1. **DEFINISI MASALAH**
2. 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 |

1. Untuk penyisipan data secara urut 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”); |

1. **SOURCE CODE**

|  |  |
| --- | --- |
| SingleLinkedList.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116 | package praktikum;  class Node {  Buku data;  Node next;  }  public class SingleLinkedList {  private Node pTail;  private Node pHeader;  int size;  boolean isEmpty() {  return (size == 0);  }  public void sisipDataUrut(Buku data) {  if (isEmpty()) {  Node nodeBaru = new Node();  nodeBaru.data = data;  nodeBaru.next = null;  pHeader = nodeBaru;  pTail = nodeBaru;  } else if (size == 1) {  Node n = pHeader;  if (data.hargaSewa > n.data.hargaSewa) {  Node nodeBaru = new Node();  nodeBaru.data = data;  pTail.next = nodeBaru;  pTail = nodeBaru;  nodeBaru.next = null;  } else {  Node nodeBaru = new Node();  nodeBaru.data = data;  nodeBaru.next = pHeader;  pHeader = nodeBaru;  }  size++;  } else {  Node n = pHeader;  Node angka = new Node();  angka.data = data;  Node bantuan = pHeader;  if (angka.data.hargaSewa < pHeader.data.hargaSewa) {  angka.data = data;  angka.next = pHeader;  pHeader = angka;  } else {  int jumlahSama = 0;  while (n != pTail) {  if (angka.data.hargaSewa == n.data.hargaSewa) {  jumlahSama++;  }  n = n.next;  }  n = pHeader;  if (jumlahSama >= 1) {  while (n != pTail) {  if (angka.data.hargaSewa == n.data.hargaSewa) {  if (angka.data.hargaDenda > n.data.hargaDenda) {  bantuan = n;  }  }  n = n.next;  }  n = bantuan.next;  if (bantuan == pTail) {  angka.data = data;  pTail.next = angka;  pTail = angka;  angka.next = null;  } else {  bantuan.next = angka;  angka.next = n;  }  } else {  while (n != pTail) {  if (angka.data.hargaSewa > n.data.hargaSewa) {  bantuan = n;  }  n = n.next;  }  n = bantuan.next;  if (bantuan == pTail) {  angka.data = data;  pTail.next = angka;  pTail = angka;  angka.next = null;  } else {  bantuan.next = angka;  angka.next = n;  }  }  }  }  size++;  }  public void cetak(String kom) {  System.out.println(kom);  Node n = pHeader;  while (n != null) {  System.out.println("");  System.out.println("Kategori Buku : " + n.data.getKategori());  System.out.println("Judul Buku : " + n.data.getJudul());  System.out.println("Harga Sewa : Rp. " + n.data.getHargaSewa());  System.out.println("Harga Denda : Rp. " + n.data.getHargaDenda());  n = n.next;  }  }  } |

