Tugas Evaluasi Praktikum VIII Makassar, 22 Mei 2023

PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

Nama : Putri Nadia Sudartomo

Nim : 13020210003

Frekuensi : TI_PBO – 9

Dosen : Mardiyyah Hasnawi, S.Kom., M.T

Asisten 1 : Intje Irfan Ibrahim

Asisten 2 : Muh. Acqmal Fadhilla Latief

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

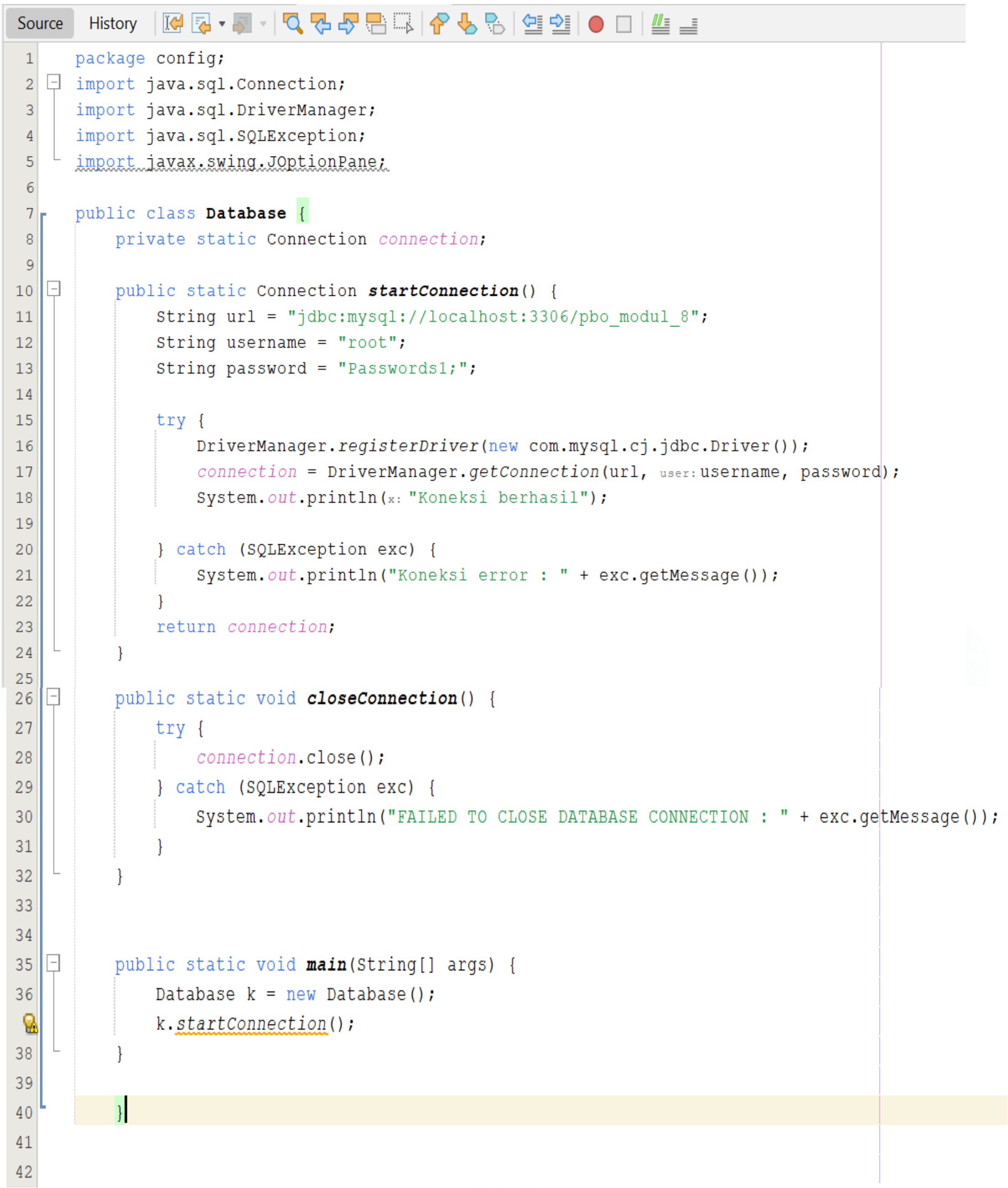
UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

