|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197  Business Application Development |
| **Computer Science** | **O212-ISYS6197-AE01-00** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Java 8  Eclipse 2020.6  XAMPP 7.4.7  MySQL Java Connection Library 5.1.49 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, proyek dan uap untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment, project, and final exam collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| JAVA, CLASS | JAVA, CLASS, SQL |

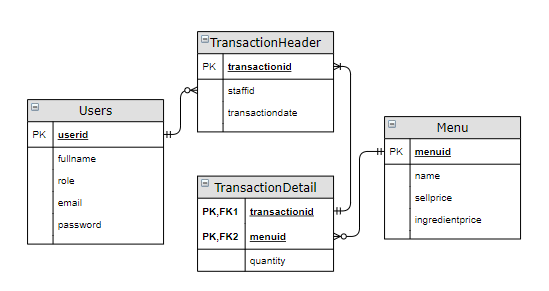
## Soal

*Case*

**Amore POS (Point of Sales)**

**Amore** is a business owner and is opening a new restaurant. To manage all the transactions, Amore asked you to create the POS system for the store. The program should be made using Java Programming Language.

There are some requirements to build this application:



Execute the provided create and insert query file on “**amorepos**” database! (“create+insert.sql”)

1. **Login Form**

This is the first form which will appear when the program is run. In this form, user can **Login** using their accounts on the program.

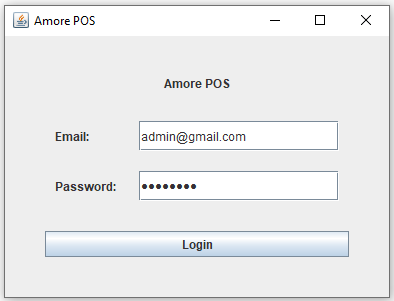


Figure 1. Login View

**Components:**

* + - **JLabel** for **Title, Email,** and **Password.**
    - **JTextField** for **input Email**.
    - **JPasswordField** for **input Password.**
    - **JButton** for **Login.**

**Description:**

* If user clicks **Login** Button, validate:
  + **Email** must be filled
  + **Password** must be filled

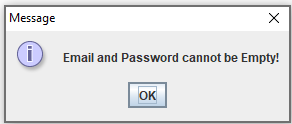


Figure 2. Email or Password is empty

* + **Email** and **Password** must be correct, according to the **existing data in the database**

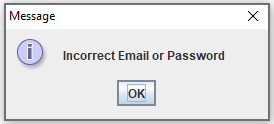


Figure 3. User inputs wrong Email or Password

* If user has successfully logged in, display a welcome message using **JOptionPane** along with the **user’s fullname**.

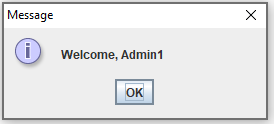


Figure 4. Welcome Message

* After user clicks the **‘OK’** button, close the **Login Form,** and open the **Main Form.**

1. **Main Form**

**Main Form** is the main frame. This form is made with **JFrame** with a **JMenu bar** above the **form**.



Figure 5. Main Form

**Components**:

* **JDesktopPane** for main pane
* **JMenu bar** for **Account**, **Manage**, **Finance**, and **Transaction**.
* **JMenuItem** for **Logout**, **Manage Account**, **Manage Restaurant Menu**, **View Monthly Report**, and **Create Transaction**.

**Description**:

* The menu bar will have an **Account menu** with a **Logout menu item**. When user **click** **Logout**, it will close the **Main Form** and open the **Login Form**

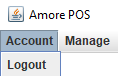


Figure 6. Account Menu

* The **rest of the Menu bar will be different**, depending on the role of the current logged in user:
  + Cashier
    - The Menu Bar will have a **Transaction Menu** with a **Create Transaction MenuItem**, when clicked it will open A **Transaction Form** using **JInternalFrame**

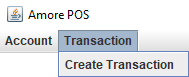


Figure 7. Transaction Menu

* + Admin
    - The Menu Bar will have a **Manage Menu** with **Manage Restaurant Menu**, and **Manage Account** MenuItem**:**

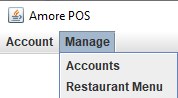


Figure 8. Transaction Menu

* + - If Restaurant Menu is clicked, it will show the **Manage Restaurant Form** using **JInternalFrame** and will be displayed inside the **Main Form.**
    - If Accounts is clicked, it will show the **Manage Account Form** using **JInternalFrame** and will be displayed inside the **Main Form.**
  + Accountant
    - The Menu Bar will have a **Finance Menu** with a **View Monthly Report** MenuItem, when clicked it will show the **Finance Form** using **JIternalFrame** and will be displayed inside the **Main Form**.

