

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Rima Ananda Nasution

Nim : 13020210238

Frekuensi : TI_PBO – 09

Dosen : Mardiyah Hasnawi, S.Kom, M.T., MTA

Asisten 1 : Intje Irfan Ibrahim

Asisten 2 : Muh. Acqmal Fadhilla Latief

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2023

-Package Config

Database :

```
package config;

import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.SQLException;

/**
 *
 * @author macbookair
 */
public class Database {
    private static Connection connection;

    public static Connection startConnection() {
        String url = "jdbc:mysql://localhost:8889/db_praktikum_7";
        String username = "root";
        String password = "root";

        try {
            DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());
            connection = DriverManager.getConnection(url, username, password);
            System.out.println("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();
    }
}
```

-Package Model

Book :

```
package model;

/**
 *
 * @author macbookair
 */
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;
    }
}
```

-Package Service

BookService :

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

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

/**
 *
 * @author macbookair
 */
public interface BookService {
    public void addBook(Book book);
    public List<Book> findBookList();
    public Book findBookById(int id);
    public void updateBook(Book book);
    public void removeBook(int id);
}
```

BookServiceImpl :

```
package service;

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

/**
 *
 * @author macbookair
 */
public class BookServiceImpl implements BookService{

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

    @Override
    public void addBook(Book book) {
        try {
            String query = "INSERT INTO book VALUES (0, ? , ?)";
            statement = (PreparedStatement) 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 book";
            statement = (PreparedStatement) connection.prepareStatement(string: query);

            ResultSet result = statement.executeQuery();
        }
    }
}
```

```

        while (result.next()) {
            Book book = new Book(title: "", authorName: "");
            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 bookId) {
    Book book = new Book();
    try {

        String query = "Select * From book Where id = ?";
        statement = (PreparedStatement) connection.prepareStatement(string: query);
        statement.setInt(i: 1, i1: bookId);

        ResultSet result = statement.executeQuery();
        if (result.next()) {
            int id = 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);
        }
        statement.close();
        return book;
    } catch (SQLException exc) {
        System.out.println("Failed To Get Book : " + exc.getMessage());
    }
    return book;
}

@Override
public void updateBook(Book book) {

    try {
        String query = "UPDATE book 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("Succesfully update book!\n" + book.getId());
        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 book Where id = ?";
        statement = (PreparedStatement) connection.prepareStatement(string: query);
        statement.setInt(i: 1, i1: id);
        statement.executeUpdate();

        System.out.println(x: "Succesfully delete book!\n");
        statement.close();
    } catch (SQLException exc) {
        System.out.println("Failed To Delete Book Data : " + exc.getMessage());
    }
}
}

```

-Package studi_kasus_8

Studi_kasus_8:

```
package studi_kasus_8;

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

/**
 *
 * @author macbookair
 */
public class Studi_kasus_8 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        BookService bookService = new BookServiceImpl();
        int menuInput = 0;

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

        do{

            Scanner scanner = new Scanner(System.in);
            System.out.println("1. Add Book");
            System.out.println("2. Find Book List");
            System.out.println("3. Find Book By Id");
            System.out.println("4. Update Book");
            System.out.println("5. Delete Book");
            System.out.println("6. Exit");

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

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

                    scanner.nextLine();

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

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

                    Book newBook = new Book(title, authorName);
                    bookService.addBook(book: newBook);
                    break;
                case 2:
                    System.out.println("=====");
                    System.out.println("Find Book List");
                    System.out.println("=====");
                    List<Book> books = bookService.findBookList();
                    if (books.size() < 1){
                        System.out.println("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("\n");
                        }
                    }
            }

        }
    }
}
```

```

        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 = scanner.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: "\n");
        }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.println(x: "Update book id : ");
        int bookIdUpdate = scanner.nextInt();

        scanner.nextLine();

        System.out.print(s: "\nBook Title : ");
        String newTitle = scanner.nextLine().substring(beginIndex: 0);
        System.out.print(s: "Author Name : ");
        String newAuthorName = scanner.nextLine();

        Book bookUpdate = new Book();
        bookUpdate.setId(id: bookIdUpdate);
        bookUpdate.setTitle(title: newTitle);
        bookUpdate.setAuthorName(authorName: newAuthorName);
        bookService.updateBook(book: bookUpdate);
        break;
    case 5:
        System.out.println(x: "=====");
        System.out.println(x: "Remove Book");
        System.out.println(x: "=====");

        System.out.print(s: "Book id : ");
        int bookIdRemove = scanner.nextInt();
        bookService.removeBook(id: bookIdRemove);
        break;
    case 6:
        System.out.println(x: "Program Finished!");
        Database.closeConnection();
        break;
    default:
        System.out.println(x: "Invalid Menu!");
        break;
    }
}while (menuInput != 6);
}
}

```

-Output :

run:

Koneksi 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 : Sangkuriang

Author Name : Joko

Book has been added!

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 : Malin Kundang

Author Name : Lala

Book has been added!

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 : 1

Title : Sangkuriang

Author Name : Joko

ID : 2

Title : Malin Kundang

Author Name : Lala

Select Menu : 3

=====

Find Book By Id

=====

Book id : 1

ID :1

Title :Sangkuriang

Author Name :Joko

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 :

1

Book Title : Timun Mas

Author Name : Lili

Succesfully update book!

1

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


```
Select Menu : 5
=====
Remove Book
=====
Book id : 2
Succesfully delete book!
```

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          : 1
Title       : Timun Mas
Author Name : Lili
```

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

```
Select Menu : 6
Program Finished!
BUILD SUCCESSFUL (total time: 32 minutes 34 seconds)
```