

**LAPORAN TUGAS PRAKTIUM  
PEMOGRAMAN BERORIENTASI OBJEK  
MEMBUAT APLIKASI PERPUSTAKAAN**



**Dosen Pembimbing :  
SLAMET TRIYANTO S.ST**

**Disusun Oleh :  
IWAN(202013033)**

**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK KAMPAR  
2021**

## **KATA PENGANTAR**

Puji dan syukur atas kehadiran Allah SWT, atas segala limpahan taufiq serta hidayah-Nya yang telah memberi penulis kesempatan untuk menyelesaikan Laporan Pratikum Tugas Akhir Pemograman Berorientasi Objek. Adapun tujuan penulisan Laporan ini adalah untuk melengkapi Tugas Perkuliahan Pemograman Berorientasi Obek

Dalam proses pembuatan Laporan ini, tentunya penulis mendapat bimbingan, arahan, koreksi dan saran. Untuk itu penulis mengucapkan terima kasih kepada Bapak Slamet Triyanto S.ST Selaku dosen pengampu mata kuliah Pemograman Berorientasi Objek.

Penulis menyadari bahwa baik dari segi penulisan maupun isi, laporan ini masih memiliki kekurangan, oleh karena itu penulis sangat mengharapkan kritik yang membangun dan saran dari pembaca agar terbentuknya kesempurnaan laporan ini. Atas partisipasinya penulis mengucapkan terima kasih.

Bangkinang, 05 agustus 2021

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	i
Daftar isi .....	ii
DAFTAR GAMBAR .....	iii
BAB I .....	1
PENDAHULUAN .....	1
A. Pengertian .....	1
a. Pengertian Pemograman Berorientasi Objek .....	1
b. Konsep Dasar PBO .....	1
B. Tujuan Praktikum .....	3
C. Alat Dan Bahan .....	3
BAB II .....	4
PEMBAHASAN .....	4
A. Rancangan Awal Aplikasi Perpustakaan .....	4
B. Flowchart .....	4
C. Tampilan Aplikasi .....	5
D. Penjelasan Source code Aplikasi perpustakaan .....	7
BAB III .....	16
PENUTUP .....	16
A. Kesimpulan .....	16
B. Saran .....	16
DAFTAR PUSTAKA .....	17

## DAFTAR GAMBAR

Gambar 2. 1 <i>Splash Screen</i> .....	5
Gambar 2. 2 jendela Login.....	6
Gambar 2. 3 jendela aplikasi perpustakaan.....	6
Gambar 2. 4 tampilan <i>game</i> tictac toe.....	7
Gambar 2. 5 source library pada <i>splash Screen</i> .....	7
Gambar 2. 6 source code <i>splashscreen</i> .....	8
Gambar 2. 7 import library jendela login.....	8
Gambar 2. 8 sourcode untuk memasukan data login .....	9
Gambar 2. 9 tampilan source code untuk menyimpan data login .....	9
Gambar 2. 10 library untuk perpustakaan .....	10
Gambar 2. 11 source code pada class tampilkan data.....	10
Gambar 2. 12 tampilan source code untuk menampilkan data ke dalam tabel	10
Gambar 2. 13 source code pada button save .....	11
Gambar 2. 14 source code pada button edit .....	11
Gambar 2. 15 source code pada button delete .....	11
Gambar 2. 16 source code pada tombol search.....	12
Gambar 2. 17 import library pada jendela game sederhana.....	12
Gambar 2. 18 source code kelas game score dan chose.....	13
Gambar 2. 19 source code pada kelas wining game .....	13
Gambar 2. 20 source code pada penyeleksian kondisi X.....	14
Gambar 2. 21 source code pada penyelksian konidi O .....	14
Gambar 2. 22 source code pada tombol reset .....	15
Gambar 2. 23 source code pada setiap tombol.....	15

# **BAB I**

## **PENDAHULUAN**

### **A. Pengertian**

#### **a. Pengertian Pemrograman Berorientasi Objek**

Pemrograman berorientasi objek (Inggris: object-oriented programming disingkat OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

#### **b. Konsep Dasar PBO**

1. Kelas — kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana

kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstraksi - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4. Enkapsulasi - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah

variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

6. Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

## **B. Tujuan Praktikum**

Dapat mengenal dan memahami konsep constructor dan overloading constructor. Ketika anda membuat sebuah objek dari class manusia (sebut saja “Orang 1”), kemudian anda dapat menge-set nilai berupa nama, alamat, tanggal lahir, dan hobi menggunakan method setter(). Berbeda halnya dengan constructor. Ketika objek “Orang 1” telah terbentuk, anda langsung memberikan nilai berupa nama, alamat, tanggal lahir, dan hobi. Hal itu ibarat anda baru lahir di dunia ini dan langsung diberi nama. Itulah merupakan konsep dari Constructor.

