

LAPORAN TUGAS BESAR
MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK
“YA-HOODIE”



Disusun oleh:

MUHAMMAD ARYA FIKRIANSYAH	1301204066
SATRIA AJI PERMANA SIWI	1301204209
RAIHAN ATSAL HAFIZH	1301204485
ALIF FAIDHIL AHMAD	1301204141
FACHRUL FEBRIANA	1301204407

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2023

DAFTAR ISI

DAFTAR ISI	i
1. PENDAHULUAN	1
1.1. LATAR BELAKANG	1
1.2. BATASAN MASALAH	1
1.3. TUJUAN	1
2. PEMBAHASAN	2
2.1. LANDASAN TEORI	2
2.1.1. JAVA	2
2.1.2. NETBEANS	2
2.1.3. MYSQL	3
3. ANALISIS APLIKASI	5
3.1. GAMBARAN UMUM APLIKASI	5
3.2. PROSES BISNIS	5
3.3. USE CASE DIAGRAM	6
3.4. CLASS DIAGRAM	7
3.5. DAFTAR CLASS YANG DIGUNAKAN	7
3.5.1. Model	7
3.5.1.1. Admin.java	7
3.5.1.2. Hoodie.java	8
3.5.1.3. User.java	8
3.5.1.4. Customer.java	8
3.5.1.5. Pembelian.java	8
3.5.1.6. Pembelian_produk.java	8
3.5.2. View	8
3.5.2.1. AdminForm.java	8
3.5.2.2. Checkout.java	8
3.5.2.3. DataPembelian.java	9
3.5.2.4. Login.java	9
3.5.2.5. PembelianUser.java	9
3.5.2.6. ProductDetailForm.java	9
3.5.2.7. ProductForm.java	9
3.5.2.8. Register.java	9
3.5.3. Controller	9
3.5.3.1. Koneksi.java	9
3.6. DAFTAR ANGGOTA DAN PERANNYA	10
3.7. LAMPIRAN SOURCE CODE	10
3.7.1. Model	10
3.7.1.1. Admin.java	10

3.7.1.2. Hoodie.java	11
3.7.1.3. User.java	12
3.7.1.4. Customer.java	13
3.7.1.5. Pembelian.java	14
3.7.1.6. Pembelian_produk.java	14
3.7.2. View	14
3.7.2.1. AdminForm.java	15
3.7.2.2. Checkout.java	17
3.7.2.3. DataPembelian.java	19
3.7.2.4. Login.java	21
3.7.2.5. PembelianUser.java	22
3.7.2.6. ProductDetailForm.java	23
3.7.2.7. ProductForm.java	24
3.7.2.8. Register.java	25
3.7.3. Controller	26
3.7.3.1. Koneksi.java	26
3.8. LAMPIRAN APLIKASI	26
3.8.1. Login	26
3.8.2. Register	27
3.8.3. Hoodie List	28
3.8.4. Detail Produk	29
3.8.5. Checkout	30
3.8.6. Riwayat Pembelian	31
3.8.7. Tampilan awal admin	32
3.8.8. Data Pembelian	33
4. PENUTUP	35
4.1. KESIMPULAN	35
4.2. SARAN	35
DAFTAR PUSTAKA	36

1. PENDAHULUAN

1.1. LATAR BELAKANG

Pemrograman berorientasi objek (Object-Oriented Programming/OOP) merupakan salah satu paradigma pemrograman yang banyak digunakan saat ini. Paradigma ini mengutamakan pemahaman tentang objek-objek yang ada dalam sistem yang akan dibuat, serta hubungan dan interaksi antar objek tersebut. Pemrograman berorientasi objek memiliki beberapa kelebihan dibandingkan dengan paradigma pemrograman lain, seperti lebih mudah dalam pemeliharaan dan pengembangan kode, lebih mudah dalam pemahaman sistem yang akan dibuat, dan lebih fleksibel dalam pengembangan sistem.

Laporan ini akan memuat deskripsi dari aplikasi atau sistem yang telah dibuat, serta penjelasan mengenai proses pembuatan dan penerapan konsep-konsep OOP yang telah dipelajari. Selain itu, laporan ini juga akan menjelaskan permasalahan yang dihadapi selama proses pembuatan, serta solusi yang telah diterapkan untuk menyelesaikan masalah tersebut.

1.2. BATASAN MASALAH

Pada pembuatan laporan ini, kami membatasi masalah sebagai berikut:

1. Customer dapat login dan logout
2. Admin dapat login dan logout
3. Customer dapat melihat daftar hoodie yang dijual
4. Customer dapat melakukan pembelian hoodie
5. Customer dapat melihat riwayat pembelian
6. Admin dapat menambah data hoodie
7. Admin dapat mengupdate stock hoodie
8. Admin dapat mendelete data hoodie
9. Admin dapat melihat data pembelian customer
10. Admin dapat mengupdate status pembelian customer

1.3. TUJUAN

Tujuan dari pengerjaan tugas ini adalah untuk memahami dan mempraktikkan konsep-konsep dasar OOP seperti inheritance, polymorphism, encapsulation, dan abstraction. Selain itu, tugas ini juga bertujuan untuk mengaplikasikan konsep-konsep tersebut dalam pembuatan sebuah aplikasi atau sistem yang sesuai dengan spesifikasi yang diberikan.

2. PEMBAHASAN

2.1. LANDASAN TEORI

2.1.1. JAVA

Java adalah bahasa pemrograman yang dikembangkan oleh James Gosling dan timnya di Sun Microsystems (sekarang merupakan bagian dari Oracle) pada tahun 1995. Java adalah bahasa pemrograman yang didesain untuk dapat berjalan diberbagai sistem operasi, seperti Windows, Mac, dan Linux.

Java merupakan bahasa pemrograman yang menggunakan paradigma pemrograman berorientasi objek (Object-Oriented Programming/OOP). Konsep dasar OOP yang terdapat pada Java adalah inheritance, polymorphism, encapsulation, dan abstraction.

Inheritance merupakan mekanisme dimana sebuah class dapat menurunkan sifat-sifat dan metode-metode yang dimilikinya kepada class lain. Hal ini berguna untuk menghindari duplikasi kode dan memudahkan proses pemeliharaan dan pengembangan kode.

Polymorphism merupakan mekanisme dimana sebuah objek dapat memiliki lebih dari satu bentuk atau comportement yang berbeda. Hal ini berguna untuk membuat sebuah kode yang lebih fleksibel dan reusable.

Encapsulation merupakan mekanisme dimana data dan method yang terkait dengan data tersebut dikelompokkan ke dalam satu unit yang disebut class. Hal ini berguna untuk menyembunyikan rincian implementasi dari user, sehingga user hanya perlu memahami interface dari class tersebut tanpa perlu tahu bagaimana class tersebut diimplementasikan.

Abstraction merupakan mekanisme dimana sebuah class hanya menunjukkan sifat-sifat yang paling esensial dari objek yang akan dibuat, serta menyembunyikan rincian-rincian yang tidak perlu diketahui oleh user. Hal ini berguna untuk membuat sebuah kode yang lebih mudah dipahami dan dikelola.

Java juga menyediakan fitur-fitur seperti exception handling, thread, dan generics yang berguna untuk menangani masalah-masalah yang terkait dengan error, parallelisme, dan reusable code.

Java dapat digunakan untuk membuat aplikasi-aplikasi desktop, web, mobile, dan game. Java juga merupakan bahasa pemrograman yang populer digunakan dalam bidang-bidang seperti data mining, big data, dan internet of things (IoT).

2.1.2. NETBEANS

NetBeans adalah sebuah Integrated Development Environment (IDE) yang dikembangkan oleh Oracle untuk mengembangkan aplikasi-aplikasi berbasis Java. NetBeans juga mendukung

pengembangan aplikasi-aplikasi berbasis lain seperti PHP, C/C++, dan HTML5.

NetBeans menyediakan berbagai fitur yang berguna untuk membantu proses pengembangan aplikasi, seperti editor kode yang powerful, debugger, profiler, dan integrasi dengan version control system seperti Git. NetBeans juga memiliki interface yang intuitif dan mudah digunakan, sehingga memudahkan bagi para developer untuk mengembangkan aplikasi dengan lebih cepat dan efisien.

NetBeans juga memiliki fitur-fitur yang berguna untuk meningkatkan produktivitas developer, seperti code completion, refactoring, dan quick fix. Fitur-fitur tersebut akan membantu developer untuk mempercepat proses pembuatan kode dan mengurangi kesalahan-kesalahan yang mungkin terjadi.

Selain itu, NetBeans juga memiliki fitur-fitur yang berguna untuk meningkatkan kualitas kode, seperti static code analysis dan unit testing. Fitur-fitur tersebut akan membantu developer untuk menemukan dan memperbaiki masalah-masalah yang ada pada kode sebelum kode tersebut dijalankan.

NetBeans juga memiliki fitur-fitur yang berguna untuk membantu proses pemeliharaan dan pengembangan aplikasi, seperti integrasi dengan sistem manajemen proyek seperti Jenkins dan integrasi dengan sistem pemetaan kode seperti JIRA. Fitur-fitur tersebut akan membantu developer untuk mengelola proyek dengan lebih efektif dan efisien.

Secara keseluruhan, NetBeans merupakan sebuah IDE yang sangat berguna bagi para developer yang ingin mengembangkan aplikasi-aplikasi berbasis Java dengan lebih cepat, efisien, dan kualitas kode yang tinggi.

2.1.3. MYSQL

MySQL adalah sebuah sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang dikembangkan oleh Oracle Corporation. MySQL merupakan salah satu sistem manajemen basis data yang paling populer dan banyak digunakan saat ini, terutama untuk aplikasi-aplikasi web.

MySQL menggunakan Structured Query Language (SQL) sebagai bahasa untuk mengakses dan memanipulasi data yang tersimpan dalam basis data. SQL merupakan bahasa standar untuk mengakses basis data relasional, sehingga MySQL dapat dengan mudah diintegrasikan dengan aplikasi-aplikasi lain yang juga menggunakan SQL.

MySQL menyediakan fitur-fitur yang berguna untuk mengelola dan mengakses data dalam jumlah yang besar, seperti indexing, locking, dan transactions. Fitur-fitur tersebut akan membantu MySQL

untuk menjalankan query dengan lebih cepat dan mengelola data dengan lebih aman.

MySQL juga memiliki fitur-fitur yang berguna untuk meningkatkan keamanan data, seperti autentikasi pengguna, enkripsi data, dan hak akses pengguna yang terkontrol. Fitur-fitur tersebut akan membantu MySQL untuk menjaga keamanan data yang tersimpan dalam basis datanya.

MySQL juga memiliki fitur-fitur yang berguna untuk membantu proses pemeliharaan dan pengembangan basis data, seperti backup dan restore data, import dan export data, dan perubahan schema basis data. Fitur-fitur tersebut akan membantu MySQL untuk mengelola basis data dengan lebih efektif dan efisien.

Secara keseluruhan, MySQL merupakan sebuah sistem manajemen basis data yang sangat berguna bagi para developer yang ingin mengelola dan mengakses data dalam jumlah yang besar dengan kecepatan yang tinggi, keamanan yang tinggi, dan fleksibilitas yang tinggi.

