

LAPORAN PROYEK PEMROGRAMAN BERORIENTASI OBJEK
JUAL BELI ONLINE



DISUSUN OLEH :

- **Sunarti Barasa (13317009)**
- **Kristina Sitinjak (13317012)**
- **Sarah E.M Barasa (13317020)**
- **Bunga Jelita Silaen (13317026)**

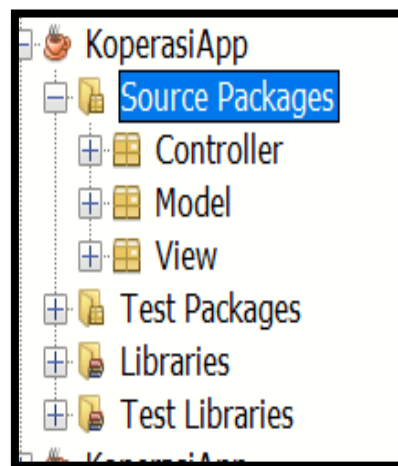
2017/2018

PEMBAHASAN

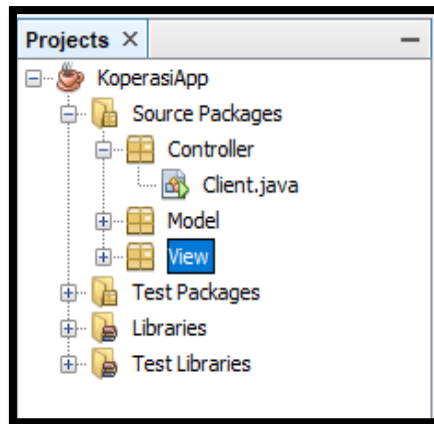
Di dalam Proyek yang berjudul Jual Beli Online ini, Kami menggunakan 2 PC dimana PC 1 sebagai Server dan PC 2 sebagai Client yang saling terhubung dengan networking, disini kita akan membahas kedua Pc tersebut. Kode program yang paling membedakan ada pada class “menu”. Class menu client dan class menu server sangat jauh berbeda dikarenakan pada class ini transaksi networking. Sehingga kode programnya sangat berbeda dan lebih diperhatikan. Dalam laporan ini class menu client dan class menu server dibahas lebih rinci. Berikut langkah – langkahnya.

A. Cara Create Project

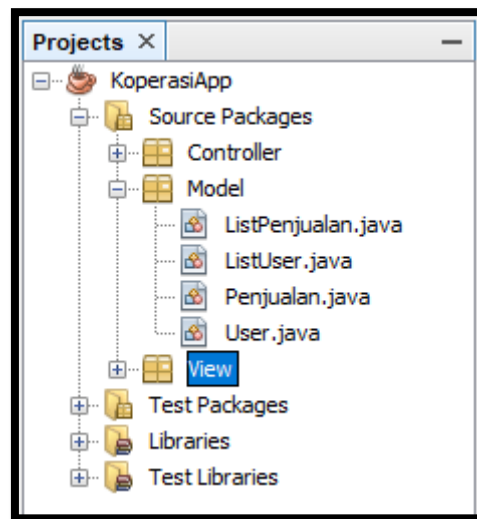
Sebelum membuat aplikasi Jual Beli Online, terlebih dahulu kita membuat package - packagenya pada netbeans. Berikut ini merupakan tampilan package – package tersebut. Pada proyek yang berjudul Aplikasi Jual Beli Online Institut Teknologi Del pada create project ini isi kodingan nya hampir sama antara server dengan client. Hanya saja yang membedakannya yakni package View class JFrame pada Menu dan Login.



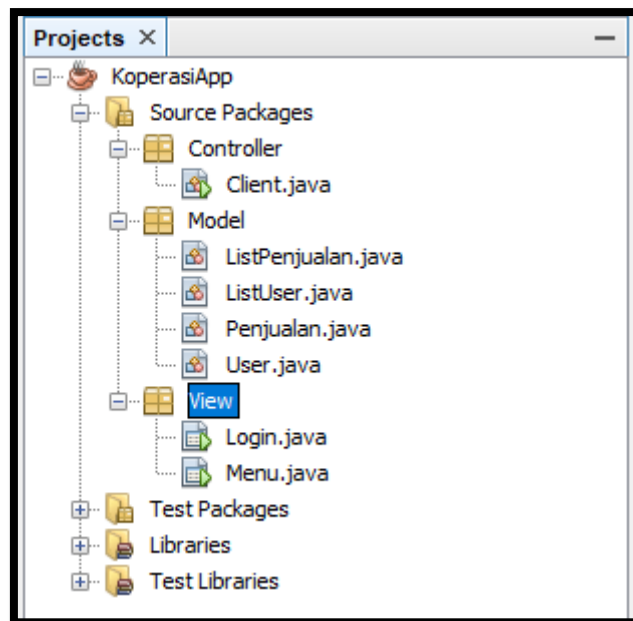
- Pada package Controller memiliki beberapa class yaitu :



- Pada package Model memiliki beberapa class :



- Pada package View memiliki beberapa class :

