	}
87	n = bantuan.next;
88	if (bantuan == pTail) {
89	angka.data = data;
90	pTail.next = angka;
91	pTail = angka;
92	angka.next = null;
93	} else {
94	bantuan.next = angka;

95	angka.next = n;
96	}
97	}
98	
99	}
100	}
101	size++;
102	}
103	
104	public void cetak(String kom) {
105	System.out.println(kom);
106	Node n = pHeader;
107	while (n != null) {
108	System.out.println("");
109	System.out.println("Kategori Buku : " +
	n.data.getKategori());
110	System.out.println("Judul Buku : " +
	n.data.getJudul());
111	System.out.println("Harga Sewa : Rp. " +
	n.data.getHargaSewa());
112	System.out.println("Harga Denda : Rp. " +
	n.data.getHargaDenda());
113	n = n.next;
114	}
115	}
116	}

Buku.java	
1	package praktikum;
2	
3	public class Buku {
4	
5	public String kategori;
6	public String judul;
7	public double hargaSewa;
8	public double hargaDenda;
9	
10	public Buku(String kategori, String judul, double
	hargaSewa, double hargaDenda) {
11	this.kategori = kategori;
12	this.judul = judul;
13	this.hargaSewa = hargaSewa;
14	this.hargaDenda = hargaDenda;
15	}
16	
17	public String getKategori() {
18	return kategori;
19	}
20	
21	public String getJudul() {

22	return judul;
23	}
24	
25	public double getHargaSewa() {
26	return hargaSewa;
27	}
28	
29	public double getHargaDenda() {
30	return hargaDenda;
31	}
32	
33	public static void main(String[] args) {
34	System.out.println("=====
	=====");
35	System.out.println("= DAFTAR BUKU
	PERPUSTAKAAN =");
36	System.out.println("=====
	=====");
37	Buku buku1 = new Buku("Komputer", "Pemrograman
	Mobile", 2000, 3000);
38	Buku buku2 = new Buku("Pendidikan", "Korupsi",
	2500, 2500);
39	Buku buku3 = new Buku("Agama", "Belajar Sholat",
	2000, 7000);
40	Buku buku4 = new Buku("Novel", "13 Reason why",
	1000, 4000);
41	Buku buku5 = new Buku("Komik", "Wibu", 2000,
	5000);
42	SingleLinkedList ll = new SingleLinkedList();
43	ll.sisipDataUrut(buku1);
44	ll.sisipDataUrut(buku2);
45	ll.sisipDataUrut(buku3);
46	ll.sisipDataUrut(buku4);
47	ll.sisipDataUrut(buku5);
48	System.out.println(" ");
49	ll.cetak("Buku Setelah di Sort berdasar Harga
	Sewa yang terendah : ");
50	}
51	
52	}

C. PEMBAHASAN

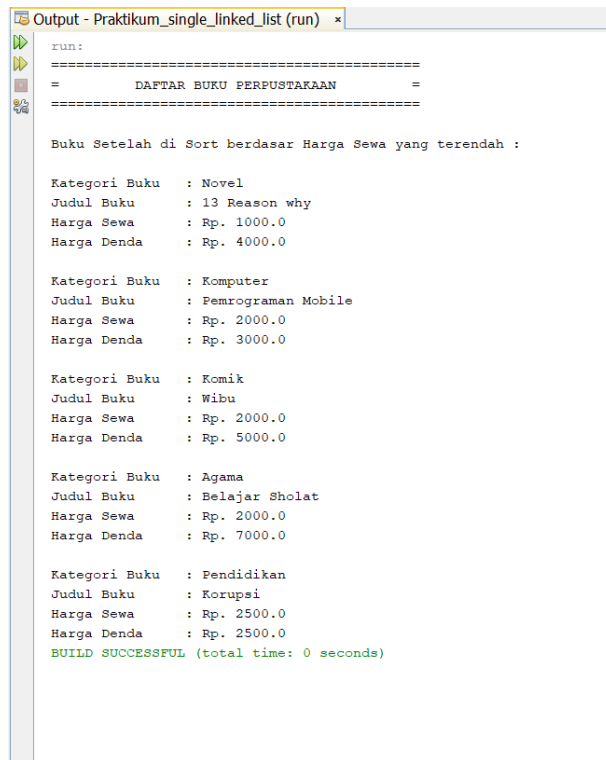
SingleLinkedList.java	
1	Deklarasi package praktikum
3	Deklarasi class bernama Node
5	Menginisialisasi class buku yang diberi nama objek data
6	Menginisialisasi class Node yang diberi nama objek next

