

**PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Putri Nadia Sudartomo

Nim : 13020210003

Frekuensi : TI\_PBO – 9

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

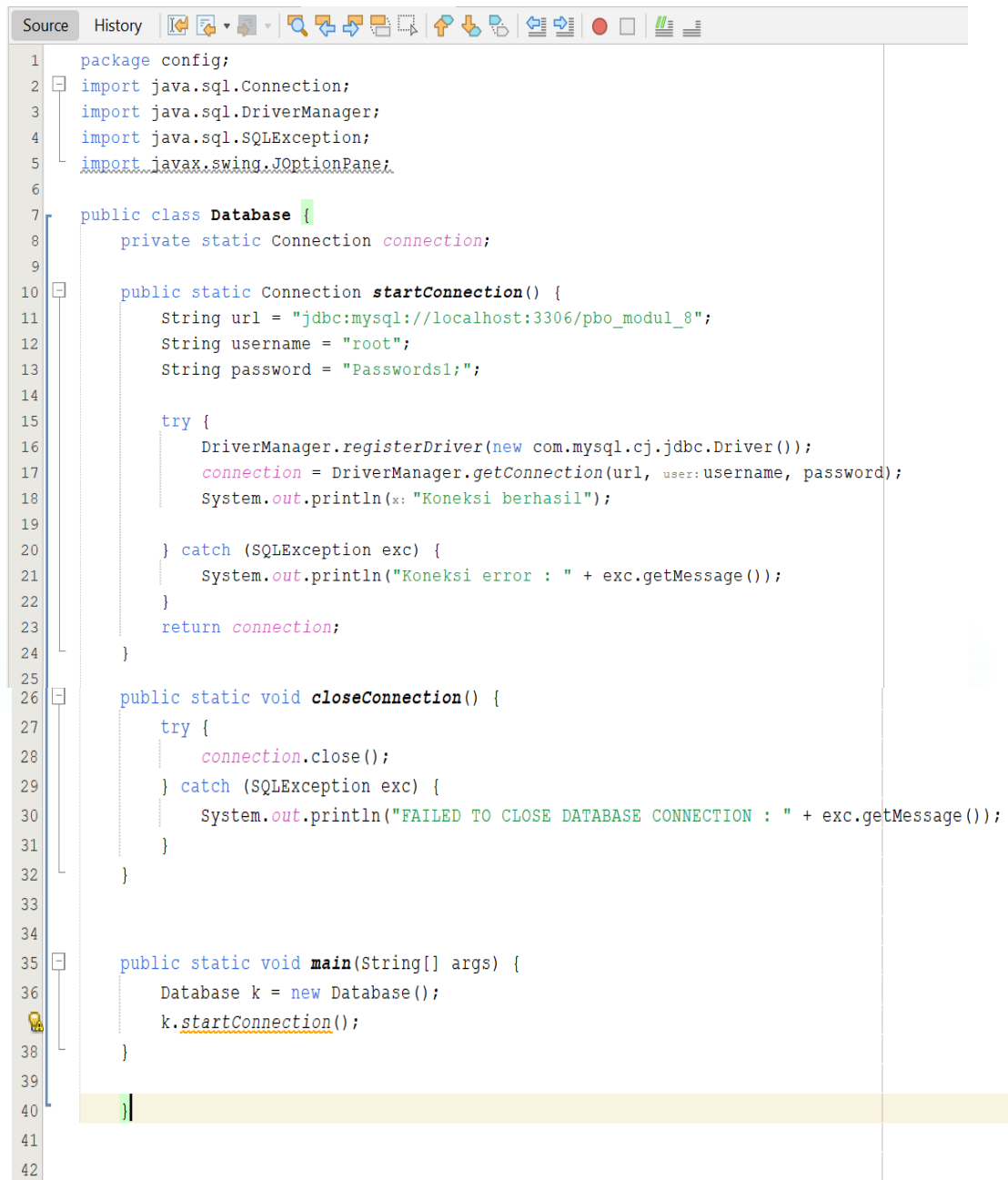
Asisten 1 : Intje Irfan Ibrahim

Asisten 2 : Muh. Acqmal Fadhilla Latief

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

## 1. Package Config

### ❖ Class DataBase



```
1 package config;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import javax.swing.JOptionPane;
6
7 public class Database {
8     private static Connection connection;
9
10    public static Connection startConnection() {
11        String url = "jdbc:mysql://localhost:3306/pbo_modul_8";
12        String username = "root";
13        String password = "Passwords1";
14
15        try {
16            DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());
17            connection = DriverManager.getConnection(url, user: username, password);
18            System.out.println("Koneksi berhasil");
19
20        } catch (SQLException exc) {
21            System.out.println("Koneksi error : " + exc.getMessage());
22        }
23        return connection;
24    }
25
26    public static void closeConnection() {
27        try {
28            connection.close();
29        } catch (SQLException exc) {
30            System.out.println("FAILED TO CLOSE DATABASE CONNECTION : " + exc.getMessage());
31        }
32    }
33
34
35    public static void main(String[] args) {
36        Database k = new Database();
37        k.startConnection();
38    }
39
40
41
42 }
```