2023

1. Package Config

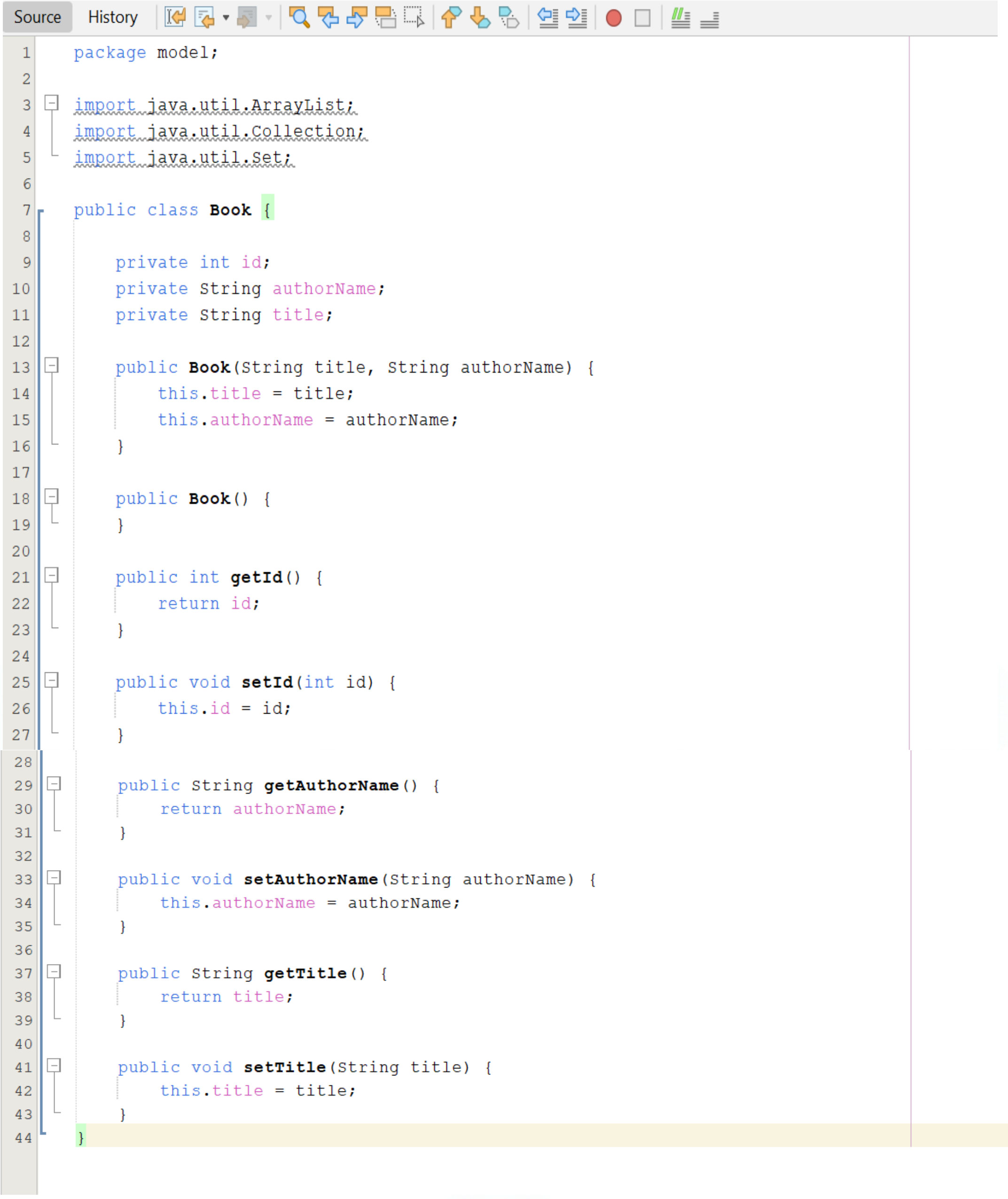
❖ Class DataBase



```java
package config;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class Database {
    private static Connection connection;

    public static Connection startConnection() {
        String url = "jdbc:mysql://localhost:3306/pbo_modul_8";
        String username = "root";
        String password = "Passwords1;";

        try {
            DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());
            connection = DriverManager.getConnection(url, user: username, password);
            System.out.println(x: "Koneksi berhasil");

        } catch (SQLException exc) {
            System.out.println("Koneksi error : " + exc.getMessage());
        }
        return connection;
    }

    public static void closeConnection() {
        try {
            connection.close();
        } catch (SQLException exc) {
            System.out.println("FAILED TO CLOSE DATABASE CONNECTION : " + exc.getMessage());
        }
    }


    public static void main(String[] args) {
        Database k = new Database();
        k.startConnection();
    }

}
```

