



Mata Kuliah : Pemograman Berbasis Objek dan Lab
Materi : JAVA SWING 4 - DAO
Kelas : TI-20-PA / TI-20-KA
Pengajar : Febri Damatraseta S.T, M.Kom

1. Create project with maven.

Membuat project JAVA dengan menggunakan maven, bagi yang belum terinstall maven pada pc atau laptopnya bisa mengikuti proses instalasi di <https://github.com/FebryFairuz/Pemograman-Object-Oriented/blob/master/src/module/Installasi-Maven.docx>

Buatlah project maven dengan nama `app_library` dan groupid adalah `'com.ibik.library.app.login'` dengan tahapan seperti dibawah ini:

- Buka command prompt atau terminal pada os pc.
- Pilih directory path yang akan menjadi tempat penyimpanan project anda. Contoh directory D:\ dan menggunakan folder bernama PBO, dimana path PBO yang akan digunakan sebagai tempat untuk penyimpanan project maven.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Febri>d:

D:\>cd D:\PBO

D:\PBO>
```

- Setelah berhasil masuk kedalam path directory yang telah ditentukan masukan argument untuk membuat project maven dengan nama `app_library` sebagai nama projectnya dan package dengan nama `'com.ibik.library.app.login'` sebagai groupId.

```
mvn -B archetype:generate -DgroupId=com.ibik.library.app.login -
DartifactId=app_library -DarchetypeArtifactId=maven-archetype-quickstart
-DarchetypeVersion=1.4
```

Setelah berhasil membuat project dengan maven silakan kalian masuk kedalam path project yang telah dibuat. Gambar dibawah ini merupakan contoh pembuatan project maven jika kondisinya sudah menampilkan tulisan **BUILD SUCCESS** maka project maven telah berhasil dibuat, namun jika menampilkan tulisan **BUILD FAILURE** maka project maven gagal dibuat.

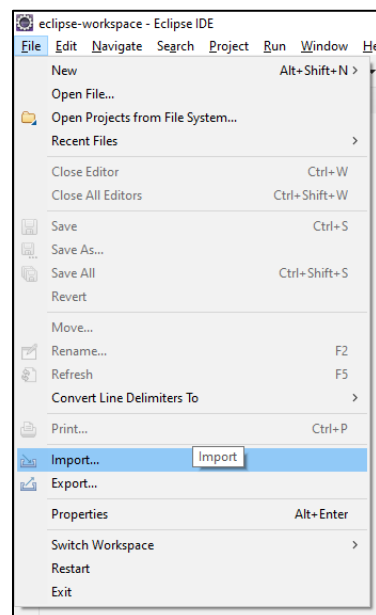
```
C:\WINDOWS\system32\cmd.exe
C:\Users\Febri>d:
D:\>cd D:\PBO

D:\PBO>mvn -B archetype:generate -DgroupId=com.ibik.library.app.login -DartifactId=app_library -DarchetypeArtifactId=
maven-archetype-quickstart -DarchetypeVersion=1.4
[INFO] Scanning for projects...
[INFO]
-----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO]
-----[ pom ]-----
[INFO]
>>> maven-archetype-plugin:3.2.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
<<< maven-archetype-plugin:3.2.0:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
--- maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO]
-----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO]
-----
[INFO] Parameter: groupId, Value: com.ibik.library.app.login
[INFO] Parameter: artifactId, Value: app_library
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.ibik.library.app.login
[INFO] Parameter: packageInPathFormat, Value: com/ibik/library/app/login
[INFO] Parameter: package, Value: com.ibik.library.app.login
[INFO] Parameter: groupId, Value: com.ibik.library.app.login
[INFO] Parameter: artifactId, Value: app_library
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: D:\PBO\app_library
[INFO]
-----
[INFO] BUILD SUCCESS
[INFO]
-----
[INFO] Total time: 25.579 s
[INFO] Finished at: 2021-12-17T12:33:25+07:00
[INFO]
-----