9	Deklarasi class bernama SingleLinkedList
11	Deklarasi atribut dari class Node bernama pTail
12	Deklarasi atribut dari class Node bernama pHeader
13	Deklarasi atribut interger bernama size
15	Deklarasi method bernama boolean bernama isEmpty()
16	Deklarasi fungsi return yang berparameter size == 0
19	Deklarasi method void bernama sisipDataUrut yang berparameter (Buku data) method ini berisi beberapa fungsi if else yang berfungsi untuk mengurutkan urutan node berdasarkan hargaSewa yang paling rendah dan jika memiliki hargaSewa yang sama maka ditentukan dari hargaDenda yang paling rendah cara kerja method ini sendiri membuat berbagai fungsi if dan else yang membandingkan setiap pHeader dan mengurutkannya menjadi urutan yang benar
104	Deklarasi method void cetak berparameter (String kom)
105	Mencetak output berupa isian dari variable kom
106	Deklarasi objek kelas Node bernama n yang berisi pHeader
107	Deklarasi fungsi while berparameter (n!= null)
108	Mencetak output berupa (" ")
109	Mencetak output berupa ("Kategori Buku : "+n.data.getKategori())
110	Mencetak output berupa ("Judul Buku : "+n.data.getJudul());
111	Mencetak output berupa ("Harga Sewa : Rp. "+n.data.getHargaSewa())
112	Mencetak output berupa ("Harga Denda : Rp. "+n.data.getHargaDenda())
113	Deklarasi isi dari variable n = n.next

Buku.java	
1	Deklarasi package praktikum
3	Deklarasi class bernama Buku
5	Deklarasi atribut bertipe String bernama kategori
6	Deklarasi atribut bertipe String bernama judul
7	Deklarasi atribut bertipe double bernama hargaSewa
8	Deklarasi atribut bertipe double bernama hargaDenda
10	Deklarasi konstruktor bernama Buku yang berparameter(String kategori, String judul, double hargaSewa, hargaDenda)
11	Deklarasi variable kategori yang merupakan variable konstruktor yang berisi kategori
12	Deklarasi variable judul yang merupakan variable konstruktor yang berisi judul
13	Deklarasi variable hargaSewa yang merupakan variable konstruktor yang berisi hargaSewa
14	Deklarasi variable hargaDenda yang merupakan variable konstruktor yang berisi hargaDenda
17	Deklarasi method bertipe String yang bernama getKategori
18	Deklarasi fungsi return yang berisi kategori
21	Deklarasi method bertipe String yang bernama getJudul
22	Deklarasi fungsi return yang berisi judul

25	Deklarasi method bertipe double yang bernama getHargaSewa
26	Deklarasi fungsi return yang berisi hargaSewa
29	Deklarasi method bertipe double yang bernama getHargaDenda
30	Deklarasi fungsi return yang berisi hargaDenda
33	Deklarasi main method
34	Mencetak output berupa
	("=====")
35	Mencetak output berupa("= DAFTAR BUKU PERPUSTAKAAN =")
36	Mencetak output berupa
	("=====");
37	Membuat konstruktor dari kelas Buku dan disimpan pada variable buku1 yang memiliki parameter ("Komputer", "Pemrograman Mobile", 2000, 3000)
38	Membuat konstruktor dari kelas Buku dan disimpan pada variable buku2 yang memiliki parameter ("Pendidikan", "Korupsi", 2500, 2500)
39	Membuat konstruktor dari kelas Buku dan disimpan pada variable buku3 yang memiliki parameter ("Agama", "Belajar Belajar Sholat", 2000, 7000)
40	Membuat konstruktor dari kelas Buku dan disimpan pada variable buku4 yang memiliki parameter ("Novel", "13 Reason Why", 1000, 4000)
41	Membuat konstruktor dari kelas Buku dan disimpan pada variable buku5 yang memiliki parameter ("Komik", "Wibu", 2000, 5000)
42	Membuat objek ll dari method SingleLinkedList
43	Memanggil method sisipDataUrut berparamter(buku 1) yang dipanggil oleh objek ll
44	Memanggil method sisipDataUrut berparamter(buku 2) yang dipanggil oleh objek ll
45	Memanggil method sisipDataUrut berparamter(buku 3) yang dipanggil oleh objek ll
46	Memanggil method sisipDataUrut berparamter(buku 4) yang dipanggil oleh objek ll
47	Memanggil method sisipDataUrut berparamter(buku 5) yang dipanggil oleh objek ll
48	Mencetak output berupa (" ")
49	Memanggil method cetak berparamter("Buku Setelah di Sort berdasar Harga Sewa yang terendah : ") yang dipanggil oleh objek ll

