


<b>[Assignment / Project] – External Document</b>	
<b>COMP6115</b> <b>Object Oriented Analysis &amp; Design</b>	
Odd Semester Year 2020/2021	

- **Project Title**

PURPLE LANE BOOKSTORE APPLICATION

- **Introduction**

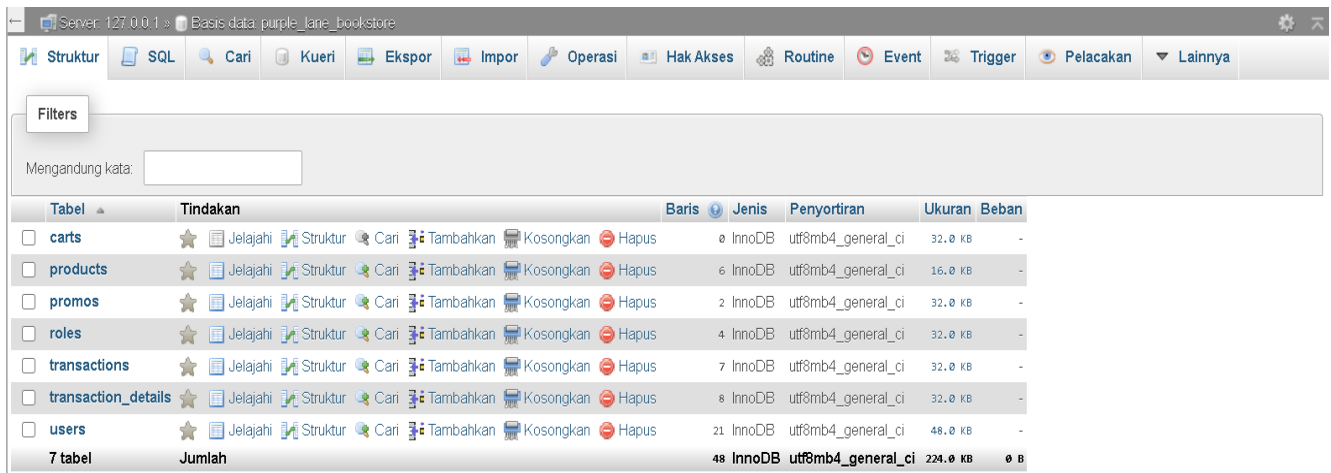
Project ini ditujukan untuk mata kuliah COMP6115-Object Oriented Analysis & Design semester 5 kelas BA-01. Kelompok kami beranggotakan Luis Genesius dan Jackie Leonardy membuat sebuah aplikasi pemesanan buku secara online. Aplikasi ini dibuat dengan Bahasa pemograman Java dengan menggunakan IDE Eclipse dan juga menggunakan DBMS (Database Management System) yaitu MySQL yang dikelola dengan menggunakan PHPMyAdmin.

Project ini kami bangun dengan menggunakan design pattern MVC atau Model-View-Controller yang dimana Model bertindak sebagai komponen yang mengurus segala urusan aplikasi dengan database seperti mengambil data dari database, mengupdate data, dan melakukan penghapusan suatu data di database. View bertindak sebagai komponen yang mengurus urusan tampilan atau layout, sehingga untuk masalah desain tampilan, interaksi aplikasi dengan user, tampilan dan pesan yang diterima oleh user semuanya di urus oleh View. Dalam pengimplementasian tampilannya atau layoutnya menggunakan extends JFrame. Controller bertindak sebagai komponen yang mengurus urusan logika bisnis sehingga semua alur kerja logika, algoritma, validasi, penerimaan data dari Model, di urus oleh Controller.

Di project ini, terbagi menjadi beberapa role (peran), yaitu ada Customer, Admin, Manager, dan Promotion Team. Customer adalah pelanggan yang dapat melakukan pemesanan buku, memilih metode pembayaran, melihat keranjang pembelian, dan mengkonfirmasi pembayaran. Admin bertugas sebagai staff yang mengurus masalah produk buku dimana Admin bisa melihat semua daftar produk buku, menambah data produk baru, mengupdate produk, dan menghapus produk (Create, Read, Update, Delete). Manager dapat melihat semua laporan transaksi serta dapat menyewa staff baru. Promotion Team bertugas sebagai staff yang mengurus masalah promo dimana Promotion Team bisa melihat semua daftar promo, menambah promo baru, mengupdate promo, dan menghapus promo (Create, Read, Update, Delete).

## • Report / Documentation

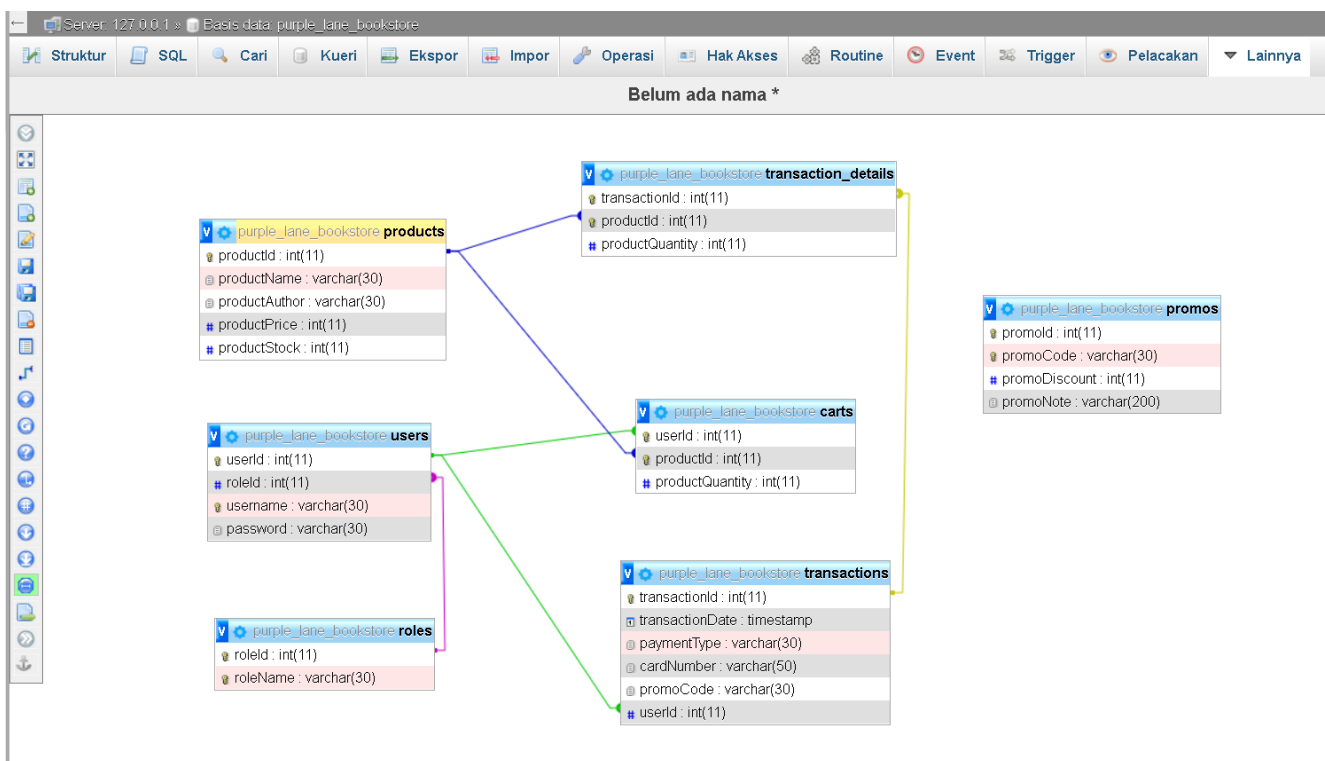
### 1. Database



The screenshot shows the phpMyAdmin interface for a database named 'purple\_lane\_bookstore'. The 'Struktur' (Structure) tab is selected, displaying a list of tables and their details. The tables are: carts, products, promos, roles, transactions, transaction\_details, and users. Each table entry includes icons for viewing, structure, search, insert, update, delete, and a summary of its columns, data type, and size.