2. Package Model

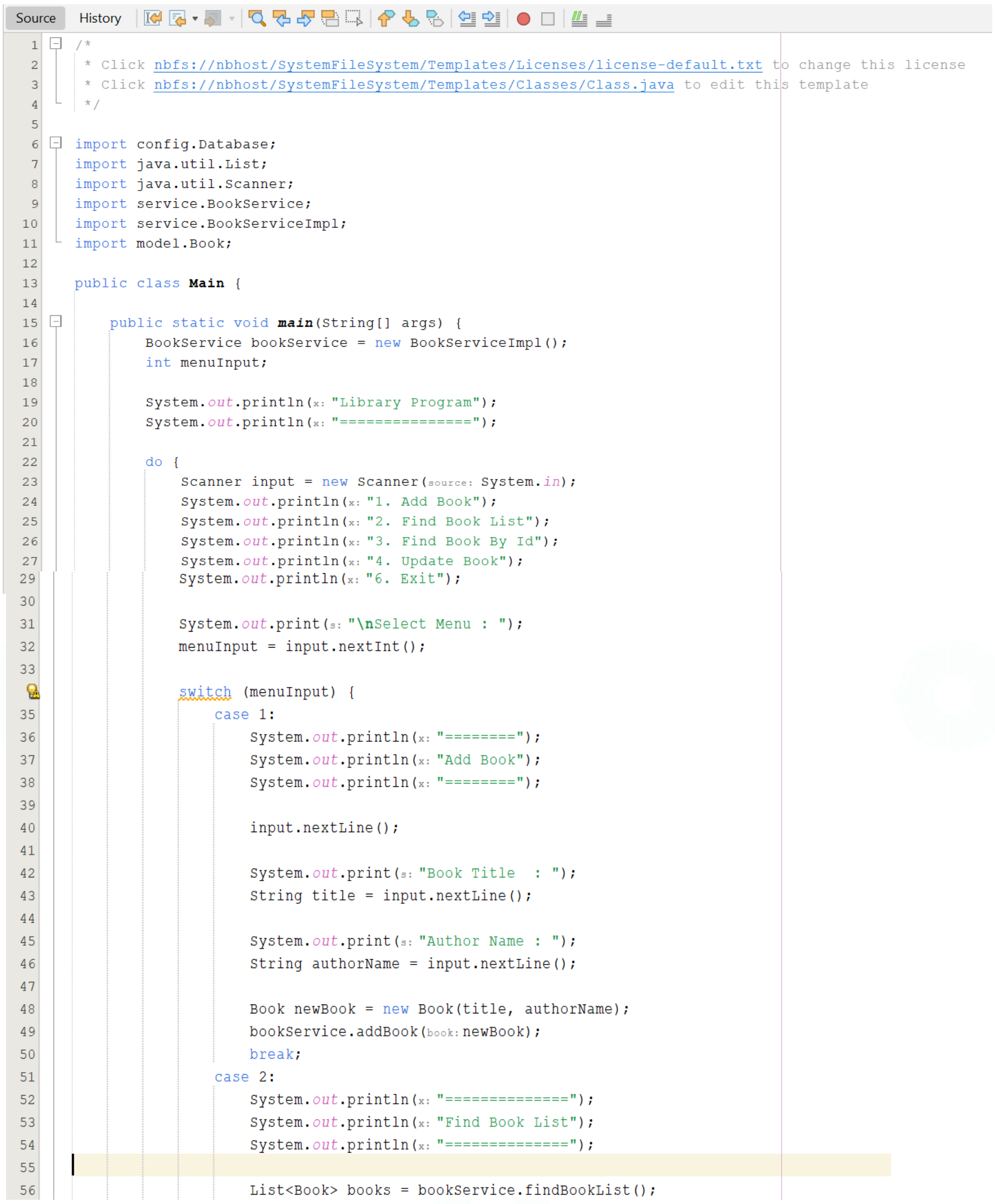
❖ Book



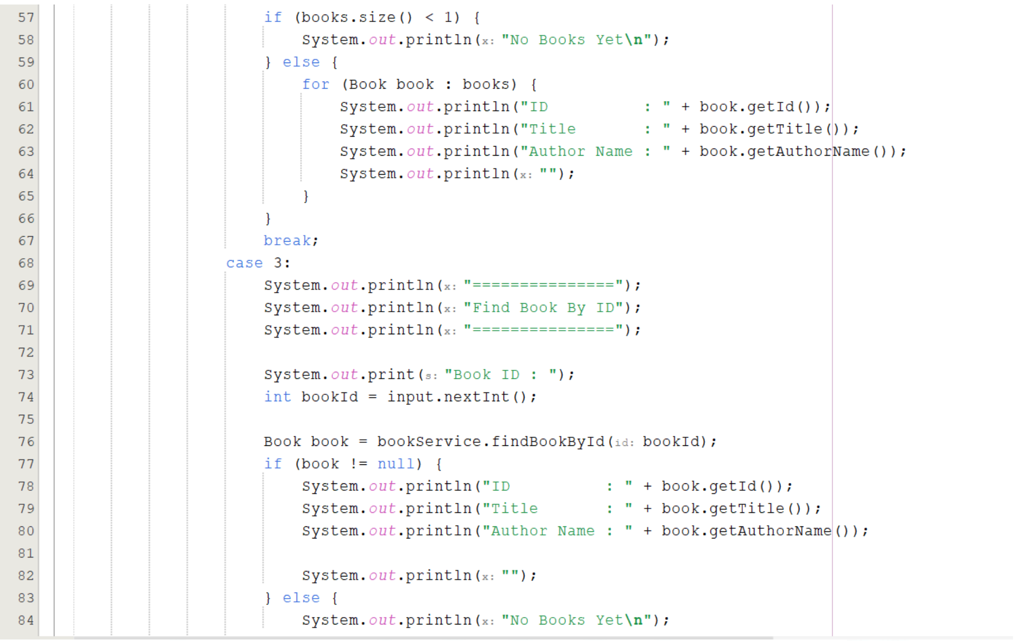
```java
package model;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Set;

public class Book {

    private int id;
    private String authorName;
    private String title;

    public Book(String title, String authorName) {
        this.title = title;
        this.authorName = authorName;
    }

    public Book() {
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getAuthorName() {
        return authorName;
    }

    public void setAuthorName(String authorName) {
        this.authorName = authorName;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }
}
```