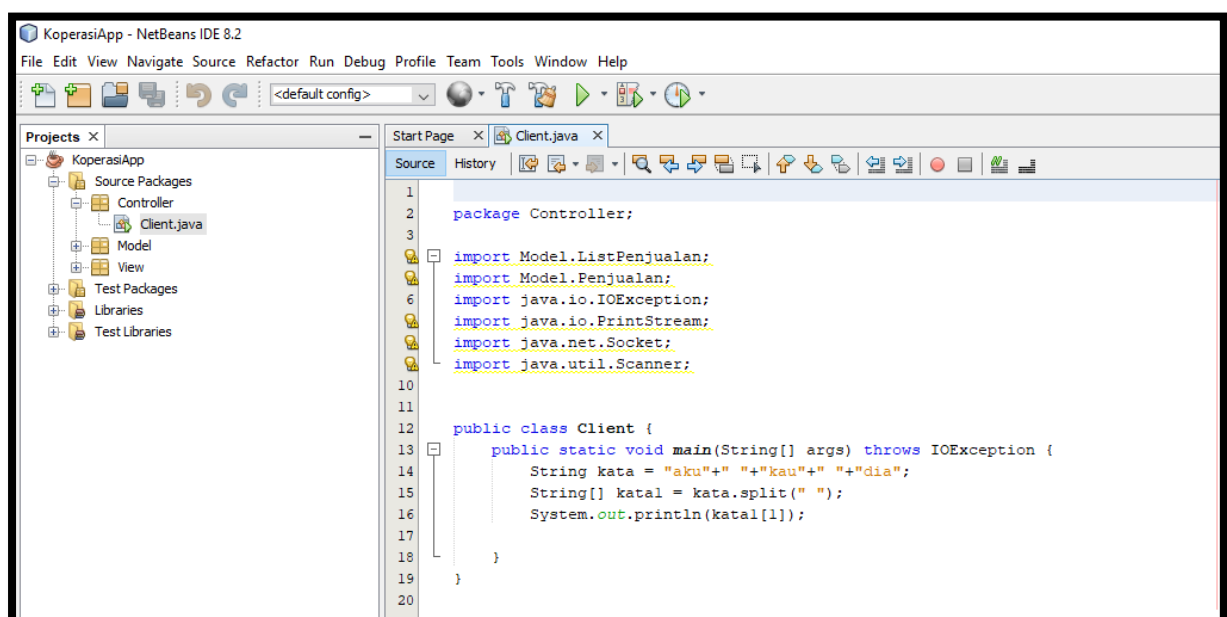


Setelah langkah tersebut telah berhasil kita buat, maka selanjutnya kita membuat kelas-kelas yang diperlukan untuk membuat aplikasi Jual Online tersebut

Program pada client

Proses yang terjadi pada bagian PC2/client. Client berfungsi untuk mengirimkan data pada server, kemudian meminta server mengirim balasan untuk total pembayaran.

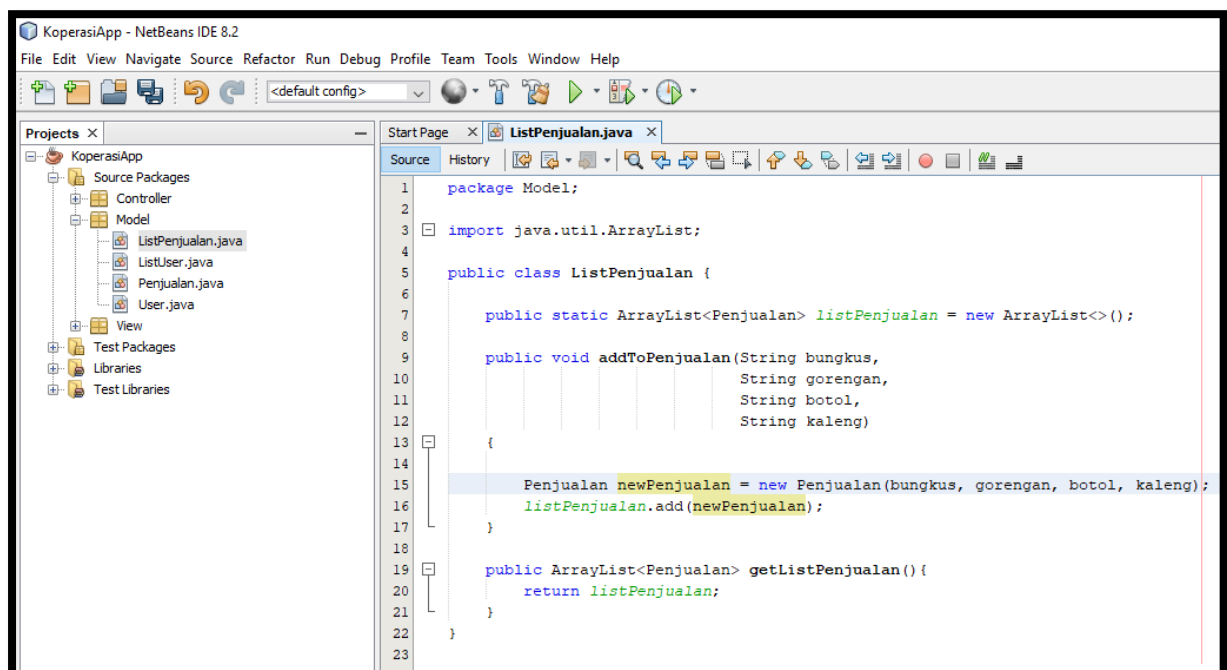
Penjelasan gambar dibawah ialah menjelaskan tampilan awal pada package controller yang dimana terdiri atas 1 class saja. Yakni class client.



Berikut isi dari class client

```
1
2  package Controller;
3
4  import Model.ListPenjualan;
5  import Model.Penjualan;
6  import java.io.IOException;
7  import java.io.PrintStream;
8  import java.net.Socket;
9  import java.util.Scanner;
10
11
12 public class Client {
13     public static void main(String[] args) throws IOException {
14         String kata = "aku"+" "+"kau"+" "+"dia";
15         String[] katal = kata.split(" ");
16         System.out.println(katal[1]);
17     }
18 }
19
20
```

Penjelasan gambar dibawah ialah menjelaskan tampilan awal pada pakacge Model yang dimana terdiri atas 4 class saja. Yakni class ListPenjualan ListUser, Penjualan, dan User .



The screenshot shows the NetBeans IDE 8.2 interface. On the left, the 'Projects' pane displays the project structure for 'KoperasiApp', including 'Source Packages' (Controller, Model, View), 'Test Packages', and 'Libraries'. The 'Model' package is expanded, showing four classes: 'ListPenjualan.java', 'ListUser.java', 'Penjualan.java', and 'User.java'. The main editor window displays the code for 'ListPenjualan.java'.

```
1  package Model;
2
3  import java.util.ArrayList;
4
5  public class ListPenjualan {
6
7      public static ArrayList<Penjualan> listPenjualan = new ArrayList<>();
8
9      public void addToPenjualan(String bungkus,
10                                String gorengan,
11                                String botol,
12                                String kaleng)
13      {
14
15          Penjualan newPenjualan = new Penjualan(bungkus, gorengan, botol, kaleng);
16          listPenjualan.add(newPenjualan);
17      }
18
19      public ArrayList<Penjualan> getListPenjualan() {
20          return listPenjualan;
21      }
22  }
23
```