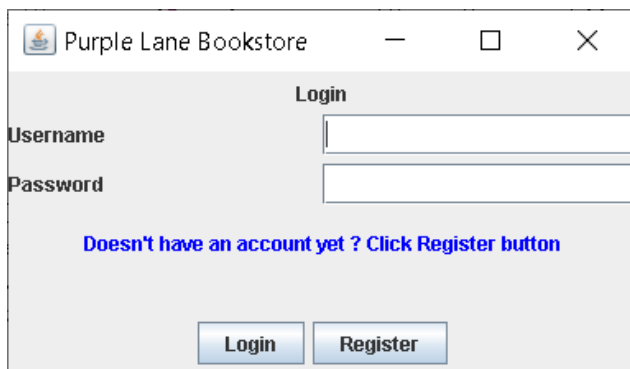
Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
<input type="checkbox"/> carts	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> products	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	6	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> promos	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> roles	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	4	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> transactions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	7	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> transaction_details	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	8	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> users	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	21	InnoDB	utf8mb4_general_ci	48.0 KB	-
7 tabel	Jumlah	48	InnoDB	utf8mb4_general_ci	224.0 KB	0 B

Ini adalah dokumentasi yang diambil di phpMyAdmin, nama databasenya yaitu purple\_lane\_bookstore yang memiliki tabel carts, products, promos, roles, transactions, transaction\_details dan users.

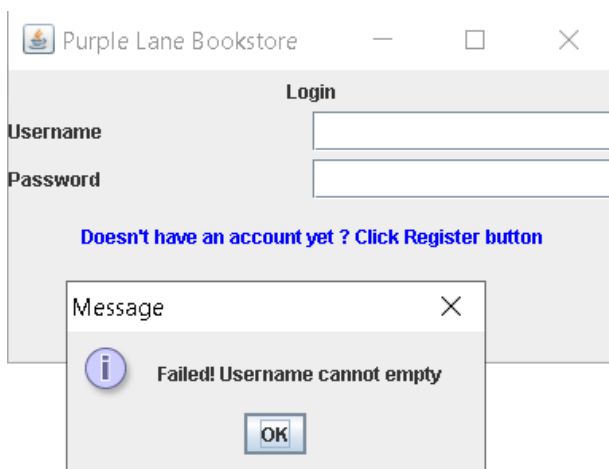


Ini adalah relationship antar tabel serta atribut-atribut pada tabel tersebut juga dengan tipe data atributnya. **Database dibuat secara manual di phpMyAdmin, dan untuk membuat tabelnya bisa mengakses file di query-list.txt sebagai referensi query nya.**

## 2. Start Menu (Login Menu)



Tampilan Code ini dapat di akses di package view dan pada class LoginView.java. Saat kalian meng-run code Java nya di Eclipse, maka tampilan paling awal yang akan kalian temui yaitu tampilan Login. Disini terdapat beberapa validasi yaitu username field dan password fieldnya tidak boleh kosong, karena tentunya untuk dapat mengakses fitur Purple Lane Bookstore di perlukan login, dan login memerlukan data user yang diinput. Password Fieldnya juga di set sebagai JPasswordField sehingga saat user menginput password, maka akan terdapat bulatan hitam. Hal ini adalah demi keamanan agar user lain tidak dapat melihat password saat lagi menginput.



Jika user tidak menginput username atau password fieldnya maka akan menerima message yang diimplementasikan menggunakan JOptionPane. Berikut adalah code validasi untuk mengecek apakah username field dan password field kosong di UserController.

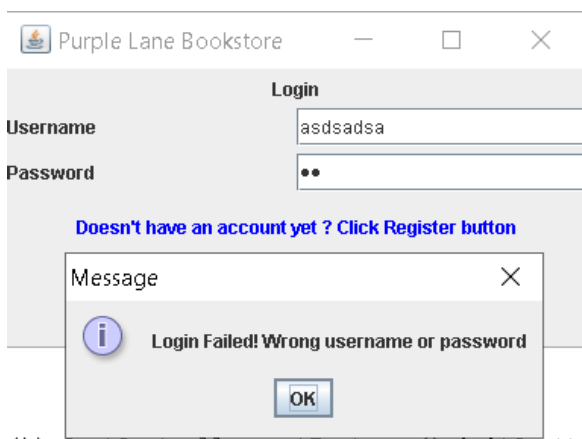
```

private boolean checkNamePass(String username, String password) {
    if(username.equals("")) {
        message = "Failed! Username cannot empty";
        return false;
    }
    if(password.equals("")) {
        message = "Failed! Password cannot empty";
        return false;
    }

    return true;
}

```

Juga terdapat validasi lain yang dimana jika username dan password tidak cocok dengan data di database maka akan menerima pesan error, dan sebaliknya jika cocok maka akan menerima pesan sukses dan langsung masuk ke main homenya sesuai dengan role masing-masing.



Berikut adalah tampilan jika data username dan password tidak cocok. Berikut adalah code method untuk loginnya di UserController.

```

public boolean login(String username, String password) {
    if(checkNamePass(username, password)) {
        UserModel tempUser = getOneUser(username, password);
        if(tempUser != null) {
            user.setUserId(tempUser.getUserId());
            user.setRoleId(tempUser.getRoleId());
            user.setUsername(tempUser.getUsername());
            user.setPassword(tempUser.getPassword());
            if(user.getRoleId() ==
                processRole("Customer").getRoleId()) CartController.getInstance().initCartList(user.getRoleId());

            message = "Success login";
        } else {
            message = "Login Failed! Wrong username or password";
            return false;
        }

        return true;
    } else {
        return false;
    }
}

```

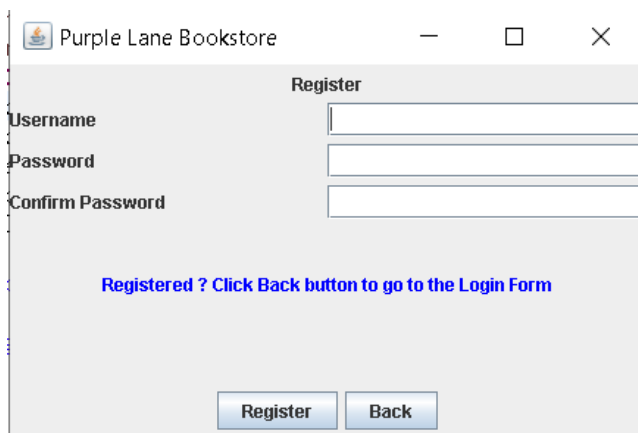
Di login ini, jika berhasil login, maka akan diarahkan user untuk ke menu yang berdasarkan rolenya seperti jika role Admin maka akan diarahkan ke menu untuk mengurus produk, jika

role Customer maka akan diarahkan ke menu customer. Berikut adalah kodenya di LoginView.java

```
loginBtn.addActionListener(new ActionListener() {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        if(UserController.getInstance().login(usernameField.getText(), passwordField.getText())) {  
            if(UserController.getInstance().getModel().getRoleId() ==  
                UserController.getInstance().processRole("Customer").getRoleId()) {  
                dispose();  
                CartController.getInstance().initCartList(UserController.getInstance().getModel().getUserId(  
                    UserController.getInstance().viewCustomerMenu());  
            } else if(UserController.getInstance().getModel().getRoleId() ==  
                UserController.getInstance().processRole("Promotion Team").getRoleId()) {  
                dispose();  
                PromoController.getInstance().viewManagePromoMenu();  
            } else if(UserController.getInstance().getModel().getRoleId() ==  
                UserController.getInstance().processRole("Manager").getRoleId()) {  
                dispose();  
                UserController.getInstance().viewManagerMenu();  
            } else if(UserController.getInstance().getModel().getRoleId() ==  
                UserController.getInstance().processRole("Admin").getRoleId()) {  
                dispose();  
                ProductController.getInstance().viewManageProductMenu();  
            }  
            System.out.println(UserController.getInstance().getMessage());  
        } else {  
            JOptionPane.showMessageDialog(loginBtn, UserController.getInstance().getMessage());  
            System.out.println(UserController.getInstance().getMessage());  
        }  
    }  
});
```

