Tugas Kegiatan Praktikum VIII Makassar, 19 Mei 2023

PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

Nama : Selvi Hidayah Johan

Nim : 13020210019

Frekuensi : TI_PBO – 9

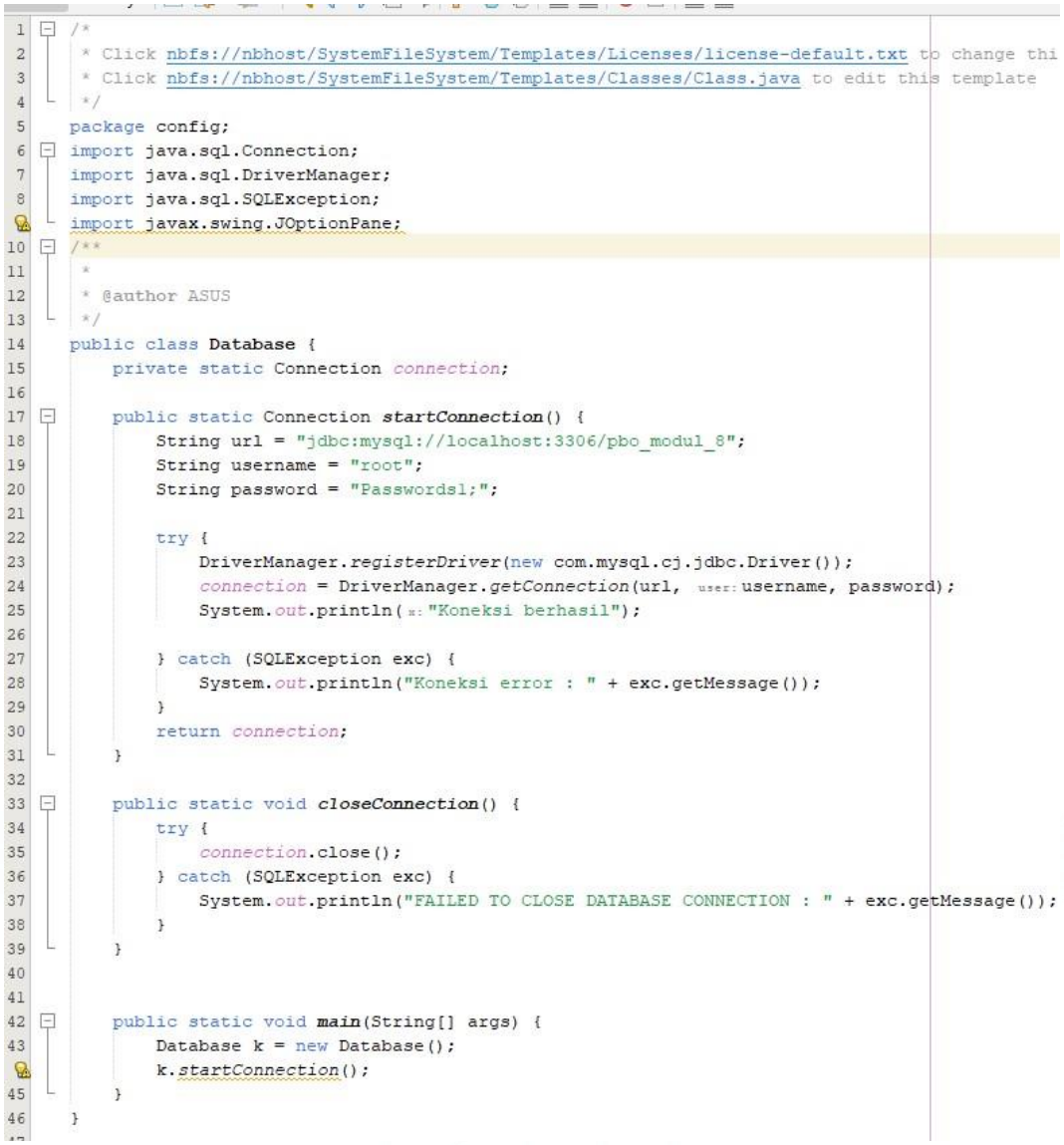
Dosen : Mardiyyah Hasnawi, S.Kom., M.T

Asisten 1 : Intje Irfan Ibrahim

Asisten 2 : Muh. Acqmal Fadhilla Latief

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change thi
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package config;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;
/**
 *
 * @author ASUS
 */
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();
    }
}
```

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/
 * Click nbfs://nbhost/SystemFileSystem/Templates/
 */
package book;

/**
 *
 * @author ASUS
 */
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;
    }
}
```

```java
package book;


import java.util.List;

/**
 *
 * @author ASUS
 */
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);
}
```

```java
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;

public class BookServiceImpl implements BookService {

    private final 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);
            }
```

