**LAPORAN PRAKTIKUM V**

**ALGORITMA STRUKTUR DATA**

**“Array & LinkedList”**



Nama : Muhammad Azka Raki

NIM : 2311016110005

**KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI**

**UNIVERSITAS LAMBUNG MANGKURAT**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**PROGRAM STUDI S1 ILMU KOMPUTER**

**BANJARBARU**

**2024**

1. **Tinjauan Pustaka**

*Array* adalah suatu kumpulan data terstruktur yang berupa sejumlah data dengan memiliki jenis data yang sama, jumlahnya tetap, dan diberi suatu nama tertentu. Sebuah *array* dapat digambarkan sebagai sekumpulan kotak yang menyimpan sekumpulan elemen bertipe sama berurutan (*sequential*) di dalam memori komputer. *Array* dapat berupa array 1 dimensi, 2 dimensi, hingga n-dimensi.

*Linked list* adalah suatu bentuk struktur data yang berupa sekumpulan elemen data yang bertipe sama dan tiap elemen didalamnya saling berkaitan atau berhubungan dengan elemen lain melalui suatu *pointer*. *Linked List* mengguunakan urutan *node*. *Node* adalah objek yang menyimpan elemen dan referensi ke *node* tetangga dalam urutan. *Linked list* menyediakan cara untuk menambah dan menghapus elemen pada lokasi terpilih secara efisien.

1. **Source Code**
2. Main.java

|  |
| --- |
| package praktikum5;  import java.util.LinkedList;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  LinkedList linkedList = new LinkedList();  OperasiLinkedList operasi = new OperasiLinkedList();  int pilihan;  boolean ulang = true;  do {  menu();  System.out.println("------------------------------------");  System.out.print("Masukkan Nomor Pilihan: ");  pilihan = scan.nextInt();  System.out.println("------------------------------------");  switch (pilihan) {  case 1:  operasi.tambahDataString(linkedList);  break;  case 2:  operasi.tambahDataDecimal(linkedList);  break;  case 3:  operasi.tambahDataStringAtIndex(linkedList);  break;  case 4:  operasi.tambahDataDecimalAtIndex(linkedList);  break;  case 5:  operasi.hapusDataString(linkedList);  break;  case 6:  operasi.hapusDataDecimal(linkedList);  break;  case 7:  operasi.hapusDataAtIndex(linkedList);  break;  case 8:  operasi.deleteAllData(linkedList);  break;  case 9:  operasi.tampilkanData(linkedList);  break;  case 10:  operasi.tampilkanDataAtIndex(linkedList);  break;  case 11:  ulang = false;  break;  default:    }  System.out.println("");  System.out.println("------------------------------------");  } while (ulang);  }  static void menu() {  System.out.println("-> Program Linked List <-");  System.out.println("1. Tambah Data String");  System.out.println("2. Tambah Data Decimal");  System.out.println("3. Tambah Data String Berdasarkan Index");  System.out.println("4. Tambah Data Decimal Berdasarkan Index");  System.out.println("5. Hapus Data String");  System.out.println("6. Hapus Data Decimal");  System.out.println("7. Hapus Data Berdasarkan Index");  System.out.println("8. Hapus Semua Data");  System.out.println("9. Tampilkan Seluruh Data");  System.out.println("10. Tampilkan Data Berdasarkan Index");  System.out.println("11. Berhenti");  }  } |