## **C. Alat Dan Bahan**

Laptop dan Instalasi aplikasi Netbens

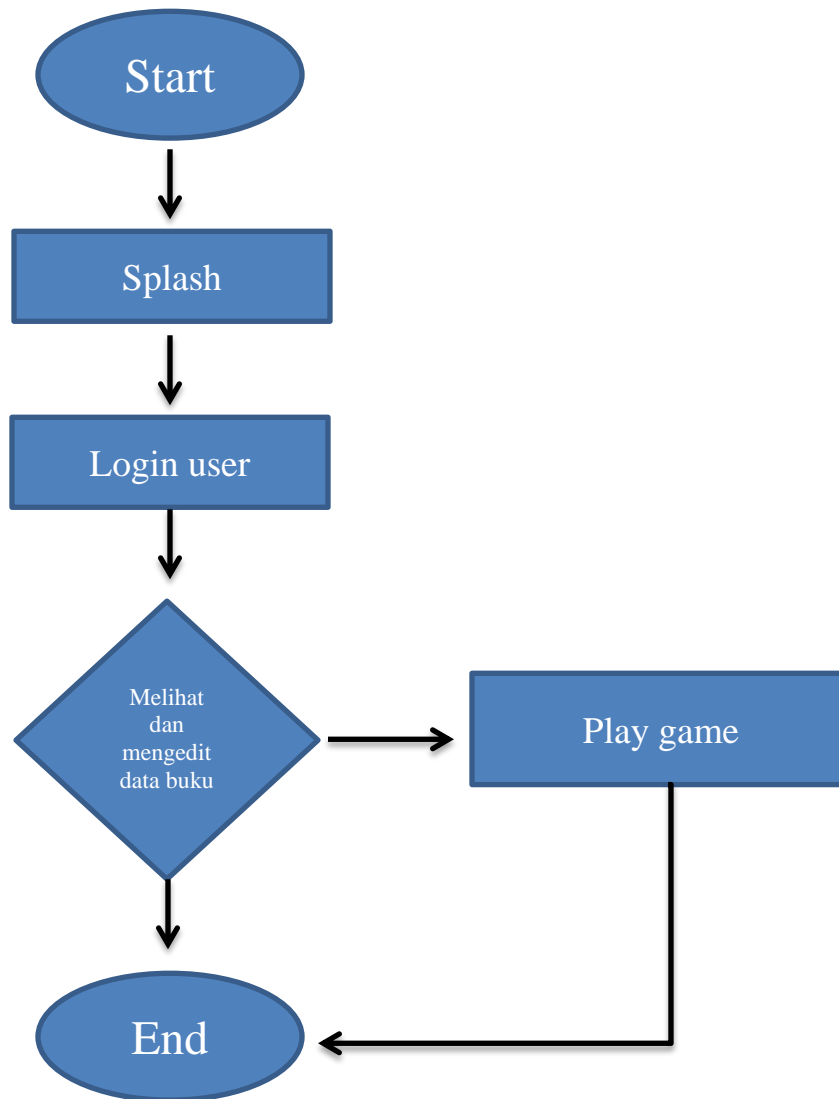
## BAB II

### PEMBAHASAN

#### A. Rancangan Awal Aplikasi Perpustakaan

Aplikasi ini digunakan untuk mengetahui tentang bagaimana menyimpan data buku, melihat data buku sekaligus menghapus data buku yang di simpan.