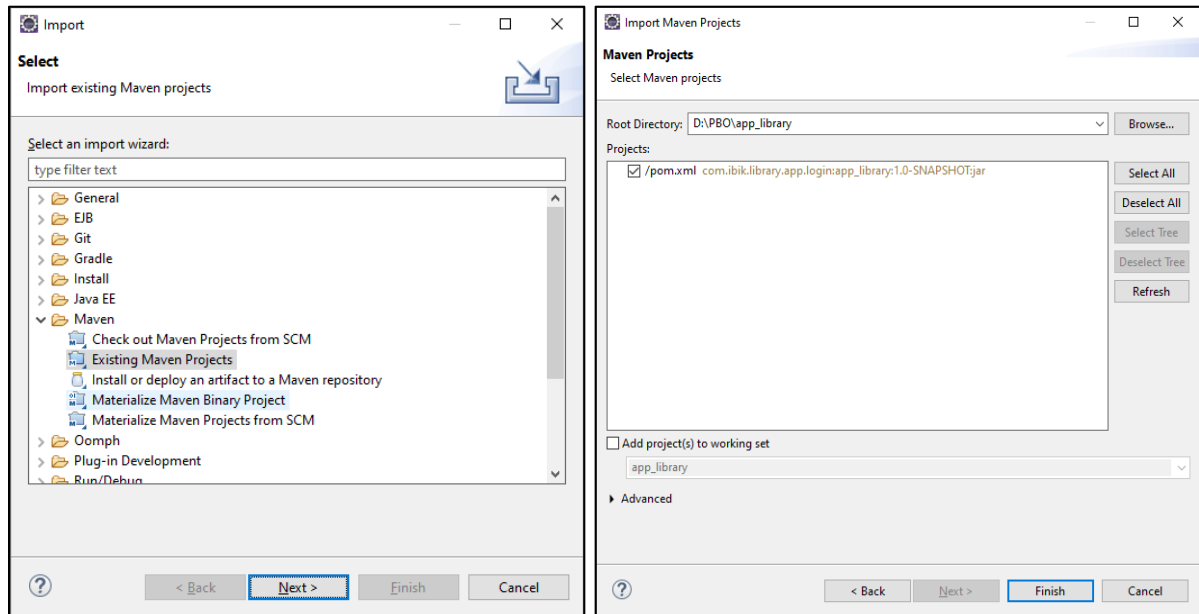
D:\PBO>cd app_library
D:\PBO\app_library>
```

2. Importing project maven to Eclipse.

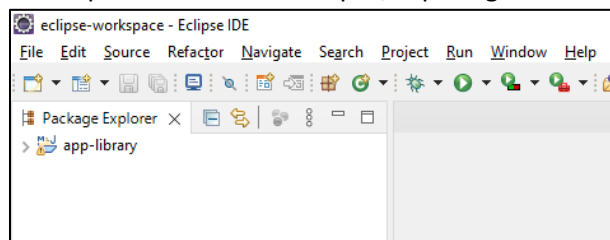
Setelah berhasil membuat project maven, silakan buka aplikasi text editor IDE Eclipse for JAVA. Selanjutnya akan melakukan import project maven kedalam IDE eclipse dengan cara klik tombol menu **File → Import...**



Maka akan tampil windows dialog dengan nama Import, pada pilihan select an import wizard pilihlah **Maven → Existing Maven Projects** lalu klik tombol Next. Maka akan muncul windows dialog dengan nama Import Maven Projects, pada windows ini untuk isian Root Directory klik tombol Browse... dan pilih path directory project maven yang telah dibuat sebelumnya.



Setelah memilih root directory path dari project maven klik tombol Finish. Maka project maven app_library telah berhasil di import kedalam IDE Eclipse, seperti gambar dibawah ini.

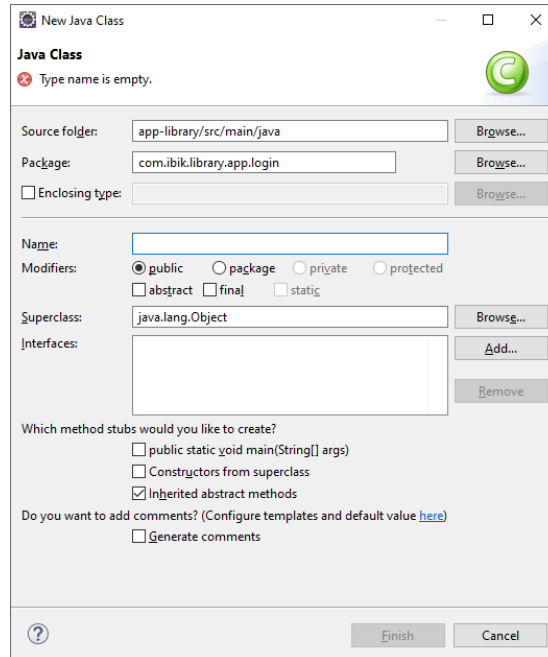


Jika pada eclipse tidak menampilkan tab window view Package Explorer, dilakan klik menu **Window → Show View → Package Explorer**

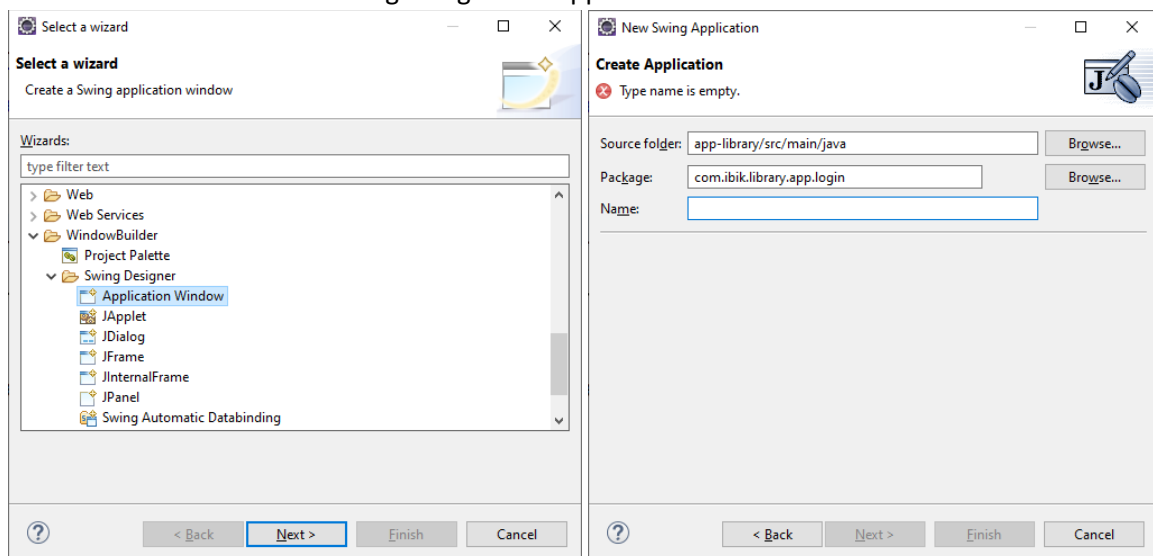
3. Create main class project.

Pada bagian ini akan membuat tiga buah class java, dua class java biasa dan satu class dengan tipe Windows Application yang disimpan pada package `com.ibik.library.app.login`. Package ini akan menjadi main class project, dalam arti ketika aplikasi ini berjalan class pada package ini lah yang akan pertama kali di execute oleh engine java pada maven.

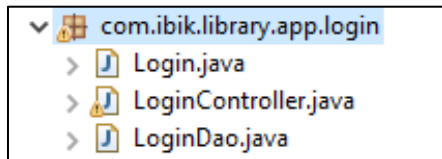