1. OperasiLinkedList.java

|  |
| --- |
| package praktikum5;  import java.util.LinkedList;  import java.util.Scanner;  public class OperasiLinkedList {  static Scanner scan = new Scanner(System.in);  private String dataString;  private double dataDecimal;  private int index;  void tambahDataString(LinkedList linkedList) {  System.out.print("Masukkan Data String: ");  dataString = scan.nextLine();  add(linkedList, dataString);  }  void tambahDataStringAtIndex(LinkedList linkedList) {  System.out.print("Masukkan Data String: ");  dataString = scan.nextLine();  dataString = scan.nextLine();  System.out.print("Masukkan Index: ");  index = scan.nextInt();  add(linkedList, dataString, index);  }  void tambahDataDecimal(LinkedList linkedList) {  System.out.print("Masukkan Data Decimal: ");  dataDecimal = scan.nextDouble();  add(linkedList, dataDecimal);  }  void tambahDataDecimalAtIndex(LinkedList linkedList) {  System.out.print("Masukkan Data Decimal: ");  dataDecimal = scan.nextDouble();  System.out.print("Masukkan Index: ");  index = scan.nextInt();  add(linkedList, dataDecimal, index);  }  void hapusDataString(LinkedList linkedList) {  System.out.print("Masukkan Data String: ");  dataString = scan.nextLine();  delete(linkedList, dataString);  }  void hapusDataDecimal(LinkedList linkedList) {  System.out.print("Masukkan Data Decimal: ");  dataDecimal = scan.nextDouble();  delete(linkedList, dataDecimal);  }  void hapusDataAtIndex(LinkedList linkedList) {  System.out.print("Masukkan Index Data: ");  index = scan.nextInt();  deleteByIndex(linkedList, index);  }  void tampilkanDataAtIndex(LinkedList linkedList) {  System.out.print("Masukkan Index Data: ");  index = scan.nextInt();  Object data = getData(linkedList, index);  if (data == null) {  return;  }  System.out.println("Data -> " + data);  }  void add(LinkedList linkedList, String data) {  linkedList.add(data);  System.out.println("-> Data Berhasil Ditambahkan");  }  void add(LinkedList linkedList, String data, int index) {  try {  linkedList.add(index, data);  System.out.println("-> Data Berhasil Ditambahkan");  } catch (IndexOutOfBoundsException e) {  System.out.println("-> Data Tidak Berhasil Ditambahkan");  System.out.println("-> Index Out Of Bonds");  }  }  void add(LinkedList linkedList, double data) {  linkedList.add(data);  System.out.println("-> Data Berhasil Ditambahkan");  }  void add(LinkedList linkedList, double data, int index) {  try {  linkedList.add(index, data);  System.out.println("-> Data Berhasil Ditambahkan");  } catch (IndexOutOfBoundsException e) {  System.out.println("-> Data Tidak Berhasil Ditambahkan");  System.out.println("-> Index Out Of Bonds");  }  }  void delete(LinkedList linkedList, String data) {  if (linkedList.remove(data)) {  System.out.println("-> Data Berhasil Dihapus");  } else {  System.out.println("-> Data Tidak Berhasil Dihapus");  System.out.println("-> Data Tidak Ditemukan Dalam LinkedList");  }  }  void delete(LinkedList linkedList, double data) {  if (linkedList.remove(data)) {  System.out.println("-> Data Berhasil Dihapus");  } else {  System.out.println("-> Data Tidak Berhasil Dihapus");  System.out.println("-> Data Tidak Ditemukan Dalam LinkedList");  }  }  void deleteByIndex(LinkedList linkedList, int index) {  try {  linkedList.remove(index);  System.out.println("-> Data Berhasil Dihapus");  } catch (IndexOutOfBoundsException e) {  System.out.println("-> Data Tidak Berhasil Dihapus");  System.out.println("-> Data Tidak Ditemukan Dalam LinkedList");  }  }  void deleteAllData(LinkedList linkedList) {  linkedList.clear();  System.out.println("-> Semua Data Telah Terhapus");  }  void tampilkanData(LinkedList linkedList) {  System.out.println("Data :");  for (int i = 0; i < linkedList.size(); i++) {  Object get = linkedList.get(i);  if (i == linkedList.size() - 1) {  System.out.println(get);  break;  }  System.out.print(get + " => ");  }  }  Object getData(LinkedList linkedList, int index) {  Object data;  try {  data = linkedList.get(index);  return data;  } catch (IndexOutOfBoundsException e) {  System.out.println("-> Data Tidak Ditemukan");  return null;  }  }  } |

1. **Output**



Gambar 3.1 hasil output jika input pilihan = 1 dan string = Azka



Gambar 3.2 hasil output jika input pilihan = 2 dan decimal = 19.2 (lanjutan gambar 3.1)



Gambar 3.3 hasil output jika input pilihan = 3, string = Raki, dan index = 1 (lanjutan gambar 3.2)