Jika user dengan role Customer, belum memiliki akun maka click tombol Register yang dimana akan mengarahkan ke page Registration Menu bagi Customer.

### 3. Registration Menu



Tampilan Code ini dapat di akses di package view dan pada class CustomerRegistrationView.java. Ini adalah tampilan untuk menu registrasi. Menu registrasi ini hanya diperuntukkan oleh Customer saja. Sehingga jika ada user yang melakukan registrasi disini, maka rolenya sudah di set secara otomatis menjadi Customer. Bagi user dengan role Admin dan Promotion Team, akun mereka hanya dapat dibuat oleh manager. **Dan bagi role Manager, akunya dapat dibuat dengan memasukkan datanya langsung ke database ke**

**dalam tabel users.** Tetapi sebelum itu, tentunya harus membuat tabel roles dan tabel users terlebih dahulu baru setelah itu memasukkan data roles ke dalam tabel roles . Berikut adalah query databasenya di file query-list.txt.

```
CREATE TABLE roles (
    roleId INT AUTO_INCREMENT PRIMARY KEY,
    roleName VARCHAR(30) NOT NULL UNIQUE
)

CREATE TABLE users (
    userId INT AUTO_INCREMENT PRIMARY KEY,
    roleId INT NOT NULL,
    username VARCHAR(30) NOT NULL UNIQUE,
    password VARCHAR(30) NOT NULL,
    FOREIGN KEY(roleId) REFERENCES roles (roleId) ON UPDATE CASCADE ON DELETE CASCADE
)

INSERT INTO roles (roleId, roleName) VALUES (NULL, "Manager"), (NULL, "Admin"), (NULL, "Promotion Team"), (NULL, "Customer")
```

**Untuk melihat query ini secara lengkap, bisa langsung mengecek di file query-list.txt**  
**Nama role yang di insert harus sesuai dengan query tersebut karena pencarian role berdasarkan dari nama rolenya. Kami juga mengeset roleName UNIQUE yang dimana agar tidak dapat meng-insert nama role yang sama.**

```
loginBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if(UserController.getInstance().login(usernameField.getText(), passwordField.getText())) {
            if(UserController.getInstance().getModel().getRoleId() ==
                UserController.getInstance().processRole("Customer").getRoleId()) {
                dispose();
                CartController.getInstance().initCartList(UserController.getInstance().getModel().getUserId(
                    UserController.getInstance().viewCustomerMenu());
            } else if(UserController.getInstance().getModel().getRoleId() ==
                UserController.getInstance().processRole("Promotion Team").getRoleId()) {
                dispose();
                PromoController.getInstance().viewManagePromoMenu();
            } else if(UserController.getInstance().getModel().getRoleId() ==
                UserController.getInstance().processRole("Manager").getRoleId()) {
                dispose();
                UserController.getInstance().viewManagerMenu();
            } else if(UserController.getInstance().getModel().getRoleId() ==
                UserController.getInstance().processRole("Admin").getRoleId()) {
                dispose();
                ProductController.getInstance().viewManageProductMenu();
            }
            System.out.println(UserController.getInstance().getMessage());
        } else {
            JOptionPane.showMessageDialog(loginBtn, UserController.getInstance().getMessage());
            System.out.println(UserController.getInstance().getMessage());
        }
    }
});
```

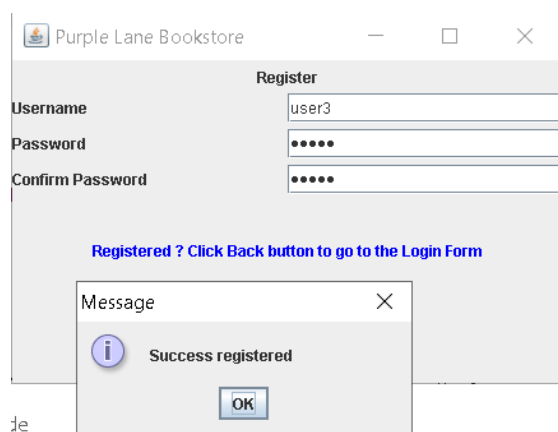
Di menu registrasi ini juga terdapat validasi yang dimana jika field username, password dan confirm password nya kosong, maka tidak dapat di proses dan akan memunculkan pesan error yang sama dengan pesan di login tadi tetapi dengan isi kalimat berbeda.

Kami juga menambahkan field confirm password yang dimana adalah demi meningkatkan user experience yang dapat membantu mencegah kesalahan user saat menginput password.

Tentunya dengan adanya field confirm password maka ada validasi baru yaitu password field dan confirm password field harus sama. Jika semua validasi tidak error, maka data akan dikirim ke database lewat UserModel

```
public void createAccount() {  
    String query = String.format("INSERT INTO %s VALUES(null, ?, ?, ?)", tableName);  
    PreparedStatement ps = con.prepareStatement(query);  
  
    try {  
        ps.setInt(1, roleId);  
        ps.setString(2, username);  
        ps.setString(3, password);  
        ps.executeUpdate();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

Dan setelah itu, user akan menerima pesan sukses dan secara otomatis akan menuju ke menu Login.



## 4. Manage Product Menu

ID	Name	Author	Price	Stock
1	Intelligent Saham	Marcus	500000	5
2	Smart English Version 2	James Larrington	100000	4
3	Business Marketing	Bill Thomas	120000	0
6	Marketing Smart	Luis	512333	3
7	Dongeng Fabel	Marzuki	55000	5
9	Buku Dagelan	Bob	30000	13

Product ID:

Product Name:

Product Author:

Product Price:

Product Stock:

Tampilan Code ini dapat di akses di package view dan pada class `ManageProductMenu.java`. Ini adalah menu untuk user dengan role Admin. Menu ini hanya dapat diakses jika anda sudah login dan memiliki role Admin. Disini akan ditampilkan tabel yang berisi daftar produk dengan kolom-kolom produk datanya. Di bawah tabel, terdapat beberapa field dan tombol yang memiliki nama sesuai dengan fungsinya. Produk ID Field di `setEditable(false)` karena agar Admin tidak bisa menambah ID sembarangan. Sehingga ini dapat mencegah kesalahan.

Untuk menginsert produk, terdapat beberapa validasi yaitu fieldnya tidak boleh ada satupun yang kosong, field product price dan product stock harus numeric, serta product stock tidak boleh di bawah atau sama dengan 0. Jika ada validasi yang dilanggar, maka process insert tidak dapat dilanjutkan, dan Admin akan menerima pesan error. Pada Insert, Product ID harus dibiarkan kosong karena di databasenya Product ID sudah di set `AUTO_INCREMENT`. Jika Product ID terlanjur ada datanya, tetapi Admin ingin menginsert produk baru, maka dapat mengklik tombol Reset yang dimana semua field akan di set menjadi kosong.



