|  |  |
| --- | --- |
| **Soal Proyek**  *Project Case* | cid:image001.png@01D03C74.0642AC10 |
| T1464  Programming for Business |
| **Teknik Informatika**  *Computer Science* | **E2-T1464-CF01** |
| **Periode Berlaku** Semester Genap 2014/2015  ***Valid on*** *Even Semester Year 2014/2015* | **Revisi 00**  *Revision 00* |

1. Seluruh kelompok tidak diperkenankan untuk :

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

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

* + - Menyadur sebagian maupun seluruh proyek dari buku,

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

* + - Mendownload sebagian maupun seluruh proyek dari internet,

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

* + - 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,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - 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. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

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* |
| Microsoft Visual Studio .NET 2010(C#.NET) |

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

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

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| Folder Project (SLN, CS, RESX, CSPROJ, USER, EXE, PDB, MANIFEST, CACHE, TLOG, TXT, RESOURCES, CSDL, MSL, SSDL, DLL, SETTINGS, image files(JPG/PNG/GIF)) | Folder Project (SLN, CS, MDF, LDF, RESX, EDMX, CSPROJ, USER, EXE, PDB, MANIFEST, CACHE, TLOG, TXT, RESOURCES, CSDL, MSL, SSDL, DLL, SETTINGS, image files(JPG/PNG/GIF)) |

## Soal

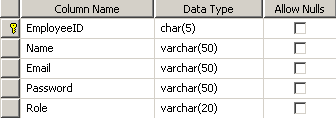
*Case*

**Bluejack BookShop**

**Bluejack BookShop** is one of the well-known bookshop in Indonesia. **Bluejack BookShop** hires you as the programmer to create an application that can manage the employees, books and transactions. You must use **C# programming language** to develop the program. The requirements are as follows:

**Database:**

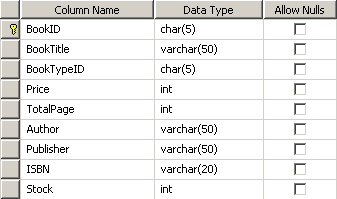
1. **MsEmployee**

****

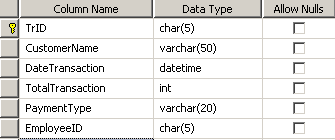
1. **MsBookType**

D:\CF\ssss\2.png

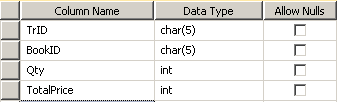
1. **MsBook**



1. **HeaderTransaction**



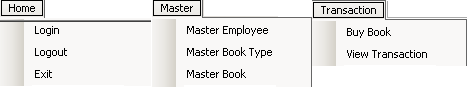
1. **DetailTransaction**

****

**Application**:

1. **Main Form**





**Components**:

* **Menu Strip** for the main menu
* **Status Strip** for the information of the logged in user.
* **Strip Status Label** for Employee Name, Employee Role, current Date and current Time
* **Menu Item** for Login, Logout, Exit, Master Employee, Master Book Type, Master Book, Buy Book and View Transaction.

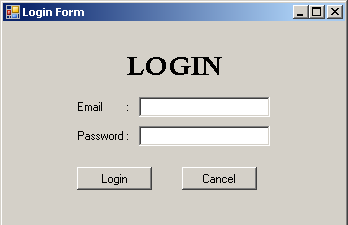
**Main menu:**

* **Home**
  + **Login =>** show **Login Form**
  + **Logout =>** for logout
  + **Exit =>** close the application
* **Master**
  + **Master Employee =>** show **Master Employee Form**
  + **Master Book Type =>** show **Master Book Type Form**
  + **Master Book =>** show **Master Book Form**
* **Transaction**
  + **Buy Book =>** show **Buy Book Form**
  + **View Transaction =>** show **View Transaction Form**

