

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Putri Nadia Sudartomo

Nim 13020210003

Frekuensi : TI_PBO – 9

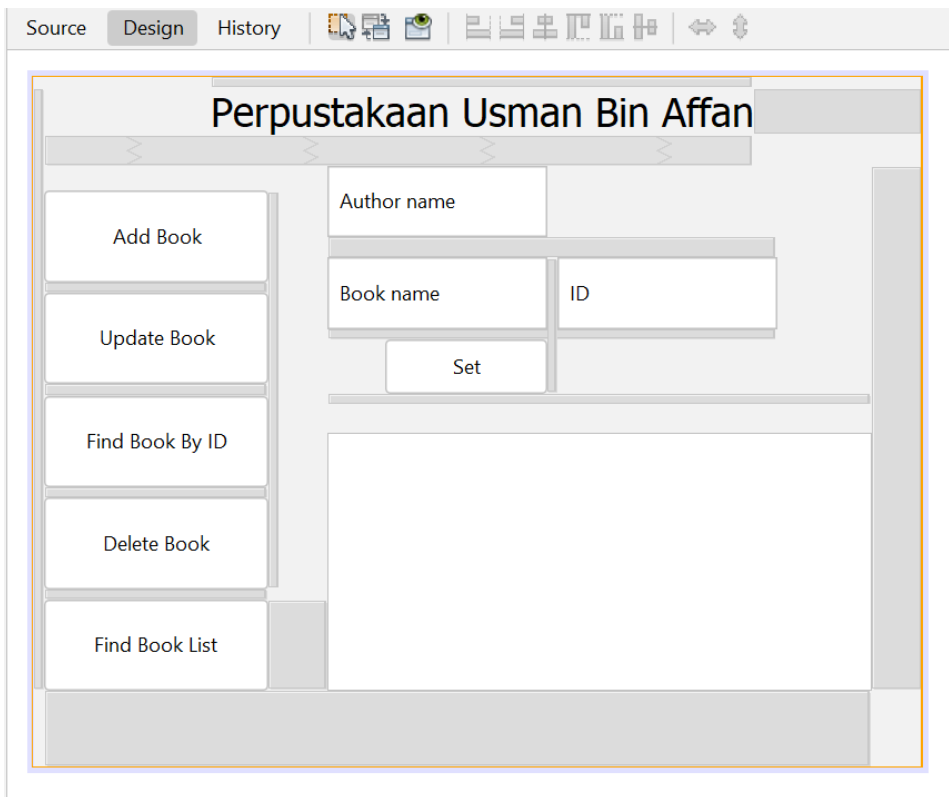
Dosen : Mardiyah Hasnawi, S.Kom., M.T

Asisten 1 : Intje Irfan Ibrahim

Asisten 2 : Muh. Acqmal Fadhilla Latief

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

Buat UI sesuai dengan gambar di bawah ini



❖ Kodingannya

```
Source Design History
1
2 import java.util.List;
3 import javax.swing.DefaultListModel;
4 import model.Book;
5 import service.BookServiceImpl;
6 import service.BookService;
7
8 public class BookApp extends javax.swing.JFrame {
9
10     private final BookService bookService = new BookServiceImpl();
11     private String menu = "";
12
13     private void changeAllVisible(boolean value) {
14         fieldBookName.setVisible(aFlag: value);
15         fieldAuthorName.setVisible(aFlag: value);
16         fieldId.setVisible(aFlag: value);
17         btnSet.setVisible(aFlag: value);
18     }
19
20     private void visibleAdd() {
21         changeAllVisible(value: false);
22         fieldBookName.setVisible(aFlag: true);
23         fieldAuthorName.setVisible(aFlag: true);
24         btnSet.setVisible(aFlag: true);
25     }
26
27     private void visibleUpdate() {
28         changeAllVisible(value: false);
```

```
Source Design History 
29     fieldBookName.setVisible(aFlag: true);
30     fieldAuthorName.setVisible(aFlag: true);
31     fieldId.setVisible(aFlag: true);
32     btnSet.setVisible(aFlag: true);
33 }
34
35 void visibleFindById() {
36     changeAllVisible(value: false);
37     fieldId.setVisible(aFlag: true);
38     btnSet.setVisible(aFlag: true);
39 }
40
41 void changeMenu(String menu) {
42     this.menu = menu;
43 }
44
45 void addBook() {
46     Book book = new Book(title: fieldBookName.getText(), authorName: fieldAuthorName.getText());
47     bookService.addBook(book);
48 }
49
50 void findListBook() {
51
52     DefaultListModel<String> model = new DefaultListModel<>();
53     List<Book> bookList = bookService.findBookList();
54
55     for (int i = 0; i < bookList.size(); i++) {
56         String outputText = "";
57         outputText += bookList.get(index: i).getId() + ". ";
58         outputText += "Nama Buku: " + bookList.get(index: i).getTitle() + ", ";
59         outputText += "Nama Author: " + bookList.get(index: i).getAuthorName();
60         model.addElement(element: outputText);
61     }
62
63     outputList.setModel(model);
64 }
65
66 void findListBookById(int i) {
67
68     DefaultListModel<String> model = new DefaultListModel<>();
69     Book book = bookService.findBookById(id: i);
70
71     String outputText = "";
72     outputText += (i) + ". ";
73     outputText += "Nama Buku: " + book.getTitle() + ", ";
74     outputText += "Nama Author: " + book.getAuthorName();
75     model.addElement(element: outputText);
76
77     outputList.setModel(model);
78 }
79
80 void deleteBook(int i) {
81     bookService.removeBook(id: i);
82 }
83
84 void updateBook(int i) {
85     Book book = new Book(title: fieldBookName.getText(), authorName: fieldAuthorName.getText());
86     bookService.updateBook(id: i, book);
87 }
88
89 public BookApp() {
90     initComponents();
91     changeAllVisible(value: false);
92 }
93
94 /**
95  * This method is called from within the constructor to initialize the form.
96  * WARNING: Do NOT modify this code. The content of this method is always
97  * regenerated by the Form Editor.
98  */
99 @SuppressWarnings("unchecked")
100 Generated Code
101