3. ANALISIS APLIKASI

3.1. GAMBARAN UMUM APLIKASI

YA-HOODIE merupakan sebuah perangkat lunak berbasis desktop yang bergerak dalam bidang penjualan hoodie. Perangkat lunak ini adalah pengganti sistem yang sudah ada, yaitu sistem penjualan hoodie di toko-toko pada umumnya. Perangkat lunak ini diharapkan dapat memudahkan pembeli dalam membeli hoodie yang ingin dibeli, tidak perlu datang ke pusat perbelanjaan atau toko hoodie, menghemat waktu pembeli, dan juga menghemat anggaran penjual untuk menyewa tempat berjualan hoodie. Pembeli dapat memilih barang-barang yang diinginkan, melakukan pembayaran via COD (Cash on Delivery), admin dapat mengelola data hoodie, menerima pesanan, dan melihat data pembelian.

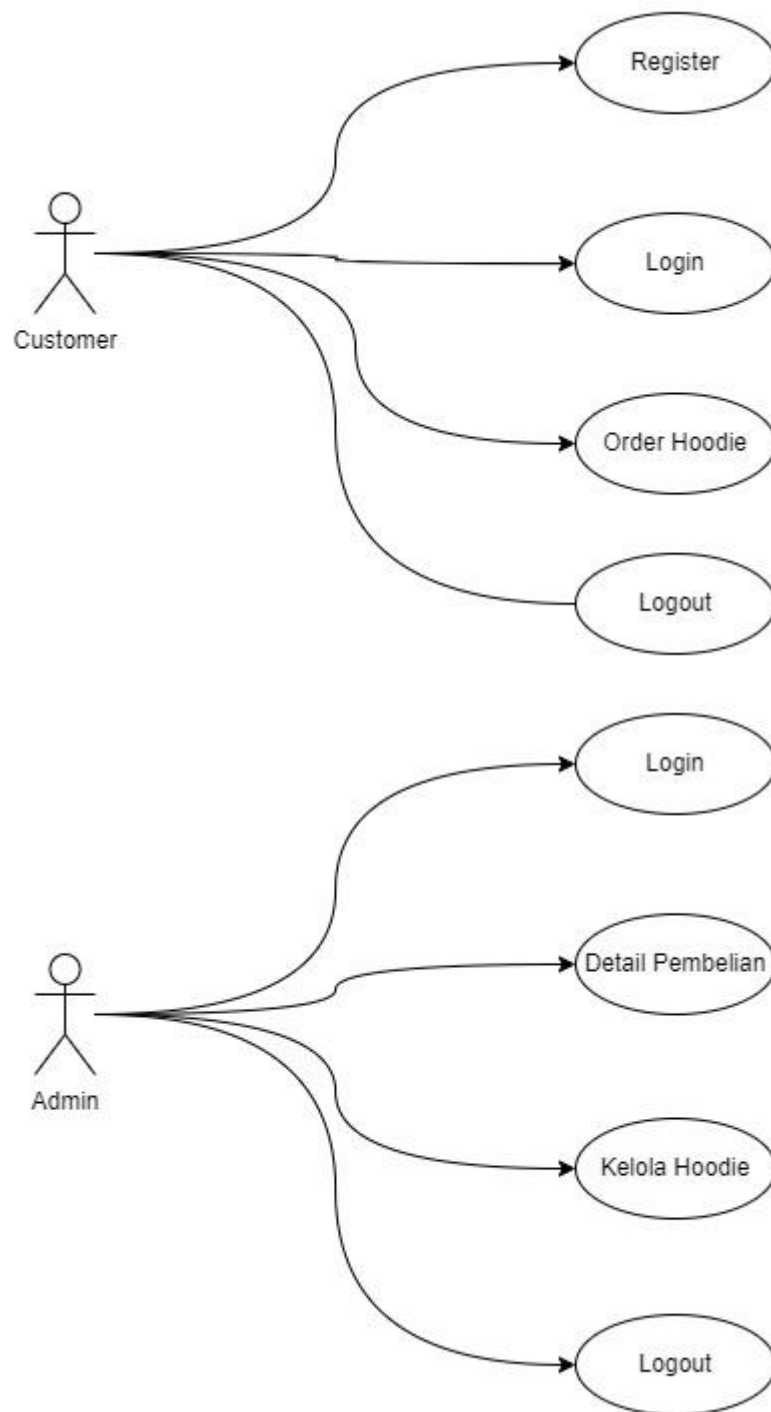
3.2. PROSES BISNIS

Berikut ini merupakan proses bisnis untuk aplikasi “Ya-Hoodie” penjualan hoodie online dengan sistem pembayaran cash on delivery:

1. Pemasaran: Admin melakukan pemasaran terhadap aplikasi dan produk hoodie yang dijual melalui media sosial, iklan online, dan promosi lainnya.
2. Penjualan: Customer mengunjungi aplikasi dan memilih hoodie yang diinginkan. Customer juga memasukkan informasi pengiriman dan memilih metode pembayaran cash on delivery.
3. Pemrosesan pesanan: Admin menerima pesanan dan melakukan pemrosesan pesanan, termasuk pengiriman pesanan ke alamat yang ditentukan oleh customer.
4. Pengiriman: Admin mengirimkan pesanan ke alamat yang ditentukan oleh customer.
5. Pembayaran: Customer membayar hoodie yang dibeli saat pengiriman dengan cash on delivery.
6. Pelacakan pesanan: Customer dapat melihat status pesanan melalui aplikasi.
7. Analisa data: Admin mengumpulkan data dari penjualan yang dilakukan oleh customer untuk menganalisa tren dan kebutuhan customer, serta meningkatkan produk dan layanan yang ditawarkan.

Secara keseluruhan, proses bisnis untuk aplikasi “Ya-Hoodie” penjualan hoodie online dengan sistem pembayaran cash on delivery meliputi tahapan-tahapan seperti pemasaran, penjualan, pemrosesan pesanan, pengiriman, pembayaran, pelacakan pesanan, dan analisa data. Setiap tahapan tersebut merupakan bagian penting dari proses bisnis yang harus dilakukan secara efektif dan efisien agar aplikasi dapat berjalan dengan lancar dan menghasilkan keuntungan.

3.3. USE CASE DIAGRAM



Gambar 1. Use Case Diagram

3.4. CLASS DIAGRAM



Gambar 2. Class Diagram

3.5. DAFTAR CLASS YANG DIGUNAKAN

3.5.1. Model

3.5.1.1. Admin.java

Pada model admin.java ini memiliki method logout yang berguna untuk admin keluar dari form admin.

3.5.1.2. Hoodie.java

Pada model hoodie.java ini memiliki beberapa method diantaranya addProduct, kodeCheck, deleteProduct, dan updateStock. addProduct berguna untuk menginputkan data hoodie baru kedalam database. KodeCheck berguna untuk pengecekan kodeHoodie yang terdapat dalam database. DeleteProduct berguna untuk menghapus data hoodie dari database. UpdateStock berguna untuk memperbarui stock berdasarkan kodeHoodienya.

3.5.1.3. User.java

Pada model user.java ini memiliki method addUser, loginCheck, getID, route. LoginCheck berguna untuk pengecekan username dan password yang diinputkan oleh user. GetID berguna untuk mendapatkan id user. Route berguna untuk menjuruskan user apakah dia menjadi pembeli atau admin.

3.5.1.4. Customer.java

Pada model customer.java ini memiliki method AddCustomer berguna untuk form registrasi, menginputkan data baru berupa username, password, nama lengkap, dan level. logout yang berguna untuk customer keluar dari form admin.

3.5.1.5. Pembelian.java

Pada model pembelian.java ini memiliki method beli. Method beli berguna untuk pembelian, method ini meminta id pembelian, id user, tanggal beli, total beli, status, dan alamat.

3.5.1.6. Pembelian_produk.java

Pada model pembelian_produk.java ini memiliki method addProduct, method ini meminta id beli produk, id pembelian, dan kodeHoodie.

3.5.2. View

3.5.2.1. AdminForm.java

Pada adminForm.java ini menampilkan tabel untuk data hoodie, field untuk kode hoodie, field untuk nama hoodie, box untuk tipe, box untuk ukuran, field untuk harga, field untuk stock, field untuk upload gambar, field untuk isi kode, field untuk ubah stock, button tambah produk, button data pembelian, button refresh, button hapus data, button edit stock, button update, dan button delete.

3.5.2.2. Checkout.java

Pada checkout.java ini menampilkan nama produk, jumlah, tanggal transaksi, total pembelian, field alamat pembelian dan pembayaran, button selesai, dan button lihat produk lainnya.

- 3.5.2.3. `DataPembelian.java`
Pada `DataPembelian.java` ini menampilkan tabel untuk data pembelian, field isi tanggal pembelian, field masukkan status, button kembali, button refresh, button edit status, dan button update status.
- 3.5.2.4. `Login.java`
Pada `Login.java` ini menampilkan field username dan field password untuk melakukan login masuk ke dalam sistem. Juga terdapat button login dan button register jika belum mempunyai akun untuk login.
- 3.5.2.5. `PembelianUser.java`
Pada `pembelianUser.java` ini menampilkan tabel yang berisi data pembelian dari user.
- 3.5.2.6. `ProductDetailForm.java`
Pada `productdetailForm` ini menampilkan detail hoodie yang dibeli berupa nama hoodie, harga, tipe ukuran, kuantiti, dan alamat pengirimannya.
- 3.5.2.7. `ProductForm.java`
Pada `productForm.java` ini menampilkan list hoodie, dan terdapat button detail product yang nantinya bisa masuk ke `productDetailForm.java` dan button riwayat pembelian yang nantinya bisa masuk ke `pembelianUser.java`.
- 3.5.2.8. `Register.java`
Pada `register.java` ini meminta user untuk memasukan username, password, dan nama lengkap, agar diinputkan ke database.

3.5.3. Controller

- 3.5.3.1. `Koneksi.java`
Pada `Koneksi.java` ini bertujuan untuk menghubungkan dengan database SQL phpMyAdmin melalui package controller.

3.6. DAFTAR ANGGOTA DAN PERANNYA

No.	Nama Lengkap	NIM	Peran
1.	Muhammad Arya Fikriansyah	1301204066	Pengelola koneksi dan database, back-end developer
2.	Satria Aji Permana Siwi	1301204209	Laporan
3.	Raihan Atsal Hafizh	1301204485	Perancangan design, front-end developer
4.	Alif Faidhil Ahmad	1301204141	Laporan
5.	Fachrul Febriana	1301204407	Laporan

3.7. LAMPIRAN SOURCE CODE

3.7.1. Model

3.7.1.1. Admin.java

```
6 package model;
7
8 public class Admin extends User {
9     public void logout() {
10         id = "";
11         usern = "";
12         pass = "";
13         System.exit(0);
14     }
15 }
```

File ini digunakan untuk user admin menginputkan idAdmin, username, dan password. id, usern, dan pass admin telah ditentukan pada database mysql. User admin hanya perlu melengkapi form sesuai dengan yang ada pada database hoodie_db. User admin dapat logout jika telah selesai melakukan kelola hoodie sesuai dengan tugasnya.