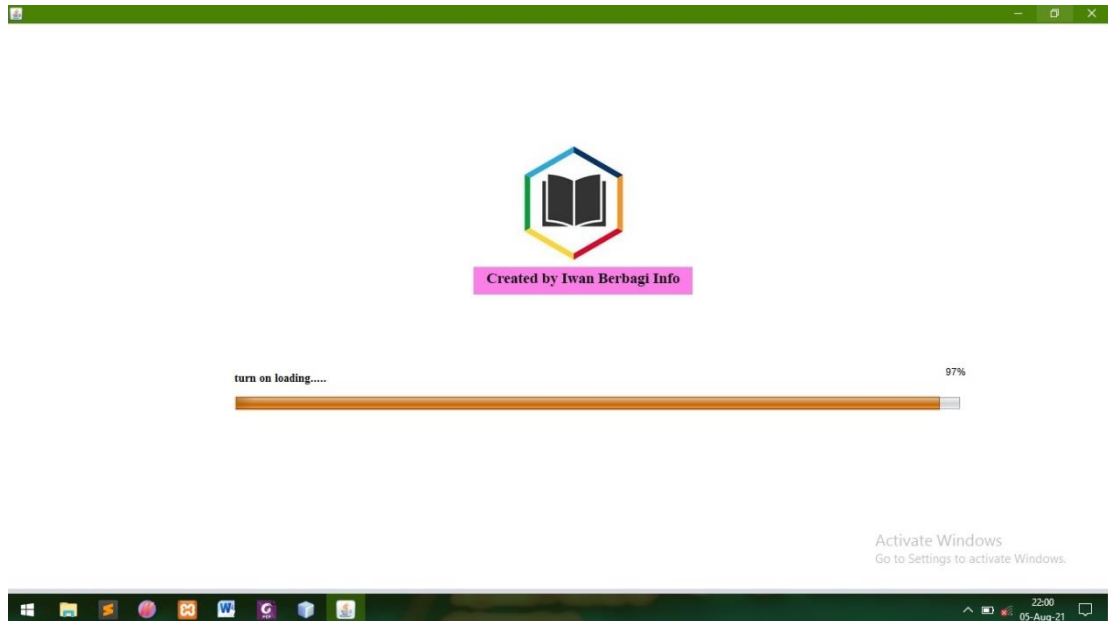
#### B. Flowchart





## C. Tampilan Aplikasi

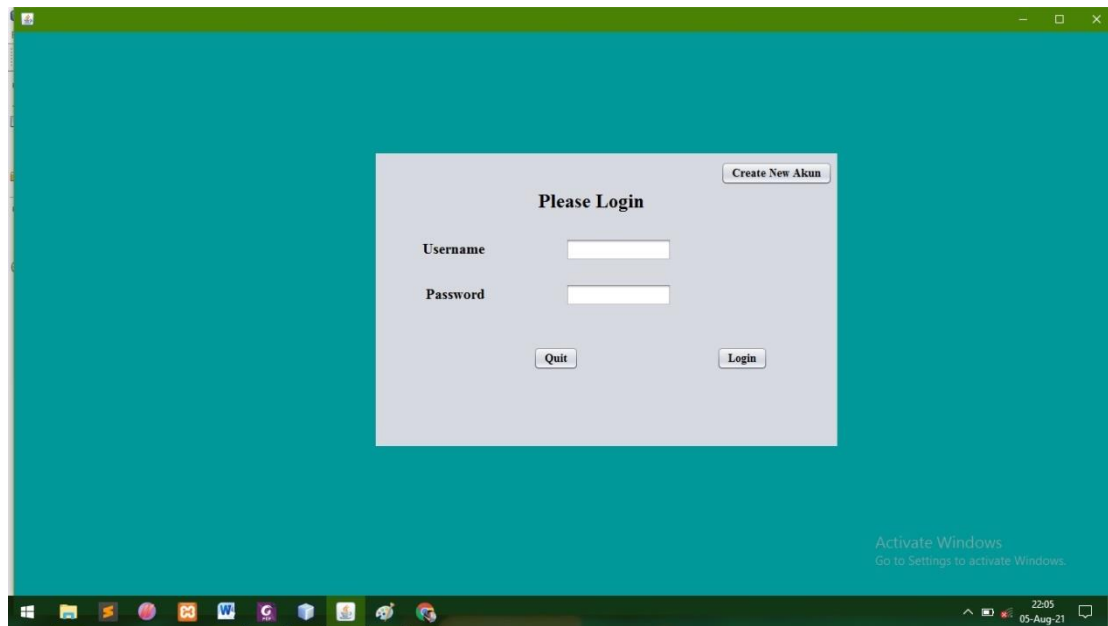
### 1. *Splash Screen*



Gambar 2. 1 *Splash Screen*

Gambar diatas merupakan tampilan Splash Screen dari aplikasi Perpustakaan. Splash Screen ini berdurasi 3 detik, dengan tampilan yang muncul ditengah-tengah layar.

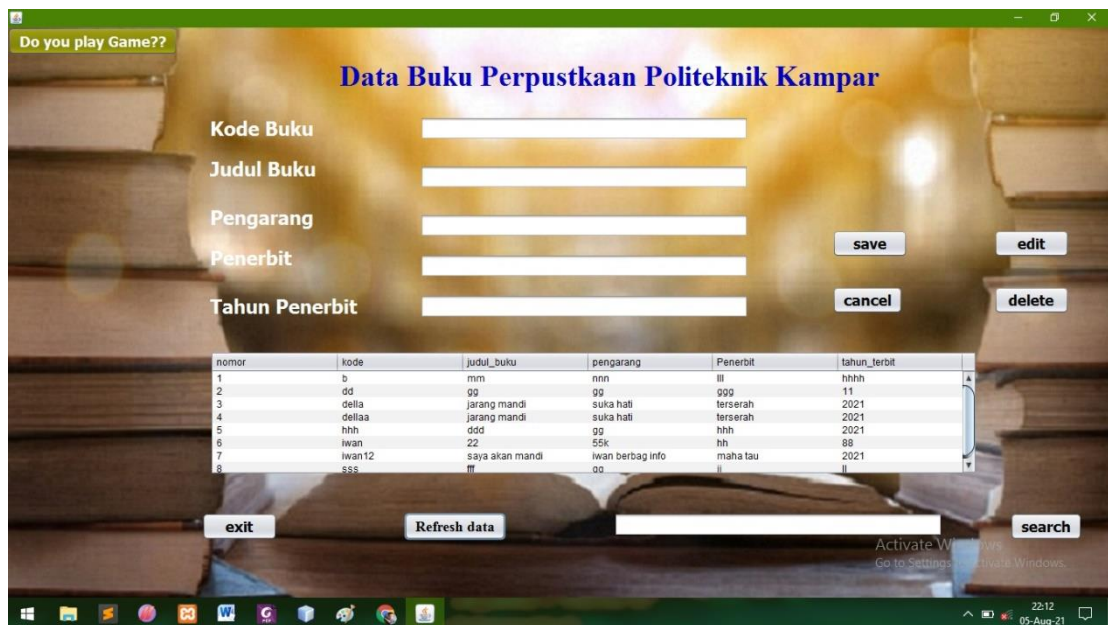
### 2. Jendela Login



Gambar 2. 2 jendela Login

Gambar diatas menampilkan Jendela Masuk dari aplikasi Perpustakaan. Tampilan ini akan muncul setelah Splash Screen selesai.