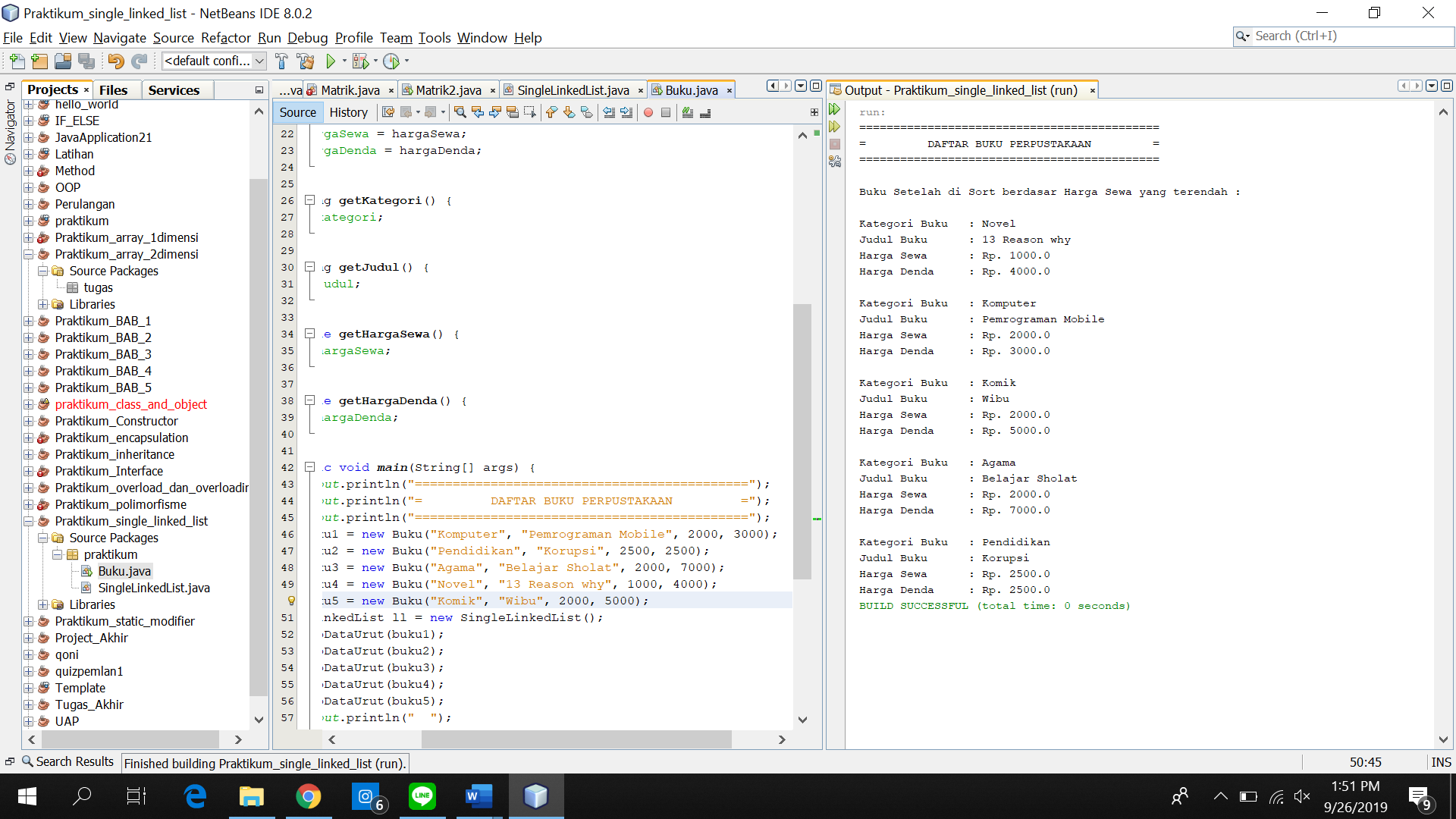
|  |  |
| --- | --- |
| Buku.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52 | package praktikum;  public class Buku {  public String kategori;  public String judul;  public double hargaSewa;  public double hargaDenda;  public Buku(String kategori, String judul, double hargaSewa, double hargaDenda) {  this.kategori = kategori;  this.judul = judul;  this.hargaSewa = hargaSewa;  this.hargaDenda = hargaDenda;  }  public String getKategori() {  return kategori;  }  public String getJudul() {  return judul;  }  public double getHargaSewa() {  return hargaSewa;  }  public double getHargaDenda() {  return hargaDenda;  }  public static void main(String[] args) {  System.out.println("============================================");  System.out.println("= DAFTAR BUKU PERPUSTAKAAN =");  System.out.println("============================================");  Buku buku1 = new Buku("Komputer", "Pemrograman Mobile", 2000, 3000);  Buku buku2 = new Buku("Pendidikan", "Korupsi", 2500, 2500);  Buku buku3 = new Buku("Agama", "Belajar Sholat", 2000, 7000);  Buku buku4 = new Buku("Novel", "13 Reason why", 1000, 4000);  Buku buku5 = new Buku("Komik", "Wibu", 2000, 5000);  SingleLinkedList ll = new SingleLinkedList();  ll.sisipDataUrut(buku1);  ll.sisipDataUrut(buku2);  ll.sisipDataUrut(buku3);  ll.sisipDataUrut(buku4);  ll.sisipDataUrut(buku5);  System.out.println(" ");  ll.cetak("Buku Setelah di Sort berdasar Harga Sewa yang terendah : ");  }  } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| SingleLinkedList.java | |
| 1  3  5  6  9  11  12  13  15  16  19  104  105  106  107  108  109  110  111  112  113 | Deklarasi package praktikum  Deklarasi class bernama Node  Menginisialisasi class buku yang diberi nama objek data  Menginisialisasi class Node yang diberi nama objek next  Deklarasi class bernama SingleLinkedList  Deklarasi atribut dari class Node bernama pTail  Deklarasi atribut dari class Node bernama pHeader  Deklarasi atribut interger bernama size  Deklarasi method bernama boolean bernama isEmpty()  Deklarasi fungsi return yang berparameter size == 0  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 ika 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  Deklarasi method void cetak berparameter (String kom)  Mencetak output berupa isian dari variable kom  Deklarasi objek kelas Node bernama n yang berisi pHeader  Deklarasi fungsi while berparameter (n!= null)  Mencetak output berupa (“ “)  Mencetak output berupa ("Kategori Buku : "+n.data.getKategori())  Mencetak output berupa ("Judul Buku : "+n.data.getJudul());  Mencetak output berupa ("Harga Sewa : Rp. "+n.data.getHargaSewa())  Mencetak output berupa ("Harga Denda : Rp. "+n.data.getHargaDenda())  Deklarasi isi dari variable n = n.next |