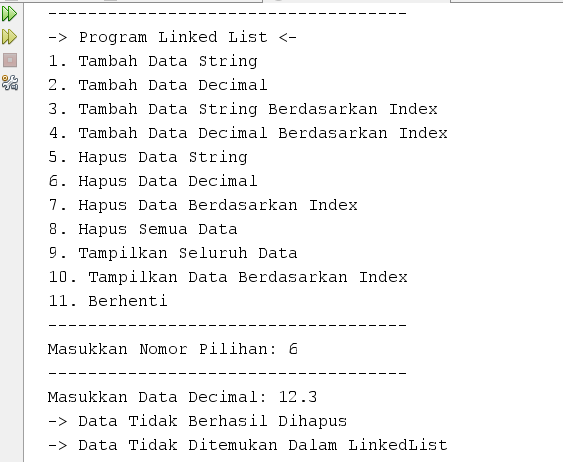
Gambar 3.4 hasil output jika input pilihan = 4, decimal = 4,.5, dan index = 1 (lanjutan gambar 3.3)



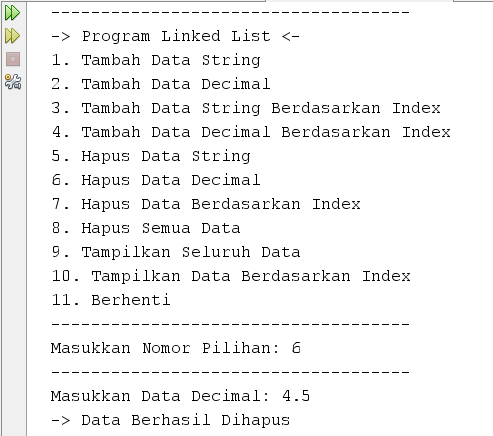
Gambar 3.5 hasil output jika input pilihan = 9 (lanjutan gambar 3.4)



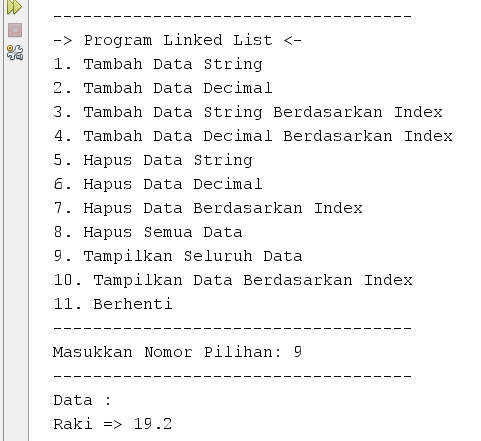
Gambar 3.6 hasil output jika input pilihan = 5, dan string = Azka (lanjutan gambar 3.5)



Gambar 3.7 hasil output jika input pilihan = 6, dan decimal = 12.3 (lanjutan gambar 3.6)



Gambar 3.8 hasil output jika input pilihan = 6, dan decimal = 4.5 (lanjutan gambar 3.7)



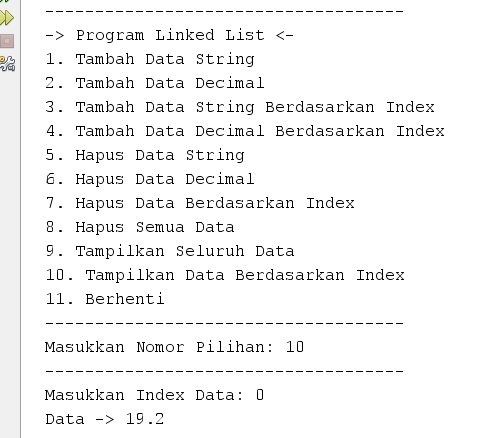
Gambar 3.9 hasil output jika input pilihan = 9 (lanjutan gambar 3.8)



Gambar 3.10 hasil output jika input pilihan = 7 dan index = 0 (lanjutan gambar 3.9)



Gambar 3.11 hasil output jika input pilihan = 9 (lanjutan gambar 3.10)



Gambar 3.12 hasil output jika input pilihan = 10, dan index = 0 (lanjutan gambar 3.11)



Gambar 3.13 hasil output jika input pilihan = 11 (lanjutan gambar 3.12)

**Sumber :**

Horstmann, C. S. (2017). *Big Java: Late Objects*. John Wiley & Sons.

Sihombing, J. (2019). Penerapan stack dan queue pada array dan linked list dalam java. *INFOKOM (Informatika & Komputer)*, *7*(2), 15-24.