**Description**:

* **Main Form** is **MDI Form** that contains submenus to open another forms. The **Main Form** initial state is maximized.
* Set **Main Form’s Control Box** to hidden.
* Set **Main Form’s background image** to “**background.jpg**”.
* **Status Strip** is filled with:
* **Employee Name Label**  : show logged in Employee Name
* **Employee Role Label** : show logged in Employee Role
* **Current Date Label** : show current Date
* **Current** **Time Label** : show current Time
* **Before login**, menus which can be accessed:
  + Home :
* Login
* Exit
* **After login**, menus which can be accessed:
  + For employee as **Admin**:
* Home
* Logout
* Exit
* Master
* Master Employee
* Master Book Type
* Master Book
* Transaction
* Buy Book
* View Transaction
  + For employee as **Shopkeeper**:
* Home
* Logout
* Exit
* Master
* Master Book Type
* Master Book
* Transaction
* Buy Book
* View Transaction

1. **Login Form**

****

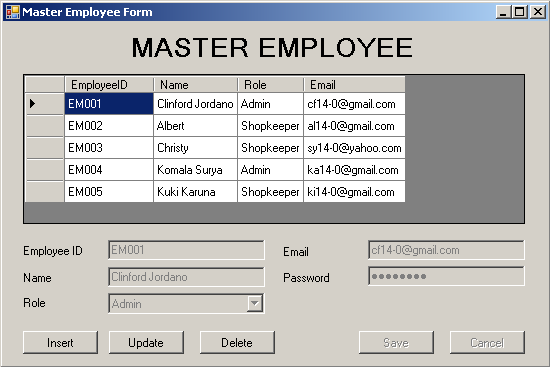
**Components**:

* **Label** for Login, Email, and Password
* **TextBox** for Email and Password
* **Button** for Login and Cancel

**Description**:

* If **Login Button** is clicked, then validate:
  + If **Email** is empty, then show error message “**Email must be filled !**”.
  + If **Password** is empty, then show error message “**Password must be filled !**”.
  + If **Email** or **Password** doesn’t match with **MsEmployee** table, then show error message “**Wrong Email and Password combination !**”.
  + If **Email** and **Password** match with **MsEmployee** table, then show message “**Login success !**”. Then, **Login Form** will be **closed** and other Menus will be active according to Employee Role (Admin or Shopkeeper).
* If **Cancel Button** is clicked, then Login Form will be closed.

1. **Master Employee Form**

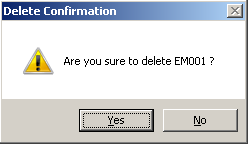


**Components**:

* **DataGridView** to show data from **MsEmployee** table
* **Label** for Master Employee, Employee ID, Name, Role, Email and Password
* **TextBox** for Employee ID, Name, Email and Password
* **ComboBox** for Role
* **Button** for Insert, Update, Delete, Save and Cancel

**Description**:

* The **initial state** of form:
  + Show all data from **MsEmployee** in **DataGridView**.
  + **DataGridView** is **enabled**.
  + Set **Role’s ComboBox** to **DropDownList**.
  + Set **Password’s Textbox** to **SystemPasswordChar**.
  + **Role ComboBox** is filled with (—Choose—, Admin and Shopkeeper).
  + All fields to input **MsEmployee** data are **disabled**.
  + Insert, Update, and Delete Buttons are **enabled**.
  + Save and Cancel Buttons are **disabled**.
* If **DataGridView** is clicked, then show all information from selected row to the existing fields.
* If **Insert Button** is clicked, then:
* **DataGridView** will be **disabled**.
* All fields to input **MsEmployee** data except **EmployeeID** are **enabled** and **empty**.
* **EmployeeID** TextBox is automatically generated from **MsEmployee** table with format **[EMxxx] plus by 1**. For example, if the last **EmployeeID** is “**EM003**”, then the program will generate it into “**EM004**”. If it is the first employee, then the program will generate it into “**EM001**”.
* Insert, Update, and Delete Button will be **disabled**.
* Save and Cancel Buttons will be **enabled**.
* If **Update Button** is clicked, then:
  + **DataGridView** will be **disabled**.
  + If there is no data chosen, then show error message “**Data must be chosen before update** !”.
  + All fields to input **MsEmployee** data except **EmployeeID** are **enabled** and **contain the last selected data**.
  + Insert, Update, and Delete Button will be **disabled**.
  + Save and Cancel Button will be **enabled**.
* If **Delete Button** is clicked, then validate:
  + If there is no data chosen, then show error message “**Data must be chosen before delete** !”.
  + Otherwise, show confirmation message “**Are you sure to delete [EmployeeID] ?**”. If the user chooses ‘**Yes**’, the data will be deleted and if the user chooses ‘**No**’, the confirmation dialog will be closed.