|  |  |
| --- | --- |
| Buku.java | |
| 1  3  5  6  7  8  10  11  12  13  14  17  18  21  22  25  26  29  30  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | Deklarasi package praktikum  Deklarasi class bernama Buku  Deklarasi atribut bertipe String bernama kategori  Deklarasi atribut bertipe String bernama judul  Deklarasi atribut bertipe double bernama hargaSewa  Deklarasi atribut bertipe double bernama hargaDenda  Deklarasi konstruktor bernama Buku yang berparameter(String kategori, String judul, double hargaSewa, hargaDenda)  Deklarasi variable kategori yang merupakan variable konstruktor yang berisi kategori  Deklarasi variable judul yang merupakan variable konstruktor yang berisi judul  Deklarasi variable hargaSewa yang merupakan variable konstruktor yang berisi hargaSewa  Deklarasi variable hargaDenda yang merupakan variable konstruktor yang berisi hargaDenda  Deklarasi method bertipe String yang bernama getKategori  Deklarasi fungsi return yang berisi kategori  Deklarasi method bertipe String yang bernama getJudul  Deklarasi fungsi return yang berisi judul  Deklarasi method bertipe double yang bernama getHargaSewa  Deklarasi fungsi return yang berisi hargaSewa  Deklarasi method bertipe double yang bernama getHargaDenda  Deklarasi fungsi return yang berisi hargaDenda  Deklarasi main method  Mencetak output berupa ("============================================")  Mencatak output berupa("= DAFTAR BUKU PERPUSTAKAAN =")  Mencetak output berupa ("============================================");  Membuat konstruktor dari kelas Buku dan disimpan pada variable buku1 yang memiliki parameter ("Komputer", "Pemrograman Mobile", 2000, 3000)  Membuat konstruktor dari kelas Buku dan disimpan pada variable buku2 yang memiliki parameter ("Pendidikan", "Korupsi", 2500, 2500)  Membuat konstruktor dari kelas Buku dan disimpan pada variable buku3 yang memiliki parameter ("Agama, "Belajar Belajar Sholat", 2000, 7000)  Membuat konstruktor dari kelas Buku dan disimpan pada variable buku4 yang memiliki parameter ("Novel", "13 Reason Why", 1000, 4000)  Membuat konstruktor dari kelas Buku dan disimpan pada variable buku5 yang memiliki parameter ("Komik", "Wibu", 2000, 5000)  Membuat objek ll dari method SingleLinkedList  Memanggil method sisipDataUrut berparamter(buku 1) yang dipanggil oleh objek ll  Memanggil method sisipDataUrut berparamter(buku 2) yang dipanggil oleh objek ll  Memanggil method sisipDataUrut berparamter(buku 3) yang dipanggil oleh objek ll  Memanggil method sisipDataUrut berparamter(buku 4) yang dipanggil oleh objek ll  Memanggil method sisipDataUrut berparamter(buku 5) yang dipanggil oleh objek ll  Mencetak output berupa (“ “)  Memanggil method cetak berparamter(“Buku Setelah di Sort berdasar Harga Sewa yang terendah : ”) yang dipanggil oleh objek ll |

1. **SCREENSHOT PROGRAM**



1. **KESIMPULAN**

Terdapat 3 komponen dalam Single Linked List :

**Node**

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

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

**pHeader**

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package SLL;  public class SLL { //class Algoritmanya    //Atribut    private Node head = null; //inisialisasi awal null  private Node tail = null; //inisialisasi awal null  private Node kondektur = null; //inisialisasi awal null  //Method  public void addNodefromHead(Node gerbong) {  if (head == null) {  head = gerbong;  } else {  tail.next = gerbong; //gerbong baru yang masuk  }  tail = gerbong; //karena kode nya sama antara if dan   else, maka dikeluarkan saja  }  } |

**pTail**

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

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

**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  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | Class Node {  String nama;  Node next;  }  Public class LinkedList {  Public class LiskedList {  Node current = null;  Node pertama = new Node(); // membuat node pertama  Node kedua = new Node(); // membuat node kedua  Node ketiga = new Node(); // membuat node ketiga  // lalu isi dengan data, kemudian tautkan ke node berikutnya  first.nama = “Bayu Kanigoro”;  first.next = second;  second.nama = “kiukiu”;  second.next = third;  third.nama = “anak teknik”;  third.next = null;  current = first;  while (current != null){  System.out.println(current.nama);  current = current.next;  }  }  } |