3.7.1.2. Hoodie.java

```
1 package model;
2
3 import java.awt.HeadlessException;
4 import javax.swing.JOptionPane;
5 import java.sql.Connection;
6 import java.sql.SQLException;
7
8 public class Hoodie {
9     String kode;
10    int id, harga, stok;
11    String nama, tipe, ukuran, path;
12
13    public void addProduct(String kode, int id, String nama, String tipe, String ukuran, int harga, int stok, String path) {
14        String level = "admin";
15        try {
16            String sql = "INSERT INTO `hoodie`(`kodehoodie`, `id`, `nama`, `tipe`, `ukuran`, `harga`, `stok`, `gambar`) VALUES "
17                + "(" + kode + ", " + id + ", " + nama + ", " + tipe + ", " + ukuran + ", " + harga + ", " + stok + ", " + path + ")";
18            java.sql.Connection conn = (java.sql.Connection) controller.koneksi.koneksiIDB();
19            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
20            pst.execute();
21            JOptionPane.showMessageDialog(null, "Data Berhasil Ditambahkan");
22        } catch (SQLException | HeadlessException e) {
23            JOptionPane.showMessageDialog(null, e);
24        }
25    }
26
27    public boolean kodeCheck(String kode) {
28        boolean res = true;
29        try {
30            Connection conn = (Connection) controller.koneksi.koneksiIDB();
31            java.sql.Statement stm = conn.createStatement();
32            java.sql.ResultSet rs = stm.executeQuery("select * from hoodie");
33            while (rs.next()) {
34                String kh = rs.getString("kodehoodie");
35                if (kode.equals(kh)) {
36                    res = true;
37                    break;
38                } else {
39                    res = false;
40                }
41            }
42        } catch (SQLException | HeadlessException e) {
43        }
44        return res;
45    }
46
47    public void deleteProduct(String kode) {
48        try {
49            String sql = "DELETE FROM `hoodie` WHERE `kodehoodie` = "
50                + "(" + kode + ")";
51            java.sql.Connection conn = (java.sql.Connection) controller.koneksi.koneksiIDB();
52            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
53            pst.execute();
54            JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");
55        } catch (SQLException | HeadlessException e) {
56            JOptionPane.showMessageDialog(null, e);
57        }
58    }
59
60    public void updateStock(String kode, int stok) {
61        try {
62            String sql = "UPDATE `hoodie` SET `stok` = " + stok + " WHERE `kodehoodie` = " + "(" + kode + ")";
63            java.sql.Connection conn = (java.sql.Connection) controller.koneksi.koneksiIDB();
64            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
65            pst.execute();
66            JOptionPane.showMessageDialog(null, "Stok Berhasil Diubah");
67        } catch (SQLException | HeadlessException e) {
68            JOptionPane.showMessageDialog(null, e);
69        }
70    }
71 }
```

File ini digunakan untuk user admin mengelola hoodie. Terdapat addProduct, fungsi add product ini untuk menambahkan data hoodie baru. User admin diminta untuk memasukan kode hoodie, id, nama, tipe ukuran harga stock, dan gambar. Terdapat fungsi kodeCheck yang berguna untuk mengecek kode hoodie. Terdapat fungsi deleteProduct yang berfungsi untuk menghapus data hoodie dari database. Fungsi UpdateStock ini berfungsi untuk memperbarui stock sesuai kode hoodie yang dipilih.

3.7.1.3. User.java

```
1 package model;
2
3 import view.AdminForm;
4 import view.ProductForm;
5 import java.awt.HeadlessException;
6 import javax.swing.JOptionPane;
7 import java.sql.Connection;
8 import java.sql.SQLException;
9
10 public class User {
11     String id="";
12     String level="";
13     String usern;
14     String pass;
15     String fnama;
16
17     public void setId(String id) {
18         this.id = id;
19     }
20
21     public String getLevel() {
22         return level;
23     }
24
25     public void setLevel(String level) {
26         this.level = level;
27     }
28
29     public String getUsern() {
30         return usern;
31     }
32
33     public void setUsern(String usern) {
34         this.usern = usern;
35     }
36
37     public String getPass() {
38         return pass;
39     }
40
41     public void setPass(String pass) {
42         this.pass = pass;
43     }
44
45     public String getFnama() {
46         return fnama;
47     }
48
49     public void setFnama(String fnama) {
50         this.fnama = fnama;
51     }
52
53     public boolean loginCheck(String usern, String pass){
54         boolean res = true;
55         try {
56             Connection conn =(Connection)controller.koneksi.koneksiDB();
57             java.sql.Statement stm = conn.createStatement();
58             java.sql.ResultSet rs = stm.executeQuery("select * from users");
59             while (rs.next())
60             {
61                 // ...
62             }
63         } catch (SQLException e) {
64             // ...
65         }
66     }
67 }
```

```

34         String iduser = rs.getString("id");
35         String us = rs.getString("username");
36         String pw = rs.getString("password");
37         String lv = rs.getString("level");
38         System.out.println(us+" "+pw+" "+lv);
39         if(us.equals(us) && pw.equals(pw)){
40             level = lv;
41             JOptionPane.showMessageDialog(null,"Berhasil Login");
42             res = true;
43             id=iduser;
44             route();
45             break;
46         }
47         else{
48             res = false;
49         }
50     }
51 }
52 catch (SQLException | HeadlessException e) {
53 }
54 return res;
55 }
56
57 public String getID(){
58     return id;
59 }
60
61 public void route() {
62     if(level.equals("pembeli")){
63         ProductForm pf = new ProductForm(id);
64         pf.setVisible(true);
65     }else if(level.equals("admin")){
66         AdminForm a = new AdminForm(id);
67         a.setVisible(true);
68     }
69 }
70 }

```

File ini dapat digunakan untuk user melakukan registrasi akun sebagai pembeli. Pada file ini terdapat fungsi route yang digunakan oleh user (admin dan customer) menampilkan masing-masing halaman sisi admin dan customer.

3.7.1.4. Customer.java

```

4 package model;
5
6 import java.awt.HeadlessException;
7 import java.sql.SQLException;
8 import javax.swing.JOptionPane;
9
10 public class Customer extends User {
11     public void addCustomer(String usern, String pass, String fnama){
12         String level = "pembeli";
13         try {
14             String sql = "INSERT INTO 'users' ('username', 'password', 'nama_lengkap', 'level') VALUES ('"+usern+"','"+pass+"','"+fnama+"','"+level+"')";
15             java.sql.Connection conn = (java.sql.Connection)controller.koneksi.koneksiDB();
16             java.sql.PreparedStatement pst = conn.prepareStatement(sql);
17             pst.execute();
18             JOptionPane.showMessageDialog(null, "Registrasi Berhasil");
19         } catch (SQLException | HeadlessException e) {
20             JOptionPane.showMessageDialog(null, e);
21         }
22     }
23
24     public void logout() {
25         id = "";
26         fnama = "";
27         usern = "";
28         pass = "";
29         System.exit(0);
30     }
31 }
32
33 }

```

File ini digunakan untuk inputan form registrasi dari user customer. Dalam file ini user customer diminta untuk menginputkan id, Fullname atau nama lengkap, username, dan password sesuai yang telah didaftarkan pada form registrasi sebelumnya.

3.7.1.5. Pembelian.java

```
6 package model;
7
8 import java.awt.HeadlessException;
9 import java.sql.SQLException;
10 import java.time.LocalDate;
11 import javax.swing.JOptionPane;
12
13 /**
14  *
15  * @author ASUS
16  */
17 public class pembelian {
18     String id_pembelian_up;
19     String id_user;
20     String status;
21     String alamat_kirim;
22     int totalnya;
23     LocalDate tgl;
24
25     public void beli(String id_pembelian_up, String id_user, LocalDate tgl, int totalnya, String status, String alamat_kirim) {
26         try {
27             String sql = "INSERT INTO 'pembelian' ('id_pembelian', 'id_user', 'tgl_beli', 'total_beli', 'status', 'alamat') VALUES "
28                 + "(" + id_pembelian_up + ", " + Integer.parseInt(id_user) + ", " + tgl + ", " + totalnya + ", " + status + ", " + alamat_kirim + ")";
29             java.sql.Connection conn = (java.sql.Connection) controller.koneksi.koneksiIDB();
30             java.sql.PreparedStatement pst = conn.prepareStatement(sql);
31             pst.execute();
32             JOptionPane.showMessageDialog(null, "Pembelian Berhasil");
33         } catch (SQLException | HeadlessException e) {
34             JOptionPane.showMessageDialog(null, e);
35         }
36     }
37 }
38
```

File ini digunakan untuk inputan pembelian dari user customer, User customer diminta untuk memasukkan data id pembelian, id user, tanggal beli, total beli, status, dan alamat. Data yang diinputkan ini nantinya akan masuk ke dalam database.

3.7.1.6. Pembelian_produk.java

```
1 /**
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package model;
7
8 import java.awt.HeadlessException;
9 import java.sql.SQLException;
10 import javax.swing.JOptionPane;
11
12 /**
13  *
14  * @author ASUS
15  */
16 public class pembelian_produk {
17     String id_beli_produk;
18     String id_pembelian;
19     String kodenya;
20
21     public void addproduk(String id_beli_produk, String id_pembelian, String kodenya) {
22         try {
23             String sql = "INSERT INTO 'pembelian_produk' ('id_beli_produk', 'id_pembelian', 'kodehoodie') VALUES "
24                 + "(" + id_beli_produk + ", " + id_pembelian + ", " + kodenya + ")";
25             java.sql.Connection conn = (java.sql.Connection) controller.koneksi.koneksiIDB();
26             java.sql.PreparedStatement pst = conn.prepareStatement(sql);
27             pst.execute();
28             JOptionPane.showMessageDialog(null, "Berhasil Simpan, Melanjutkan ke Checkout");
29         } catch (SQLException | HeadlessException e) {
30             JOptionPane.showMessageDialog(null, e);
31         }
32     }
33 }

```

File ini digunakan untuk form pembelian produk hoodie. Di dalam file ini terdapat data terkait id_beli_produk, id_pembelian, kodehoodie. Jika user telah berhasil melakukan pembelian produk akan muncul popup “Berhasil Simpan, Melanjutkan ke Checkout”. Popup itu menandakan bahwa pembelian hoodie yang telah dilakukan oleh user pembeli berhasil dan user pembeli diharapkan melanjutkan transaksi pembayaran untuk menyelesaikan pembelian hoodie.