****

* If **Save Button** is clicked, then validate:
* If **Name** isempty, then show error message “**Name must be filled !**”.
* If **Email** isempty, then show error message “**Email must be filled !**”.
* If **Email** is **less** than 10 characters, then show error message “**Email must 10 characters minimum !**”.
* If **Email** is not contain ‘**.**’ (**dot**), then show error message “**Email must contain at least 1 ‘.’ (dot) !**”.
* If **Email** is not contain exactly **1** ’**@**’ (**at**), then show error message “**Email must contain exactly 1 ’@’ (at) !**”.
* If **Email** is started or ended with ‘**@**’ (**at**) or ‘**.**’ (**dot**) then show error message “**Email must not be started or ended with '@' (at) or '.' (dot) !**”.
* If **Email**’s ‘**@**’ (**at**) and ‘**.**’ (**dot**) is **side by side**, then show error message “**Email's '@' (at) and '.' (dot) must not be side by side !**”.
* If **Email**’s ‘**@**’ (**at**) is **after** the last ‘**.**’ (**dot**), then show error message “**Email's '@' (at) may not after the last '.' (dot) !**”.
* If **Email’s domain** is **not** ‘**yahoo**’or‘**gmail**’, then show error message “**Email domain must be 'yahoo' or 'gmail' !**”.

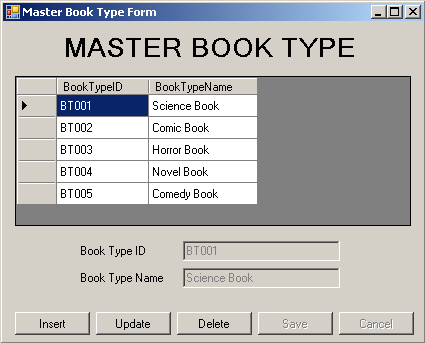
Example:

[**bambang@yahoo.com**](mailto:bambang@yahoo.com)

The email’s domain is “yahoo”

* If **Password** isempty, then show error message “**Password must be filled** **!**”.
* If **Password** length is **less** than 8 characters, then show error message “**Password must be 8 characters minimum !**”.
* If **Password** is not contain 5 alphabets minimum, then show error message “**Password at least contain 5 alphabets minimum !**”.
* If **Password** is contain space, then show error message “**Password may not contain space !**”.
* If **Role** isn’t chosen, then show error message “**Role must be chosen !**”.
* If all data have been inputted correctly, then the application will insert or update into **MsEmployee** and will show message“**Data succesfuly inserted !**” or “**Data succesfuly updated !**” according to the selected button before and the form will return to initial state.
* If **Cancel Button** is clicked, then the form will **return to initial state**.