- a. Buatlah dua buah file class java dengan nama Login dan LoginDao pada package `com.ibik.library.app.login`, dengan cara klik kanan pada package tersebut pilih **New** → **Class** dan akan muncul window dialog New Java Class. Pada isian Name silakan isi dengan nama Login, lalu klik tombol Finish. Lakukan hal yang sama untuk membuat class LoginDao.



- b. Buatlah class dengan tipe Windows Application didalam package yang sama dengan kedua class sebelumnya. Klik kanan pada package pilih **New** → **Other...** lalu muncul window **Select a wizard**. Pilihlah **WindowBuilder** → **Swing Designer** → **Application Window** lalu klik tombol **Next**.

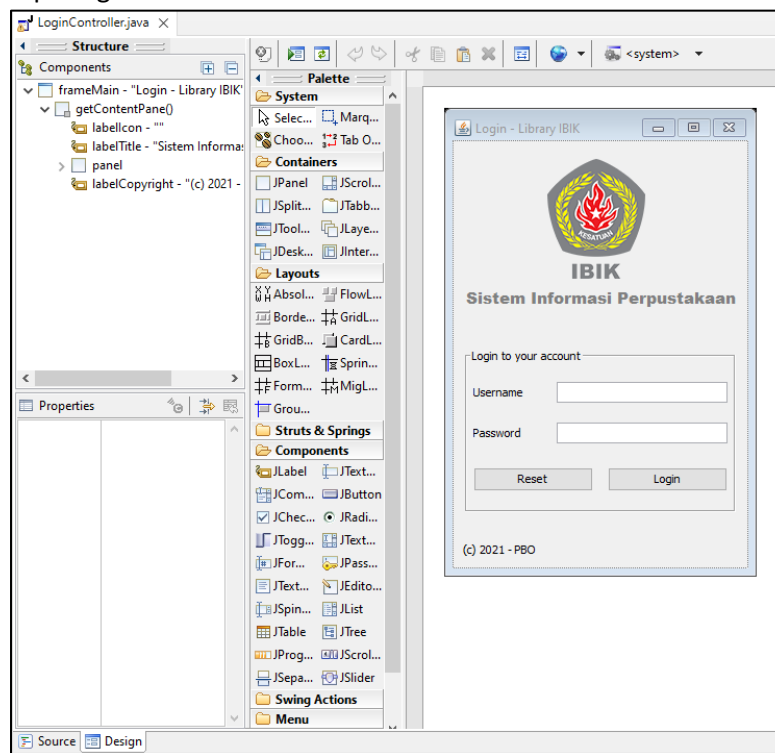


Pada window New Swing Application pada isian Name silakan isi dengan nama LoginController lalu klik tombol Finish. Maka dapat dilihat pada Package Explorer pada package `com.ibik.library.app.login` telah terbentuk tiga buah file java yaitu Login, LoginDAO dan LoginController.



4. Design layout class LoginController.

Bukalah file LoginController pada package explorer dengan cara klik kanan pilih **Open with → WindowBuilder Editor**. Setelah terbuka pada tab paling bawah pilihlah tab Design, maka akan muncul tampilan component dan struktur untuk mendesain sebuah aplikasi desktop. Silakan membuat desain seperti gambar dibawah ini:



Desainlah halaman login seperti gambar diatas dengan menggunakan Palette yang disediakan. Cara menggunakannya cukup drag and drop item yang akan digunakan.

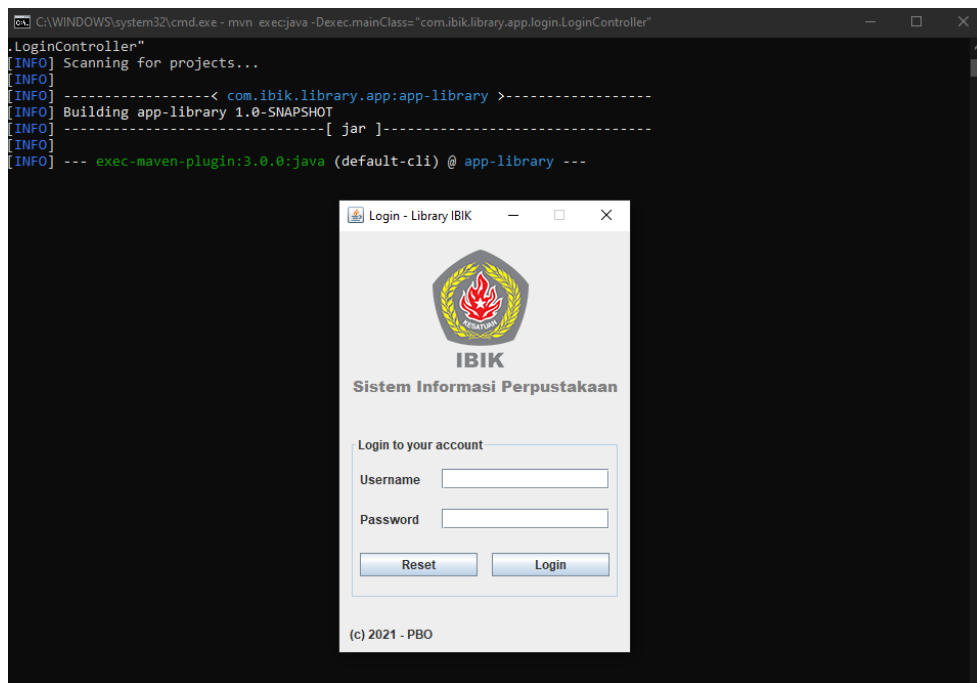
5. Running project with maven.

Cara menjalankan project yang telah dibuat silakan buka command prompt dan masuk kedalam directory path project yang telah dibuat.

Masukan argument `mvn clean package` untuk mengkompile project terlebih dahulu. Lalu masukan argument `maven` untuk running dengan menuliskan `mvn exec:java -Dexec.mainClass="namaPackage.namaClass"`. Contoh:

```
D:\PBO\app_library>mvn clean package
D:\PBO\app_library>mvn exec:java -Dexec.mainClass="com.ibik.library.app.login.LoginController"
```

Jika pada saat mengkompile project mendapatkan tulisan BUILD SUCCESS maka program berhasil dikompile dan akan menghasilkan sebuah file berextensi `.jar` pada folder target. Jika yang didapatkan BUILD FAILURE maka program memiliki error code dan perlu diperbaiki kesalahan yang telah ditentukan.



Pada gambar diatas merupakan tampilan jika setelah mengeksekusi argument running project telah berhasil dikerjakan. Maka akan menampilkan class LoginController yang telah dibuat.

6. Create validation for login form.

Buka file LoginController pada IDE Eclipse dan pilih tampilan source pada tab paling bawah. Pada tampilan tersebut akan menampilkan scripting source code dari desain layout yang telah dibuat. Pada bagian ini silakan membuat sebuah validasi dengan ketentuan seperti:

- Isian dari username atau password tidak boleh kosong
- Isian dari username dan password harus menggunakan angka
- Isian dari username tidak boleh lebih dari 10 digit
- Isian dari password tidak boleh lebih dari 6 digit

Jika ketentuan diatas terpenuhi maka akan menampilkan sebuah window dialog error dengan menampilkan info dari masing-masing ketentuan tersebut. Buatlah scripting tersebut sesuai dengan ketentuan-ketentuan diatas dan masukan kedalam sebuah function bernama `ValidationForm()`.

7. Connection to database with JDBC.

Bagian ini akan mengkoneksikan aplikasi dengan database. Pada module sebelumnya telah dijelaskan untuk mengkoneksikan aplikasi java dengan database menggunakan JDBC. Disini akan menggunakan database dari module sebelumnya yang bernama **IBIK_AppLibrary** dimana database tersebut memiliki satu buah table yaitu **table User**.

Nama	Tipe/Value	Keterangan
ID	int(11)	Primary, uniq
NIK	Varchar(10)	Secondary, uniq
Fullname	Varchar(50)	
PlaceBirth	Varchar(20)	
Birthdate	Date	
Gender	Enum('m','f')	m = Male; f = female
Address	Textarea	
Email	Varchar(25)	
Password	Varchar(100)	
CreatedAt	Timestamp	On insert current date
CreatedBy	Varchar(20)	
UpdatedAt	Timestamp	On update current date
UpdatedBy	Varchar(20)	

- a. Memasang plugin API sql pada project maven.

Buka file pom.xml dan masukan script dibawah ini untuk menambahkan library API MYSQL.

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.27</version>
</dependency>
```