3.7.2.1. AdminForm.java

```

1  *
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4  */
5
6  package view;
7  import model.Hoodie;
8  import model.Admin;
9  import com.mysql.cj.xdevapi.Statement;
10 import java.awt.HeadlessException;
11 import java.sql.Connection;
12 import java.sql.SQLException;
13 import java.io.ByteArrayOutputStream;
14 import java.io.File;
15 import java.io.FileInputStream;
16 import javax.swing.ImageIcon;
17 import javax.swing.JFileChooser;
18 import java.awt.Image;
19 import java.sql.Connection;
20 import java.sql.PreparedStatement;
21 import java.sql.ResultSet;
22 import javax.swing.JLabel;
23 import java.sql.ResultSet;
24 import javax.swing.JOptionPane;
25 import net.proteanit.sql.DbUtils;
26
27
28 public class AdminForm extends javax.swing.JFrame {
29     String idAdmin = "";
30     String path="";
31     private ImageIcon format=null;
32     private JLabel lblimg;
33     String fname=null;
34     int a=0;
35     byte[] pinageme=null;
36     Connection conn=null;
37     public boolean databaru;
38
39     /**
40      * Creates new form AdminForm
41      */
42     public AdminForm(String id) {
43         idAdmin = id;
44         initComponents();
45         GetData(); // tampilkan ke grid
46         databaru=true;
47     }
48
49     public AdminForm() {
50     }
51
52     private void GetData() { // menampilkan data dari database
53         try {
54             Connection conn = (Connection)controller.koneksi.koneksiDB();
55             java.sql.Statement stm = conn.createStatement();
56             java.sql.ResultSet sql = stm.executeQuery("SELECT `kodehoodie` , `nama` , `tipe` , `ukuran` , `harga` , `stok` , `gambar` FROM `hoodie`");
57             jTable1.setModel(DbUtils.resultSetToTableModel(sql));
58         }
59         catch (SQLException | HeadlessException e) {
60         }
61     }
62
63     private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
64         if (txtKode.getText().isEmpty() || txtNama.getText().isEmpty() || Integer.parseInt(txtHarga.getText()) == 0 || Integer.parseInt(txtStok.getText()) == 0) {
65             throw new Exception();
66         } catch (Exception ex) {
67             JOptionPane.showMessageDialog(null, "Solusi Kode, nama dan gambar dan lainnya tidak boleh kosong");
68         }
69     }
70     else {
71         Hoodie p = new Hoodie();
72         p.addProduct(txtKode.getText(), Integer.parseInt(txtNama.getText()), Integer.parseInt(txtTipe.getText()), Integer.parseInt(txtUkuran.getText()), Integer.parseInt(txtHarga.getText()), Integer.parseInt(txtStok.getText()), path);
73         txtKode.setText("");
74         txtNama.setText("");
75         txtHarga.setText("");
76         txtStok.setText("");
77         dispose();
78         this.setVisible(true);
79     }
80 }
81
82 private void btnUploadActionPerformed(java.awt.event.ActionEvent evt) {
83     JFileChooser file = new JFileChooser();
84     file.setMultiSelectionEnabled(true);
85     file.setSelectionMode(JFileChooser.FILES_AND_DIRECTORIES);
86     file.setFileHidingEnabled(false);
87     if (file.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
88         java.io.File f = file.getSelectedFile();
89         path = f.getPath().toString();
90         txtPath.setText(path);
91         System.err.println(f.getPath().toString());
92     }
93 }
94
95 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
96     // TODO add your handling code here:
97     databaru = false; // menampilkan data ke textbox
98     try {
99         int row = jTable1.getSelectedRow();
100         String tabel_klik=(jTable1.getModel().getValueAt(row, 0).toString());
101         java.sql.Connection conn = (java.sql.Connection)controller.koneksi.koneksiDB();
102         java.sql.Statement stm = conn.createStatement();
103         java.sql.ResultSet sql = stm.executeQuery("select * from karyawan where id_karyawan='"+tabel_klik+"'");
104         if(sql.next()){
105             String id = sql.getString("kodehoodie");
106             txtKode.setText(id);
107             String nama = sql.getString("nama");
108             txtNama.setText(nama);
109             String nik = sql.getString("tipe");
110             cmbTipe.setSelectedItem(nik);
111             String jabatan = sql.getString("ukuran");
112             cmbUkuran.setSelectedItem(jabatan);
113             String no_telp = sql.getString("harga");
114             txtHarga.setText(no_telp);
115             String alamat = sql.getString("stok");
116             txtStok.setText(alamat);
117             String path = sql.getString("gambar");
118             txtPath.setText(path);
119         }
120     }
121     catch (Exception e) {}
122 }

```

```

414 private void btnCekDataActionPerformed(java.awt.event.ActionEvent evt) {
415     GetData();
416     // TODO add your handling code here:
417 }

419 private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
420     Hoodie p = new Hoodie();
421     if(!p.kodeCheck(txtKodel.getText())){
422         JOptionPane.showMessageDialog(null, "Data Produk Tidak Ditemukan, Gagal Menghapus Data");
423     }else{
424         p.deleteProduct(txtKodel.getText());
425         txtKodel.setText("");
426         GetData();
427     }
428 }
429
431 private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
432     Hoodie p = new Hoodie();
433     if(!p.kodeCheck(txtKodel.getText())){
434         JOptionPane.showMessageDialog(null, "Data Produk Tidak Ditemukan");
435     }else{
436         btnUpdate.setEnabled(true);
437         txtStok1.setEditable(true);
438     }
439 }
440
441 private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
442     Hoodie p = new Hoodie();
443     p.updateStock(txtKodel.getText(), Integer.parseInt(txtStok1.getText()));
444     GetData();
445     txtKodel.setText("");
446     txtStok1.setText("");
447 }
448
449 private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
450     // TODO add your handling code here:
451     Admin a = new Admin();
452     a.logout();
453 }
454
455 private void btnPembelianActionPerformed(java.awt.event.ActionEvent evt) {
456     // TODO add your handling code here:
457     DataPembelian dp = new DataPembelian(idAdmin);
458     dp.setVisible(true);
459     dispose();
460 }

```

```

466 /**
467  * @param args the command line arguments
468  */
469 public static void main(String args[]) {
470     /* Set the Nimbus look and feel */
471     Look and feel setting code (optional)
472
473     /* Create and display the form */
474     java.awt.EventQueue.invokeLater(new Runnable() {
475         public void run() {
476             new AdminForm().setVisible(true);
477         }
478     });
479 }
480
481 // Variables declaration - do not modify
482 private javax.swing.JButton btnCekData;
483 private javax.swing.JButton btnEdit;
484 private javax.swing.JButton btnHapus;
485 private javax.swing.JButton btnLogout;
486 private javax.swing.JButton btnPembelian;
487 private javax.swing.JButton btnTambah;
488 private javax.swing.JButton btnUpdate;
489 private javax.swing.JButton btnUpload;
490 private javax.swing.JComboBox<String> cmbTipe;
491 private javax.swing.JComboBox<String> cmbUkuran;
492 private javax.swing.JLabel jLabel1;
493 private javax.swing.JLabel jLabel10;
494 private javax.swing.JLabel jLabel12;
495 private javax.swing.JLabel jLabel13;
496 private javax.swing.JLabel jLabel14;
497 private javax.swing.JLabel jLabel15;
498 private javax.swing.JLabel jLabel16;
499 private javax.swing.JLabel jLabel17;
500 private javax.swing.JLabel jLabel18;
501 private javax.swing.JLabel jLabel19;
502 private javax.swing.JScrollPane jScrollPane1;
503 private javax.swing.JScrollPane jScrollPane2;
504 private javax.swing.JTable jTable1;
505 private javax.swing.JTextField txtHarga;
506 private javax.swing.JTextField txtKode;
507 private javax.swing.JTextField txtKodel;
508 private javax.swing.JTextField txtNama;
509 private javax.swing.JTextArea txtPath;
510 private javax.swing.JTextField txtStok;
511 private javax.swing.JTextField txtStok1;
512 // End of variables declaration

```

File ini digunakan untuk menampilkan halaman hoodie store dimana hanya dapat dilihat oleh admin. Pada halaman ini terdiri atas tampilan data hoodie yang tersedia. User admin dapat menginputkan tambah produk yang terdiri dari kode hoodie, nama hoodie, tipe hoodie, ukuran, harga, stok, dan mengupload gambar hoodie. Admin dapat mengedit stok hoodie dengan menghapus data, dan juga dapat mengubah stok. Admin dapat melihat detail pembelian dan sistem akan mengalihkan ke halaman data pembelian.

3.7.2.2. Checkout.java

```
5 package view;
6
7 import model.pembelian;
8 import java.awt.HeadlessException;
9 import java.sql.SQLException;
10 import java.time.LocalDate;
11 import java.util.Date;
12 import javax.swing.JOptionPane;
13
14 /**
15  *
16  * @author janggawijaya
17  */
18 public class Checkout extends javax.swing.JFrame {
19
20     /**
21      * Creates new form Checkout
22      */
23     String id_pembelian_up = "";
24     String id_user = "";
25     String status = "waiting";
26     String alamatkirin = "";
27     int totalnya = 0;
28     LocalDate tgl = LocalDate.now();
29     public Checkout() {}
30     public Checkout(String nama, String jumlah, String total, String id_pembelian, String id, String alamat) {
31         initComponents();
32         id_pembelian_up = id_pembelian;
33         id_user = id;
34         totalnya = Integer.parseInt(total);
35         lblNama.setText("Produk: " + nama);
36         lblJumlah.setText("Jumlah: " + jumlah);
37         lblTotal.setText("Rp. " + total);
38         lblTgl.setText("Tanggal Pembelian: " + java.time.LocalDate.now().toString());
39         alamatkirin = alamat;
40         lblAlamat.setText(alamat);
41     }
}
```

```

166 private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
167     pembelian p = new pembelian();
168     p.beli(id_pembelian_up, id_user, tgl, totalnya, status, alamatkirim);
169 }
170
171 private void btnProdukActionPerformed(java.awt.event.ActionEvent evt) {
172     // TODO add your handling code here:
173     ProductForm pf = new ProductForm(id_user);
174     pf.setVisible(true);
175     dispose();
176 }
177
178 /**
179  * @param args the command line arguments
180  */
181 public static void main(String args[]) {
182     /* Set the Nimbus look and feel */
183     Look and feel setting code (optional)
184
185     /* Create and display the form */
186     java.awt.EventQueue.invokeLater(new Runnable() {
187         public void run() {
188             new Checkout().setVisible(true);
189         }
190     });
191 }
192
193 // Variables declaration - do not modify
194 private javax.swing.JToggleButton btnKeluar;
195 private javax.swing.JToggleButton btnProduk;
196 private javax.swing.JLabel jLabel1;
197 private javax.swing.JLabel jLabel2;
198 private javax.swing.JLabel jLabel3;
199 private javax.swing.JLabel jLabel7;
200 private javax.swing.JLabel lblAlamat;
201 private javax.swing.JLabel lblJumlah;
202 private javax.swing.JLabel lblNama;
203 private javax.swing.JLabel lblTgl;
204 private javax.swing.JLabel lblTotal;
205 // End of variables declaration
206 }

```

File ini digunakan untuk menampilkan halaman checkout setelah user benar-benar akan melakukan pembelian hoodie. Sebelumnya, user pembeli dapat melihat detail pembelian hoodie untuk memastikan yang akan dibeli sudah sesuai. Pada halaman checkout ini terdapat nama produk, jumlah, tanggal pembelian, total pembelian, alamat pengiriman, dan juga jenis pengiriman. Pada halaman ini user dapat menekan button selesai untuk mengakhiri checkout dan button lihat produk lainnya untuk menampilkan halaman product form kembali.

