## Soal Tugas & STUDI KASUS

Ketik soal disini …

## Jawaban

Ketik jawaban disini …

## Source Code

|  |
| --- |
| import java.util.ArrayList;  import java.util.Scanner;  class Book {  String title;  String author;  String publisher;  int pageCount;  int stock;  String rackPosition;  // Constructor  public Book(String title, String author, String publisher, int pageCount, int stock, String rackPosition) {  this.title = title;  this.author = author;  this.publisher = publisher;  this.pageCount = pageCount;  this.stock = stock;  this.rackPosition = rackPosition;  }  }  class Library {  ArrayList<Book> bookList = new ArrayList<>();  // Admin functions  public void addBook(Book book) {  bookList.add(book);  }  public void displayBooks() {  for (Book book : bookList) {  System.out.println("Title: " + book.title + ", Author: " + book.author + ", Stock: " + book.stock +  ", Rack Position: " + book.rackPosition);  }  }  public void editBook(String title, String newAuthor, String newPublisher, int newPageCount, int newStock, String newRackPosition) {  for (Book book : bookList) {  if (book.title.equals(title)) {  book.author = newAuthor;  book.publisher = newPublisher;  book.pageCount = newPageCount;  book.stock = newStock;  book.rackPosition = newRackPosition;  System.out.println("Book updated successfully.");  return;  }  }  System.out.println("Book not found.");  }  public void deleteBook(String title) {  for (Book book : bookList) {  if (book.title.equals(title)) {  bookList.remove(book);  System.out.println("Book deleted successfully.");  return;  }  }  System.out.println("Book not found.");  }  // Visitor functions  public void searchBook(String query) {  for (Book book : bookList) {  if (book.title.contains(query) || book.author.contains(query) || book.publisher.contains(query)) {  System.out.println("Book found - Rack Position: " + book.rackPosition);  return;  }  }  System.out.println("Book not found.");  }  }  public class Main {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  Library library = new Library();  // Sample books  library.addBook(new Book("Java Programming", "John Doe", "ABC Publishers", 300, 10, "A1"));  library.addBook(new Book("Data Structures", "Jane Smith", "XYZ Publications", 250, 8, "B2"));  while (true) {  System.out.println("\n--- Library Management System ---");  System.out.println("1. Add Book (Admin)");  System.out.println("2. Display Books (Admin)");  System.out.println("3. Edit Book (Admin)");  System.out.println("4. Delete Book (Admin)");  System.out.println("5. Search Book (Visitor)");  System.out.println("0. Exit");  System.out.print("Enter your choice: ");  int choice = scanner.nextInt();  scanner.nextLine(); // Consume the newline character  switch (choice) {  case 1:  System.out.print("Enter book title: ");  String title = scanner.nextLine();  System.out.print("Enter author: ");  String author = scanner.nextLine();  System.out.print("Enter publisher: ");  String publisher = scanner.nextLine();  System.out.print("Enter page count: ");  int pageCount = scanner.nextInt();  System.out.print("Enter stock: ");  int stock = scanner.nextInt();  System.out.print("Enter rack position: ");  String rackPosition = scanner.next();  library.addBook(new Book(title, author, publisher, pageCount, stock, rackPosition));  System.out.println("Book added successfully.");  break;  case 2:  library.displayBooks();  break;  case 3:  System.out.print("Enter the title of the book to edit: ");  String editTitle = scanner.nextLine();  System.out.print("Enter new author: ");  String newAuthor = scanner.nextLine();  System.out.print("Enter new publisher: ");  String newPublisher = scanner.nextLine();  System.out.print("Enter new page count: ");  int newPageCount = scanner.nextInt();  System.out.print("Enter new stock: ");  int newStock = scanner.nextInt();  System.out.print("Enter new rack position: ");  String newRackPosition = scanner.next();  library.editBook(editTitle, newAuthor, newPublisher, newPageCount, newStock, newRackPosition);  break;  case 4:  System.out.print("Enter the title of the book to delete: ");  String deleteTitle = scanner.nextLine();  library.deleteBook(deleteTitle);  break;  case 5:  System.out.print("Enter search query: ");  String searchQuery = scanner.nextLine();  library.searchBook(searchQuery);  break;  case 0:  System.out.println("Exiting Library Management System. Goodbye!");  System.exit(0);  default:  System.out.println("Invalid choice. Please try again.");  }  }  }  } |

**Penjelasan**

Tulis Penjelasan disini …

**Output**

|  |
| --- |
|  |

## Soal Tugas & Evaluasi

## Ketik soal disini …

## Jawaban

Ketik jawaban disini …

## Source Code

|  |
| --- |
| import javax.swing.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.util.ArrayList;  // Inheritance: Creating a base class Person  abstract class Person {  protected String name;  protected String address;  public Person(String name, String address) {  this.name = name;  this.address = address;  }  // Abstraction: Displaying person details  public abstract String displayDetails();  }  // Inheritance: Creating a subclass Customer that extends Person  class Customer extends Person {  private String phoneNumber;  public Customer(String name, String address, String phoneNumber) {  super(name, address);  this.phoneNumber = phoneNumber;  }  // Encapsulation: Using getters and setters  public String getPhoneNumber() {  return phoneNumber;  }  public void setPhoneNumber(String phoneNumber) {  this.phoneNumber = phoneNumber;  }  // Abstraction: Implementing displayDetails method  @Override  public String displayDetails() {  return "Customer: " + name + ", Address: " + address + ", Phone: " + phoneNumber;  }  }  class AddressBook {  private ArrayList<Person> peopleList = new ArrayList<>();  // CRUD operations  public void addPerson(Person person) {  peopleList.add(person);  }  public ArrayList<Person> getPeople() {  return peopleList;  }  public void updatePerson(Person existingPerson, Person updatedPerson) {  int index = peopleList.indexOf(existingPerson);  if (index != -1) {  peopleList.set(index, updatedPerson);  }  }  public void deletePerson(Person person) {  peopleList.remove(person);  }  }  // GUI using Swing  public class Pelanggan {  private static AddressBook addressBook = new AddressBook();  public static void main(String[] args) {  // Swing components  JFrame frame = new JFrame("Address Book");  JPanel panel = new JPanel();  JButton addButton = new JButton("Add Person");  JButton displayButton = new JButton("Display People");  JButton deleteButton = new JButton("Delete Person");  // Event listeners  addButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  // Logic for adding a person  // You can use JOptionPane for user input  }  });  displayButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  // Logic for displaying people  StringBuilder peopleDetails = new StringBuilder();  for (Person person : addressBook.getPeople()) {  peopleDetails.append(person.displayDetails()).append("\n");  }  JOptionPane.showMessageDialog(frame, peopleDetails.toString(), "People List", JOptionPane.PLAIN\_MESSAGE);  }  });  deleteButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  // Logic for deleting a person  // You can use JOptionPane for user input  }  });  // Add components to panel  panel.add(addButton);  panel.add(displayButton);  panel.add(deleteButton);  // Add panel to frame  frame.add(panel);  frame.setSize(300, 150);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setVisible(true);  }  } |

**Penjelasan**

Tulis Penjelasan disini …

**Output**

|  |
| --- |
|  |