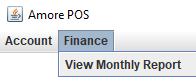


Figure 9. Finance Menu

1. **Manage Account Form**

The **Manage Account Form** will use **JInternalFrame** and will be displayed within **Main Form.**

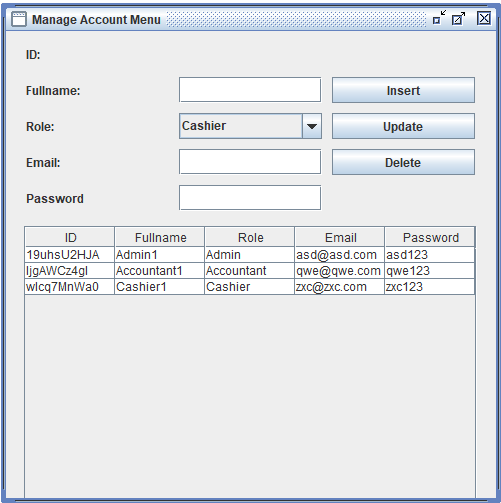


Figure 10. Manage Account Form

**Components:**

* **JLabel** for **ID Title**, **UserID**, **Fullname Title**, **Role Title**, **Email Title**, and **Password Title**.
* **JTextBox** for **input Fullname**, **input Email,** and **input Password.**
* **JComboBox** for **selecting Role:**
  + **Accountant**
  + **Admin**
  + **Cashier**
* **JButton** for **Insert**, **Update**, **Delete**
* **JTable** for **Accounts Table**

**Description:**

* If a table row is selected, the form will **fill UserId, Fullname, Role, Email, Password** with the **selected row’s data**

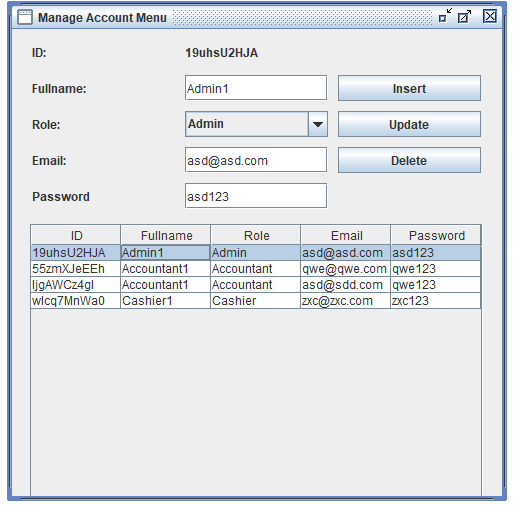


Figure 11. Table Row Click

* If **insert button** is clicked, validate:
  + **Fullname** must be **filled.**
  + **Email** must be **filled.**
  + **Password** must be **filled.**



Figure 12. Empty Field Validation

* + **Email**
    - **Does not starts** with **‘@’**
    - **Does not ends** with **‘@’**
    - **Does not ends** with **‘.’**
    - **Have only 1** **‘@’**
    - **‘.’ must not** be directly **after** or **before** **‘@’** (ex: “hello@.com” is incorrect)
    - Otherwise display a message

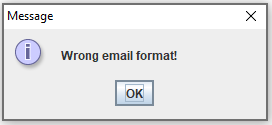


Figure 13. Wrong Email Format Validation

* + - **Must not exists** in the current database, otherwise display a message

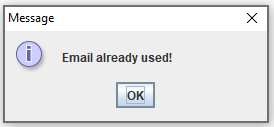


Figure 14. Email Already Used Validation

* + **Password** must be **alphanumeric**, otherwise display a message

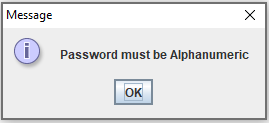


Figure 15. Password Alphanumeric Validation

* + Generate user id with the following format:



* + Show message if **insert is successful, insert the account into the database**

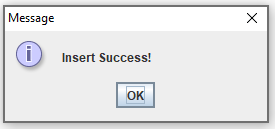


Figure 16. Insert Success Message

* + And **clear the fields** in the form
* If **update button** is clicked, validate:
  + Row to be updated **must be selected** first, otherwise display a message

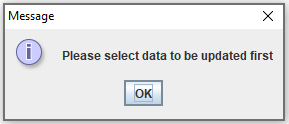


Figure 17. Update Row Not Selected

* + **All Fields** must be **filled**



Figure 18. Empty Field Validation

* + **Email**
    - **Does not starts** with **‘@’**
    - **Does not ends** with **‘@’**
    - **Does not ends** with **‘.’**
    - **Have only 1** **‘@’**
    - **‘.’ must not** be directly **after** or **before** **‘@’** (ex: “hello@.com” is incorrect)
    - Otherwise display a message

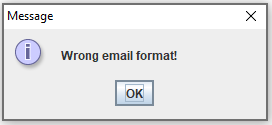


Figure 19. Wrong Email Format Validation

* + - **Must not exists** in the current database, otherwise display a message

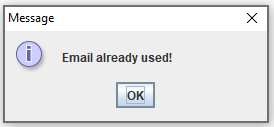


Figure 20. Email Already Used Validation

* + **Password** must be **alphanumeric**, otherwise display a message

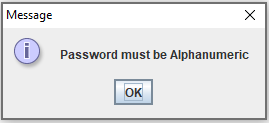


Figure 21. Password Alphanumeric Validation

* + Show message if **update is successful, update** the **account** from the database **based on the selected User ID**

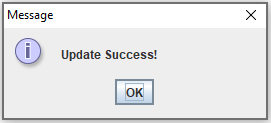


Figure 22. Update Success Message

* + And **clear the fields** in the form, and **deselect** the table row
* If **delete button** is clicked, validate:
  + Row to be deleted **must be selected** first, otherwise display a message:

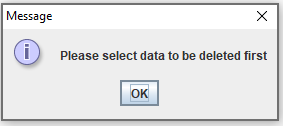


Figure 23. Delete Row Not Selected

* + Show message if **delete is successful, delete** the **account** from the database **based on the selected User ID.** After that **clear, the fields** in the form, and **deselect** the table row

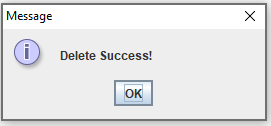


Figure 24. Delete Row Not Selected

1. **Manage Menu Form**

The **Manage Menu Form** will use **JInternalFrame** and will be displayed within **Main Form**

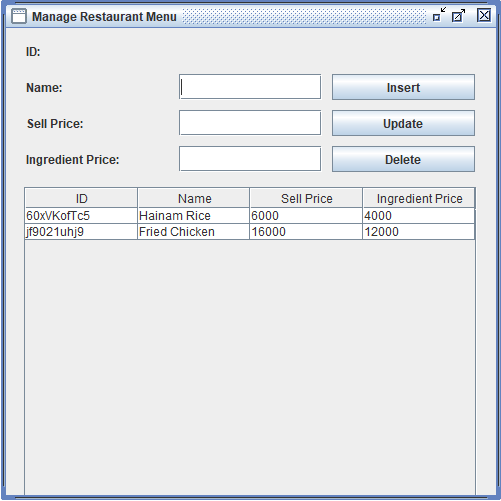


Figure 25. Manage Menu Form

**Components:**

* **JLabel** for **ID, MenuID, Name, Sell Price, Ingredient Price**
* **JTextBox** for **input Name, input Sell Price, input Ingredient Price**
* **JButton** for **Insert, Update, Delete**
* **JTable** for displaying **Menu Data**

**Description:**

* If a table row is selected, the form will **fill itself** with the **selected row’s data**