## 2. Package Model

### ❖ Book

```
Source History 
1 package model;
2
3 import java.util.ArrayList;
4 import java.util.Collection;
5 import java.util.Set;
6
7 public class Book {
8
9     private int id;
10    private String authorName;
11    private String title;
12
13    public Book(String title, String authorName) {
14        this.title = title;
15        this.authorName = authorName;
16    }
17
18    public Book() {
19    }
20
21    public int getId() {
22        return id;
23    }
24
25    public void setId(int id) {
26        this.id = id;
27    }
28
29    public String getAuthorName() {
30        return authorName;
31    }
32
33    public void setAuthorName(String authorName) {
34        this.authorName = authorName;
35    }
36
37    public String getTitle() {
38        return title;
39    }
40
41    public void setTitle(String title) {
42        this.title = title;
43    }
44 }
```

### 3. Package modul\_8\_13020210003

#### ❖ Class main

```
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  import config.Database;
7  import java.util.List;
8  import java.util.Scanner;
9  import service.BookService;
10 import service.BookServiceImpl;
11 import model.Book;
12
13 public class Main {
14
15     public static void main(String[] args) {
16         BookService bookService = new BookServiceImpl();
17         int menuInput;
18
19         System.out.println(x: "Library Program");
20         System.out.println(x: "=====");
21
22         do {
23             Scanner input = new Scanner(source: System.in);
24             System.out.println(x: "1. Add Book");
25             System.out.println(x: "2. Find Book List");
26             System.out.println(x: "3. Find Book By Id");
27             System.out.println(x: "4. Update Book");
28             System.out.println(x: "6. Exit");
29
30             System.out.print(s: "\nSelect Menu : ");
31             menuInput = input.nextInt();
32
33             switch (menuInput) {
34                 case 1:
35                     System.out.println(x: "=====");
36                     System.out.println(x: "Add Book");
37                     System.out.println(x: "=====");
38
39                     input.nextLine();
40
41                     System.out.print(s: "Book Title : ");
42                     String title = input.nextLine();
43
44                     System.out.print(s: "Author Name : ");
45                     String authorName = input.nextLine();
46
47                     Book newBook = new Book(title, authorName);
48                     bookService.addBook(book: newBook);
49                     break;
50                 case 2:
51                     System.out.println(x: "=====");
52                     System.out.println(x: "Find Book List");
53                     System.out.println(x: "=====");
54
55                     List<Book> books = bookService.findBookList();
56
57             }
```


```

57         if (books.size() < 1) {
58             System.out.println(x: "No Books Yet\n");
59         } else {
60             for (Book book : books) {
61                 System.out.println("ID           : " + book.getId());
62                 System.out.println("Title        : " + book.getTitle());
63                 System.out.println("Author Name : " + book.getAuthorName());
64                 System.out.println(x: "");
65             }
66         }
67         break;
68     case 3:
69         System.out.println(x: "=====");
70         System.out.println(x: "Find Book By ID");
71         System.out.println(x: "=====");
72
73         System.out.print(s: "Book ID : ");
74         int bookId = input.nextInt();
75
76         Book book = bookService.findBookById(id: bookId);
77         if (book != null) {
78             System.out.println("ID           : " + book.getId());
79             System.out.println("Title        : " + book.getTitle());
80             System.out.println("Author Name : " + book.getAuthorName());
81
82             System.out.println(x: "");
83         } else {
84             System.out.println(x: "No Books Yet\n");
85         }
86         break;
87     case 4:
88         System.out.println(x: "=====");
89         System.out.println(x: "Update Book");
90         System.out.println(x: "=====");
91
92         System.out.print(s: "Update Book Id : ");
93         int bookIdUpdate = input.nextInt();
94
95         input.nextLine();
96
97         System.out.print(s: "Book Title : ");
98         String newTitle = input.nextLine();
99
100        System.out.print(s: "Author Name : ");
101        String newAuthorName = input.nextLine();
102
103        Book bookUpdate = new Book();
104        bookUpdate.setId(id: bookIdUpdate);
105        bookUpdate.setTitle(title: newTitle);
106        bookUpdate.setAuthorName(authorName: newAuthorName);
107        bookService.updateBook(bookUpdate);
108
109        break;
110    case 5:
111        System.out.println(x: "=====");
112        System.out.println(x: "Remove Book");
113    default:
114        System.out.println(x: "Invalid Menu!");
115        break;
116    }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }


```

## 4. Package Service

### ❖ BookService

```
Source History 
1 package service;
2
3 import java.util.List;
4 import model.Book;
5
6 public interface BookService {
7
8     public void addBook(Book book);
9
10    public List<Book> findBookList();
11
12    public Book findBookById(int id);
13
14    public void updateBook(int id, Book book);
15
16    public void removeBook(int id);
17
18    public void updateBook(Book bookUpdate);
19 }
20
```

### ❖ BookServiceImpl

```
Source History 
1 package service;
2
3 import config.Database;
4 import java.sql.Connection;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
9 import java.util.List;
10 import model.Book;
11
12 public class BookServiceImpl implements BookService {
13
14     private Connection connection = Database.startConnection();
15     private PreparedStatement statement;
16
17     @Override
18     public void addBook(Book book) {
19         try {
20             String query = "INSERT INTO books (title, author_name) VALUES (?, ?)";
21             statement = connection.prepareStatement(string: query);
22             statement.setString(i: 1, string: book.getTitle());
23             statement.setString(i: 2, string: book.getAuthorName());
24             statement.executeUpdate();
25
26             System.out.println("Book has been added!\n");
27             statement.close();
28         } catch (SQLException exc) {
```

```

29         System.out.println("FAILED TO ADD BOOK " + exc.getMessage());
30     }
31 }
32
33 @Override
34 public List<Book> findBookList() {
35     List<Book> books = new ArrayList<>();
36
37     try {
38
39         String query = "SELECT * FROM books";
40         statement = (PreparedStatement) connection.prepareStatement(string: query);
41         ResultSet result = statement.executeQuery();
42         while (result.next()) {
43             Book book = new Book();
44             book.setId(id: result.getInt(string: "id"));
45             book.setTitle(title: result.getString(string: "title"));
46             book.setAuthorName(authorName: result.getString(string: "author_name"));
47
48             books.add(e: book);
49         }
50         statement.close();
51         return books;
52     } catch (SQLException exc) {
53         System.out.println("FAILED TO GET BOOK LIST: " + exc.getMessage());
54     }
55     return books;
56 }
57
58 @Override
59 public Book findBookById(int id) {
60     Book book = new Book();
61     try {
62
63         String query = "SELECT * FROM books WHERE id = ?";
64         statement = (PreparedStatement) connection.prepareStatement(string: query);
65         statement.setInt(i: 1, i1: id);
66
67         ResultSet result = statement.executeQuery();
68         if (result.next()) {
69             int bookId = result.getInt(string: "id");
70             String title = result.getString(string: "title");
71             String authorName = result.getString(string: "author_name");
72
73             book.setId(id);
74             book.setTitle(title);
75             book.setAuthorName(authorName);
76
77         } else {
78             return null;
79         }
80         statement.close();
81         return book;
82     } catch (SQLException exc) {
83         System.out.println("FAILED TO GET BOOK : " + exc.getMessage());

```

```

83         System.out.println("FAILED TO GET BOOK : " + exc.getMessage());
84     }
85     return book;
86 }
87
88 @Override
89 public void updateBook(Book book) {
90     try {
91
92         String query = "UPDATE books SET title = ?, author_name = ? WHERE id = ?";
93         statement = (PreparedStatement) connection.prepareStatement(string: query);
94         statement.setString(i: 1, string: book.getTitle());
95         statement.setString(i: 2, string: book.getAuthorName());
96         statement.setInt(i: 3, i1: book.getId());
97         statement.executeUpdate();
98
99         System.out.println("Successfully updated the book with id = " + book.getId());
100         System.out.println(x: "\n");
101         statement.close();
102     } catch (SQLException exc) {
103         System.out.println("FAILED TO UPDATE BOOK DATA : " + exc.getMessage());
104     }
105 }
106
107 @Override
108 public void removeBook(int id) {
109     try {
110
111         String query = "DELETE FROM books WHERE id = ?";
112         statement = (PreparedStatement) connection.prepareStatement(string: query);
113         statement.setInt(i: 1, i1: id);
114         statement.executeUpdate();
115
116         System.out.println(x: "Successfully delete book!\n");
117         statement.close();
118     } catch (SQLException exc) {
119         System.out.println("FAILED TO DELETE BOOK DATA : " + exc.getMessage());
120     }
121 }
122
123 @Override
124 public void updateBook(int id, Book book) {
125     throw new UnsupportedOperationException(message: "Not supported yet."); // Generated from nbfs://nbhost/Syste
126 }
127
128 }
129
130

```





## 5. OUTPUT

### Menu 1

```
Output - Modul_8_13020210003B1 (run) ×  
  
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

```
Output - Modul_8_13020210003B1 (run) ×  
  
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

Output - Modul\_8\_13020210003B1 (run) ×

```
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

Output - Modul\_8\_13020210003B1 (run) ×

```
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
```