Masukan script tersebut pada bagian tag `<dependencies> ... </dependencies>` pada tag ini lah semua plugin atau library yang dibutuhkan pada project ditambahkan. Untuk menemukan library maven dapat mengunjungi situs : <https://mvnrepository.com/>

```
19 </properties>
20
21<dependencies>
22  <dependency>
23    <groupId>junit</groupId>
24    <artifactId>junit</artifactId>
25    <version>4.11</version>
26    <scope>test</scope>
27  </dependency>
28
29  <dependency>
30    <groupId>mysql</groupId>
31    <artifactId>mysql-connector-java</artifactId>
32    <version>8.0.27</version>
33  </dependency>
34
35 </dependencies>
36
```

- b. Buatlah package baru dengan nama `com.ibik.library.app.connection` dan buatlah satu file class java dengan nama `ConnectionDB`. Pada class `ConnectionDB` silakan buat script seperti dibawah ini untuk membuat koneksi dengan database *ibik_applibrary*.

```
package com.ibik.library.app.connection;

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

public class ConnectionDB {
    private Connection conn;

    public Connection connect() throws SQLException {
        String host = "localhost:3306";
        String dbName = "ibik_applibrary";
        String dbuser = "root";
        String dbpassword = "";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }

        Connection conn = DriverManager.getConnection("jdbc:mysql://" + host + "/" + dbName,
            dbuser, dbpassword);

        return conn;
    }

    public Connection close() throws SQLException {
        conn.close();
        return conn;
    }
}
```