102 private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
103
104     visibleAdd();
105     changeMenu(menu: "add");
106 }
107
108 private void btnFindBookActionPerformed(java.awt.event.ActionEvent evt) {
109     changeAllVisible(value: false);
110     outputList.setVisible(aFlag: true);
111     findListBook();
112 }
```

```

287     }
288
289     private void btnFindByIdActionPerformed(java.awt.event.ActionEvent evt) {
290
291         visibleFindById();
292         changeMenu(menu: "findById");
293         findListBook();
294     }
295
296     private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
297
298         visibleUpdate();
299         changeMenu(menu: "update");
300     }
301
302     private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
303
304         visibleFindById();
305         changeMenu(menu: "delete");
306     }
307
308     private void fieldBookNameActionPerformed(java.awt.event.ActionEvent evt) {
309
310         // TODO add your handling code here:
311     }
312
313     private void fieldAuthorNameActionPerformed(java.awt.event.ActionEvent evt) {
314
315         // TODO add your handling code here:
316     }
317
318     private void btnSetActionPerformed(java.awt.event.ActionEvent evt) {
319
320         switch (menu) {
321             case "add":
322                 addBook();
323                 changeAllVisible(value: false);
324                 findListBook();
325                 break;
326             case "delete":
327                 deleteBook(i: Integer.parseInt(s: fieldId.getText()));
328                 changeAllVisible(value: false);
329                 findListBook();
330                 break;
331             case "update":
332                 updateBook(i: Integer.parseInt(s: fieldId.getText()));
333                 changeAllVisible(value: false);
334                 findListBookById(i: Integer.parseInt(s: fieldId.getText()));
335                 break;
336             case "findById":
337                 findListBookById(i: Integer.parseInt(s: fieldId.getText()));
338                 changeAllVisible(value: false);
339                 break;
340         }
341     }
342
343     private void fieldIdActionPerformed(java.awt.event.ActionEvent evt) {
344
345         // TODO add your handling code here:
346     }
347
348     /**
349      * @param args the command line arguments
350      */
351     public static void main(String args[]) {
352
353         /* Set the Nimbus look and feel */
354         Look and feel setting code (optional)
355         //</editor-fold>
356
357         /* Create and display the form */
358         java.awt.EventQueue.invokeLater(new Runnable() {
359             public void run() {

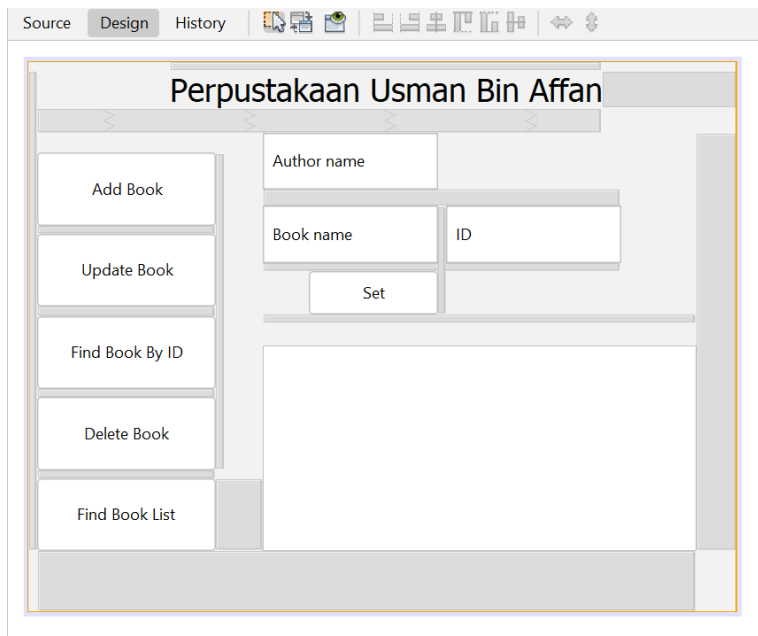
```

```

377         new BookApp().setVisible(b: true);
378     }
379     });
380 }
381
382 // Variables declaration - do not modify
383 private javax.swing.JButton btnAdd;
384 private javax.swing.JButton btnDelete;
385 private javax.swing.JButton btnFindBook;
386 private javax.swing.JButton btnFindById;
387 private javax.swing.JButton btnSet;
388 private javax.swing.JButton btnUpdate;
389 private javax.swing.JTextField fieldAuthorName;
390 private javax.swing.JTextField fieldBookName;
391 private javax.swing.JTextField fieldId;
392 private javax.swing.JLabel jLabel1;
393 private javax.swing.JScrollPane jScrollPane1;
394 private javax.swing.JList<String> outputList;
395 private javax.swing.JPanel panelOutput;
396 // End of variables declaration
397 }
398

```

Desain



Output