3.7.2.3. DataPembelian.java

```

1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package view;
7
8   import java.awt.HeadlessException;
9   import java.sql.Connection;
10  import java.sql.SQLException;
11  import javax.swing.ImageIcon;
12  import javax.swing.JLabel;
13  import javax.swing.JOptionPane;
14  import net.proteanit.sql.DbUtils;
15
16  /**
17   *
18   * @author ASUS
19   */
20  public class DataPembelian extends javax.swing.JFrame {
21      String idAdmin = "";
22      String path="";
23      private ImageIcon format=null;
24      private JLabel lblimg;
25      String fname=null;
26      int s=0;
27      byte[] pimage=null;
28      Connection conn=null;
29      public boolean databaru;
30
31      /**
32       * Creates new form DetailPembelian
33       */
34      public DataPembelian() {
35          initComponents();
36      }
37
38      /**
39       * Creates new form AdminForm
40       */
41      public DataPembelian(String id) {
42          idAdmin = id;
43          initComponents();
44          GetDataPembelian(); // tampilkan ke grid
45          databaru=true;
46      }
47
48      private void GetDataPembelian() { // menampilkan data dari database
49          try {
50              Connection conn =(Connection)controller.koneksi.koneksiDB();
51              java.sql.Statement stmt = conn.createStatement();
52              java.sql.ResultSet rsl = stmt.executeQuery("SELECT nama_langkap as 'Nama Lengkap', 'tgl_beli' as 'Tanggal Pembelian', 'nama' as 'Produk', "
53                  + "'harga' as 'Harga', 'total_beli' as 'Total Belanja', 'alamat', 'status' FROM users us "
54                  + "INNER JOIN pembelian pm ON us.id=pm.iduser INNER JOIN pembelian_produk pp ON pm.id_pembelian=pp.id_pembelian "
55                  + "INNER JOIN hoodie hd ON pp.kodehoodie=hd.kodehoodie");
56              jTable1.setModel(DbUtils.resultSetToTableModel(rsl));
57          } catch (SQLException | HeadlessException e) {
58              //
59          }
60
61      }
62
63      public boolean kodeCheck(String kode){
64          boolean res = true;
65          try {
66              Connection conn =(Connection)controller.koneksi.koneksiDB();
67              java.sql.Statement stmt = conn.createStatement();
68              java.sql.ResultSet rs = stmt.executeQuery("select * from pembelian");
69              while (rs.next()) {
70                  String ip = rs.getString("tgl_beli");
71                  if(kode.equals(ip)){
72                      res=true;
73                      break;
74                  }
75                  else{
76                      res = false;
77                  }
78              }
79          } catch (SQLException | HeadlessException e) {
80              //
81          }
82          return res;
83      }

```

```

84 public void updateStatus(String tgl_beli, String status){
85     try {
86         String sql = "UPDATE 'pembelian' SET 'status'='"+status+"' WHERE 'tgl_beli' = '"+tgl_beli+"'";
87         java.sql.Connection conn = (java.sql.Connection)controller.koneksi.koneksiDB();
88         java.sql.PreparedStatement pst = conn.prepareStatement(sql);
89         pst.executeUpdate();
90         JOptionPane.showMessageDialog(null, "Status Berhasil Diubah");
91     } catch (SQLException | HeadlessException e) {
92         JOptionPane.showMessageDialog(null, e);
93     }
94 }
95
96 /**
97  * This method is called from within the constructor to initialize the form.
98  * WARNING: Do NOT modify this code. The content of this method is always
99  * regenerated by the Form Editor.
100  */
101 @SuppressWarnings("unchecked")
102 Generated Code
255
256 private void btnKembaliActionPerformed(java.awt.event.ActionEvent evt) {
257     // TODO add your handling code here:
258     AdminForm af = new AdminForm(idAdmin);
259     af.setVisible(true);
260     dispose();
261 }
262
263 private void btnEditStatusActionPerformed(java.awt.event.ActionEvent evt) {
264     // TODO add your handling code here:
265     DataPembelian dp = new DataPembelian();
266     if(!dp.kodeCheck(txtTgl_beli.getText())){
267         JOptionPane.showMessageDialog(null, "Tanggal Pembelian Tidak Ditemukan");
268     }else{
269         btnUpdateStatus.setEnabled(true);
270         txtStatus.setEditable(true);
271     }
272 }
273
274 private void btnUpdateStatusActionPerformed(java.awt.event.ActionEvent evt) {
275     // TODO add your handling code here:
276     DataPembelian dp = new DataPembelian();
277     dp.updateStatus(txtTgl_beli.getText(), txtStatus.getText());
278     GetDataPembelian();
279     txtTgl_beli.setText("");
280     txtStatus.setText("");
281 }
282
283 private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
284     // TODO add your handling code here:
285     GetDataPembelian();
286 }
287
288 /**
289  * @param args the command line arguments
290  */
291 public static void main(String args[]) {
292     /* Set the Nimbus look and feel */
293     Look and feel setting code (optional)
294     //</editor-fold>
295
296     /* Create and display the form */
297     java.awt.EventQueue.invokeLater(new Runnable() {
298         public void run() {
299             new DataPembelian().setVisible(true);
300         }
301     });
302 }
303
304 // Variables declaration - do not modify
305 private javax.swing.JButton btnEditStatus;
306 private javax.swing.JButton btnKembali;
307 private javax.swing.JButton btnRefresh;
308 private javax.swing.JButton btnUpdateStatus;
309 private javax.swing.JLabel jLabel1;
310 private javax.swing.JLabel jLabel2;
311 private javax.swing.JLabel jLabel3;
312 private javax.swing.JLabel jLabel4;
313 private javax.swing.JScrollPane jScrollPane1;
314 private javax.swing.JTable jTable1;
315 private javax.swing.JTextField txtStatus;
316 private javax.swing.JTextField txtTgl_beli;
317 // End of variables declaration
318 }

```

File ini digunakan untuk menampilkan data pembelian. Data yang ditampilkan berupa nama lengkap, tanggal pembelian, produk, harga, total belanja, dan alamat. Terdapat

button kembali untuk kembali ke halaman sebelumnya, tombol refresh untuk untuk menampilkan data terbaru jika ada, dan juga bisa melakukan edit status dan update status untuk mengubah status data pembelian.

3.7.2.4. Login.java

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4  */
5  package view;
6
7  import model.User;
8  import javax.swing.JOptionPane;
9
10 /**
11  *
12  * @author User
13  */
14 public class Login extends javax.swing.JFrame {
15
16     /**
17     * Creates new form Login
18     */
19     public Login() {
20         initComponents();
21     }
22
23     /**
24     * This method is called from within the constructor to initialize the form.
25     * WARNING: Do NOT modify this code. The content of this method is always
26     * regenerated by the Form Editor.
27     */
28     @SuppressWarnings("unchecked")
29     Generated Code
30
31     @SuppressWarnings("unchecked")
32     Generated Code
33
34     private void txtUsernameActionPerformed(java.awt.event.ActionEvent evt) {
35         // TODO add your handling code here:
36     }
37
38     private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
39         User u = new User();
40         boolean ok = u.loginCheck(txtUsername.getText(), txtPassword.getText());
41         if(!ok){
42             JOptionPane.showMessageDialog(null,"Maaf Username dan Password tidak sesuai");
43         }else{
44             dispose();
45         }
46     }
47
48     private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
49         new Register().setVisible(true);
50         dispose();
51     }
52
53     public static void main(String args[]) {
54         /* Set the Nimbus look and feel */
55         Look and feel setting code (optional)
56
57         /* Create and display the form */
58         java.awt.EventQueue.invokeLater(new Runnable() {
59             public void run() {
60                 new Login().setVisible(true);
61             }
62         });
63
64         // Variables declaration - do not modify
65         private javax.swing.JButton btnLogin;
66         private javax.swing.JButton btnRegister;
67         private javax.swing.JLabel jLabel1;
68         private javax.swing.JLabel jLabel2;
69         private javax.swing.JLabel jLabel3;
70         private javax.swing.JPasswordField txtPassword;
71         private javax.swing.JTextField txtUsername;
72         // End of variables declaration
73     }
74
75 }

```

File ini digunakan untuk tampilan login halaman admin dan customer. Pada login ini terdapat inputan berupa username dan password. User dapat menekan tombol login jika telah menginputkan username dan password sesuai dengan yang telah diregistrasikan sebelumnya.

3.7.2.5. PembelianUser.java

```

6 package view;
7
8 import java.awt.HeadlessException;
9 import java.sql.Connection;
10 import java.sql.SQLException;
11 import net.proteanit.sql.DbUtils;
12
13 /**
14  *
15  * @author ASUS
16  */
17 public class PembelianUser extends javax.swing.JFrame {
18     String id_user = "";
19
20     public PembelianUser(String id) {
21         id_user = id;
22         initComponents();
23         GetDataPembelianUser();
24     }
25
26
27     /**
28      * Creates new form Pembelian
29      */
30     public PembelianUser() {
31         initComponents();
32     }
33
34     private void GetDataPembelianUser() { // menampilkan data dari database
35         try {
36             Connection conn = (Connection)controller.koneksi.koneksiDB();
37             java.sql.Statement stm = conn.createStatement();
38             java.sql.ResultSet sql = stm.executeQuery("SELECT 'nama_lengkap' as 'Nama Lengkap', 'tgl_beli' as 'Tanggal Pembelian', "
39                 + "'nama' as 'Produk', 'harga' as 'Harga', 'total_beli' as 'Total Belanja', 'alamat' as 'Alamat Pengiriman', "
40                 + "'status' as 'Status' FROM users us INNER JOIN pembelian pm ON us.id=pm.iduser "
41                 + "INNER JOIN pembelian_produk pp ON pm.id_pembelian=pp.id_pembelian "
42                 + "INNER JOIN hoodie hd ON pp.kodehoodie=hd.kodehoodie WHERE iduser = '"+id_user+"'");
43             jTable1.setModel(DbUtils.resultSetToTableModel(sql));
44         }
45         catch (SQLException | HeadlessException e) {}
46     }
47
48
49     private void btnKembaliActionPerformed(java.awt.event.ActionEvent evt) {
50         // TODO add your handling code here:
51         ProductForm pf = new ProductForm(id_user);
52         pf.setVisible(true);
53         dispose();
54     }
55
56     private void btnlogoutActionPerformed(java.awt.event.ActionEvent evt) {
57         // TODO add your handling code here:
58         //exit
59         System.exit(0);
60     }
61
62     /**
63      * @param args the command line arguments
64      */
65     public static void main(String args[]) {
66         /* Set the Nimbus look and feel */
67         Look and feel setting code (optional)
68         //</editor-fold>
69
70         /* Create and display the form */
71         java.awt.EventQueue.invokeLater(new Runnable() {
72             public void run() {
73                 new PembelianUser().setVisible(true);
74             }
75         });
76     }
77
78     // Variables declaration - do not modify
79     private javax.swing.JButton btnKembali;
80     private javax.swing.JButton btnlogout;
81     private javax.swing.JLabel jLabel1;
82     private javax.swing.JLabel jLabel2;
83     private javax.swing.JScrollPane jScrollPane1;
84     private javax.swing.JTable jTable1;
85     // End of variables declaration
86 }

```

File ini digunakan untuk menampilkan data pembelian user. Data yang ditampilkan antara lain nama lengkap, tanggal pembelian, produk, harga, total belanja, alamat pengiriman, dan status. Terdapat juga button untuk kembali ke halaman sebelumnya dan button logout untuk keluar tidak sebagai user.

3.7.2.6. ProductDetailForm.java

```
Start Page x Checkout.java x DataPembelian.java x Login.java x ProductDetailForm.java x index.php x koneksi.java x Admin.java x AdminForm.java x User.js
Source Design History

