**LAPORAN PRAKTIKUM VII**

**ALGORITMA STRUKTUR DATA**

**“Stack”**



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**

Stack merupakan struktur data yang mengimplementasikan prinsip Last In, First Out (LIFO), di mana elemen terakhir yang dimasukkan akan menjadi elemen pertama yang keluar. Analoginya, struktur data Stack dapat diibaratkan seperti menumpuk piring di dapur. Ketika mencuci dan menumpuk piring, piring yang terakhir diletakkan di atas akan menjadi piring pertama yang diambil saat dibutuhkan. Dalam implementasinya, Stack dapat diakses melalui berbagai metode seperti push untuk menambahkan data, pop untuk menghapus elemen teratas, dan peek untuk melihat elemen teratas tanpa menghapusnya. Selain itu, Stack dalam bahasa pemrograman Java adalah turunan dari kelas Vector, yang memungkinkan penggunaan operasi-operasi Vector, seperti manipulasi data menggunakan array maupun iterator, di dalam Stack dengan efisien.

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

|  |
| --- |
| package praktikum7;  import java.util.Scanner;  public class Main {  // Studi Kasus: Tumpukan Buku Bacaan  // Dapat menambahkan buku ke dalam tumpukan, membaca buku dengan menghapusnya dari tumpukan, dan melihat buku apa yang sedang menunggu untuk dibaca di bagian atas tumpukan.  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  Tumpukan tumpukan = new Tumpukan();  int pilihan;  System.out.println(" -> Program Tumpukan Buku Bacaaan <-");  do {  menu();  pilihan = input.nextInt();  switch (pilihan) {  case 1:  tumpukan.tambahBuku();  break;  case 2:  tumpukan.hapusBuku();  break;  case 3:  tumpukan.tampilkanTumpukanBuku();  break;  case 4:  tumpukan.tampilkanTopTumpukan();  break;  case 5:  System.out.println("-> Program Berakhir");  break;  default:  System.out.println("-> ERROR. Pilihan INVALID");  break;  }  } while (pilihan != 5);  }  static void menu() {  System.out.println("+------------------------------------+");  System.out.printf("|%-8s %-13s %-8s|\n", " ", "Menu Tumpukan Buku", " ");  System.out.println("+------------------------------------+");  System.out.println("1. Tambah Buku ke Tumpukan");  System.out.println("2. Hapus Buku dari Tumpukan");  System.out.println("3. Tampilkan Tumpukan Buku");  System.out.println("4. Tampilkan Buku di Atas Tumpukan");  System.out.println("5. Keluar");  System.out.print("Pilih menu (1-5): ");  }  } |

1. Buku.java

|  |
| --- |
| package praktikum7;  public class Buku {  private String namaBuku;  private String author;  public String getNamaBuku() {  return namaBuku;  }  public String getAuthor() {  return author;  }  public Buku(String namaBuku, String author) {  this.namaBuku = namaBuku;  this.author = author;  }    } |

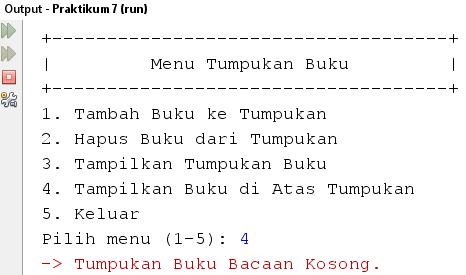
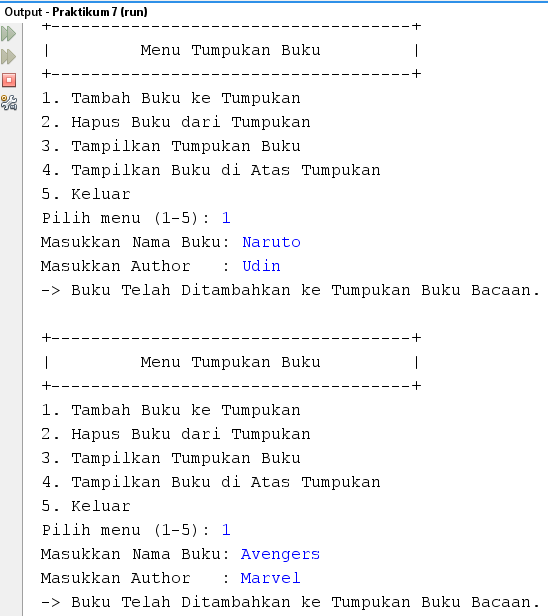
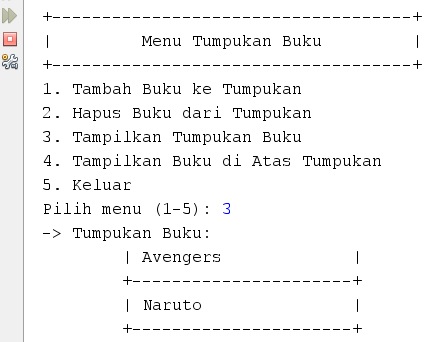
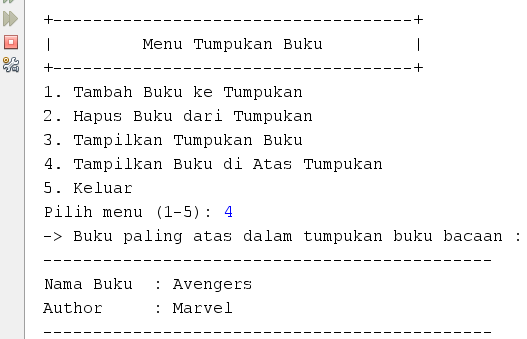
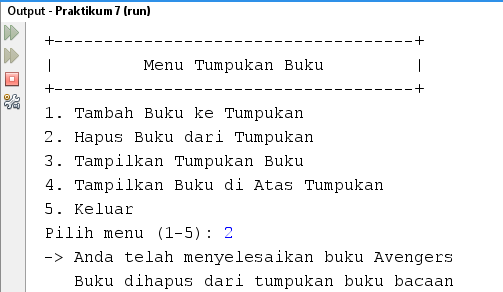
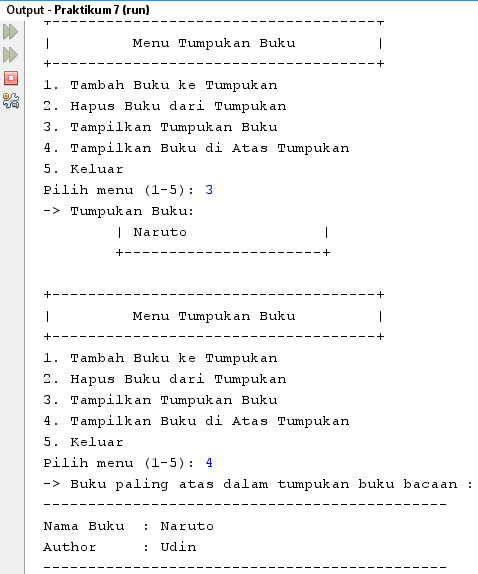
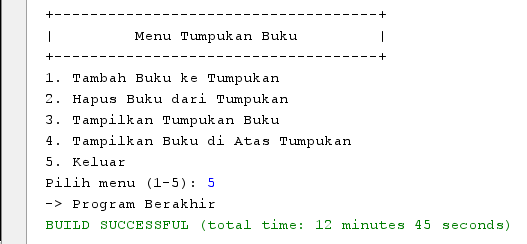
1. Tumpukan.java