Dari script diatas terdapat class `ConnectionDB` memiliki satu buah atribute bernama `conn` dengan tipe data `Connection`. Dan memiliki dua buah function bernama `connect()` dan `close()`.

- c. Setelah membuat file koneksi bernama class `ConnectionDB`, selanjutnya akan memasangkan file koneksi tersebut kedalam file `LoginController`. Silakan buka file `LoginController` lalu pada function `LoginController()` silakan tambahkan script dibawah ini:

```
public LoginController() {
    //call the ConnectionDB class using error handling : try_catch
    ConnectionDB conn = new ConnectionDB(); //init object class ConnectionDB
    //error handling for if connection is connected will show initialize() function
    try {
        conn.connect();
        initialize();

        //error handling for if connection is connected will show Window Dialog, the connection is failed
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        e.printStackTrace();
    }
    //end call
}
```

Dari scripting diatas dapat dilihat bahwasannya jika berhasil terkoneksi ke dalam database maka akan menampilkan function `initialize()` yang isinya adalah code untuk menjalankan GUI class

LoginController. Sedangkan jika tidak berhasil terkoneksi maka akan menampilkan window dialog error dengan info gagal terkoneksi ke database.

8. Connection to table with DAO.

Data Access Object (DAO) adalah sebuah *pattern* yang umum digunakan oleh para *developer* java ketika mengkoneksikan aplikasi dengan database. Konsep kerja dari DAO ini memisahkan kode program berdasarkan fungsinya, sifat dari DAO itu sendiri untuk memanipulasikan data yang terdapat di dalam database sehingga aplikasi yang dibuat menjadi *Loose coupling*. Seluruh proses *Create Retrive Update Delete* (CRUD) database ada didalam *class-class* DAO tersebut. Cara kerja dari DAO itu sendiri diilustrasikan seperti berikut:



Penjelasan lebih lanjut dapat dilihat di: <http://tutorials.jenkov.com/java-persistence/index.html>

Setelah membuat koneksi ke database berhasil, selanjutnya membuat koneksi kedalam tabel User. Dimana ketentuan dari point nomor 6, jika ketentuan tersebut tidak ada yang terpenuhi. Aplikasi akan mengecek jika username sama dengan NIK dan password sama dengan Password pada table User maka akan masuk ke halaman windows frame Dashboard.