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
4  */
5  package view;
6  import model.pembelian_produk;
7  import java.awt.HeadlessException;
8  import java.sql.Statement;
9
10 import java.sql.SQLException;
11 import java.sql.ResultSet;
12 import java.awt.Image;
13 import javax.swing.ImageIcon;
14 import javax.swing.JOptionPane;
15
16 /**
17  *
18  * @author janggawijaya
19  */
20 public class ProductDetailForm extends javax.swing.JFrame {
21     String kodenya = "";
22     String gambar = "";
23     String nama = "";
24     String tipe = "";
25     String harga = "";
26     String ukuran = "";
27     String idnya = "";
28     int min = 50;
29     int max = 9999;
30     int min2 = 10000;
31     int max2 = 20000;
32
33     //Generate random int value from 50 to 100
34     // System.out.println("Random value in int from "+min+" to "+max+ " :");
35     int random_int = (int) Math.floor(Math.random() * (max-min+1)+min);
36     int random_int2 = (int) Math.floor(Math.random() * (max2-min2+1)+min2);
37     // System.out.println(random_int);
38     String id_beli_produk = "PR-"+Integer.toString(random_int);
39     String id_pembelian = "PB-"+Integer.toString(random_int2);
40
41     public ProductDetailForm() {
42     }
43
44     public ProductDetailForm(String kode,String id){
45         kodenya = kode;
46         idnya = id;
47         initComponents();
48         getProduct();
49     }
50
51     public void getProduct(){
52         try{
53             Statement stat = (Statement) controller.koneksi.koneksiDB().createStatement( );
54             String sql = "Select * from hoodie where kodehoodie = '"+kodenya+"'";
55             ResultSet res = stat.executeQuery(sql);
56             while(res.next ()){
57                 nama = res.getString("nama");
58                 tipe = res.getString("tipe");
59                 ukuran = res.getString("ukuran");
60                 harga = res.getString("harga");
61                 gambar = res.getString("gambar");
62             }
63         }catch(SQLException err){
64             JOptionPane.showMessageDialog(null, err.getMessage() );
65         }
66         ImageIcon im = new ImageIcon(gambar);
67         Image image = im.getImage();
68         Image newimg = image.getScaledInstance(300, 300, java.awt.Image.SCALE_SMOOTH); // scale it the smooth way
69         im = new ImageIcon(newimg);
70         jLabel9.setIcon(im);
71         lblNama.setText(nama);
72         lblHarga.setText("Rp. "+harga);
73         lblTipe.setText(tipe);
74         lblUkuran.setText(ukuran);
75     }
76
77     public boolean cekQty(){
78         boolean status = false;
79         if(Integer.parseInt(txtQty.getText())>=20){
80             status = false;
81         }else{
82             status = true;
83         }
84         return status;
85     }
86 }
```

File ini berfungsi untuk menampilkan form detail dari data yang akan dibeli oleh customer. Data yang ditampilkan adalah nama hoodie, harga tipe, dan ukuran. Customer diminta untuk memasukkan kuantiti hoodie yang ingin dibeli dan alamat buat pengiriman. Setelah menginputkan kuantiti dan alamat customer tinggal menekan button beli sekarang. Button beli sekarang akan membawa customer ke form checkout.

3.7.2.7. ProductForm.java

```

1  // * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2  // * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
3  //
4  package view;
5
6  //import com.mysql.cj.xdevapi.Statement;
7
8  import model.customer;
9  import model.hoodie;
10 import java.awt.HeadlessException;
11 import java.awt.Image;
12 import java.io.File;
13 import java.sql.Connection;
14 import java.sql.Statement;
15 import java.sql.ResultSet;
16 import java.sql.SQLException;
17 import javax.swing.ImageIcon;
18 import javax.swing.JLabel;
19 import javax.swing.JOptionPane;
20 import javax.swing.table.DefaultTableModel;
21 import javax.swing.table.TableColumnModel;
22 import net.proteanit.sql.DbUtils;
23 import controller.koneksi;
24
25 /**
26  *
27  * @author User
28  */
29 public class ProductForm extends javax.swing.JFrame {
30     String idnya = "";
31     private DefaultTableModel model;
32
33     /**
34      * Creates new form ProductForm
35      */
36     public ProductForm() {}
37     public ProductForm(String id) {
38         initComponents();
39         idnya = id;
40
41         model = new DefaultTableModel ( ){
42             @Override
43             public Class<?> getColumnClass(int column) {
44                 switch(column) {
45                     case 0: return ImageIcon.class;
46                     default: return Object.class;
47                 }
48             }
49         };
50
51         jTable1.setModel(model);
52
53         model.addColumn("Gambar");
54         model.addColumn("Kode");
55         model.addColumn("Nama");
56         model.addColumn("Tipe");
57         model.addColumn("Ukuran");
58         model.addColumn("Harga");
59         jTable1.setRowHeight(120);
60         jTable1.setRowSelectionAllowed(false);
61         TableColumnModel col = jTable1.getColumnModel();
62         col.getColumn(0).setPreferredWidth(180);
63         col.getColumn(0).setResizable(true);
64         getData();
65
66     public void getData() {
67         //menghapus isi table tblGaji
68         model.getDataVector().removeAllElements();
69         model.fireTableDataChanged();
70
71         try {
72             //membuat statemen pemanggilan data pada table hoodie dari database
73             Statement stat = (Statement) koneksi.koneksiDB().createStatement();
74             String sql = "Select * from hoodie";
75             ResultSet res = stat.executeQuery(sql);
76
77             //penelusuran baris pada tabel tblGaji dari database
78             while(res.next()) {
79                 Object[] obj = new Object[6];
80                 ImageIcon im = new ImageIcon(res.getString("gambar"));
81                 Image image = im.getImage();
82                 Image newimg = Image.getScaledInstance(120, 120, java.awt.Image.SCALE_SMOOTH); // scale it the smooth way
83                 im = new ImageIcon(newimg);
84                 obj[0] = im;
85                 obj[1] = res.getString("kodehoodie");
86                 obj[2] = res.getString("nama");
87                 obj[3] = res.getString("tipe");
88                 obj[4] = res.getString("ukuran");
89                 obj[5] = res.getString("harga");
90
91                 model.addRow(obj);
92             }
93         } catch (SQLException ex) {
94             JOptionPane.showMessageDialog(this, ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
95         }
96     }
97 }

```

```

91         catch (SQLException err) {
92             JOptionPane.showMessageDialog(null, err.getMessage());
93         }
94     }
95     private void GetData() { // menampilkan data dari database
96         try {
97             Connection conn = (Connection) koneksi.koneksiDB();
98             java.sql.Statement stm = conn.createStatement();
99             java.sql.ResultSet sql = stm.executeQuery("SELECT 'kodehoodie', 'nama', 'tipe', 'ukuran', 'harga', 'stock' FROM 'hoodie'");
100             jTable1.setModel(DbUtils.resultSetToTableModel(sql));
101         }
102         catch (SQLException | HeadlessException e) {
103             //
104         }
105     }

```

File ini digunakan untuk menampilkan list dari hoodie untuk menampilkan pada user. Pada product form ini terdapat title yang dimana terdapat gambar, kode hoodie, nama, tipe ukuran, dan harga. Pada tampilan product form ini juga user dapat melihat riwayat pembelian yang telah dilakukan, dan detail pembelian dengan menginputkan kode hoodie yang akan dibeli lalu halaman dilanjutkan pada product detail form. User dapat menekan logout untuk keluar dari sistem.

3.7.2.8. Register.java

```

5     package view;
6
7     import java.util.logging.Level;
8     import java.util.logging.Logger;
9     import javax.swing.JOptionPane;
10    import model.User;
11
12    /**
13     *
14     * @author janggawijaya
15     */
16    public class Register extends javax.swing.JFrame {
17
18        /**
19         * Creates new form Register
20         */
21        public Register() {
22            initComponents();
23        }
24
25        private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
26            User u = new User();
27            if (txtUsername.getText().isEmpty() || txtPassword.getText().isEmpty() || txtNama.getText().isEmpty()) {
28                try {
29                    throw new Exception();
30                } catch (Exception ex) {
31                    JOptionPane.showMessageDialog(null, "Kolom username, password dan nama tidak boleh kosong");
32                }
33            } else {
34                u.addUser(txtUsername.getText(), txtPassword.getText(), txtNama.getText());
35                new Login().setVisible(true);
36                dispose();
37            }
38        }
39
40        private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
41            new Login().setVisible(true);
42            dispose();
43        }
44
45        /**
46         * @param args the command line arguments
47         */
48        public static void main(String args[]) {
49            /* Set the Nimbus look and feel */
50            /* Look and feel setting code (optional) */
51
52            /* Create and display the form */
53            java.awt.EventQueue.invokeLater(new Runnable() {
54                public void run() {
55                    new Register().setVisible(true);
56                }
57            });
58        }
59
60        // Variables declaration - do not modify
61        private javax.swing.JButton btnLogin;
62        private javax.swing.JButton btnRegister;
63        private javax.swing.JLabel jLabel1;
64        private javax.swing.JLabel jLabel2;
65        private javax.swing.JLabel jLabel3;
66        private javax.swing.JLabel jLabel4;
67        private javax.swing.JLabel jLabel5;
68        private javax.swing.JTextField txtNama;
69        private javax.swing.JPasswordField txtPassword;
70        private javax.swing.JTextField txtUsername;
71        // End of variables declaration
72    }

```

File ini digunakan untuk menampilkan halaman register yang dimana user dapat menginputkan username, password dan juga nama lengkap. Lalu akan disimpan di dalam database hoodie_db yaitu pada tabel user.

3.7.3. Controller

3.7.3.1. Koneksi.java