Berikut adalah isi dari class ListPenjualan pada package Model.

```
1 package Model;
2
3 import java.util.ArrayList;
4
5 public class ListPenjualan {
6
7     public static ArrayList<Penjualan> listPenjualan = new ArrayList<>();
8
9     public void addToPenjualan(String bungkus,
10                                String gorengan,
11                                String botol,
12                                String kaleng)
13     {
14
15         Penjualan newPenjualan = new Penjualan(bungkus, gorengan, botol, kaleng);
16         listPenjualan.add(newPenjualan);
17     }
18
19     public ArrayList<Penjualan> getListPenjualan() {
20         return listPenjualan;
21     }
22 }
23
```

Berikut adalah isi dari class ListUser pada package Model.

```
1 package Model;
2
3 import java.util.ArrayList;
4
5 public class ListUser {
6     public static ArrayList<User> listUser = new ArrayList<>();
7
8     public void addUserList(){
9         User newUser = new User("user", "user", "1");
10        listUser.add(newUser);
11    }
12
13    public boolean isAuth(String username, String password){
14        boolean isAuth = false;
15        for(int i = 0; i < listUser.size(); i++){
16            return username.equals(listUser.get(i).getUsername()) && password.equals(listUser.get(i).getPassword());
17        }
18        return false;
19    }
20
21 }
22
```

Berikut adalah isi dari class User pada package Model.

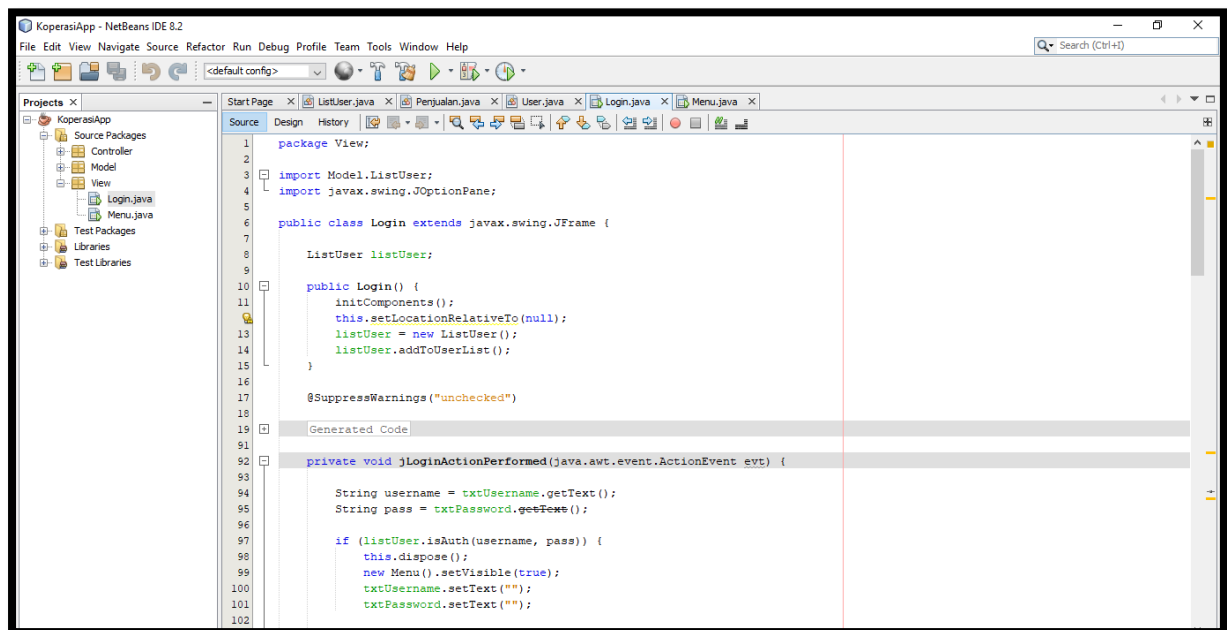
```
1 package Model;
2
3 public class User {
4     private String username, password;
5
6     public User(String username, String password, String saldo) {
7         this.username = username;
8         this.password = password;
9     }
10
11     public User() {
12     }
13
14     User(String user, String user0, String sarah, String barasa, String string) {
15         System.out.println("");
16     }
17
18     User(String user, String user0) {
19         System.out.println("");
20     }
21
22     public String getUsername() {
23         return username;
24     }
25
26     public void setUsername(String username) {
27         this.username = username;
28     }
29
30     public String getPassword() {
31         return password;
```

Berikut adalah isi dari class Penjualan.

```
1 package Model;
2
3 public class Penjualan {
4     private String bungkus,goreng, botol, kaleng;
5
6     public Penjualan(String bungkus, String goreng, String botol, String kaleng) {
7         this.bungkus = bungkus;
8         this.goreng = goreng;
9         this.botol = botol;
10        this.kaleng = kaleng;
11    }
12
13    public String getBungkus() {
14        return bungkus;
15    }
16
17    public void setBungkus(String bungkus) {
18        this.bungkus = bungkus;
19    }
20
21    public String getGoreng() {
22        return goreng;
23    }
24
25    public void setGoreng(String goreng) {
26        this.goreng = goreng;
27    }
28
29    public String getBotol() {
30        return botol;
31    }
```

```
32
33    public void setBotol(String botol) {
34        this.botol = botol;
35    }
36
37    public String getKaleng() {
38        return kaleng;
39    }
40
41    public void setKaleng(String kaleng) {
42        this.kaleng = kaleng;
43    }
44
45    String getNoTransaksi() {
46        throw new UnsupportedOperationException("Not supported yet.");
47    }
48 }
49
```


Penjelasan gambar dibawah ialah menjelaskan tampilan awal pada pakacge View yang dimana terdiri atas 2 class JFrame. Yakni class JFrame login dan class JFrame Menu (digunakan sebagai menu client).