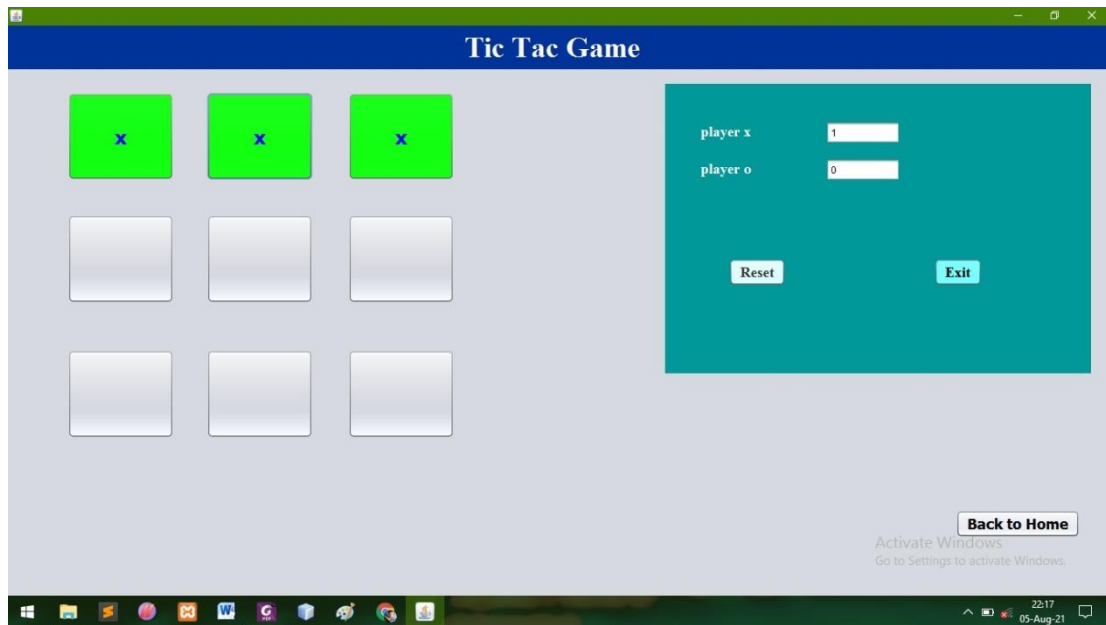
### 3. Jendela Tampilan aplikasi perpustakaan



Gambar 2. 3 jendela aplikasi perpustakaan

Design diatas menampilkan jendela aplikasi perpustakaan untuk melihat ,menginputkan data buku, mengedit dan menghapus data buku dan di pojok kiri atas saya menambahkan menu game sederhana

#### 4. Jendela tampilan game sederhana



Gambar 2. 4 tampilan *game* tictac toe

Aplikasi di atas merupakan tampilan game sederhana untuk percobaan dalam membuat aplikasi lebih lanjut yang lebih keren

#### D. Penjelasan Source code Aplikasi perpustakaan

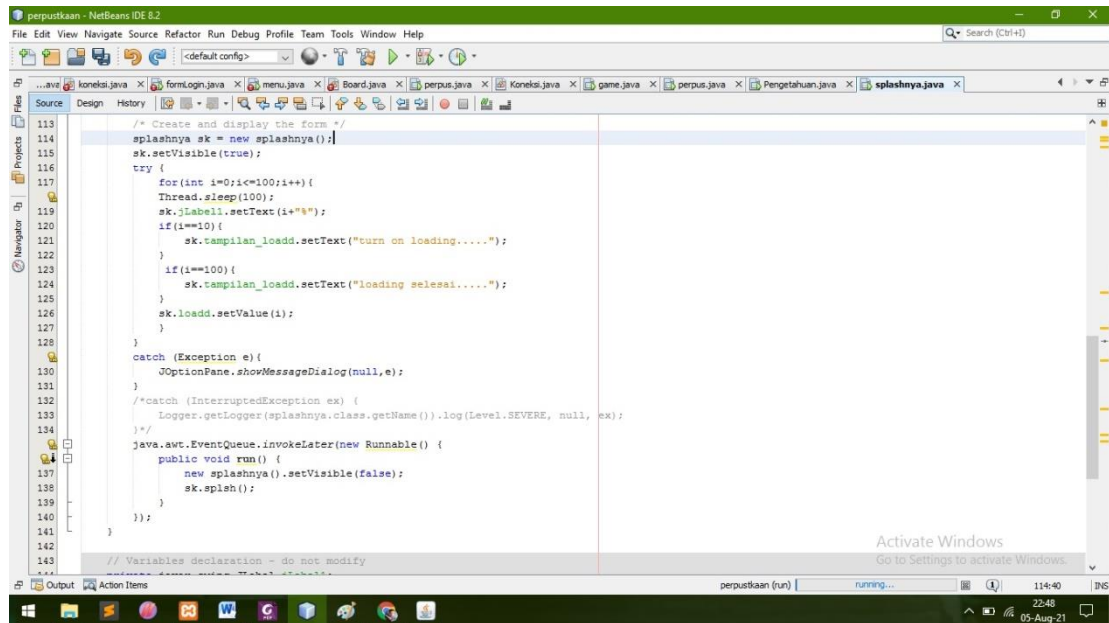
Dalam pembuatan sebuah aplikasi kita wajib untuk memasukan source code atau bisa di sebut sebagai perintah untuk menjalankan logika manusia. Source code yang saya gunakan kali ini menggunakan bahasa java.