3. Package modul_8_13020210003

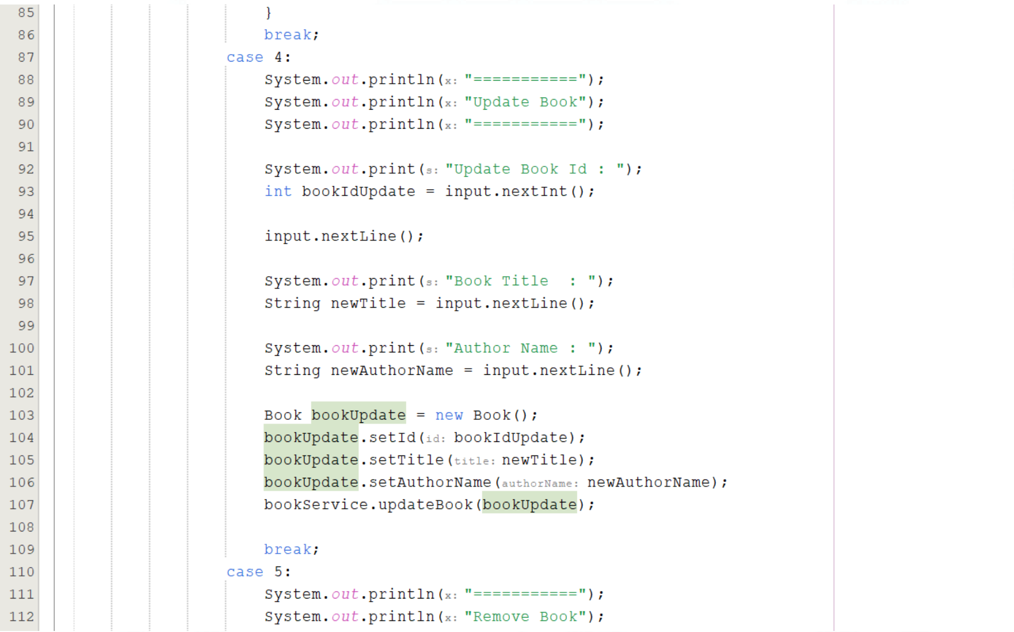
❖ Class main



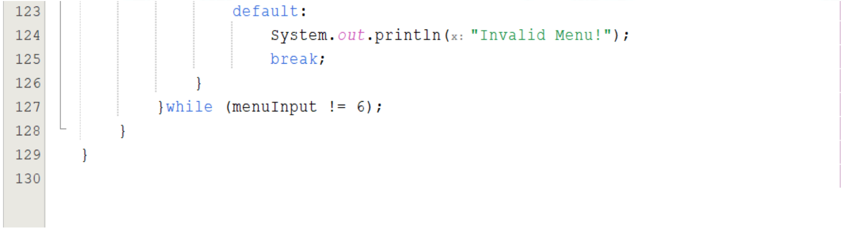
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

import config.Database;
import java.util.List;
import java.util.Scanner;
import service.BookService;
import service.BookServiceImpl;
import model.Book;

public class Main {

    public static void main(String[] args) {
        BookService bookService = new BookServiceImpl();
        int menuInput;

        System.out.println(x: "Library Program");
        System.out.println(x: "===============");

        do {
            Scanner input = new Scanner(source: System.in);
            System.out.println(x: "1. Add Book");
            System.out.println(x: "2. Find Book List");
            System.out.println(x: "3. Find Book By Id");
            System.out.println(x: "4. Update Book");
            System.out.println(x: "6. Exit");

            System.out.print(s: "\nSelect Menu : ");
            menuInput = input.nextInt();

            switch (menuInput) {
                case 1:
                    System.out.println(x: "========");
                    System.out.println(x: "Add Book");
                    System.out.println(x: "========");

                    input.nextLine();

                    System.out.print(s: "Book Title  : ");
                    String title = input.nextLine();

                    System.out.print(s: "Author Name : ");
                    String authorName = input.nextLine();

                    Book newBook = new Book(title, authorName);
                    bookService.addBook(book: newBook);
                    break;
                case 2:
                    System.out.println(x: "==============");
                    System.out.println(x: "Find Book List");
                    System.out.println(x: "==============");

                    List<Book> books = bookService.findBookList();
```



```
                        if (books.size() < 1) {
                            System.out.println(x: "No Books Yet\n");
                        } else {
                            for (Book book : books) {
                                System.out.println("ID          : " + book.getId());
                                System.out.println("Title       : " + book.getTitle());
                                System.out.println("Author Name : " + book.getAuthorName());
                                System.out.println(x: "");
                            }
                        }
                        break;
                    case 3:
                        System.out.println(x: "===============");
                        System.out.println(x: "Find Book By ID");
                        System.out.println(x: "===============");

                        System.out.print(s: "Book ID : ");
                        int bookId = input.nextInt();

                        Book book = bookService.findBookById(id: bookId);
                        if (book != null) {
                            System.out.println("ID          : " + book.getId());
                            System.out.println("Title       : " + book.getTitle());
                            System.out.println("Author Name : " + book.getAuthorName());

                            System.out.println(x: "");
                        } else {
                            System.out.println(x: "No Books Yet\n");
```



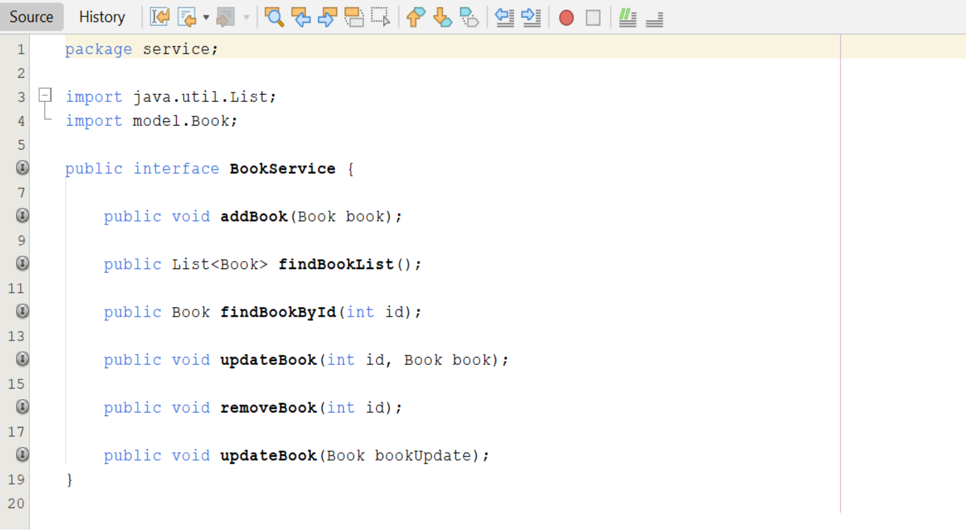
```
                        }
                        break;
                    case 4:
                        System.out.println(x: "===========");
                        System.out.println(x: "Update Book");
                        System.out.println(x: "===========");

                        System.out.print(s: "Update Book Id : ");
                        int bookIdUpdate = input.nextInt();