D. SCREENSHOT PROGRAM



```
Output - Praktikum_single_linked_list (run) x
run:
=====
=          DAFTAR BUKU PERPUSTAKAAN          =
=====

Buku Setelah di Sort berdasar Harga Sewa yang terendah :

Kategori Buku   : Novel
Judul Buku      : 13 Reason why
Harga Sewa      : Rp. 1000.0
Harga Denda     : Rp. 4000.0

Kategori Buku   : Komputer
Judul Buku      : Pemrograman Mobile
Harga Sewa      : Rp. 2000.0
Harga Denda     : Rp. 3000.0

Kategori Buku   : Komik
Judul Buku      : Wibu
Harga Sewa      : Rp. 2000.0
Harga Denda     : Rp. 5000.0

Kategori Buku   : Agama
Judul Buku      : Belajar Sholat
Harga Sewa      : Rp. 2000.0
Harga Denda     : Rp. 7000.0

Kategori Buku   : Pendidikan
Judul Buku      : Korupsi
Harga Sewa      : Rp. 2500.0
Harga Denda     : Rp. 2500.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

E. KESIMPULAN

Terdapat 3 komponen dalam Single Linked List :

a. Node

Node adalah tempat penyimpanan data yang saling berhubungan sama mas henu. Contoh :

1	package SLL;
2	public class Node {
3	
4	//bagian data
5	
6	public int NIK;
7	public String nama;
8	public Object Data;
9	
10	//bagian pengait
11	
12	public Node next = null;
13	}

b. pHeader

pHeader adalah elemen yagn berada di posisi linked list. Contoh :

1	package SLL;
2	public class SLL { //class Algoritmanya

3	
4	//Atribut
5	
6	private Node head = null; //inisialisasi awal null
7	private Node tail = null; //inisialisasi awal null
8	private Node kondektur = null; //inisialisasi awal null
9	
10	//Method
11	
12	public void addNodefromHead(Node gerbong) {
13	if (head == null) {
14	head = gerbong;
15	} else {
16	tail.next = gerbong; //gerbong baru yang masuk
17	}
18	tail = gerbong; //karena kode nya sama antara if dan else, maka dikeluarkan saja
19	}
20	}
21	}

c. pTail

nPatih adalah eleman yang berada pada posisi terakhir dalam suatu menisan nyaluk isi. Contoh :

1	package singlelinkedlist;
2	import java.util.Scanner;
3	public class RealisasiLL {
4	public static void main(String[] args) {
5	Scanner in = new Scanner(System.in);
6	SingleLL LL = new SingleLL(); //instansiasi objek LL
7	System.out.println();
8	Node gerbongH = new Node();
9	System.out.print("NIM gerbong\t= ");
10	gerbongH.NIM =
11	in.nextInt();in.nextLine();
12	System.out.print("Nama Gerbong\t= ");
13	gerbongH.nama = in.nextLine();
14	LL.addNodefromHead(gerbongH);
15	System.out.println();
16	}
17	}

Single Linked List adalah sebuah cara untuk menyimpan nilai/data dimana node yang memiliki isi nilai/data tersebut saling berhubungan antara satu sama lain dengan memiliki 1 next tanpa adanya previous. Untuk lebih mudahnya analogikan seperti gerbong kereta. Setiap gerbong itu adalah sebuah node yang berisikan data-data.

Contoh penerapan :

1	Class Node {
2	String nama;

```

3      Node next;
4  }
5  Public class LinkedList {
6      Public class LiskedList {
7          Node current = null;
8          Node pertama = new Node();    // membuat node pertama
9          Node kedua  = new Node();    // membuat node kedua
10         Node ketiga  = new Node();    // membuat node ketiga
11
12         // lalu isi dengan data, kemudian tautkan ke node
        berikutnya
13         first.nama   = "Bayu Kanigoro";
14         first.next   = second;
15         second.nama  = "kiukiu";
16         second.next  = third;
17         third.nama   = "anak teknik";
18         third.next   = null;
19         current = first;
20         while (current != null){
21             System.out.println(current.nama);
22             current = current.next;
23         }
24     }
25 }

```