```

private boolean checkZeroAndNumVal(String productPrice, String productStock) {
    int tempPrice;
    try {
        tempPrice = Integer.parseInt(productPrice);
    } catch (NumberFormatException e) {
        message = "Cannot process! Product Price must be numeric!";
        return false;
    }
    if(tempPrice <= 0) {
        message = "Cannot process! Product Price must be more than zero!";
        return false;
    }
    int tempStock;
    try {
        tempStock = Integer.parseInt(productStock);
    } catch (NumberFormatException e) {
        message = "Cannot process! Product Stock must be numeric!";
        return false;
    }
    if(tempStock <= 0) {
        message = "Cannot process! Product Stock must be more than zero!";
        return false;
    }

    return true;
}

```

Jika semua validasi berhasil dilewati dengan benar, maka Admin akan menerima pesan sukses insert produk.

Untuk update, Admin dapat mengklik salah satu baris di tabel tersebut, maka data-data di baris tersebut akan secara otomatis masuk ke fieldnya.

ID	Name	Author	Price	Stock
1	Intelligent Saham	Marcus	500000	5
2	Smart English Version 2	James Larrington	100000	4
3	Business Marketing	Bill Thomas	120000	0
6	Marketing Smart	Luis	512333	3
7	Dongeng Fabel	Marzuki	55000	5
9	Buku Dagelan	Bob	30000	13

Product ID	6
Product Name	Marketing Smart
Product Author	Luis
Product Price	512333
Product Stock	3

Insert
Update
Delete
Reset

Admin dapat mengupdate nama produk, nama penulis, harga, dan stock tetapi tidak bisa mengupdate ID produk. Ini dikarenakan agar Admin tidak dapat sembarangan mengubah ID produk yang dimana ID Produk adalah atribut yang krusial. Validasi pada update juga sama dengan validasi pada insert tetapi ada tambahan validasi baru yaitu Product ID nya tidak boleh

kosong. Setelah semua validasi dilewati dengan benar maka akan ada pesan sukses update, dan produk yang sudah diupdate akan langsung tampil di tabel.

```
public boolean updateProduct(String productId, String productName, String productAuthor, String productPrice, String productStock) {  
    if(updateProductVal(productId, productName, productAuthor, productPrice, productStock)) {  
        int tempProdId = Integer.parseInt(productId);  
        int tempProdPrice = Integer.parseInt(productPrice);  
        int tempProdStock = Integer.parseInt(productStock);  
        update(tempProdId, productName, productAuthor, tempProdPrice, tempProdStock);  
        message = "Success Update";  
        return true;  
    }  
  
    return false;  
}
```

Untuk melakukan delete, caranya sama seperti saat mengupdate, yaitu tinggal memilih salah satu baris di tabel tersebut dan mengklik tombol Delete. Delete memiliki validasi yaitu bahwa produk ID field tidak boleh kosong. Kemudian akan muncul JOptionPane.showConfirmDialog yang dimana menanyakan apakah user yakin menghapus produk ini ?. Jika Yes, maka produk akan langsung di delete, tetapi jika No, maka tidak jadi menghapus.

```
public boolean deleteProduct(String productId) {  
    if(deleteProductVal(productId)) {  
        int tempProdId = Integer.parseInt(productId);  
        delete(tempProdId);  
        message = "Success Delete";  
        return true;  
    }  
  
    return false;  
}
```

Disini juga dapat melakukan pencarian produk, yang dimana harus menginput ID produknya ke dalam field Search di pojok kanan atas dan mengklik tombol Search.

Purple Lane Bookstore

Logout

Search product. Search

Admin Product View

ID	Name	Author	Price	Stock
1	Intelligent Saham	Marcus	500000	5
2	Smart English Version 2	James Larrington	100000	4
3	Business Marketing	Bill Thomas	120000	0
6	Marketing Smart	Luis	512333	3
7	Dongeng Fabel	Marzuki	55000	5
9	Buku Dagelan	Bob	30000	13

Product ID: 6

Product Name: Marketing Smart

Product Author: Luis

Product Price: 512333

Product Stock: 3

Insert Update Delete Reset

Terdapat validasi yaitu Search field tidak boleh kosong. Jika search field kosong, atau user menginput bukan numeric, atau jika user menginput ID Produk yang tidak ada di database, maka akan muncul pesan No Result. Jika sistem mengecek ID Produk ada di database maka, tabel akan langsung menampilkan datanya.

Purple Lane Bookstore

Logout

2 Search

Admin Product View

ID	Name	Author	Price	Stock
2	Smart English Version 2	James Larrington	100000	4

Product ID: 6

Product Name: Marketing Smart

Product Author: Luis

Product Price: 512333

Product Stock: 3

Click Reset button to view all product

Insert Update Delete Reset

Jika ingin melihat semua daftar produknya lagi, maka tinggal mengklik tombol Reset.

Jika Admin ingin logout, maka dapat mengklik tombol Logout dan akan langsung menuju ke

menu login.

## 5. Manage Promo Menu

ID	Code	Discount	Note
1	PURPLEBLESS	30000	Code yang diberikan untuk peraya...
4	BOOKSTORE	25000	Code bookstore

Promo ID

Promo Code

Product Discount

Product Note

Tampilan Code ini dapat di akses di package view dan pada class

ManagePromoMenuView.java. Ini adalah menu untuk user dengan role Promotion Team.

Menu ini dapat diakses dengan melakukan login terlebih dahulu dan user memiliki role Promotion Team. Tampilan menu ini memiliki kesamaan dengan tampilan pada Manage Product Menu. Dimana terdapat tabel dan dibawahnya terdapat field dan tombol.

Pada field, Promo ID juga di `setEditable(false)` karena agar mencegah kesalahan, dan Promo ID tidak dapat di rubah.