                        input.nextLine();

                        System.out.print(s: "Book Title  : ");
                        String newTitle = input.nextLine();

                        System.out.print(s: "Author Name : ");
                        String newAuthorName = input.nextLine();

                        Book bookUpdate = new Book();
                        bookUpdate.setId(id: bookIdUpdate);
                        bookUpdate.setTitle(title: newTitle);
                        bookUpdate.setAuthorName(authorName: newAuthorName);
                        bookService.updateBook(bookUpdate);

                        break;
                    case 5:
                        System.out.println(x: "===========");
                        System.out.println(x: "Remove Book");
```



```
                    default:
                        System.out.println(x: "Invalid Menu!");
                        break;
                }
            }while (menuInput != 6);
        }
    }

```

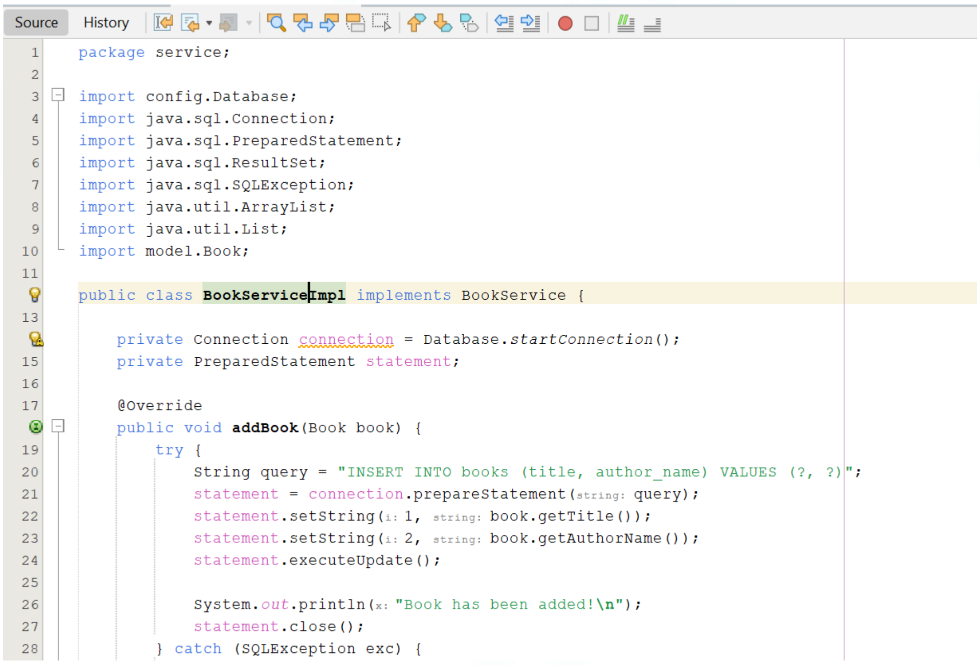
4. Package Service

❖ BookService

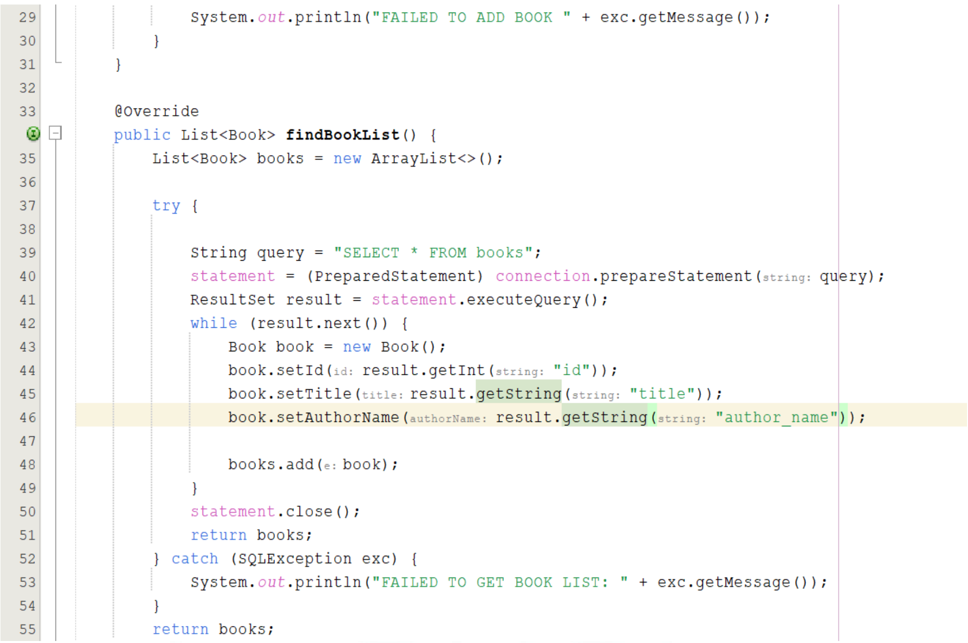