Berikut adalah isi code dari program Login, sebelumnya untuk membuat Login kita harus membuat design GUI pada class login.

```

1  package View;
2
3  import Model.ListUser;
4  import javax.swing.JOptionPane;
5
6  public class Login extends javax.swing.JFrame {
7
8      ListUser listUser;
9
10     public Login() {
11         initComponents();
12         this.setLocationRelativeTo(null);
13         listUser = new ListUser();
14         listUser.addToUserList();
15     }
16
17     @SuppressWarnings("unchecked")
18
19     Generated Code
20
21     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
22
23         String username = txtUsername.getText();
24         String pass = txtPassword.getText();
25
26         if (listUser.isAuth(username, pass)){
27             this.dispose();
28             new Menu().setVisible(true);
29             txtUsername.setText("");
30             txtPassword.setText("");
31         }
32     }
33 }

```

```

99
100     } else {
101         JOptionPane.showMessageDialog(rootPane, "Password Salah");
102         txtPassword.setText("");
103         txtPassword.requestFocus();
104     }
105 }
106
107
108 private void txtUsernameActionPerformed(java.awt.event.ActionEvent evt) {
109
110 }
111
112 private void txtPasswordActionPerformed(java.awt.event.ActionEvent evt) {
113     // TODO add your handling code here:
114     String user = ("freakprogrammer");
115     String pass = ("12345");
116     String usname = txtUsername.getText();
117     String passw = txtPassword.getText();
118     if (usname.equals(user) && passw.equals(pass)) {
119         this.setVisible(false);
120         new Menu().setVisible(true);
121     } else {
122         JOptionPane.showMessageDialog(null, "User Name atau Password Salah \n");
123         txtUsername.setText("");
124         txtPassword.setText("");
125     }
126 }

```

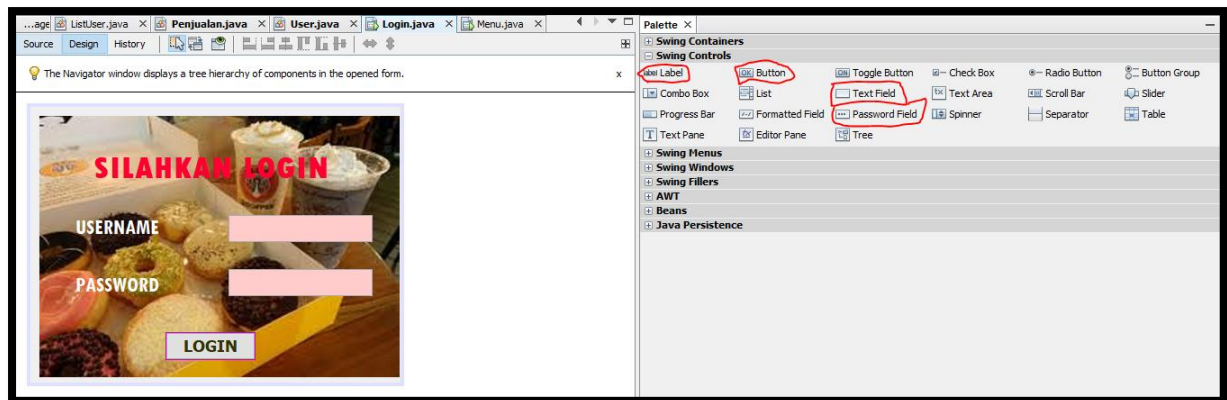
Potongan program dibawah adalah bawaan dari GUI pada saat melakukan Design

```

127
128 /**
129  * @param args the command line arguments
130  */
131 public static void main(String args[]) {
132     /* Set the Nimbus look and feel */
133     Look and feel setting code (optional)
134
135     /* Create and display the form */
136     java.awt.EventQueue.invokeLater(new Runnable() {
137         @Override
138         public void run() {
139             new Login().setVisible(true);
140         }
141     });
142 }
143
144 // Variables declaration - do not modify
145 private javax.swing.JButton jButton1;
146 private javax.swing.JLabel jLabel1;
147 private javax.swing.JLabel jLabel2;
148 private javax.swing.JLabel jLabel3;
149 private javax.swing.JLabel jLabel4;
150 private javax.swing.JPasswordField txtPassword;
151 private javax.swing.JTextField txtUsername;
152 // End of variables declaration
153 }
154

```

Gambar dibawah ini menjelaskan tentang bagaimana cara membuat Design dari Login GUI.



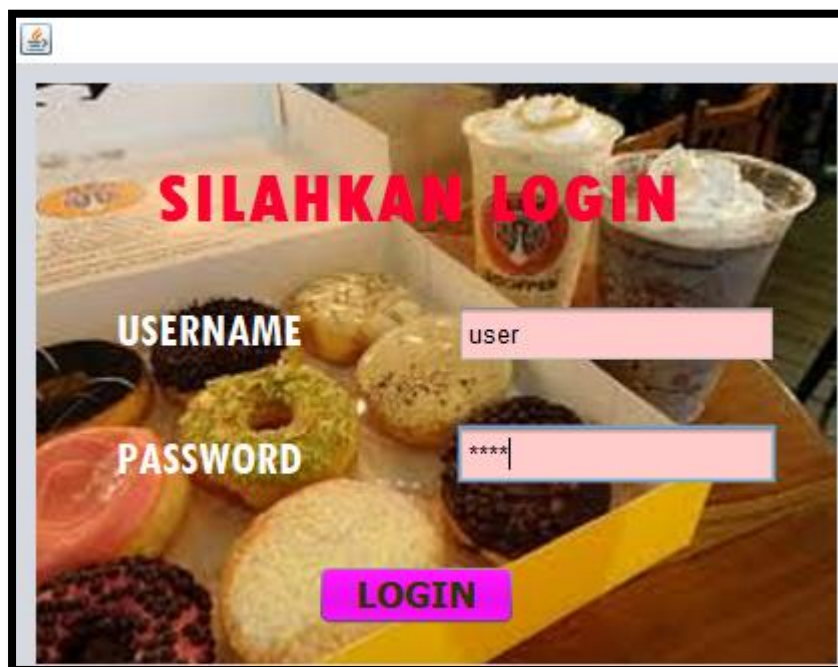
Keterangan gambar diatas ialah :

Untuk membuat design pada GUI ita menggunakan komponen dari Pallet pada Swing controls. Berikut penjelasannya

1. Pilih Label -> digunakan pada tulisan Silahkan Login, Username, Password.
2. Pilih TextField -> digunakan pada kotak disebelah kanan pada username.
3. Pillih PassworField -> digunakan pada kotak disebelah kanan pada Password.
4. Pilih Button -> digunakan pada "login".