- Buka file Login dan buatlah teknik *encapsulation* berdasarkan **table User**.
- Buka file LoginDAO dan buatlah satu fungsi bernama `checkLogin()` seperti dibawah ini:

```
package com.ibik.library.app.login;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import com.ibik.library.app.connection.ConnectionDB;

public class LoginDao {

    private String sqlLogin = "select * from user where NIK = ? and password = ?";

    public Login checkLogin(Login login) throws Exception{
        if(login == null) { //check if class login is empty
            return null; //will return null
        }

        ConnectionDB conn = new ConnectionDB();
        Connection c = conn.connect();//open connection to db by class ConnectionDB
        PreparedStatement pst = c.prepareStatement(sqlLogin);
        pst.setLong(1, login.getNIK()); //entering parameter NIK to index 1
        pst.setString(2, login.getPassword()); //entering parameter Password to index 2
        ResultSet rs = pst.executeQuery(); //execute query base on var sqlLogin
        if(!rs.next()) { //check if there is no result from database
            return null; //will return null
        }

        //if it has result from query it will put the result to class encapsulation class Login
        Login l = new Login();
        l.setID(rs.getInt("ID"));
        l.setNIK(rs.getLong("NIK"));
        l.setFullname(rs.getString("Fullname"));
        l.setPlaceBirth(rs.getString("PlaceBirth"));
        l.setBirthdate(rs.getString("Birthdate"));
        l.setGender(rs.getString("Gender"));
        l.setAddress(rs.getString("Address"));
        l.setEmail(rs.getString("Email"));
        l.setPassword(rs.getString("Password"));

        c.close(); //closing connection to db
        return l; //return class Login value
    }
}
```

- c. Buka class LoginController, dan masukan script-script dibawah ini untuk mengecek apakah isian dari form Login ada pada database.

➤ Pada Action Click button Login

```
//action for btnLogin if having click event
JButton btnLogin = new JButton("Login");
btnLogin.setBounds(139, 114, 115, 23);
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String username = textUsername.getText();
        String password = String.valueOf(textPassword.getPassword());
        //sending variabel username and password to function ValidationForm() and save it in to variabel result
        MyResults result = new LoginController().ValidationForm(username, password);
        if(result.getResponse()) {
            frameMain.dispose();
            new Dashboard().setVisible(true);
        }else {
            JOptionPane.showMessageDialog(null, result.getLabel(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
panel.add(btnLogin);
//end action for btnLogin if having click event
```

➤ Pada function ValidationForm()

```
public MyResults ValidationForm(String username, String password) { //sending parameter username & password
    MyResults m = new MyResults();
    //condition for if username or password is empty
    if (username.isEmpty() || password.isEmpty()) {
        m.setResponse(false);
        m.setLabel("Fill both the username and password fields");
    }

    //condition for if username has length more than 10
    else if(username.length() > 10) {
        m.setResponse(false);
        m.setLabel("Keyword must not exceed 10 characters");
    }

    //condition for if password has length more than 6
    else if(password.length() > 6) {
        m.setResponse(false);
        m.setLabel("Keyword must not exceed 6 characters");
    }

    //condition for if username or password has alphabet value => only number can be pass it
    else if( username.matches("[a-z]*") || password.matches("[a-z]*") ) {
        m.setResponse(false);
        m.setLabel("This entry can only contain numbers");
    }

    //condition if username and password is not empty
    else {
        //sending username and password to DAO
        //saving to class Login
        Long NIK = Long.parseLong(username);

        Login login = new Login();
        login.setNIK( NIK );
        login.setPassword(password);
        //end saving to class Login
        //sending class Login to DAO
        LoginDao loginDao = new LoginDao();
        try {
            Login l = loginDao.checkLogin(login);
            System.out.println("NIK "+l.getNIK());
            System.out.println("Fullname "+l.getFullname());
            m.setResponse(true);
            m.setLabel("Wellcome to this application");
        } catch (Exception e) {
            e.printStackTrace();
            m.setResponse(false);
            m.setLabel("You have entered an invalid username or password.");
        }
    }
    return m;
}
```

- d. Buatlah file class JFrame didalam package `com.ibik.library.app.dashboard` dengan nama Dashboard. Silakan desain bebas isi dari page ini.

9. Result of the apps.

Setelah selesai menambahkan scripting diatas silakan di run projectnya dan berikan hasil capture dari project yang telah dibuat. Beserta menampilkan hasil dari ketentuan form pada point 6 dan tampilan jika kondisi username dan password sesuai dengan table User. Kumpulkan dalam bentuk wording yang berisi hasil capture program beserta source code setiap capture program. Source code project dapat disimpan melalui versioning GIT dengan tag @FebryFairuz atau dapat di upload di gdrive dengan share ke febridmf@ibik.ac.id

Jika ada kendala dalam pengerjaan silakan hubungi langsung saya secara pribadi atau melalui groupnya.