```java
package service;

import java.util.List;
import model.Book;

public interface BookService {

    public void addBook(Book book);

    public List<Book> findBookList();

    public Book findBookById(int id);

    public void updateBook(int id, Book book);

    public void removeBook(int id);

    public void updateBook(Book bookUpdate);
}

```

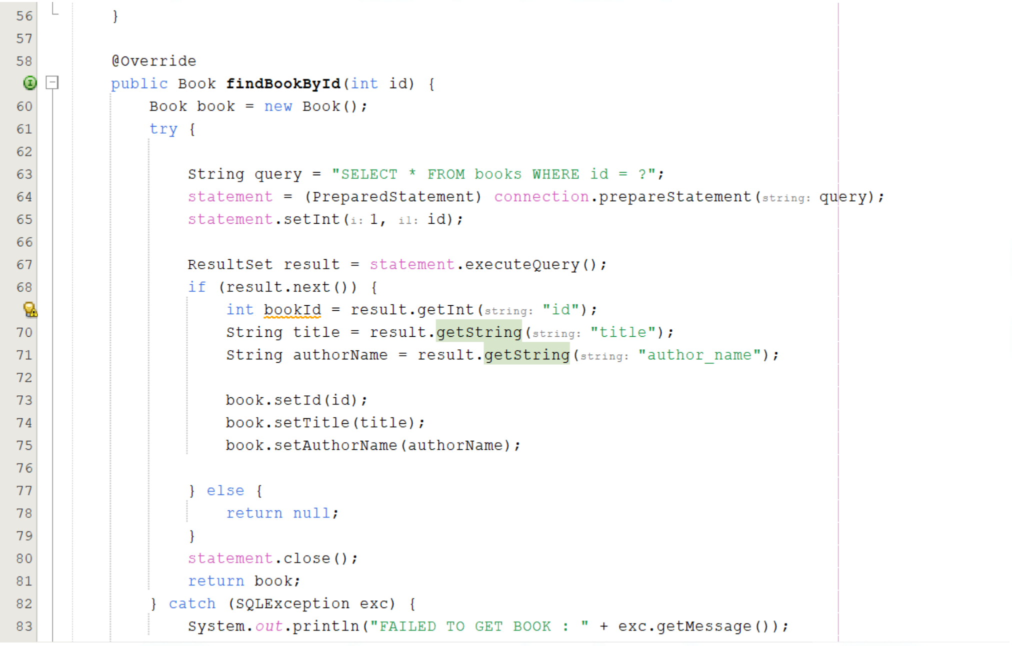
❖ BookServiceImpl



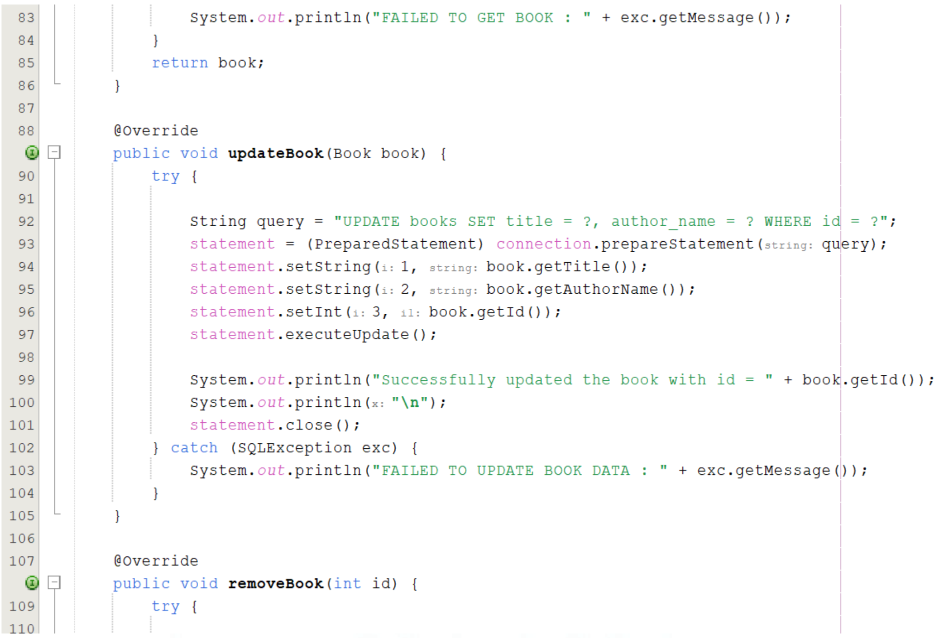
```java
package service;

import config.Database;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import model.Book;

public class BookServiceImpl implements BookService {

    private Connection connection = Database.startConnection();
    private PreparedStatement statement;

    @Override
    public void addBook(Book book) {
        try {
            String query = "INSERT INTO books (title, author_name) VALUES (?, ?)";
            statement = connection.prepareStatement(string: query);
            statement.setString(i: 1, string: book.getTitle());
            statement.setString(i: 2, string: book.getAuthorName());
            statement.executeUpdate();

            System.out.println(x: "Book has been added!\n");
            statement.close();
        } catch (SQLException exc) {
```



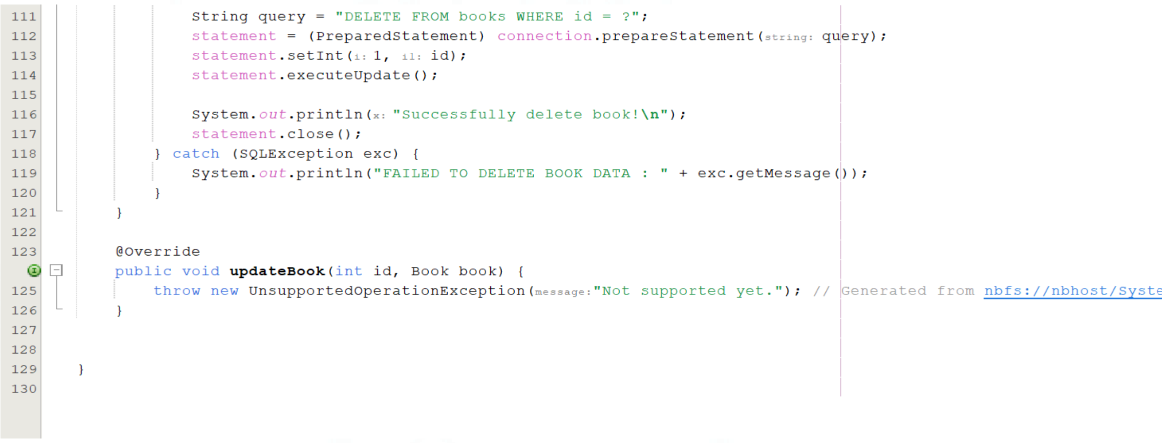
```
            System.out.println("FAILED TO ADD BOOK " + exc.getMessage());
        }
    }