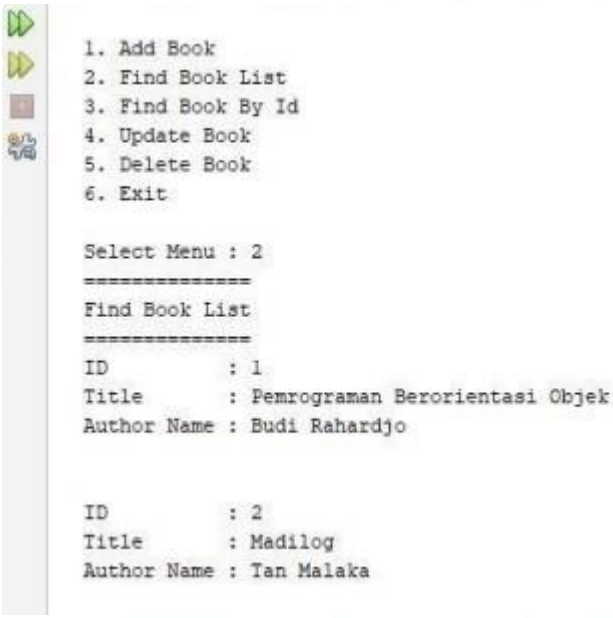
```
53              statement.close();
54              return books;
55          } catch (SQLException exc) {
56              System.out.println("FAILED TO GET BOOK LIST: " + exc.getMessage());
57          }
58          return books;
59      }
60
61      @Override
62      public Book findBookById(int id) {
63          Book book = new Book();
64          try {
65
66              String query = "SELECT * FROM books WHERE id = ?";
67              statement = (PreparedStatement) connection.prepareStatement( string: query);
68              statement.setInt( i: 1,  i1: id);
69
70              ResultSet result = statement.executeQuery();
71              if (result.next()) {
72                  String title = result.getString( string: "title");
73                  String authorName = result.getString( string: "author_name");
74
75                  book.setId(id);
76                  book.setTitle(title);
77                  book.setAuthorName(authorName);
78
79              } else {
80                  return null;
81              }
82              statement.close();
83              return book;
84          } catch (SQLException exc) {
85              System.out.println("FAILED TO GET BOOK : " + exc.getMessage());
86          }
87          return book;
88      }
89
90      @Override
91      public void updateBook(int id, Book book) {
92          try {
93
94              String query = "UPDATE books SET title = ?, author_name = ? WHERE id = ?";
95              statement = (PreparedStatement) connection.prepareStatement( string: query);
96              statement.setString( i: 1,  string: book.getTitle());
97              statement.setString( i: 2,  string: book.getAuthorName());
98              statement.setInt( i: 3,  i1: id);
99              statement.executeUpdate();
100
```

```
100                 System.out.println("Successfully updated the book with id = " + id);
101                 System.out.println( x: "\n");
102                 statement.close();
103             } catch (SQLException exc) {
104                 System.out.println("FAILED TO UPDATE BOOK DATA : " + exc.getMessage());
105             }
106         }
107
108         /**
109          *
110          * @param id
111          */
112         @Override
113         public void removeBook(int id) {
114             try {
115
116                 String query = "DELETE FROM books WHERE id = ?";
117                 statement = (PreparedStatement) connection.prepareStatement( string: query);
118                 statement.setInt( i: 1,  i1: id);
119                 statement.executeUpdate();
120
121                 System.out.println( x: "Successfully delete book!\n");
122                 statement.close();
123             } catch (SQLException exc) {
124                 System.out.println("FAILED TO DELETE BOOK DATA : " + exc.getMessage());
125             }
126         }
127     }
128 }
```

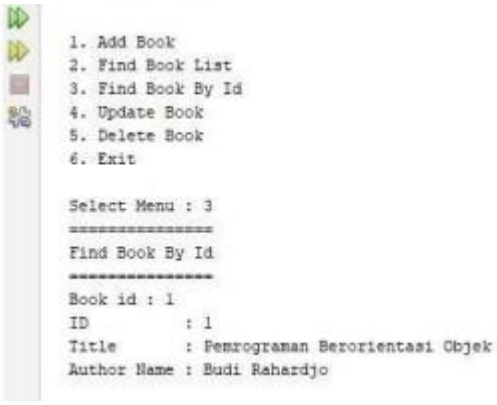
```
run:
Library Program
===============
1. Add Book
2. Find Book List
3. Find Book By Id
4. Update Book
5. Delete Book
6. Exit

Select Menu : 1
========
Add Book
========
Book Title : Pemrograman Berorientasi Objek
Author Name : Budi Rahardjo
Book has been added!

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

-outpu

```
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 : Madilog
Author Name : Tan Malaka
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       : Pemrograman Berorientasi Objek
Author Name : Budi Rahardjo


ID          : 2
Title       : Madilog
Author Name : Tan Malaka
```



```
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 : 1
ID          : 1
Title       : Pemrograman Berorientasi Objek
Author Name : Budi Rahardjo
```

```
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
Successfully 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       : Pemrograman Berorientasi Objek
Author Name : Budi Rahardjo

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: 2 minutes 18 seconds)
```