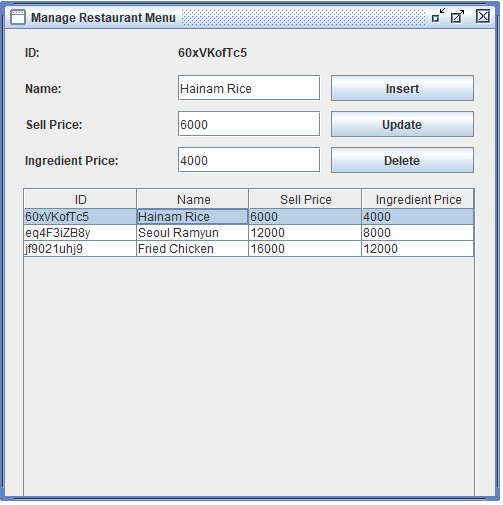


Figure 26. Manage Menu Form

* If user clicked the **insert** button
  + **Name** must be **filled.**
  + **Sell Price** must be **filled.**
  + **Ingredient Price** must be **filled.**

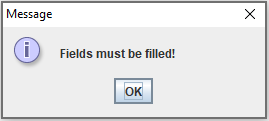


Figure 27. Empty Field Validation

* + Validate that the **prices** must be a number (both **Sell Price** and **Ingredient Price**)

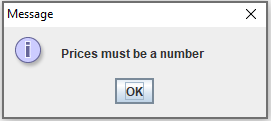


Figure 28. Price and Ingredient Validation

* + Generate menu id with the following format:



* + Show message if **insert is successful, insert the item to the database**, clear the fields, and clear the table’s selected row

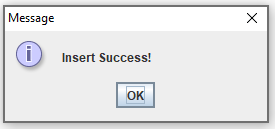


Figure 29. Insert Success Message

* If user clicked the **update** button
  + Validate that a row data must be selected

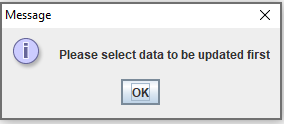


Figure 30. Update Row Not Selected

* + Validate that the **fields** are **not empty**

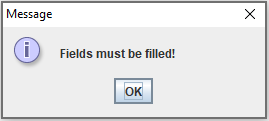


Figure 31. Empty Field Validation

* + Validate that the **prices** must be a number (both **Sell Price** and **Ingredient Price**)

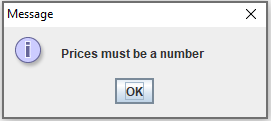


Figure 32. Price and Ingredient Validation

* + Show message if **update** is **successful, update** the **item** from database **based on the selected item**, clear the fields, and clear the table’s selected row

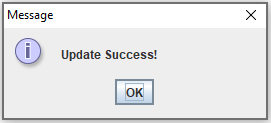


Figure 33. Update Success Message

* If user clicked the **delete** button
  + Validate that a row data must be selected

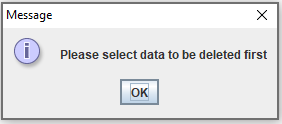


Figure 34. Delete Row Not Selected

* + Show message if **delete** is **successful, delete** the **item** from database **based on the selected item**, clear the fields, and clear the table’s selected row

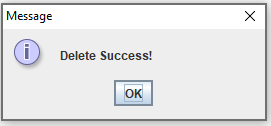


Figure 35. Delete Success Message

1. **Transaction Form**

The **Transaction Form** will use **JInternalFrame** and will be displayed within **Main Form**