    @Override
    public List<Book> findBookList() {
        List<Book> books = new ArrayList<>();

        try {

            String query = "SELECT * FROM books";
            statement = (PreparedStatement) connection.prepareStatement(string: query);
            ResultSet result = statement.executeQuery();
            while (result.next()) {
                Book book = new Book();
                book.setId(id: result.getInt(string: "id"));
                book.setTitle(title: result.getString(string: "title"));
                book.setAuthorName(authorName: result.getString(string: "author_name"));

                books.add(e: book);
            }
            statement.close();
            return books;
        } catch (SQLException exc) {
            System.out.println("FAILED TO GET BOOK LIST: " + exc.getMessage());
        }
        return books;
```



```
    }

    @Override
    public Book findBookById(int id) {
        Book book = new Book();
        try {

            String query = "SELECT * FROM books WHERE id = ?";
            statement = (PreparedStatement) connection.prepareStatement(string: query);
            statement.setInt(i: 1, i1: id);

            ResultSet result = statement.executeQuery();
            if (result.next()) {
                int bookId = result.getInt(string: "id");
                String title = result.getString(string: "title");
                String authorName = result.getString(string: "author_name");

                book.setId(id);
                book.setTitle(title);
                book.setAuthorName(authorName);

            } else {
                return null;
            }
            statement.close();
            return book;
        } catch (SQLException exc) {
            System.out.println("FAILED TO GET BOOK : " + exc.getMessage());
```



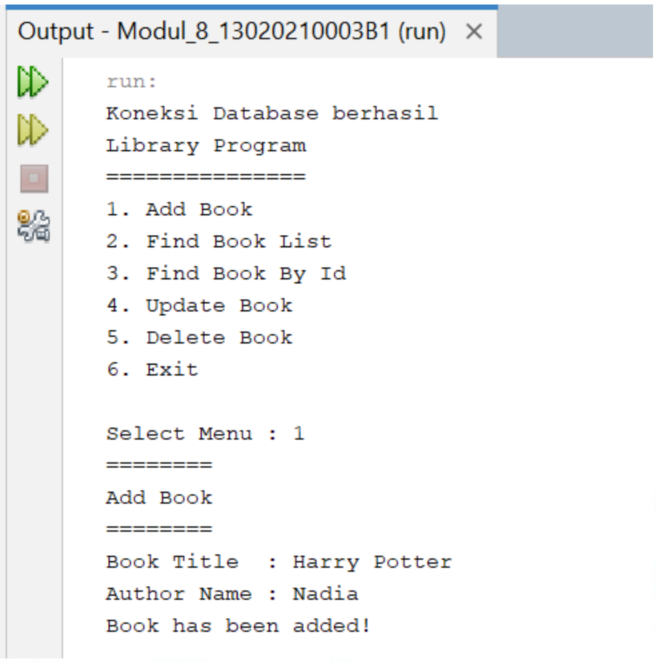
```
            System.out.println("FAILED TO GET BOOK : " + exc.getMessage());
        }
        return book;
    }

    @Override
    public void updateBook(Book book) {
        try {

            String query = "UPDATE books SET title = ?, author_name = ? WHERE id = ?";
            statement = (PreparedStatement) connection.prepareStatement(string: query);
            statement.setString(i: 1, string: book.getTitle());
            statement.setString(i: 2, string: book.getAuthorName());
            statement.setInt(i: 3, i1: book.getId());
            statement.executeUpdate();

            System.out.println("Successfully updated the book with id = " + book.getId());
            System.out.println(x: "\n");
            statement.close();
        } catch (SQLException exc) {
            System.out.println("FAILED TO UPDATE BOOK DATA : " + exc.getMessage());
        }
    }

    @Override
    public void removeBook(int id) {
        try {

```



```
            String query = "DELETE FROM books WHERE id = ?";
            statement = (PreparedStatement) connection.prepareStatement(string: query);
            statement.setInt(i: 1, i1: id);
            statement.executeUpdate();

            System.out.println(x: "Successfully delete book!\n");
            statement.close();
        } catch (SQLException exc) {
            System.out.println("FAILED TO DELETE BOOK DATA : " + exc.getMessage());
        }
    }

    @Override
    public void updateBook(int id, Book book) {
        throw new UnsupportedOperationException(message:"Not supported yet."); // Generated from nbfs://nbhost/Syste
    }