Berikut gambar ketika Class JFrame Login DI run

Ketika di run pengguna harus memasukkan username dan password sesuai dengan yang diset. Username dan password telah di set pada List User.



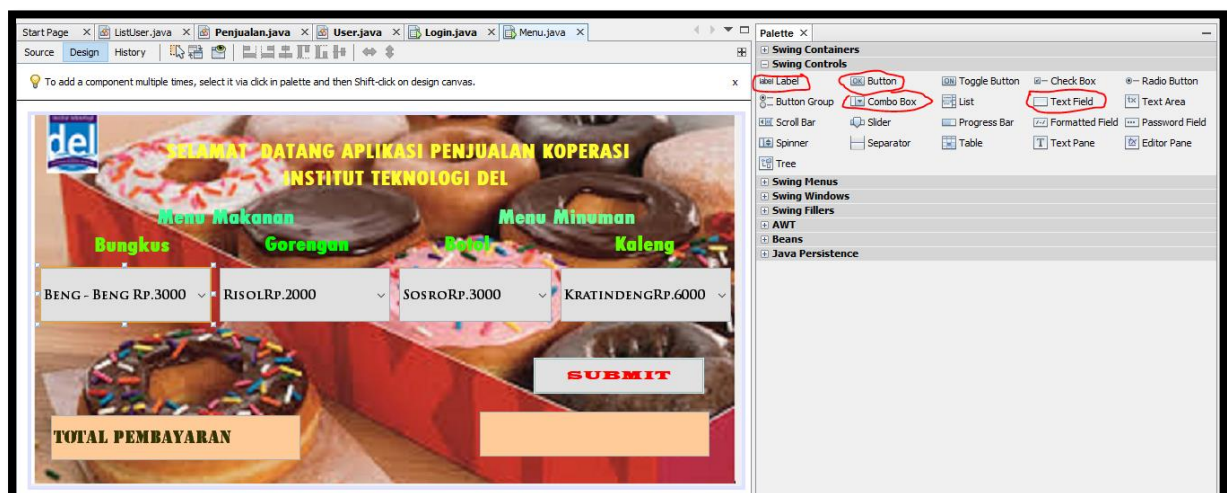
Kemudian berikut adalah Desain GUI pada class Menu (Client)

Untuk membuat design pada GUI kita menggunakan komponen dari Pallet pada Swing controls. Berikut penjelasannya

1. Pilih Label -> digunakan pada bagian judul, Menu Makanan dan Menu Minuman, bungkus, gorengan, Botol kaleng.
2. Pilih TextField -> digunakan pada kotak disebelah kanan total pembayaran
3. Pilih Button -> digunakan pada "submit".

Berikut adalah potongan code pada class login .

```
89 private void jLoginActionPerformed(java.awt.event.ActionEvent evt) {  
90  
91     String username = txtUsername.getText();  
92     String pass = txtPassword.getText();  
93  
94     if (listUser.isAuth(username, pass)) {  
95         this.dispose();  
96         new Menu().setVisible(true);  
97         txtUsername.setText("");  
98         txtPassword.setText("");  
99  
100    } else {  
101        JOptionPane.showMessageDialog(rootPane, "Password Salah");  
102        txtPassword.setText("");  
103        txtPassword.requestFocus();  
104    }  
105  
106 }
```



Ketika di run maka akan muncul seperti gambar dibawah ini.



Gambar dibawah ketika pada saat di run dan mulai memilih menu makanan apa saja yang di beli. Dan pada gambar ini program sudah terhubung dengan server si penjual.