##### a) *Splash Screen*

```
import java.util.logging.Level; //untuk mengontrol output logging logger
import java.util.logging.Logger; //untuk membuat type data BigDecimal
import javax.swing.JFrame; //untuk memanggil library swing
import javax.swing.JOptionPane; //untuk mengambil dan menampilkan informasi
```

Gambar 2. 5 source library pada *splash Screen*

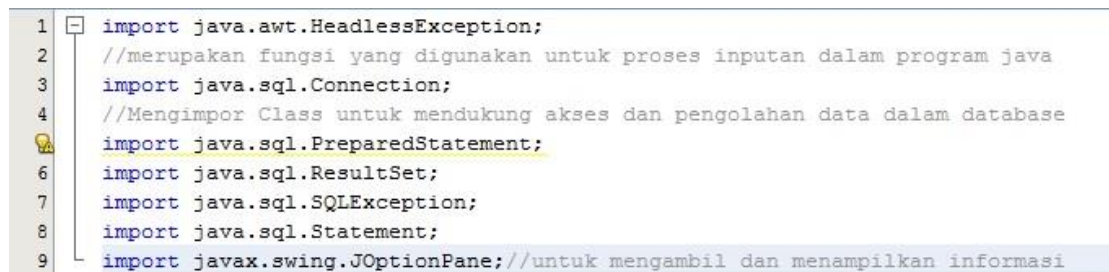
Tampilan di atas adalah source code library untuk memanggil perintah yang kita butuhkan dalam sebuah program



Gambar 2. 6 source code splashscreen

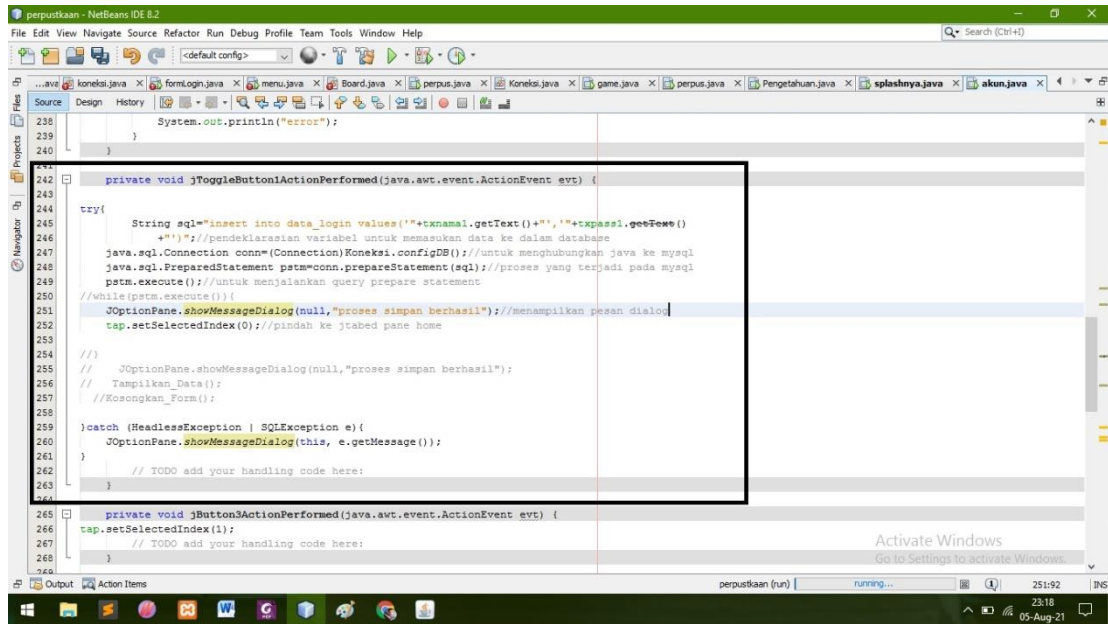
Tampilan di atas terdapat perintah untuk memanggil splash screen dan melanjutkan ke frame selanjutnya