```
prodIdField = new JTextField();  
prodIdField.setEditable(false);
```

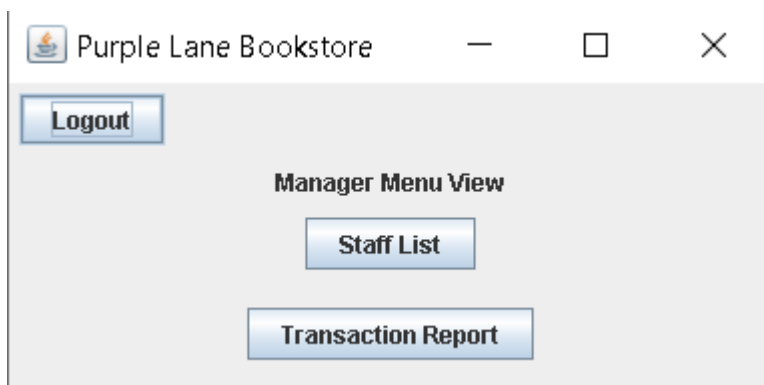
Jika ingin menginsert promo baru, maka terdapat beberapa validasi yaitu Promo ID harus kosong, jika Promo ID terlanjur ada datanya, maka bisa mengklik tombol Reset. Kemudian Promo Code, Discount field dan Note text area tidak boleh kosong. Kemudian validasi berikutnya yaitu Promo Discount harus minimal 15000, dan tidak boleh di bawah itu. Jika ada validasi yang dilanggar, maka akan memunculkan pesan error. Jika semua validasi sukses terlewati maka akan ada pesan sukses yang muncul, dan promo baru secara otomatis akan tampil di tabel.

Jika ingin mengupdate promo, maka Promotion Team dapat mengklik salah satu baris dari tabel tersebut, maka otomatis data-data di baris tersebut sesuai kolomnya akan muncul di field dan product notenya. Promotion Team tidak bisa mengupdate Promo ID, yang dimana hal ini demi mencegah kesalahan karena Promo ID adalah atribut yang krusial. Kemudian validasi lainnya sama dengan validasi saat menginsert promo baru. Jika semua validasi sukses, maka akan ada pesan sukses update, dan promo yang diupdate akan otomatis di ubah di tabel tersebut, serta field dan area datanya akan di set menjadi kosong.

Jika ingin menghapus promo, maka Promotion Team dapat mengklik salah satu baris dari tabel tersebut, maka otomatis data-data di baris tersebut sesuai kolomnya akan muncul di field dan product notenya. Kemudian Promotion Team klik tombol Delete, dan akan muncul JOptionPane.showConfirmDialog yang dimana menanyakan apakah user yakin menghapus promo ini ?. Jika Yes, maka promo akan langsung di hapus, tetapi jika No, maka tidak jadi menghapus. Berikut adalah code untuk menghapus produk di ProductController.

Jika Promotion Team ingin logout, maka tinggal mengklik tombol Logout, dan akan menuju ke menu Login.

## 6. Manager Menu



Tampilan Code ini dapat di akses di package view dan pada class ManagerMenuView.java. Ini adalah menu yang pertama kali akan diakses jika masuk sebagai role Manager. Untuk membuat akun Manager, dapat menambah langsung datanya di database, di tabel users dan pilih roleId yang sesuai dengan Manager. Bisa dengan query atau menambahkan nya menggunakan fungsi di phpMyAdmin nya.

Server: 127.0.0.1 » Basis data: purple\_lane\_bookstore » Tabel: users

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor

Kolom	Jenis	Fungsi	Tak Ternilai	Nilai
userid	int(11)	<input type="text"/>		<input type="text"/>
roleid	int(11)	<input type="text"/>		Manager - 1 <input type="text"/>
username	varchar(30)	<input type="text"/>		manager2
password	varchar(30)	<input type="text"/>		manager2

Di menu Manager ini, Manager dapat memilih untuk mengakses menu laporan transaksi atau ingin menambah data staff baru. Jika Manager ingin logout, maka dapat mengklik tombol Logout, dan akan langsung menuju ke login menu.

## 7. Staff List Menu

Purple Lane Bookstore

Back

Staff List View

ID	Username	Role
17	admin1	Admin
18	promo team1	Promotion Team
21	admin2	Admin
22	promo team2	Promotion Team
23	promo team3	Promotion Team
24	admin3	Admin
25	admin4	Admin

Username

Role

Password

Hire

Tampilan Code ini dapat di akses di package view dan pada class ManageStaffMenuView.java. Ini adalah tampilan menu yang diakses jika mengklik tombol Staff List di Manager Menu. Disini Manager dapat melihat jumlah staff dan rolenya masing-masing. Ini ditambahkan demi user experience Manager. Manager juga dapat membuat akun untuk staff yang memiliki role Admin atau Promotion Team. Terdapat validasi yaitu username tidak boleh kosong, dan password juga tidak boleh kosong. Kemudian dapat memilih role di Combo Box role tersebut.

ID	Username	Role
17	admin1	Admin
18	promo team1	Promotion Team
21	admin2	Admin
22	promo team2	Promotion Team
23	promo team3	Promotion Team
24	admin3	Admin
25	admin4	Admin

Username:

Role: Admin ▼

Password:

Jika semua validasi sukses, maka akan ada pesan sukses dan akun Admin atau Promotion Team sudah siap digunakan.

Kemudian terdapat tombol Back yang dimana akan menuju ke menu Manager.

## 8. Transaction Report Menu

Purple Lane Bookstore Transaction Report

Select Month: Desember ▼

Select Year: 2020 ▼

Tampilan Code ini dapat di akses di package view dan pada class TransactionReportMenuView.java. Ini adalah menu untuk user dengan role Manager. Menu ini dapat diakses jika Manager mengklik tombol Transaction Report di menu Manager. Pada saat

awal tampilan menu ini, terdapat tabel yang merupakan daftar semua transaksi history dari user dengan role Customer.

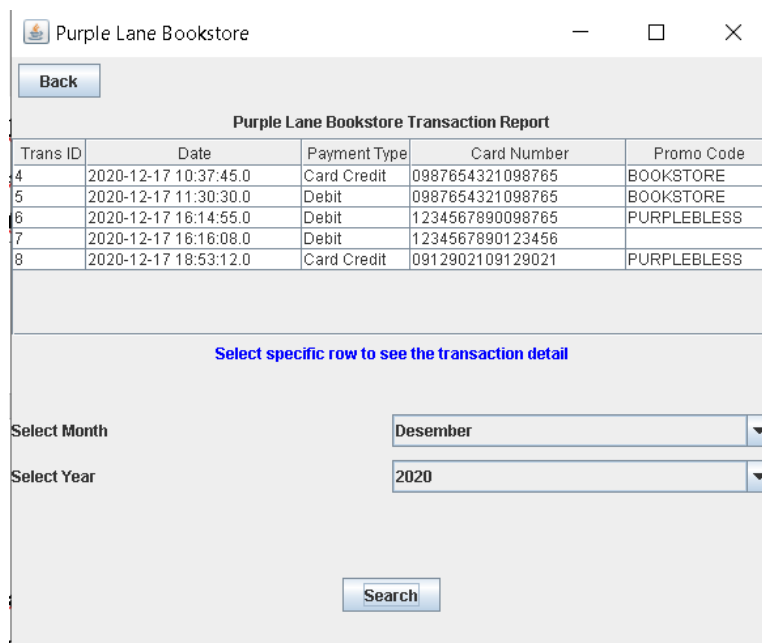
Dan di bawahnya terdapat Combo Box Month dan Year. Kami membuat agar Combo Box Month dan Year nya itu di tampilkan sesuai dengan bulan dan tahun tanggal sekarang.

Code untuk mengambil bulan dan tahun sekarang yaitu di TransactionController.

```
public String getCurrentMonth() {
    LocalDate currentdate = LocalDate.now();
    int currentMonth = currentdate.getMonthValue();
    return getMonthName(currentMonth);
}