Berikut potongan code pada JbtSubmit.


```

private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //mengambil data dari combo button
    String bungkus = cb_Bungkus.getSelectedItem().toString();
    String gorengan = cb_gorengan.getSelectedItem().toString();
    String botol = cb_botol.getSelectedItem().toString();
    String kaleng = cb_Kaleng.getSelectedItem().toString();

    try { //dikumpulkan dari yang sudah diambil dari atas

        String all = bungkus + "\n" + gorengan + "\n" + botol + "\n" + kaleng;
        oos.writeObject(all); //networkingnya untuk transaksi
        System.out.println(bungkus);
        System.out.println(gorengan);
        System.out.println(botol);
        System.out.println(kaleng);
        String message = (String) ois.readObject(); //kasih balasan untuk penjumlahan
        hargaTotal = message;
        txtTotal.setText(message);
        System.out.println(message);

    } catch (IOException ex) {
        Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Berikut adalah tampilan setelah client menerima balasan dari server berupa total pembayaran dari server.



Program create pada server

Proses yang terjadi pada bagian PC2/Server. Server berfungsi untuk menerima data pemesanan dan menyimpan transaksi yang dilakukan dengan client /PC1.

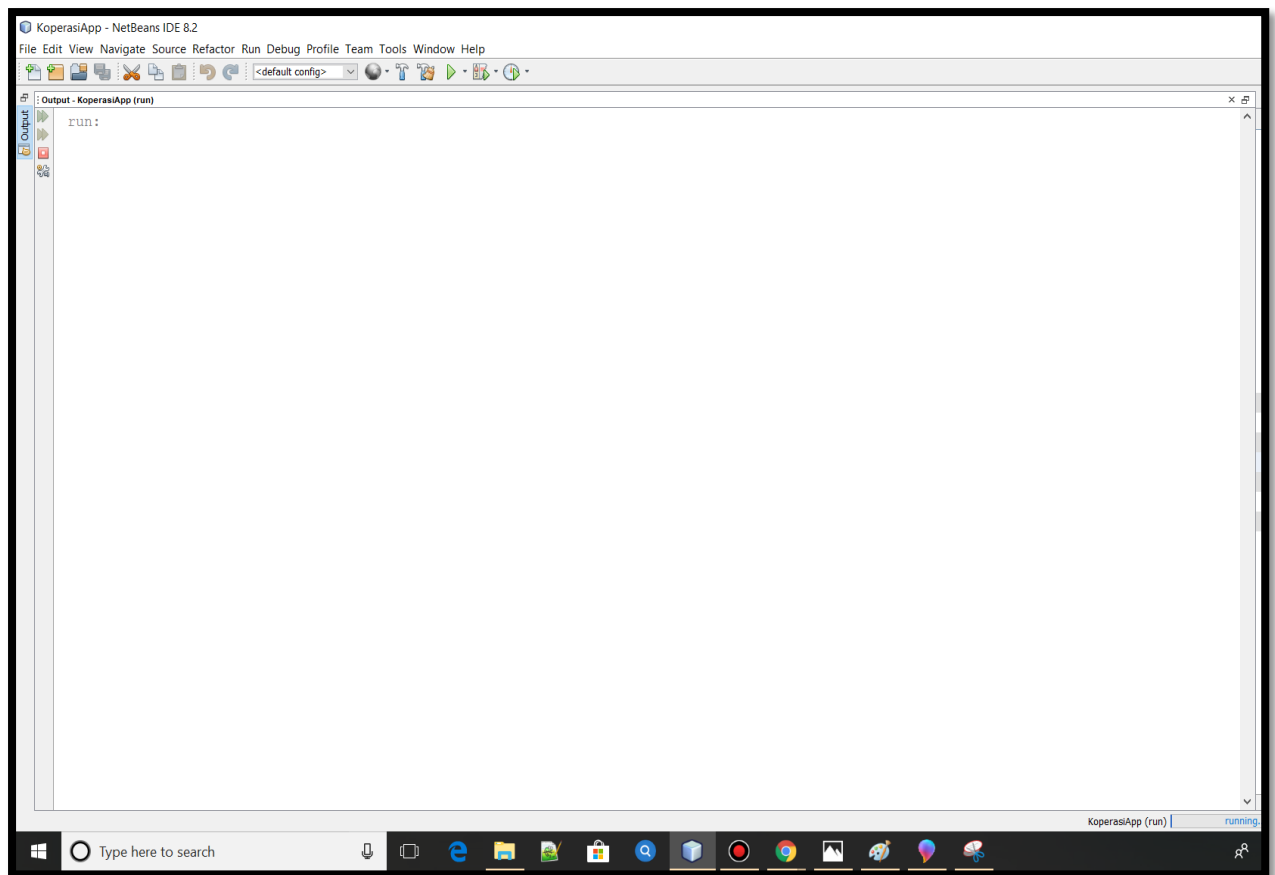
Berikut proses yang terjadi pada server.

1. Untuk menjalankan aplikasi ini, pertama sekali server login terlebih dahulu. File login dijalankan kemudian akan muncul form login. Isi form login dengan username dan password yang telah diset. Username dan password KoperasiApp adalah admin. Kodingan untuk form login server sendiri sama dengan kodingan untuk form login client.



Gbr1. Login Server

2. Bagian yang pertama kali men-run adalah bagian server, kemudian server akan menunggu pesan dari client (PC1). Saat menunggu layar kosong seperti pada gambar dibawah yang akan terjadi pada server.



Gbr.2 Tampilan GUI pada bagian server saat menunggu pesanan dari client/PC1.

```

public Menu() {
    try {

        sl = new ServerSocket(1342);
        s = sl.accept();
        System.out.println("cek");
        ois = new ObjectInputStream(s.getInputStream());
        oos = new ObjectOutputStream(s.getOutputStream());

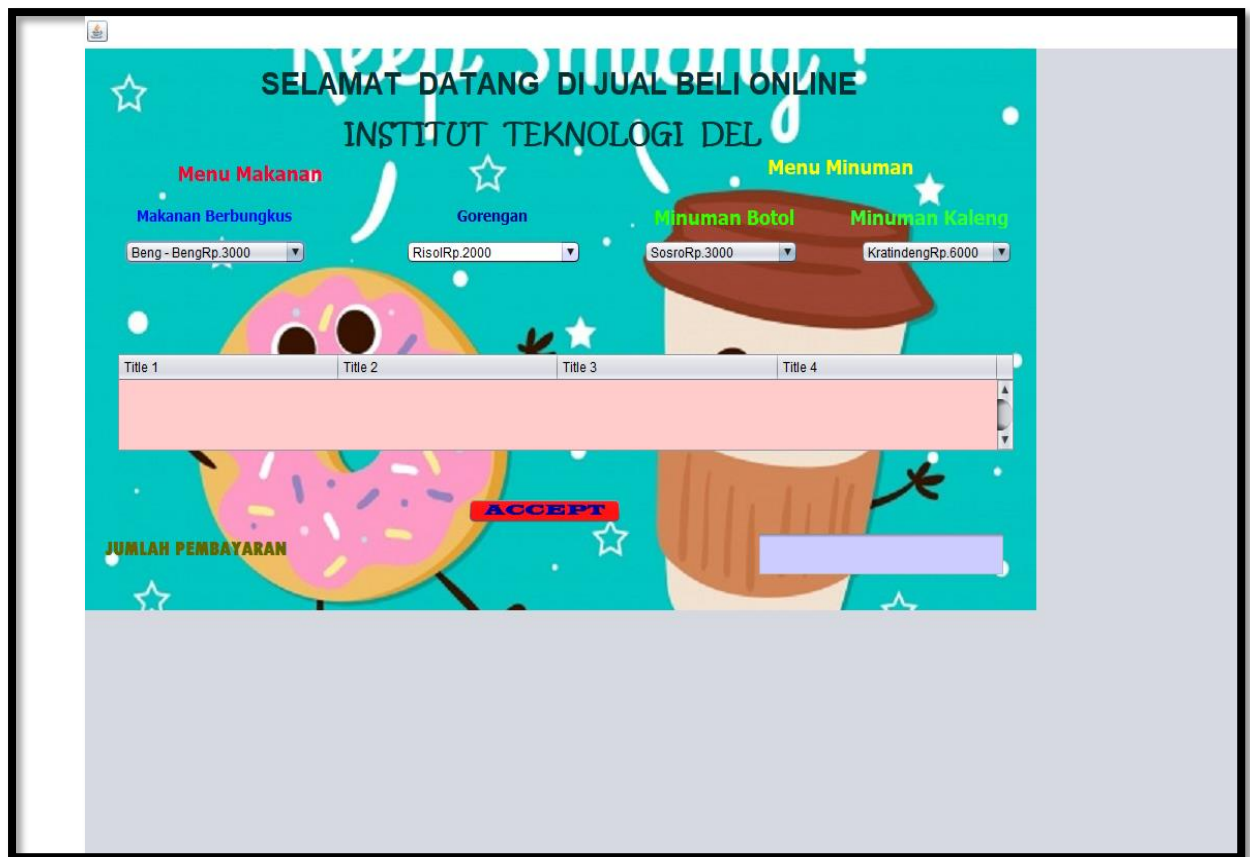
    } catch (IOException ex) {
        System.out.println("error");
    }
    initComponents();
}