#### b) Jendela Login

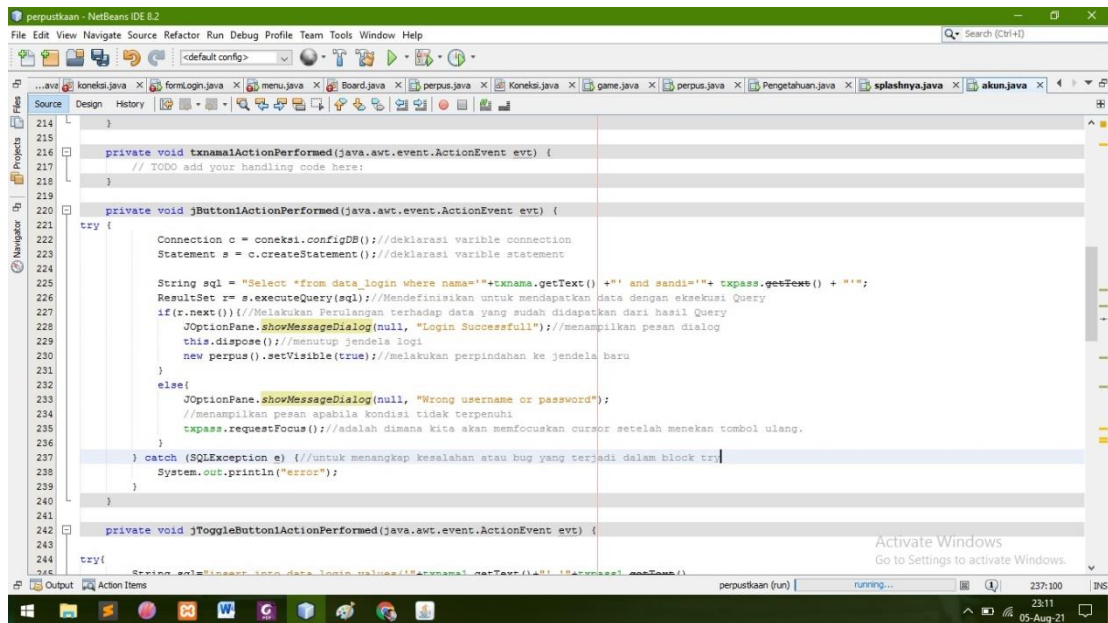


Gambar 2. 7 import library jendela login

Tampilan di atas adalah library yang ada pada menu login dimana untuk memanggil data base maka kita harus mengimpor library untuk memanggilnya



Gambar 2. 8 sourcode untuk memasukan data login



Gambar 2. 9 tampilan source code untuk menyimpan data login

c) Jendela buku perpustakaan

```

1 import java.awt.Image;
2 import java.awt.Toolkit;
3 import java.awt.HeadlessException;
4 import java.sql.Connection;
5 import java.sql.SQLException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8 import javax.swing.JOptionPane;
9 import javax.swing.table.DefaultTableModel;

```

Gambar 2. 10 library untuk perpustakaan

Tampilan di atas merupakan library yang ada pada jendela buku perpustakaan yang mana semua library tersebut di perlukan dalam pembuatan jendela buku perpustakaan

```

31 public void Tampilkan_Data(){
32     DefaultTableModel model=new DefaultTableModel();
33     model.addColumn("nomor");
34     model.addColumn("kode");
35     model.addColumn("judul_buku");
36     model.addColumn("pengarang");
37     model.addColumn("penerbit");
38     model.addColumn("tahun_terbit");
39     try{
40         int no=1;
41
42         String sql="select*from data_buku where kode like '%"
43             +txtcari.getText()+"%' or judul_buku like '%"
44             +txtcari.getText()+"%' or pengarang like '%" +txtcari.getText()+"%' or penerbit like '%" +txtcari.getText()+"%' or
45             +txtcari.getText()+"%' or tahun_terbit like '%" +txtcari.getText()+"%'";
46
47         java.sql.Connection conn=(Connection)Koneksi.configDB();
48         java.sql.Statement stm = conn.createStatement();
49         java.sql.ResultSet res=stm.executeQuery(sql);
50
51         while (res.next()){
52             //DefaultTableModel model=new DefaultTableModel();
53             model.addRow(new Object[]{no++, res.getString(1), res.getString(2), res.getString(3), res.getString(4),res.getString(5)});
54             tabperpustakaan.setModel(model);
55         }
56         // tabperpustakaan.setModel(model);
57
58     }catch (SQLException e){
59         System.out.println("error: " + e.getMessage());
60     }
61 }

```

Gambar 2. 11 source code pada class tampilkan data

Gambar 2. 12 tampilan source code untuk menampilkan data ke dalam tabel

Pada baris ke 32 berfungsi untuk membuat variabel baru untuk pemanggilan tabel

Pada baris ke 33 sampai 8 berfungsi untuk menampilkan isi yang kita inputkan ke dalam tabel yang telah kita buat

Pada baris ke 42 sampai k3 53 merupakan proses untuk memasukan dan menampilkan data ke dalam jTable



```

262 private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
263     try{
264         String sql="insert into data_buku values('"+txtkode.getText()+"','"+txtjudul.getText()
265             + "','"+ txtpengarang.getText()+"','"+
266             +txtpenerbit.getText()+"','"+txttahun.getText()+"')";
267         java.sql.Connection conn=(Connection)Koneksi.configDB();
268         java.sql.PreparedStatement pstmt=conn.prepareStatement(sql);
269         pstmt.execute();
270         //while(pstmt.execute()){
271             JOptionPane.showMessageDialog(null,"proses simpan berhasil");
272             Tampilkan_Data();
273         //}
274         // JOptionPane.showMessageDialog(null,"proses simpan berhasil");
275         // Tampilkan_Data();
276         //Kosongkan_Form();
277     }catch (HeadlessException | SQLException e){
278         JOptionPane.showMessageDialog(this, e.getMessage());
279     }
280     // TODO add your handling code here:
281 }
282