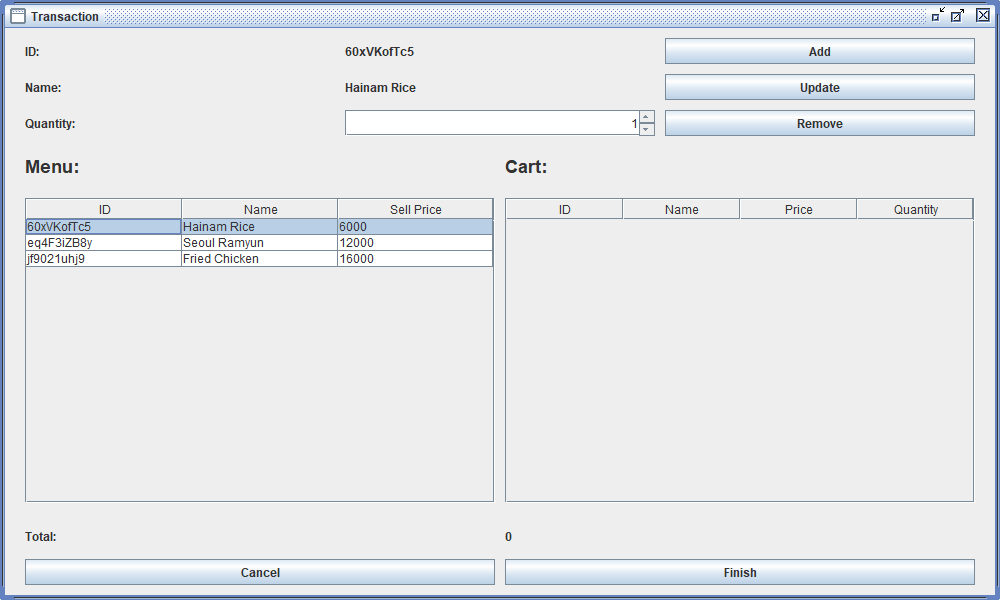


Figure 36. Transaction Form

**Components:**

* **JLabel** for **ID Title, MenuId, Name Title, MenuName, Menu Title, Cart Title, Total Title, Total Price, Quantity Title.**
* **JSpinner** for **Quantity** input.
* **JButton** for **Add, Update, Remove, Finish, Cancel.**
* **JTable** for **Menu Table**, and **Transaction Cart Table.**

**Description:**

* If user clicks a row from the **Menu Table** or **Cart Table**, the fields will **fill itself with the row data**

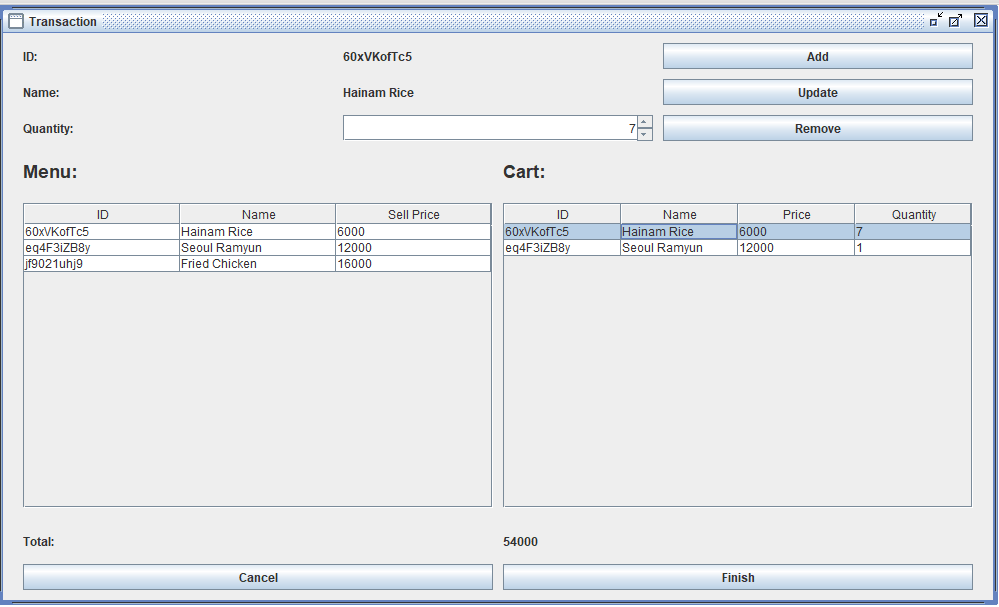


Figure 38. Manage Menu Form

* If user clicks **add**:
  + Validate that a **menu row** must be selected

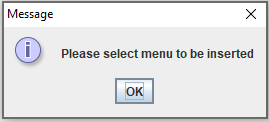


Figure 39. Delete Success Form

* + If **the selected item** is **not in cart** yet, **add** it to the cart
  + If **the selected item** is **already in a cart**, then **increases the quantity of the Selected Product in Cart’s Table** based on **the Quantity input.**
  + **Display a message** if add is **successful**, clear the fields, and **clear the table’s** selected row

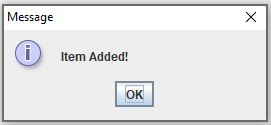


Figure 40. Item Added Message

* If user clicks **update**:
  + Validate that a **cart row** must be **selected**