public String totalHarga = null;
FileOutputStream fileout = null;

```

Gbr 2b. Code program yang digunakan untuk mengecek apakah server dan client berada pada koneksi yang sama dandapat melakukan transaksi.

3. Saat client telah melakukan pesanan atau bagia client ada pada kondisi run, Tampilan GUI pada server akan muncul tetapi data masih kosong Karena server belum menerima atau mengklik tombol accept. Seperti pada gambar 3.



Gbr.3 Tampilan server saat pesanan/data belum diterima (accept)

4. Ketika client melakukan pemesanan untuk pertama kalinya dan server telah meng-accept data maka server akan terisi oleh data pesanan client, seperti pada gambar 3. Table pada server akan terisi

data dari client.

The screenshot shows a web application interface for 'INSTITUT TEKNOLOGI DEL'. The header includes the text 'SELAMAT DATANG DI JUAL BELI ONLINE' and 'INSTITUT TEKNOLOGI DEL'. The interface is divided into two main sections: 'Menu Makanan' (Food Menu) and 'Menu Minuman' (Drink Menu). The 'Menu Makanan' section includes 'Makanan Berbungkus' (Packaged Food) with a dropdown menu showing 'Beng - BengRp.3000' and 'Gorengan' (Fried Food) with a dropdown menu showing 'RisolRp.2000'. The 'Menu Minuman' section includes 'Minuman Botol' (Bottle Drink) with a dropdown menu showing 'SosroRp.3000' and 'Minuman Kaleng' (Canned Drink) with a dropdown menu showing 'KratindengRp.6000'. Below the menu items, there is a table with four columns: 'bungkus', 'gorengan', 'botol', and 'kaleng'. The table contains the following data: 'Beng - Beng Rp.3000', 'RisolRp.2000', 'SosroRp.3000', and 'KratindengRp.6000'. A red 'ACCEPT' button is visible. At the bottom, the 'JUMLAH PEMBAYARAN' (Total Payment) is displayed as '14000'.

bungkus	gorengan	botol	kaleng
Beng - Beng Rp.3000	RisolRp.2000	SosroRp.3000	KratindengRp.6000

JUMLAH PEMBAYARAN: 14000

Gbr4a. Tampilan pada server saat data/pesanan telah diterima untuk satu kali pemesanan.

```

public void tampilkandata() {
    try {
        String log = "";
        fileout = new FileOutputStream("log.txt");
        DefaultTableModel tbjual = new DefaultTableModel();
        // tbjual.addTableModelListener(tabelTransaksi);
        tbjual.addColumn("bungkus");
        tbjual.addColumn("gorengan");
        tbjual.addColumn("botol");
        tbjual.addColumn("kaleng");
        tbjual.setRowCount(0);

        for (int i = 0; i < ListPenjualan.listPenjualan.size(); i++) { //membaca
            tbjual.addRow(new Object[]{
                ListPenjualan.listPenjualan.get(i).getBungkus(),
                ListPenjualan.listPenjualan.get(i).getGoreng(),
                ListPenjualan.listPenjualan.get(i).getBotol(),
                ListPenjualan.listPenjualan.get(i).getKaleng()
            });

            log = ListPenjualan.listPenjualan.get(i).getBungkus() + "|"
                + ListPenjualan.listPenjualan.get(i).getGoreng() + "|"
                + ListPenjualan.listPenjualan.get(i).getBotol() + "|"
                + ListPenjualan.listPenjualan.get(i).getKaleng() + "\r\n";
            fileout.write(log.getBytes());
        }
    }

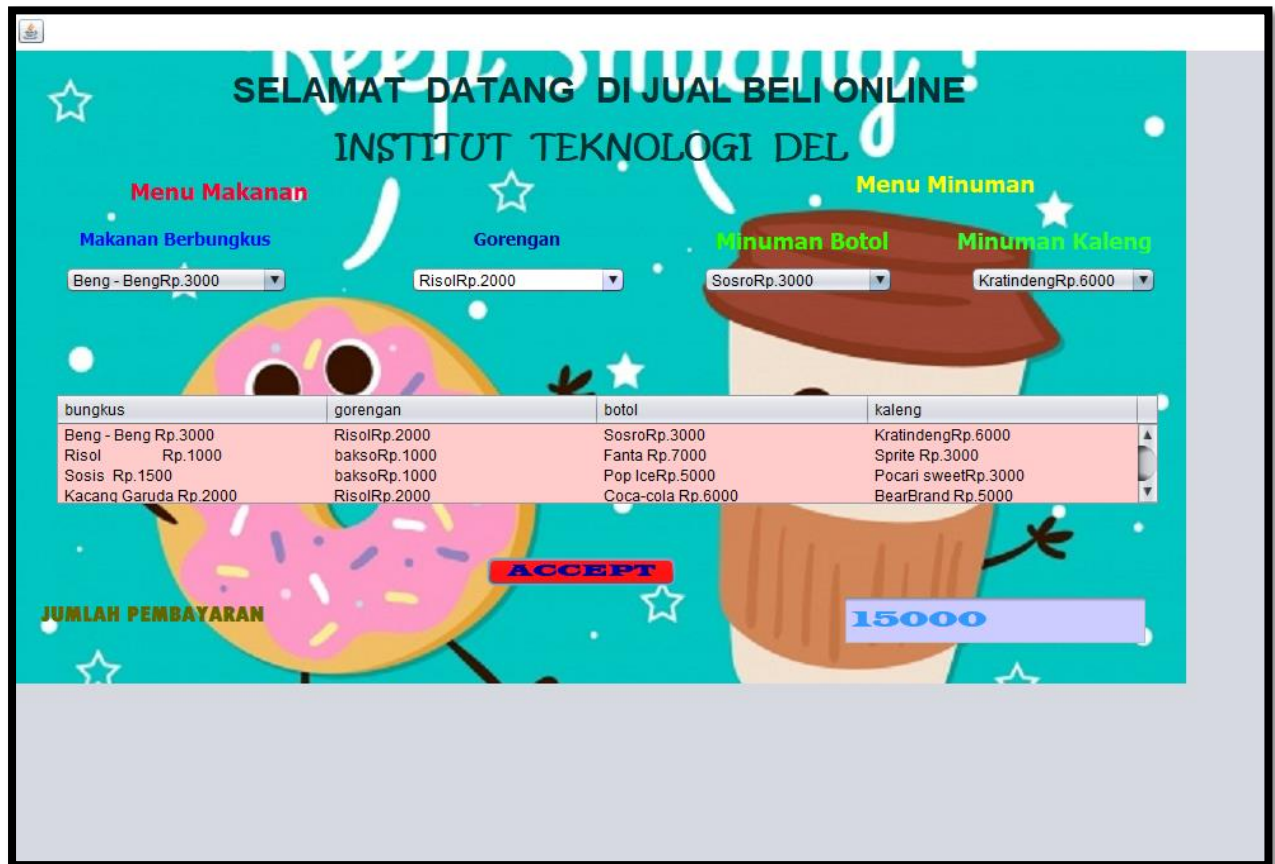
    Penjualan newPenjualan = new Penjualan(bungkus, gorengan, botol, kaleng);
    ListPenjualan.listPenjualan.add(newPenjualan);