public String getCurrentYear() {
    LocalDate currentdate = LocalDate.now();
    int currentYear = currentdate.getYear();
    return String.valueOf(currentYear);
}
```

Jika ingin melihat sejarah transaksi Customer di bulan dan tahun tertentu, maka dapat memilih bulannya di Combo Box Month dan memilih tahunnya di Combo Box Year, maka selanjutnya mengklik Search, dan jika ada sejarah transaksi di bulan dan tahun tersebut, maka data tersebut akan tampil di tabel dan label informasi biru akan muncul yang dimana memberikan pesan kepada user.



Purple Lane Bookstore

Back

Purple Lane Bookstore Transaction Report

Trans ID	Date	Payment Type	Card Number	Promo Code
4	2020-12-17 10:37:45.0	Card Credit	0987654321098765	BOOKSTORE
5	2020-12-17 11:30:30.0	Debit	0987654321098765	BOOKSTORE
6	2020-12-17 16:14:55.0	Debit	1234567890098765	PURPLEBLESS
7	2020-12-17 16:16:08.0	Debit	1234567890123456	
8	2020-12-17 18:53:12.0	Card Credit	0912902109129021	PURPLEBLESS

Select specific row to see the transaction detail

Select Month: Desember

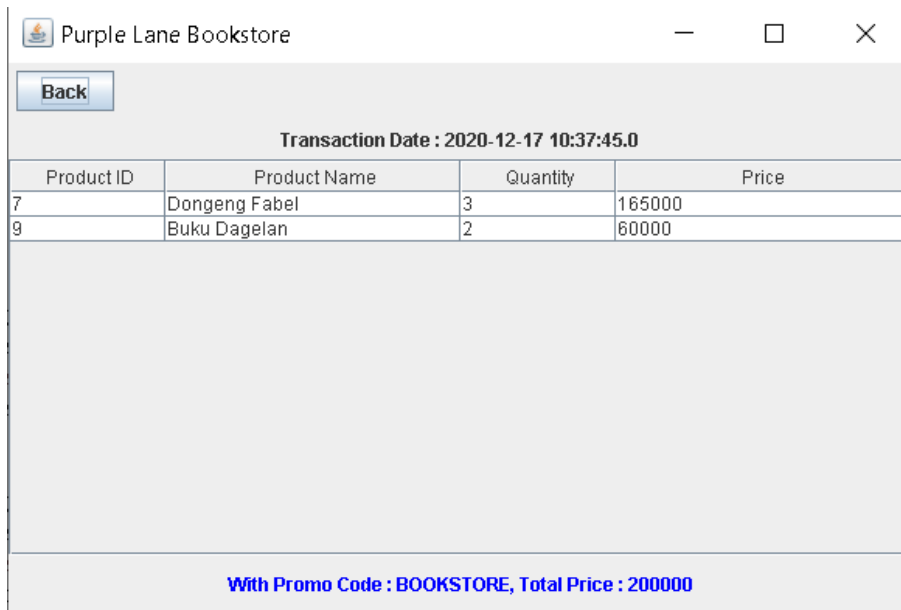
Select Year: 2020

Search

Kemudian Manager bisa mengklik salah satu baris dari tabel tersebut untuk melihat detail transaksinya.



## 9. Transaction Detail Menu



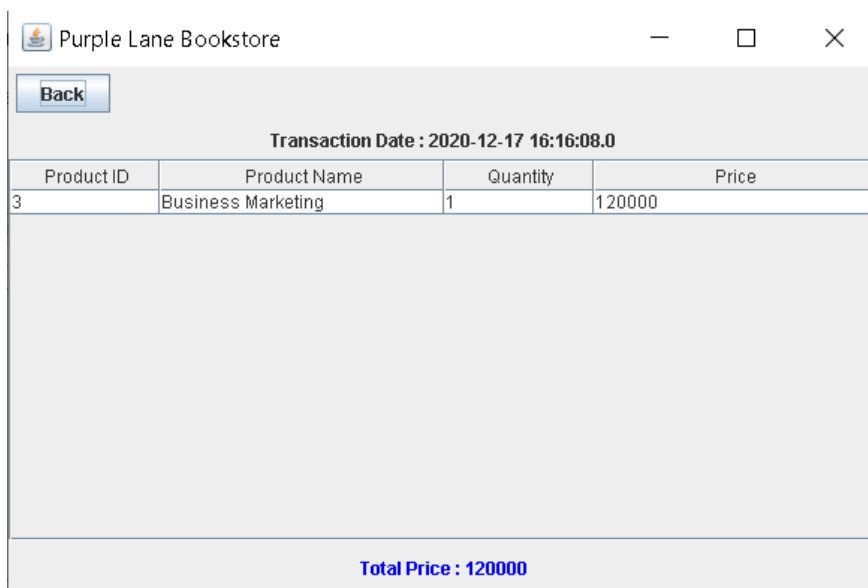
The screenshot shows a window titled "Purple Lane Bookstore" with a "Back" button in the top left. Below the button, the text "Transaction Date : 2020-12-17 10:37:45.0" is displayed. A table with four columns (Product ID, Product Name, Quantity, Price) contains two rows of transaction data. Below the table, a large empty rectangular area is present. At the bottom, a blue text label indicates a discount applied using a promo code.

Product ID	Product Name	Quantity	Price
7	Dongeng Fabel	3	165000
9	Buku Dagelan	2	60000

With Promo Code : BOOKSTORE, Total Price : 200000

Tampilan Code ini dapat di akses di package view dan pada class TransactionDetailView.java. Ini adalah menu detail transaksi yang diakses melalui jika user Manager atau Customer mengklik salah satu baris di tabel Transaction Report atau Transaction History. Disini menampilkan tanggal transaksinya, serta menampilkan data-data seperti nama produk yang di beli, jumlah dan harga berdasarkan quantitynya tersebut. Kemudian di bawah tabel ada label yang Customer menggunakan promo code, maka total harganya akan langsung di kurangi dengan discount yang disediakan oleh promo codenya.

Jika Customer tidak menggunakan promo code maka tidak ada discount serta tidak ada label promo codenya.



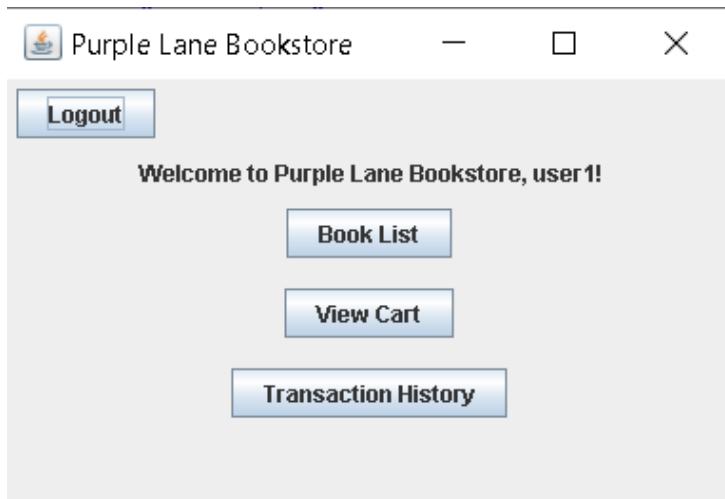
The screenshot shows a window titled "Purple Lane Bookstore" with a "Back" button in the top left. Below the button, the text "Transaction Date : 2020-12-17 16:16:08.0" is displayed. A table with four columns (Product ID, Product Name, Quantity, Price) contains one row of transaction data. Below the table, a large empty rectangular area is present. At the bottom, a blue text label shows the total price without any discount.

Product ID	Product Name	Quantity	Price
3	Business Marketing	1	120000

Total Price : 120000

Tombol Back untuk menuju kembali ke Transaction Report jika user tersebut dengan role Manager, dan kembali ke Transaction History jika user tersebut dengan role Customer.

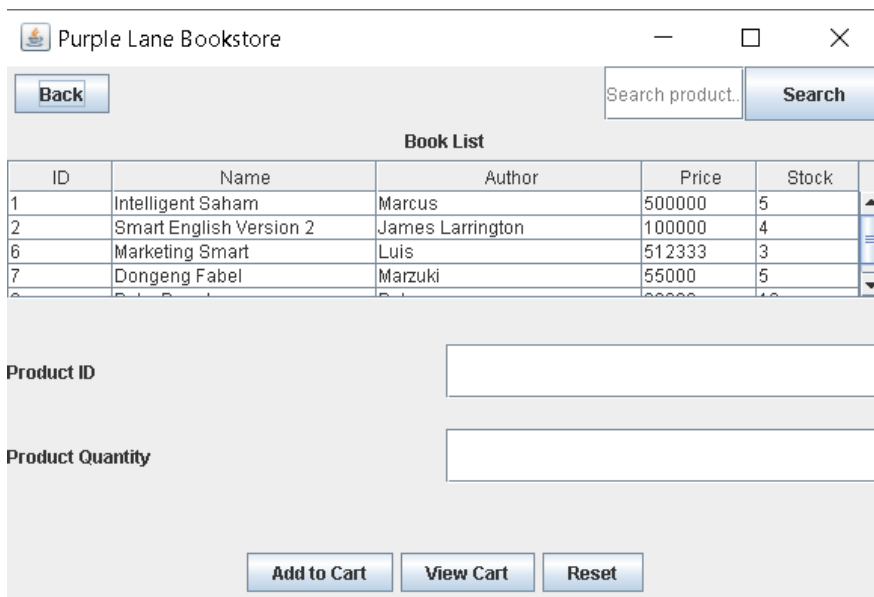
## 10. Customer Menu



Tampilan Code ini dapat di akses di package view dan pada class CustomerMenuView.java. Setelah kita login sebagai Customer, maka tampilan inilah yang akan pertama kali tampil saat setelah kita sukses login. Terdapat tombol Book List untuk ke menu Book List, tombol View Cart untuk melihat keranjang belanja, dan Transaction History untuk ke sejarah pembelian.

Jika Customer ingin logout, maka dapat mengklik tombol Logout, dan akan langsung menuju ke menu Login.

## 11. Add To Cart Menu



ID	Name	Author	Price	Stock
1	Intelligent Saham	Marcus	500000	5
2	Smart English Version 2	James Larrington	100000	4
6	Marketing Smart	Luis	512333	3
7	Dongeng Fabel	Marzuki	55000	5

Tampilan Code ini dapat di akses di package view dan pada class AddToCartMenuView.java. Menu ini dapat di akses jika anda mengklik tombol Book List pada menu Customer. Disini Customer dapat melihat daftar buku yang tersedia dan Customer dapat mencari produk dengan menginput produk ID. Fungsi search ini sama seperti pada fungsi search di Manage Product Menu. Di tabel tersebut, semua produk akan di tampilkan kecuali dengan produk yang memiliki stok 0. Sehingga jika ada produk yang memiliki stok 0, maka tidak akan tampil di menu Add To Cart, tetapi masih tampil di menu Manage Product yang di operasikan oleh Admin.

Berikut adalah contoh validasi jika produk stok 0, maka tidak akan tampil di Add To Cart Menu. Code ini dapat di akses di AddToCartMenuView.java.