```

5
6 package controller;
7
8 import java.sql.Connection;
9 import java.sql.DriverManager;
10 import java.sql.SQLException;
11 import javax.swing.JOptionPane;
12
13 public class koneksi { // koneksi ke database
14     private static Connection mysqlkoneksi;
15     public static Connection koneksiDB() throws SQLException {
16         if(mysqlkoneksi==null){
17             try {
18                 String DB="jdbc:mysql://localhost:3306/hoodie_db"; // hoodie_db database
19                 String user="root"; // user database
20                 String pass=""; // password database
21                 DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());
22                 mysqlkoneksi = (Connection) DriverManager.getConnection(DB,user,pass);
23             } catch (Exception e) {
24                 JOptionPane.showMessageDialog(null,"gagal koneksi");
25             }
26         }
27         return mysqlkoneksi;
28     }
29 }

```

File ini digunakan untuk melakukan pengkoneksian dari file java yang berada pada netbeans ke dalam sql phpMyAdmin. Langkah pengkoneksian database diharuskan untuk menjalankan xampp control panel. Adapun yang harus dihidupkan adalah Apache dan MySQL. Pada bagian string DB, user perlu mengcopy link jdbc dari mysql dan disesuaikan dengan port dan nama database yang akan digunakan.

3.8. LAMPIRAN APLIKASI

3.8.1. Login

Jframe “Login” digunakan sebagai halaman user (Admin dan customer) untuk melakukan proses login dan masuk ke dalam aplikasi. Komponen:

1. “YA-HOODIE” menggunakan “JLabel”

2. "Username" menggunakan "JLabel"
3. "Password" menggunakan "JLabel"
4. Kolom "Username" menggunakan "JTextField"
5. Kolom "Password" menggunakan "JTextField"
6. Tombol "Login" menggunakan "JButton"
7. Tombol "Register" menggunakan "JButton"

Class yang bertindak adalah Login.java

Penjelasan:

1. JTextField "Username" berguna untuk memasukkan username yang telah terdaftar di database.
2. JTextField "Password" berguna untuk memasukkan username yang telah terdaftar di database.
3. JButton "Login" berguna untuk submit username dan password yang telah dimasukkan pada kolom username dan password.
4. JButton "Register" berguna untuk mengakses halaman register untuk user (customer) melakukan registrasi agar dapat melakukan proses login.
5. Halaman ini merupakan tampilan pertama dari aplikasi.

3.8.2. Register



The image shows a Java Swing window titled "Register". Inside the window, the text "YA-HOODIE" is displayed in a large, bold, black font, with "REGISTER" in a smaller, bold, black font below it. There are three text input fields: "Username", "Password", and "Nama Lengkap". At the bottom of the window, there are two buttons: "Register" and "Login".

Jframe "Login" digunakan sebagai halaman user (dan customer) untuk melakukan proses registrasi dan mendaftarkan data diri ke dalam sistem aplikasi.

Komponen:

1. "YA-HOODIE" menggunakan "JLabel"
2. "Register" menggunakan "JLabel"
3. "Username" menggunakan "JLabel"
4. "Password" menggunakan "JLabel"
5. "Nama Lengkap" menggunakan "JLabel"
6. Kolom "Username" menggunakan "JTextField"

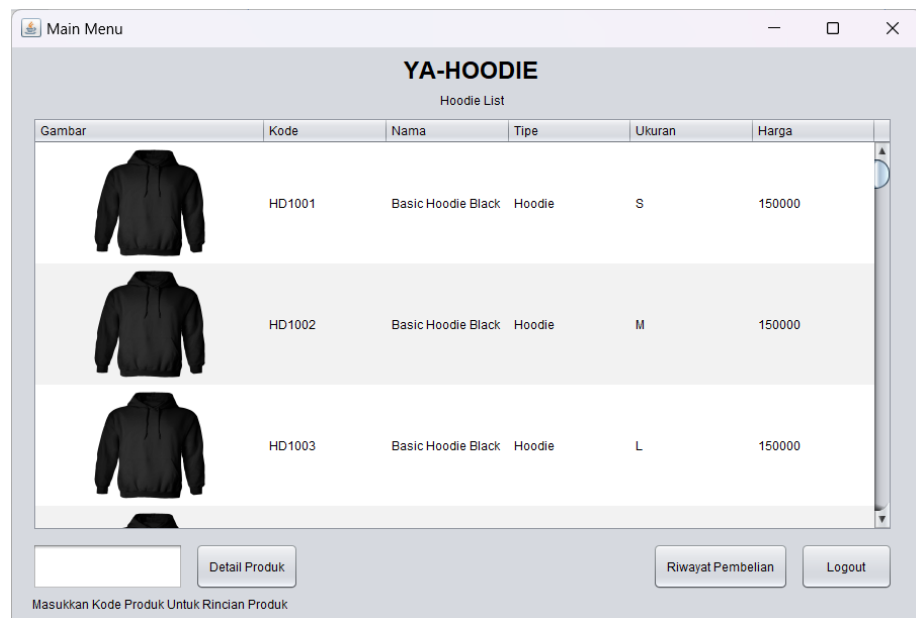
7. Kolom “Password” menggunakan “JTextField”
8. Kolom “Nama Lengkap” menggunakan “JTextField”
9. Tombol “Register” menggunakan “JButton”
10. Tombol “Login” menggunakan “JButton”

Class yang bertindak adalah Register.java

Penjelasan:

1. JTextField “Username” berguna untuk memasukkan username yang akan didaftarkan ke database.
2. JTextField “Password” berguna untuk memasukkan password yang akan didaftarkan ke database.
3. JTextField “Nama Lengkap” berguna untuk memasukkan nama lengkap yang akan didaftarkan ke database.
4. JButton “Login” berguna untuk mengakses halaman login.
5. JButton “Register” berguna untuk submit username, password, dan nama lengkap yang telah dimasukkan pada kolom username dan password.

3.8.3. Hoodie List



Jframe “Main Menu” digunakan sebagai halaman Customer untuk melihat daftar hoodie tersedia yang ingin dibeli dan merupakan tampilan awal setelah proses login.

Komponen:

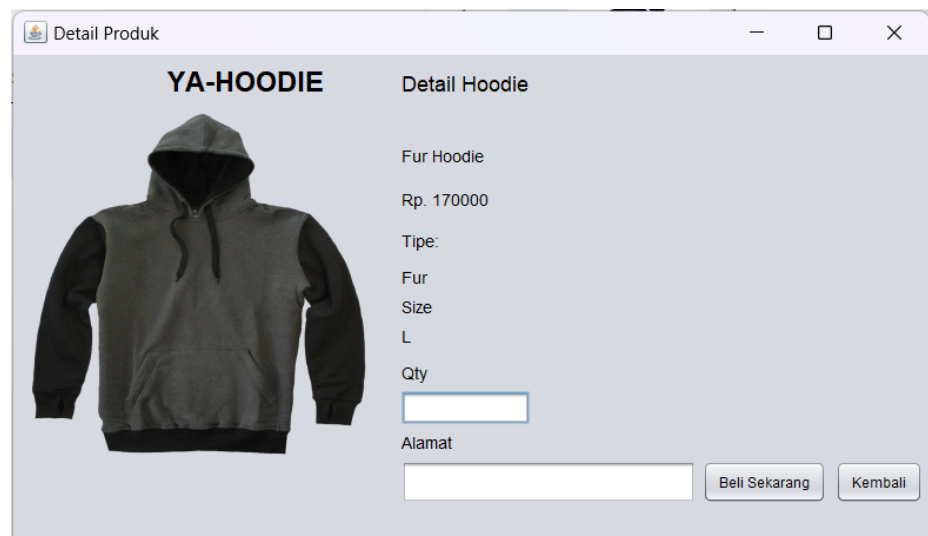
1. “YA-HOODIE” menggunakan “JLabel”
2. “Hoodie List” menggunakan “JLabel”
3. Kolom produk menggunakan “JScrollPane” dan “JTabel”
4. Kode produk menggunakan “JTextField”
5. Tombol “Detail Produk” menggunakan “JButton”
6. Tombol “Riwayat Pembelian” menggunakan “JButton”
7. Tombol “Logout” menggunakan “JButton”

Class yang bertindak adalah ProductForm.java

Penjelasan:

1. “JScrollPane” dan “JTable” kolom produk berguna untuk menampilkan daftar hoodie yang terdapat di database.
2. JTextField “Kode produk” berguna untuk memasukkan kode hoodie yang ingin dibeli oleh customer.
3. JButton “Detail Produk” berguna untuk submit kode produk yang dimasukkan oleh customer dan mengakses halaman detail produk jika kode produk yang dimasukkan benar.
4. JButton “Riwayat Pembelian” berguna untuk mengakses halaman riwayat pembelian yang telah dilakukan oleh user.
5. JButton “Logout” berguna untuk customer menutup dan keluar aplikasi.

3.8.4. Detail Produk



Jframe “Detail Produk” digunakan sebagai halaman Customer untuk melihat detail produk dari kode hoodie yang telah dimasukkan pada halaman main menu.

Komponen:

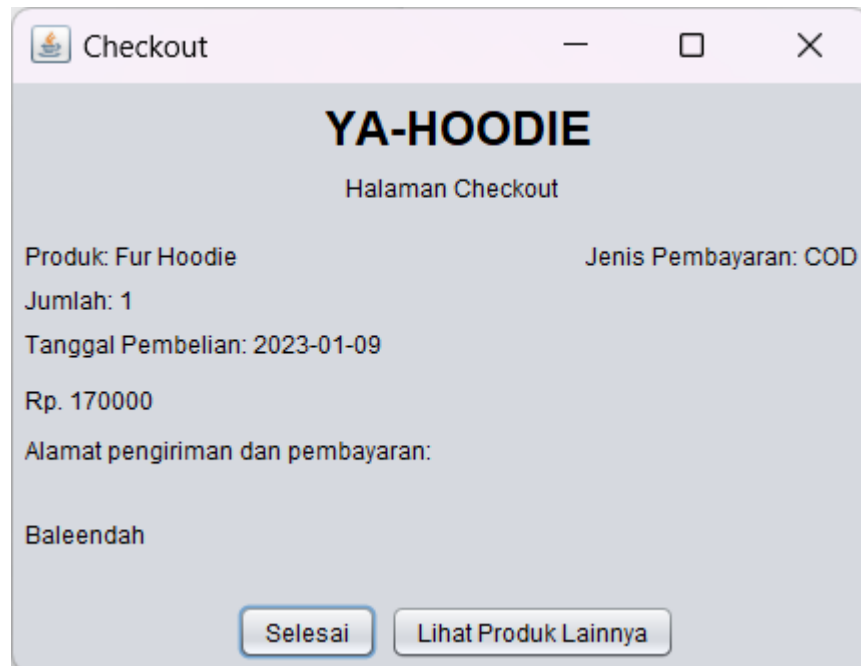
1. “YA-HOODIE” menggunakan “JLabel”
2. “Detail Hoodie” menggunakan “JLabel”
3. Nama hoodie menggunakan “JLabel”
4. Harga hoodie menggunakan “JLabel”
5. Tipe hoodie menggunakan “JLabel”
6. Size hoodie menggunakan “JLabel”
7. Qty hoodie menggunakan “JLabel”
8. Kolom Qty menggunakan “JTextField”
9. Alamat menggunakan “JLabel”
10. Kolom alamat menggunakan “JTextField”
11. Tombol “Beli Sekarang” menggunakan “JButton”
12. Tombol “Kembali” menggunakan “JButton”

Class yang bertindak adalah ProductDetailForm.java

Penjelasan:

1. “JLabel” nama hoodie, harga hoodie, tipe hoodie, Size hoodie berguna untuk menampilkan data nama hoodie, harga hoodie, tipe hoodie, size hoodie yang dipilih.
2. “JTextField” kolom Qty berguna untuk memasukkan jumlah pembelian hoodie yang dipilih.
3. “JTextField” kolom alamat berguna untuk memasukkan alamat pengiriman dan pembayaran hoodie yang dipilih customer.
4. “JButton” Beli Sekarang berguna untuk submit pembelian yang dilakukan customer.
5. “JButton” Kembali berguna untuk menampilkan halaman sebelumnya.

3.8.5. Checkout



Jframe “Halaman Checkout” digunakan sebagai halaman Customer untuk melihat detail pembelian hoodie yang telah dimasukkan pada halaman sebelumnya.

Komponen:

1. “YA-HOODIE” menggunakan “JLabel”
2. “Halaman Checkout” menggunakan “JLabel”
3. Nama produk menggunakan “JLabel”
4. Jumlah menggunakan “JLabel”
5. Tanggal pembelian menggunakan “JLabel”
6. Alamat pengiriman menggunakan “JLabel”
7. Jenis pembayaran menggunakan “JLabel”
8. Tombol “Lihat Produk Lainnya” menggunakan “JButton”
9. Tombol “Selesai” menggunakan “JButton”

Class yang bertindak adalah Checkout.java

Penjelasan:

1. JButton “Lihat Produk Lainnya” berguna untuk kembali ke halaman main menu
2. JButton “Selesai” berguna untuk menyelesaikan pembelian

3.8.6. Riwayat Pembelian



Jframe “Main Menu” digunakan sebagai halaman Customer untuk melihat daftar hoodie tersedia yang ingin dibeli dan merupakan tampilan awal setelah proses login.

Komponen:

1. “YA-HOODIE” menggunakan “JLabel”
2. “Riwayat Pembelian” menggunakan “JLabel”
3. Kolom riwayat pembelian menggunakan “JScrollPane” dan “JTable”
4. Tombol “Kembali” menggunakan “JButton”
5. Tombol “Logout” menggunakan “JButton”

Class yang bertindak adalah PembelianUser.java

Penjelasan:

1. “JScrollPane” dan “JTable” kolom riwayat pembelian berguna untuk menampilkan riwayat pembelian dari customer yang telah dilakukan dan tersimpan di database.
2. JButton “Kembali” berguna untuk kembali ke halaman main menu.
3. JButton “Logout” berguna untuk customer menutup dan keluar aplikasi.

3.8.7. Tampilan awal admin

YA-HOODIE

Kode Hoodie

Nama Hoodie

Tipe

Ukuran

Harga

Stok

Upload Gambar

kodeho...	nama	tipe	ukuran	harga	stock	gambar
HD1001	Basic H...	Hoodie	S	150000	20	C:\User...
HD1002	Basic H...	Hoodie	M	150000	30	C:\User...
HD1003	Basic H...	Hoodie	L	150000	32	C:\User...
HD1004	Basic H...	Hoodie	XL	150000	35	C:\User...
HD1006	Black Zi...	Zipper	S	75000	23	C:\User...
HD1007	Black Zi...	Zipper	M	75000	22	C:\User...