    this.tampilkandata();
}

```

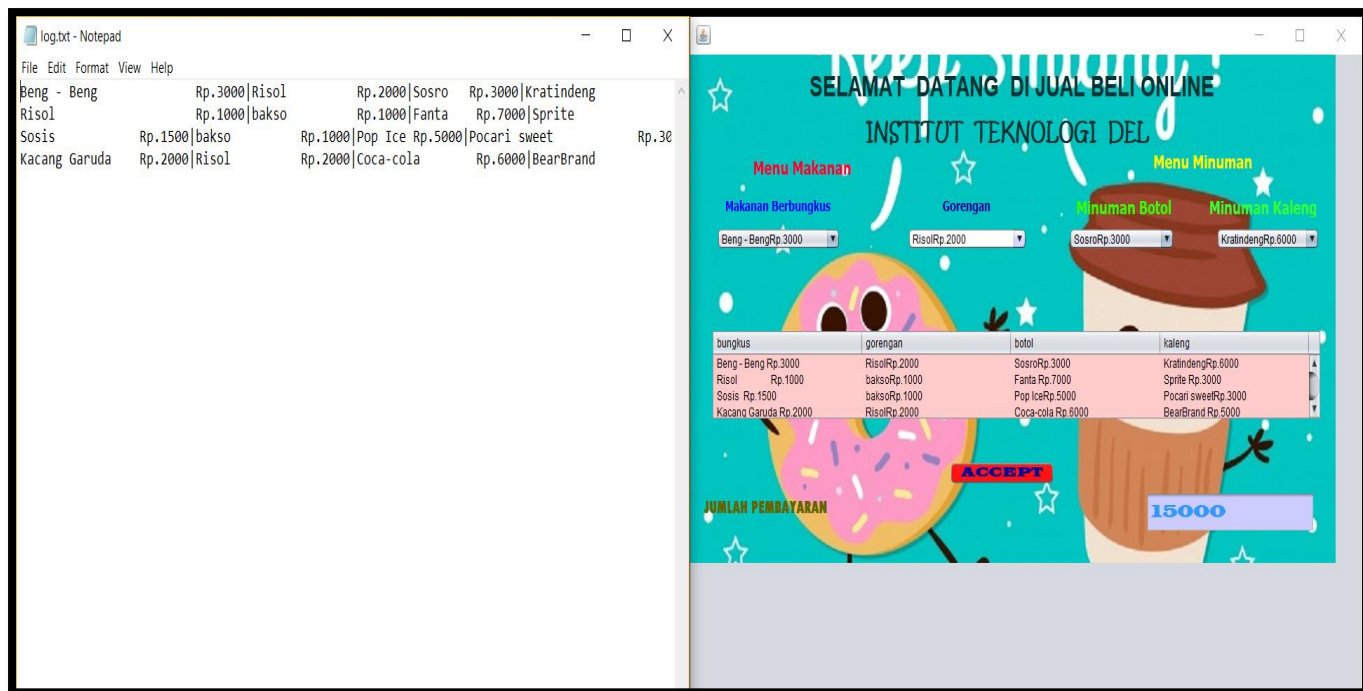
Gbr4b. Code program untuk menampilkan data dari si client

5. Ketika client /PC1 telah melakukan beberapa kali pemesanan dan server menerima semuanya, data pesanan akan tampil pada table server.



Gbr5. Tampilan server saat telah menerima beberapa pesanan

6. Data pemesanan tersebut akan disimpan pada sebuah file txt. Pada proyek ini file txt bernama log.txt. Seperti pada gambar dibawah. Ketika kita membuka file log.txt yang tersimpan dalam folder KoperasiApp, Data-data pemesanan yang telah terjadi akan tersimpan pada file tersebut.



Gbr.6a Tampilan ketika data dari client diterima dan disimpan pada file log.txt

Kode program gambar6b. dutuliskan sebelum kode program kondisi membaca list dari penjualan,

```
public void tampilkandata() {
    try {
        String log = "";
        fileout = new FileOutputStream("log.txt");
        DefaultTableModel tbjual = new DefaultTableModel();
```

Gbr6b.Code program unruk menyimpan transaksi di file log.txt

Kode program gambar6b. dutuliskan setelah kode program kondisi membaca list dari penjualan.

```

        fileout.write(log.getBytes());
    }
    System.out.println(log);
    tabelTransaksi.setModel(tbjual);
    Harga.setText(totalHarga); //menampilkan harga
// <editor-fold defaultstate="collapsed" desc="Generated Code">
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);

    } catch (IOException ex) {
        Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);

    } finally {
        if (fileout != null) {

            try {
                fileout.close(); //telah selesai penulisan file
            } catch (IOException ex) {
                System.out.println(ex);
            }
        }
    }
}

```

Gbr6c.Code program unruk menyimpan transaksi di file log.txt