```

public void fetchTableData() {
    Vector<Object> tProducts = new Vector<>();

    for (Model model : ProductController.getInstance().getAllProduct()) {
        Vector<String> tempStringList = new Vector<>();
        ProductModel tempProduct = (ProductModel) model;
        if(tempProduct.getProductStock() == 0) continue;

        tempStringList.add(String.valueOf(tempProduct.getProductId()));
        tempStringList.add(tempProduct.getProductName());
        tempStringList.add(tempProduct.getProductAuthor());
        tempStringList.add(String.valueOf(tempProduct.getProductPrice()));
        tempStringList.add(String.valueOf(tempProduct.getProductStock()));

        tProducts.add(tempStringList);
    }

    DefaultTableModel dtm = new DefaultTableModel(tProducts, tHeader) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };

    table.setModel(dtm);
    setColumnWidth();
}

```

Customer dapat menambah produk ke Cart dengan mengklik salah satu baris di tabel produk tersebut atau bisa dengan manual mengisi product ID dan product Quantity field nya. Terdapat validasi jika ingin Add To Cart yaitu Product ID dan Product Quantity field tidak boleh kosong, serta Product Quantity tidak boleh lebih dari produk stok nya. Kemudian jika Customer ada menginput product ID yang tidak ada di database, maka akan muncul pesan error bahwa produk dengan ID tersebut tidak ada di database.

Jika Customer menambah produk ke Cart yang dimana produk tersebut sudah ada di Cart, maka quantity produk di Cart tersebut akan di tambah. Disini juga terdapat validasi dimana quantity product di Cart + quantity yang di input di Add to Cart Menu ini tidak boleh lebih dari produk stoknya.

Berikut code yang dapat dilihat di CartController.

```

public boolean addToCart(String productId, String quantity) {
    if(CartValidation(productId, quantity)) {
        int tempUserId = UserController.getInstance().getModel().getUserId();
        int tempProductId = Integer.parseInt(productId);
        int tempQuantity = Integer.parseInt(quantity);
        CartModel cart = getOneCart(tempUserId, tempProductId);
        if(cart == null) {
            //insert
            insert(tempUserId, tempProductId, tempQuantity);
            cartList.add(new CartModel(tempUserId, tempProductId, tempQuantity));
        } else {
            ProductModel product = ProductController.getInstance().getOneProduct(tempProductId);
            if((tempQuantity + cart.getQuantity()) > product.getProductStock()) {
                message = "The inputed quantity must be below product stock!";
                return false;
            }
            //update
            tempQuantity = tempQuantity + cart.getQuantity();
            update(tempUserId, tempProductId, tempQuantity);
            setCartListQuantity(tempUserId, tempProductId, tempQuantity);
        }