HD1008	Black Zi...	Zipper	L	75000	26	C:\User...
HD1009	Black Zi...	Zipper	XL	75000	28	C:\User...
HD1011	Basic H...	Hoodie	S	150000	32	C:\User...
HD1012	Basic H...	Hoodie	M	150000	29	C:\User...
HD1013	Basic H...	Hoodie	L	150000	25	C:\User...
HD1014	Basic H...	Hoodie	XL	150000	27	C:\User...
HD1016	Dry Hoo...	Hoodie	S	250000	20	C:\User...
HD1017	Dry Hoo...	Hoodie	M	250000	17	C:\User...
HD1018	Dry Hoo...	Hoodie	L	250000	30	C:\User...
HD1019	Dry Hoo...	Hoodie	XL	250000	27	C:\User...
HD1021	Relaxed...	Hoodie	S	20000	19	C:\User...
HD1022	Relaxed...	Hoodie	M	20000	16	C:\User...
HD1023	Relaxed...	Hoodie	L	20000	22	C:\User...
HD1024	Relaxed...	Hoodie	XL	20000	20	C:\User...
HD1026	Cloud R...	Hoodie	S	250000	24	C:\User...
HD1027	Cloud R...	Hoodie	M	250000	20	C:\User...
HD1028	Cloud R...	Hoodie	L	250000	26	C:\User...
HD1029	Cloud R...	Hoodie	XL	250000	23	C:\User...
HD1031	Street H...	Street	S	300000	25	C:\User...

Isi Kode Ubah Stok

Jframe “Admin” digunakan oleh admin untuk menambah, menghapus, mengupdate stock hoodie yang tersedia serta dapat mengakses halaman data pembelian dari user dan mengubah statusnya. Komponen:

1. “YA-HOODIE” menggunakan “JLabel”
2. “Kode Hoodie” menggunakan “JLabel”
3. “Nama Hoodie” menggunakan “JLabel”
4. “Tipe” menggunakan “JLabel”
5. “Ukuran” menggunakan “JLabel”
6. “Harga” menggunakan “JLabel”
7. “Stok” menggunakan “JLabel”
8. “Upload Gambar” menggunakan “JLabel”
9. “Isi kode” menggunakan “JLabel”
10. “Ubah Stock” menggunakan “JLabel”
11. Kolom “Kode Hoodie” menggunakan “JTextField”
12. Kolom “Nama Hoodie” menggunakan “JTextField”
13. Kolom “Tipe” menggunakan “JComboBox”
14. Kolom “Ukuran” menggunakan “JComboBox”
15. Kolom “Harga” menggunakan “JTextField”
16. Kolom “Stok” menggunakan “JTextField”
17. Kolom “Upload Gambar” menggunakan “JTextArea”
18. Kolom “Isi kode” menggunakan “JTextField”
19. Kolom “Ubah Stock” menggunakan “JTextField”
20. Kolom produk hoodie menggunakan “JScrollPane” dan “JTable”
21. Tombol “Tambah Produk” menggunakan “JButton”
22. Tombol “Upload Gambar” menggunakan “JButton”
23. Tombol “Refresh” menggunakan “JButton”
24. Tombol “Detail Pembelian” menggunakan “JButton”
25. Tombol “Logout” menggunakan “JButton”
26. Tombol “Hapus Data” menggunakan “JButton”
27. Tombol “Edit Stock” menggunakan “JButton”

28. Tombol “Update” menggunakan “JButton”

Class yang bertindak adalah AdminForm.java

Penjelasan:

1. JTextField kolom “Kode Hoodie” berguna untuk memasukkan kode hoodie yang ingin ditambahkan.
2. JTextField kolom “Nama Hoodie” berguna untuk memasukkan nama hoodie yang ingin ditambahkan.
3. JComboBox kolom “Tipe” berguna untuk memasukkan tipe hoodie yang ingin ditambahkan.
4. JComboBox kolom “Ukuran” berguna untuk memasukkan ukuran hoodie yang ingin ditambahkan.
5. JTextField kolom “Harga” berguna untuk memasukkan harga hoodie yang ingin ditambahkan.
6. JTextField kolom “Stock” berguna untuk memasukkan stock hoodie yang ingin ditambahkan.
7. JTextArea kolom “Upload Gambar” berguna untuk memasukkan alamat gambar hoodie yang ingin dimasukkan.
8. JTextField kolom “Isi Kode” berguna untuk memasukkan kode hoodie yang ingin dipilih.
9. JTextField kolom “Ubah Stock” berguna untuk memasukkan stock hoodie yang ingin diupdate.
10. “JScrollPane” dan “JTable” kolom produk berguna untuk menampilkan produk yang tersimpan di database.
11. JButton “Tambah Produk” berguna untuk submit data hoodie.
12. JButton “Refresh” berguna untuk mengupdate data di kolom produk.
13. JButton “Data Pembelian” berguna untuk mengakses halaman pembelian dari customer.
14. JButton “Hapus Data” berguna untuk menghapus data hoodie.
15. JButton “Edit Stock” berguna untuk submit kode hoodie.
16. JButton “Update” berguna untuk submit stock hoodie terbaru.
17. JButton “Upload File” berguna untuk memilih gambar yang ingin diupload.

3.8.8. Data Pembelian

The screenshot shows a Java Swing window titled "Admin - Data Pembelian". Inside the window, there is a header section with the text "YA-HOODIE" and "Data Pembelian Customer". Below this is a table with the following columns: Nama Lengkap, Tanggal Pembelian, Produk, Harga, Total Belanja, alamat, and status. The table contains 8 rows of data. At the bottom of the window, there are two input fields labeled "Isi Tanggal Pembelian" and "Masukkan status", and two buttons labeled "Edit Status" and "Update Status". There are also two buttons labeled "Kembali" and "Refresh" on the left side.

Nama Lengkap	Tanggal Pembelian	Produk	Harga	Total Belanja	alamat	status
Arya Fikriansyah	2023-01-08	Basic Hoodie Black	150000	150000	Jl. Sukabirus	Terkirim
raihan1234	2023-01-08	Basic Hoodie Red	150000	150000	Sukapura	Terkirim
Fachrul Febriana	2023-01-09	Fur Hoodie	170000	170000	Baleendah	waiting
Satria Aji PS	2023-01-08	Basic Hoodie Brown	150000	150000	Kost Bougenville	Terkirim
Arya Fikriansyah	2023-01-08	Basic Hoodie Black	150000	150000	Jl. Sukabirus, Gg D...	Terkirim
raihan1234	2023-01-08	Flanel Hoodie	170000	170000	Jl. Sukapura, Bojon...	Terkirim
raihan1234	2023-01-08	Relaxed Fit Hoodie	20000	40000	Sukapura	Terkirim
Arya Fikriansyah	2023-01-09	Crewneck Hoodie	160000	160000	Jl. Sukabirus, Gg D...	waiting

Jframe “Admin - Data Pembelian” digunakan sebagai halaman admin untuk melihat daftar pembelian customer yang telah dilakukan.

Komponen:

1. “YA-HOODIE” menggunakan “JLabel”
2. “Data Pembelian Customer” menggunakan “JLabel”
3. “Isi tanggal pembelian” menggunakan “JLabel”
4. “Masukkan Status” menggunakan “JLabel”
5. Kolom Data Pembelian Customer menggunakan “JScrollPane”, dan “JTabel”
6. Kolom “Isi tanggal pembelian” menggunakan “JTextField”
7. Kolom “Masukkan Status” menggunakan “JTextField”
8. Tombol “Kembali” menggunakan “JButton”
9. Tombol “Refresh” menggunakan “JButton”
10. Tombol “Edit Status” menggunakan “JButton”
11. Tombol “Update Status” menggunakan “JButton”

Class yang bertindak adalah DataPembelian.java

Penjelasan:

1. JScrollPane, dan JTabel kolom Data Pembelian Customer berguna untuk menampilkan Data Pembelian dari Customer telah dilakukan dan tersimpan di database.
2. JTextField kolom “Isi tanggal pembelian” berguna untuk memasukkan tanggal pembelian yang ingin diubah statusnya oleh admin.
3. JTextField kolom “Masukkan status” berguna untuk memasukkan status yang ingin diubah oleh admin.
4. JButton “Kembali” berguna untuk kembali ke halaman main menu.
5. JButton “Refresh” berguna untuk mengupdate data di kolom Data Pembelian Customer.
6. JButton “Edit Status” berguna untuk mensubmit tanggal pembelian hoodie yang ingin diubah oleh admin.
7. JButton “Update Status” berguna untuk mengupdate status di kolom Data Pembelian Customer.

4. PENUTUP

4.1. KESIMPULAN

Aplikasi penjualan hoodie online ini membantu penjual hoodie dan pembeli untuk melakukan penjualan dan pembelian hoodie dengan mudah. Dengan aplikasi ini penjual dapat dengan mudah memasukkan data hoodie, mengelola data hoodie, mengupdate data hoodie, menghapus data hoodie, melihat data pembelian dengan mudah sehingga proses penjualan dan pembelian hoodie lebih cepat dan efisien.

4.2. SARAN

Aplikasi penjualan hoodie online ini masih belum sempurna. Untuk itu kedepannya diharapkan adanya pengembangan untuk aplikasi ini agar menjadi lebih baik lagi. Kami menyarankan beberapa saran bagi pengembang dan pengguna dengan dibangunnya aplikasi ini. Adapun saran-saran tersebut adalah:

1. Aplikasi ini diharapkan menambahkan fitur review untuk customer agar customer dapat memberikan penilaian kepada penjual.
2. Aplikasi ini diharapkan menambahkan beberapa jenis pembayaran lain selain Cash on Delivery untuk memudahkan Customer dalam melakukan pembayaran hoodie.

DAFTAR PUSTAKA

- MJS. (2022). “*Salindia 09 - Static and Inner Class*”. Pemrograman Berorientasi Objek. Fakultas Informatika. Telkom University.
- MJS. (2022). “*Salindia 10 - Collection and Generic*”. Pemrograman Berorientasi Objek. Fakultas Informatika. Telkom University.
- MJS. (2022). “*Salindia 11 - Collection and Generic*”. Pemrograman Berorientasi Objek. Fakultas Informatika. Telkom University.
- MJS. (2022). “*Salindia 12 - MVC and GUI*”. Pemrograman Berorientasi Objek. Fakultas Informatika. Telkom University.
- MJS. (2022). “*Salindia 14 - JDBC-MJS*”. Pemrograman Berorientasi Objek. Fakultas Informatika. Telkom University.