```

Gambar 2. 13 source code pada button save

```

284 private void btneditActionPerformed(java.awt.event.ActionEvent evt) {
285     try{
286         String sql="update data_buku set kode='"+txtkode.getText()+"',judul_buku='"+txtjudul.getText()
287             + "','pengarang'+ txtpengarang.getText()+"',penerbit'"
288             +txtpenerbit.getText()+"',tahun_terbit='"+txttahun.getText()+"' where kode='"+txtkode.getText()+"'";
289
290         java.sql.Connection conn=(Connection)Koneksi.configDB();
291         java.sql.PreparedStatement pstmt=conn.prepareStatement(sql);
292         pstmt.execute();
293         JOptionPane.showMessageDialog(null,"Edit data berhasil....");
294         Tampilkan_Data();
295         //Kosongkan_Form();
296     }catch (HeadlessException | SQLException e){
297         JOptionPane.showMessageDialog(this, e.getMessage());
298     }
299     // TODO add your handling code here:
300 }
301

```

Gambar 2. 14 source code pada button edit

```

308 private void btndeleteActionPerformed(java.awt.event.ActionEvent evt) {
309     try{
310         String sql="delete from data_buku where kode='"+txtkode.getText()+"'";
311         java.sql.Connection conn=(Connection)Koneksi.configDB();
312         java.sql.PreparedStatement pstmt=conn.prepareStatement(sql);
313         pstmt.execute();
314         JOptionPane.showMessageDialog(null,"Hapus data berhasil....");
315         Tampilkan_Data();
316         //Kosongkan_Form();
317     }catch (HeadlessException | SQLException e){
318         JOptionPane.showMessageDialog(this, e.getMessage());
319     }
320 }
321
322

```

Gambar 2. 15 source code pada button delete

```

329 private void btnsearchActionPerformed(java.awt.event.ActionEvent evt) {
330     try{//untuk menjalankan sebuah blok program
331         String sql="select*from data_buku where kode like '%"
332             +txtocari.getText()+"%' + "or judul_buku like '%"
333             +txtocari.getText()+"%' + "or pengarang like '%" +txtocari.getText()+"%' + "or penerbit like '%" +txtocari.getText()+"%' +
334             "or tahun_terbit like '%" +txtocari.getText()+"%'";
335
336         java.sql.Connection conn=(Connection)Koneksi.configDB();
337         java.sql.PreparedStatement pstmt=conn.prepareStatement(sql);
338         pstmt.executeQuery();
339         // JOptionPane.showMessageDialog(null,"pencarian selesai...");
340         Tampilkan_Data();
341         //Kosongkan_Form();
342     }
343 }
344
345 catch (HeadlessException | SQLException e){
346     JOptionPane.showMessageDialog(this, e.getMessage());
347 }
348 // TODO add your handling code here:
349
350 // TODO add your handling code here:
351 }
352

```

Gambar 2. 16 source code pada tombol search

#### d) Jendela game sederhana

```

2 import java.awt.Color;
3 import javax.swing.JFrame;
4 import javax.swing.JOptionPane;
5

```

Gambar 2. 17 import library pada jendela game sederhana

Pada baris ke dua berfungsi untuk menambahkan color pada button

Pada baris ke tiga berfungsi untuk menggunakan pallete pada netbens

Pada baris ke empat berfungsi untuk mengambil atau menampilkan informasi



```

16 public class game extends javax.swing.JFrame {
17
18     private String StartGame="x";
19     private int xCount=0;
20     private int oCount=0;
21
22
23     private void game_score() {
24         px.setText(String.valueOf(xCount));
25         po.setText(String.valueOf(oCount));
26     }
27
28     private void chose() {
29         if(StartGame.equalsIgnoreCase("x")) {
30             StartGame="O";
31         }
32         else {
33             StartGame="x";
34         }
35     }
36 }

```

Gambar 2. 18 source code kelas game score dan chose

```

37     private void wininggame() {
38         String b1=btn1.getText();
39         String b2=btn2.getText();
40         String b3=btn3.getText();
41
42         String b4=btn4.getText();
43         String b5=btn5.getText();
44         String b6=btn6.getText();
45
46         String b7=btn7.getText();
47         String b8=btn8.getText();
48         String b9=btn9.getText();

```

Gambar 2. 19 source code pada kelas wining game

```

52  if(b1=="x" && b2=="x" && b3=="x"){
53      JOptionPane.showMessageDialog(this, "player x win", "tic tac toe",
54          JOptionPane.INFORMATION_MESSAGE);
55      xCount++;
56      game_score();
57      btn1.setBackground(Color.green);
58      btn2.setBackground(Color.green);
59      btn3.setBackground(Color.green);
60  }
61  else if(b4=="x" && b5=="x" && b6=="x"){
62      JOptionPane.showMessageDialog(this, "player x win", "tic tac toe",
63          JOptionPane.INFORMATION_MESSAGE);
64      xCount++;
65      game_score();
66      btn4.setBackground(Color.green);
67      btn5.setBackground(Color.green);
68      btn6.setBackground(Color.green);
69  }
70  else if(b7=="x" && b8=="x" && b9=="x"){
71      JOptionPane.showMessageDialog(this, "player x win", "tic tac toe",
72          JOptionPane.INFORMATION_MESSAGE);
73      xCount++;
74      game_score();
75      btn7.setBackground(Color.green);
76      btn8.setBackground(Color.green);
77      btn9.setBackground(Color.green);
78  }