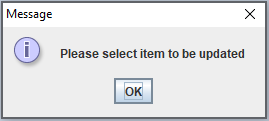


Figure 41. Cart Row not Selected

* + Validate **quantity** must be **more than 0**

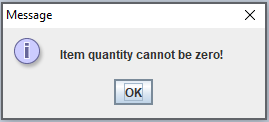


Figure 42. Quantity Validation

* + Otherwise, it will execute the **Update** query and display message if **update** is success, **update the item in cart based on the selected item**, clear the fields, and clear the table’s selected row

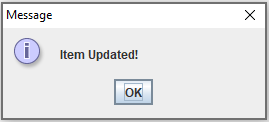


Figure 43. Update Menu Form

* If user clicks **remove**:
  + Validate that a **cart row** must be selected

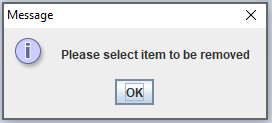


Figure 44. Cart Row not Selected

* + Display message if **delete** is success, **delete the item from cart based on the selected item**, clear the fields, and clear the table’s selected row

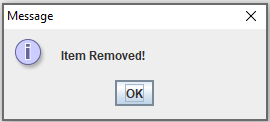


Figure 45. Delete Menu Message

* If user clicks **finish,** save the transaction data to the database with the following id:



* If user clicks **cancel**, the **transaction will be canceled** and the **form will close**

1. **Finance Form**

The **Finance Form** will use **JInternalFrame** and will be displayed within **Main Form**

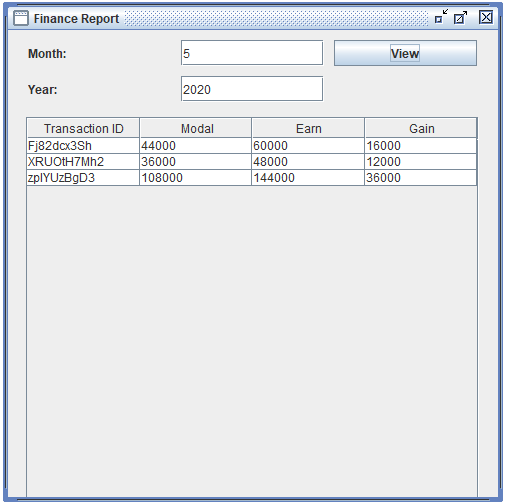


Figure 46. Finance Form

**Components:**

* **JLabel** for **Month**, **Year**
* **JTextField** for **input Month**, **input Year**
* **JButton** for **View**
* **JTable** for **Sales Report** obtained by Trasanctions that occurs in the inputed **Month** and **Year**
  + **TransactionID**
  + **Modal** = **Quantity** \* **Ingredient Price**
  + **Earn** = **Quantity** \* **Sell Price**
  + **Gain** = **Earn - Modal**

**Description:**

* If user clicked the **view button:**
  + Validate that the inputted month and year must be a number

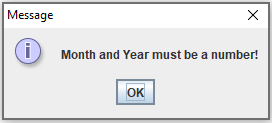


Figure 47. Number Validation

* + Validate that the month value must be between 1-12

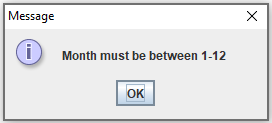


Figure 48. Month Validation

* + Otherwise, display **TransactionID**, **Modal** (obtained from the total of **quantity** bought multiplied by the **menu ingredient price)**, **Earn** (obtained from the total of **quantity** bought multiplied by the **menu sell price)**, **Gain** (obtained from **Earn** - **Modal),** for each transaction that occurred in the inputted month and year.

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
2. Additional rules:
   * All connections and data manipulation to the database must use the JDBC driver. The database file (.sql) must be submitted as well.
   * Use **LayoutManager** to design your application (You are **not allowed** to use **free** / **absolute** / **null** layout).
   * All validations must use the techniques taught during practicum. (Another language or Regular Expression is prohibited).
3. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
4. Include the other files that can support your project, such as:
   * All files in your project.
   * Other files (image, audio, video, etc.) used in your project.
   * \*.DOC file (documentation of your project) that contains all pages in your project, reference links of additional files (image, audio, video, etc.) used in your project, the description about how to use your application, etc.

**Please ask your teaching assistant if there are any project related questions.**