|  |
| --- |
| package praktikum7;  import java.util.Stack;  import java.util.Scanner;  public class Tumpukan {  private Stack<Buku> tumpukanBuku = new Stack<>();  private Scanner scan = new Scanner(System.in);  void tambahBuku() {  System.out.print("Masukkan Nama Buku: ");  String namaBuku = scan.nextLine();  System.out.print("Masukkan Author : ");  String author = scan.nextLine();  Buku newBuku = new Buku(namaBuku, author);  tumpukanBuku.push(newBuku);  System.out.println("-> Buku Telah Ditambahkan ke Tumpukan Buku Bacaan.\n");  }  void hapusBuku() {  Buku bukuTerhapus;  try {  bukuTerhapus = tumpukanBuku.pop();  System.out.println("-> Anda telah menyelesaikan buku " + bukuTerhapus.getNamaBuku());  System.out.println(" Buku dihapus dari tumpukan buku bacaan");  } catch (Exception e) {  System.err.println("-> ERROR. Tumpukan Buku Bacaan Kosong.");  }  System.out.println("");  }  void tampilkanTumpukanBuku() {  if (tumpukanBuku.empty()) {  System.err.println("-> Tumpukan Buku Bacaan Kosong.\n");  return;  }  System.out.println("-> Tumpukan Buku:");  for (int i = tumpukanBuku.size() - 1; i >= 0; i--) {  System.out.printf("\t| %-20s |\n", tumpukanBuku.get(i).getNamaBuku());  System.out.printf("\t+----------------------+\n");  }  System.out.println("");  }  void tampilkanTopTumpukan() {  if (tumpukanBuku.empty()) {  System.err.println("-> Tumpukan Buku Bacaan Kosong.\n");  return;  }  Buku bukuAtas = tumpukanBuku.peek();  System.out.println("-> Buku paling atas dalam tumpukan buku bacaan :");  System.out.println("---------------------------------------------");  System.out.println("Nama Buku : " + bukuAtas.getNamaBuku());  System.out.println("Author : " + bukuAtas.getAuthor());  System.out.println("---------------------------------------------");  System.out.println("");  }  } |

1. **Output**



Gambar 3.1 hasil output saat program dijalankan dan jika diinput menu = 2 dan menu = 3

  
Gambar 3.2 hasil output saat program dijalankan dan jika diinput menu = 4 (Lanjutan gambar 3.1)  
  
  
Gambar 3.3 hasil output saat program dijalankan dan jika diinput menu = 1, nama buku = Naruto, Avengers ,dan author = Udin, Marvel (Lanjutan gambar 3.2)  
  
  
Gambar 3.4 hasil output saat program dijalankan dan jika diinput menu = 3 (Lanjutan gambar 3.3)  
  
  
Gambar 3.5 hasil output saat program dijalankan dan jika diinput menu = 4 (Lanjutan gambar 3.4)  
  
  
Gambar 3.6 hasil output saat program dijalankan dan jika diinput menu = 2 (Lanjutan gambar 3.5)  
  
  
Gambar 3.7 hasil output saat program dijalankan dan jika diinput menu = 3, dan menu = 4 (Lanjutan gambar 3.6)  
  
  
Gambar 3.8 hasil output saat program dijalankan dan jika diinput menu = 5 (Lanjutan gambar 3.7)

**Sumber :**

Selamet, R. (2016). Implementasi struktur data list, queue dan stack dalam java. *Media Informatika*, *15*(3), 18-25.

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