```

Gambar 2. 20 source code pada penyeleksian kondisi X

```

129  if(b1=="O" && b2=="O" && b3=="O"){
130      JOptionPane.showMessageDialog(this, "player o win", "tic tac toe",
131          JOptionPane.INFORMATION_MESSAGE);
132      oCount++;
133      game_score();
134      btn1.setBackground(Color.green);
135      btn2.setBackground(Color.green);
136      btn3.setBackground(Color.green);
137  }
138  else if(b4=="O" && b5=="O" && b6=="O"){
139      JOptionPane.showMessageDialog(this, "player o win", "tic tac toe",
140          JOptionPane.INFORMATION_MESSAGE);
141      oCount++;
142      game_score();
143      btn4.setBackground(Color.green);
144      btn5.setBackground(Color.green);
145      btn6.setBackground(Color.green);
146  }

```

Gambar 2. 21 source code pada penyeleksian kondisi O

```

602 private void btnresetActionPerformed(java.awt.event.ActionEvent evt) {
603     btn1.setText(null);
604     btn2.setText(null);
605     btn3.setText(null);
606
607     btn4.setText(null);
608     btn5.setText(null);
609     btn6.setText(null);
610
611     btn7.setText(null);
612     btn8.setText(null);
613     btn9.setText(null);
614
615
616     btn1.setBackground(Color.LIGHT_GRAY);
617     btn2.setBackground(Color.LIGHT_GRAY);
618     btn3.setBackground(Color.LIGHT_GRAY);
619
620     btn4.setBackground(Color.LIGHT_GRAY);
621     btn5.setBackground(Color.LIGHT_GRAY);
622     btn6.setBackground(Color.LIGHT_GRAY);
623
624     btn7.setBackground(Color.LIGHT_GRAY);
625     btn8.setBackground(Color.LIGHT_GRAY);
626     btn9.setBackground(Color.LIGHT_GRAY);
627     // TODO add your handling code here:
628 }

```

Gambar 2. 22 source code pada tombol reset

```

484 private void btn7ActionPerformed(java.awt.event.ActionEvent evt) {
485     btn7.setText(StartGame);
486     if(StartGame.equalsIgnoreCase("x")){
487         btn7.setForeground(Color.blue);
488     }else{
489         btn7.setForeground(Color.red);
490     }
491     chose();
492     wininggame();
493     // TODO add your handling code here:
494 }

```

Gambar 2. 23 source code pada setiap tombol

## **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

Aplikasi Data Buku Perpustakaan adalah aplikasi yang digunakan untuk menyimpan dan melihat tampilan buku pada netbens. Aplikasi ini sudah menggunakan Database,penyeleksian kondisi if dan else serta menggunakan komponen Swing Aplikasi ini di buat berdasarkan kebutuhan perpustakaan dalam menyimpan data buku yang baru masuk dan menghapus data buku

#### **B. Saran**

Penulis menyadari masih banyak terdapat kelemahan dan kekurangan didalam aplikasi data buku perpustakaan salah satu contohnya adalah tidak ada menu tanggal peminjaman,dan pembembalian buku. Oleh karna itu penulis berharap aplikasi ini dapat dikembangkan dan diinovasi sedemikian rupa agar lebih menarik perhatian pengguna. Penulis juga mengucapkan rasa terimakasih kepada Bapak Slamet Triyanto, S.ST yang telah memberikan arahan dan bimbingan selama proses pembuatan aplikasi sehingga penulis dapat menyelesaikan aplikasi dan laporan dengan baik. Namun penulis menyadari masih terdapat banyak kesalahan dalam pembuatan laporan ini, untuk itu penulis berharap saran dan keritik yang membangun dari pembaca agar penulis dapat memperbaiki kesalahan dan dapat membuat laporan praktikum yang lebih baik lagi

## DAFTAR PUSTAKA

<https://repository.bsi.ac.id/index.php/unduh/item/258812/1.MODUL-PEMROGRAMAN-VISUAL-II-DENGAN-BAHASA-JAVA.pdf>. Diakses tanggal 5 agustus 2001 pukul 20:30

<https://www.slideshare.net/bayuvins/koneksi-database-mysql-dengan-netbean>. Diakses tanggal 5 agustus 2001 pukul 20:30

<https://medium.com/@yunitapujiyanti0/memahami-struktur-dan-aturan-penulisan-sintaks-java-bagian-1-a0d38ed2cd2b>. Diakses tanggal 5 agustus 2001 pukul 20:30

<https://core.ac.uk/download/pdf/293465667.pdf>. Diakses tanggal 5 agustus 2001 pukul 20:30