        message = "Add to cart success!";
        return true;
    }
    return false;
}

```

Jika ingin melihat keranjang maka dapat mengklik tombol View Cart.  
Terdapat tombol Back yang dimana dapat kembali ke Customer menu.

## 12. Manage Cart Menu

Purple Lane Bookstore

Add More Main Menu

user1's Cart Menu

ID	Name	Quantity	Total Price
2	Smart English Version 2	1	100000
7	Dongeng Fabel	2	110000

Product ID

Product Quantity

Click Checkout button if you ready to checkout

Update Checkout

Tampilan Code ini dapat di akses di package view dan pada class ManageCartMenuView.java. Disini adalah menu yang dimana Customer dapat melihat daftar keranjang yang ingin dipesan. Terdapat tombol Update dimana jika Customer ingin mengupdate produk quantity nya maka dapat mengklik manual salah satu baris di tabel tersebut atau mengisi secara manual dengan validasi Product ID field tidak boleh kosong dan juga Product Quantity field tidak boleh kosong. Serta produk quantity yang ingin dirubah tidak boleh lebih dari product stock.

Purple Lane Bookstore

Add More Main Menu

user1's Cart Menu

ID	Name	Quantity	Total Price
2	Smart English Version 2	1	100000
7	Dongeng Fabel	2	110000

Product ID: 7

Product Quantity: 2

Click Checkout button if you ready to checkout

Update Checkout

Setelah itu Customer dapat checkout dengan mengklik tombol Checkout, Customer tidak dapat checkout jika tidak ada produk sama sekali di menu Cart.

Terdapat tombol Main Menu yang akan kembali ke menu Customer, dan tombol Add More untuk menuju ke menu Add To Cart.

### 13. Checkout Menu

Purple Lane Bookstore

Back

user1's Checkout Menu

ID	Name	Quantity	Total Price	Select
2	Smart English Version 2	1	100000	<input type="checkbox"/>
7	Dongeng Fabel	2	110000	<input type="checkbox"/>

Select which product(s) you want to pay, if you ready click Payment button

Payment

Tampilan Code ini dapat di akses di package view dan pada class CheckoutMenuView.java. Disini adalah menu Checkout yang dapat diakses di tombol Checkout pada Manage Cart Menu. Disini Customer dapat memilih produk-produk yang akan dibayar dengan mengklik

baris di tabel tersebut.

Purple Lane Bookstore

Back

user1's Checkout Menu

ID	Name	Quantity	Total Price	Select
2	Smart English Version 2	1	100000	<input type="checkbox"/>
7	Dongeng Fabel	2	110000	<input checked="" type="checkbox"/>

Select which product(s) you want to pay, if you ready click Payment button

Payment

Jika Customer tidak men-select produk satupun di menu Checkout ini maka Customer tidak dapat menuju ke menu Payment.

Purple Lane Bookstore

Back

user1's Checkout Menu

ID	Name	Quantity	Total Price	Select
2	Smart English Version 2	1	100000	<input type="checkbox"/>
7	Dongeng Fabel	2	110000	<input type="checkbox"/>

Select which product(s) you want to pay, if you ready click Payment button

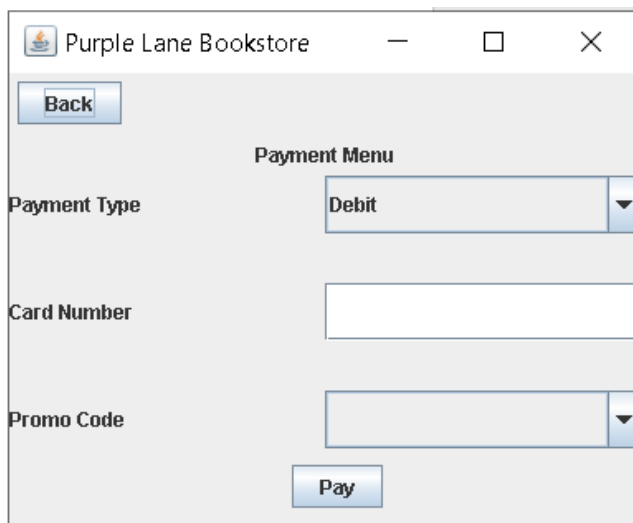
Message

Cannot pay! There is no selected cart!

OK

Terdapat tombol Back untuk menuju ke Manage Cart Menu.

## 14. Payment Menu

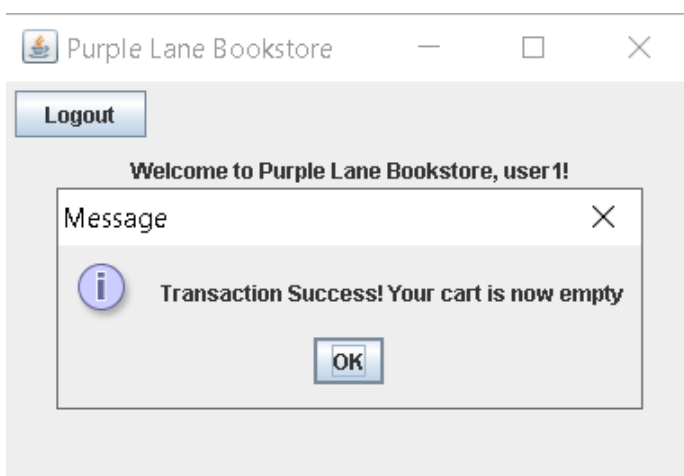


The screenshot shows a window titled "Purple Lane Bookstore" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a "Back" button in the top left corner. The main area is titled "Payment Menu". It contains three input fields: "Payment Type" with a dropdown menu currently showing "Debit", "Card Number" with a text input field, and "Promo Code" with a dropdown menu. At the bottom right of the form is a "Pay" button.

Tampilan Code ini dapat di akses di package view dan pada class PaymentMenuView.java. Ini adalah menu Payment dimana Customer dapat memilih Payment Type, dan Promo Code. Promo Code Combo Box dapat dibiarkan kosong karena Customer dapat memilih apakah ingin menggunakan promo atau tidak.

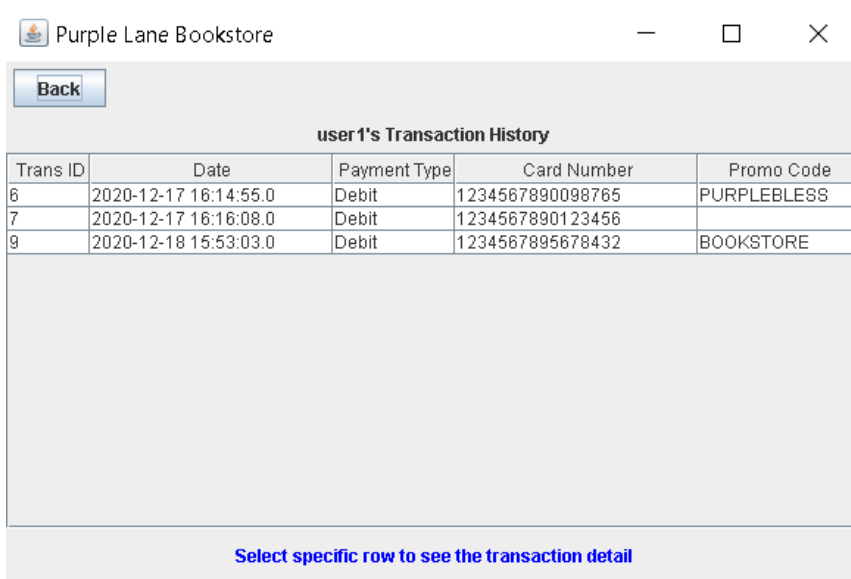
Untuk melakukan pembayaran, Customer wajib mengisi Card Number fieldnya dengan minimal 16 digit. Jika Customer tidak jadi melakukan pembayaran, maka dapat mengklik tombol Back untuk menuju ke menu Checkout.

Jika tombol Pay di klik, dan semua validasi sukses maka akan mendapat pesan sukses dan langsung otomatis menuju ke Customer menu. Data di Cart serta data di checkout akan kembali kosong.





## 15. Transaction History Menu



Tampilan Code ini dapat di akses di package view dan pada class TransactionHistoryMenuView.java. Ini adalah menu Transaction History. Menu ini dapat diakses oleh Customer saat mengklik tombol Transaction History di Customer Menu. Data-data di tabel tersebut merupakan transaksi yang sudah dilakukan oleh user1. Jika ingin melihat detail transaksi maka dapat mengklik salah satu baris di tabel tersebut.

Terdapat tombol Back untuk menuju kembali ke Customer Menu.

### • Reference

- <https://www.tutorialspoint.com/how-to-get-current-day-month-and-year-in-java-8> Link/
- [https://www.w3schools.com/sql/func\\_sqlserver\\_month.asp/](https://www.w3schools.com/sql/func_sqlserver_month.asp/)
- <https://stackoverflow.com/questions/37803739/how-to-get-month-from-timestamp-value/>
- <https://stackoverflow.com/questions/2997364/select-by-month-of-a-field>
- <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- <https://docs.oracle.com/javase/tutorial/uiswing/components/table.html>
- <https://docs.oracle.com/javase/8/docs/api/javax/swing/JFrame.html>
- [https://www.w3schools.com/sql/sql\\_and\\_or.asp](https://www.w3schools.com/sql/sql_and_or.asp)
- <https://alvinalexander.com/java/edu/pj/jdbc/jdbc0003/>
- [https://www.w3schools.com/sql/sql\\_primarykey.ASP](https://www.w3schools.com/sql/sql_primarykey.ASP)
- <https://stackoverflow.com/questions/16213836/java-swing-jtextfield-set-placeholder>
- <https://stackoverflow.com/questions/11999560/get-combobox-value-in-java-swing>
- <https://mkyong.com/swing/java-swing-how-to-make-a-confirmation-dialog/>

- <https://stackoverflow.com/questions/953972/java-jtable-setting-column-width>
- <https://stackoverflow.com/questions/18803298/how-to-make-textarea-unclickable-in-desktop-applicationjava>

- **Group Member**

- 2201758303 – Luis Genesis
- 2201757093 – Jackie Leonardy