}

```

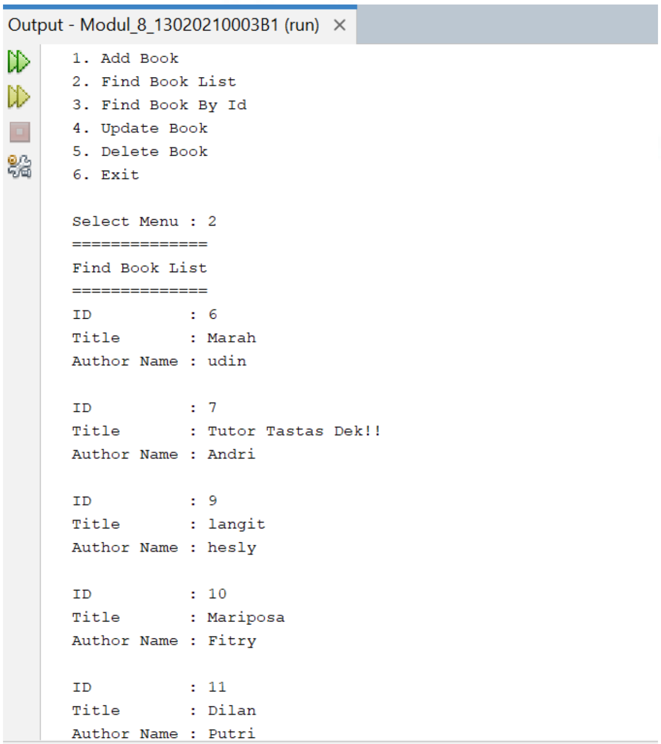
5. OUTPUT

Menu 1



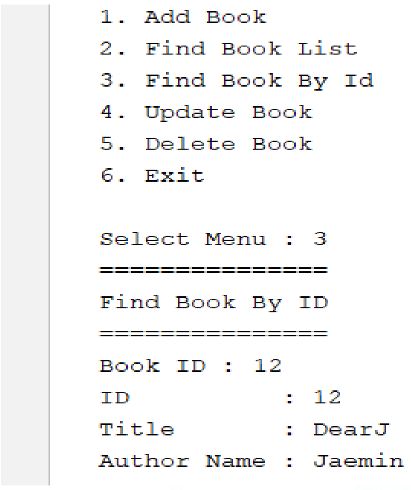
```
run:
Koneksi Database berhasil
Library Program
===============
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 1
========
Add Book
========
Book Title  : Harry Potter
Author Name : Nadia
Book has been added!
```

Menu 2



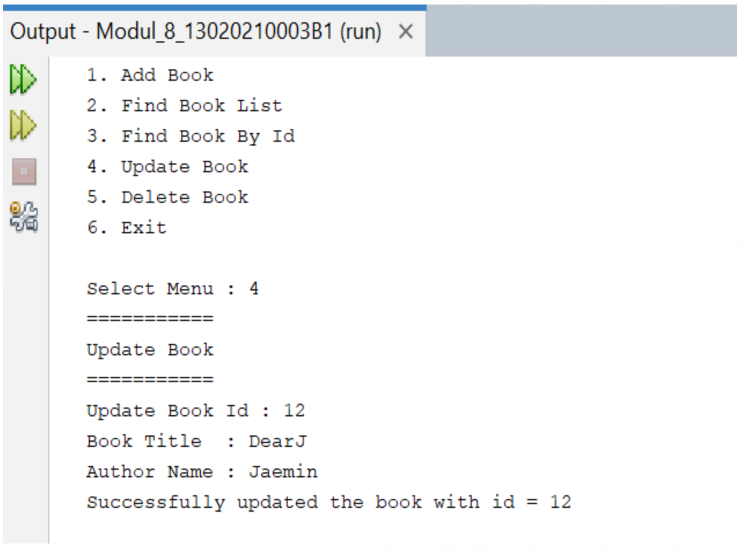
```
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 2
==============
Find Book List
==============
ID          : 6
Title       : Marah
Author Name : udin

ID          : 7
Title       : Tutor Tastas Dek!!
Author Name : Andri

ID          : 9
Title       : langit
Author Name : hesly

ID          : 10
Title       : Mariposa
Author Name : Fitry

ID          : 11
Title       : Dilan
Author Name : Putri
```

Menu 3



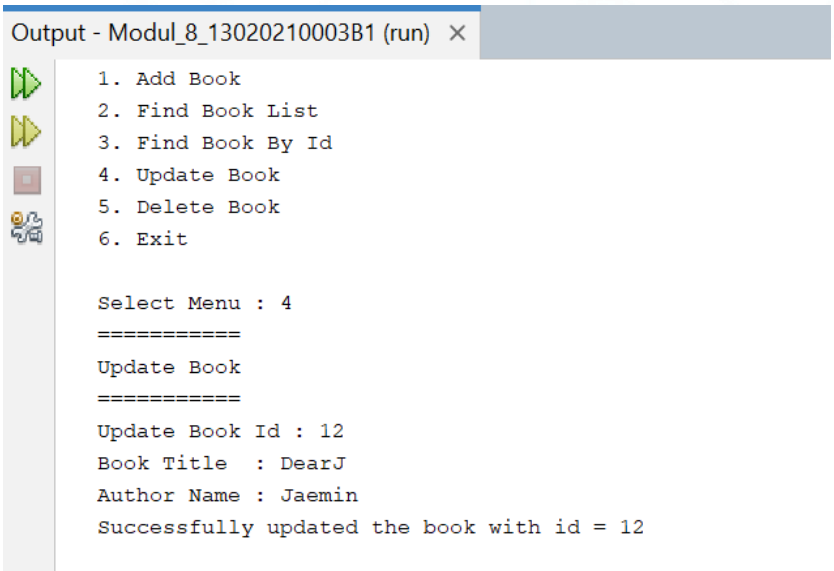
```
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 3
===============
Find Book By ID
===============
Book ID : 12
ID          : 12
Title       : DearJ
Author Name : Jaemin
```

Menu 4



```
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 4
===========
Update Book
===========
Update Book Id : 12
Book Title  : DearJ
Author Name : Jaemin
Successfully updated the book with id = 12
```

Menu 5



```
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 4
===========
Update Book
===========
Update Book Id : 12
Book Title  : DearJ
Author Name : Jaemin
Successfully updated the book with id = 12
```