1. **Master Book Type Form**

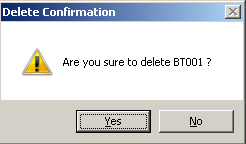
****

**Components**:

* **DataGridView** to show data from **MsBookType** table.
* **Label** for Master Book Type, Book Type ID and Book Type Name.
* **TextBox** for Book Type ID and Book Type Name.
* **Button** for Insert, Update, Delete, Save and Cancel

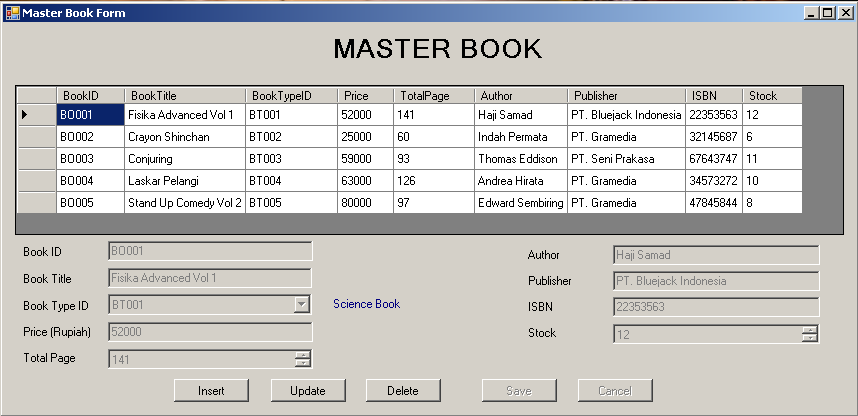
**Description**:

* The **initial state** of form:
  + Show all data from **MsBookType** in **DataGridView**.
  + **DataGridView** is **enabled**.
  + All fields to input **MsBookType** data are **disabled**.
  + Insert, Update, and Delete Buttons are **enabled**.
  + Save and Cancel Buttons are **disabled**.
* If **DataGridView** is clicked, then show all information from selected row to the existing fields.
* If **Insert Button** is clicked, then:
* **DataGridView** will be **disabled**.
* All fields to input **MsBookType** data except **BookTypeID** are **enabled** and **empty**.
* **BookTypeID** TextBox is automatically generated from **MsBookType** table with format **[BTxxx] plus by 1**. For example, if the last **BookTypeID** is “**BT003**”, then the program will generate it into “**BT004**”. If it is the first book type, then the program will generate it into “**BT001**”.
* Insert, Update, and Delete Button will be **disabled**.
* Save and Cancel Buttons will be **enabled**.
* If **Update Button** is clicked, then:
  + **DataGridView** will be **disabled**.
  + If there is no data chosen, then show error message “**Data must be chosen before update** !”.
  + All fields to input **MsBookType** data except **BookTypeID** are **enabled** and **contain the last selected data**.
  + Insert, Update, and Delete Button will be **disabled**.
  + Save and Cancel Buttons will be **enabled**.
* If **Delete Button** is clicked, then validate:
  + If there is no data chosen, then show error message “**Data must be chosen before delete** !”.
  + Otherwise, show confirmation message “**Are you sure to delete [BookTypeID] ?**”. If the user chooses ‘**Yes**’, the data will be deleted and if the user chooses ‘**No**’, the confirmation dialog will be closed.

****

* If **Save Button** is clicked, then validate:
* If **Book Type Name** is empty, then show error message “**Book Type** **Name must be filled !**”.
* If **Book Type Name** is **less** than 2 words, then show error message “**Book Type** **Name must 2 words minimum !**”.
* If all data have been inputted correctly, then the application will insert or update into **MsBookType** and will show message“**Data succesfully inserted** **!**” or “**Data succesfully updated !**” according to the selected button before and the form will return to initial state.
* If **Cancel Button** is clicked, then the form will **return to initial state**.

1. **Master Book Form**

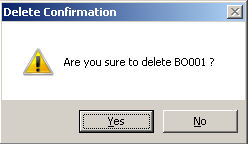


**Components**:

* **DataGridView** to show data from **MsBook** table.
* **Label** for Master Book, Book ID, Book Title, Book Type ID, Detail Book Type ID, Price (Rupiah), Total Page, Author, Publisher, ISBN and Stock.
* **TextBox** for BookID, BookTitle, Price (Rupiah), Author, Publisher and ISBN.
* **ComboBox** for Book Type ID.
* **NumericUpDown** for Total Page and Stock.
* **Button** for Insert, Update, Delete, Save and Cancel.

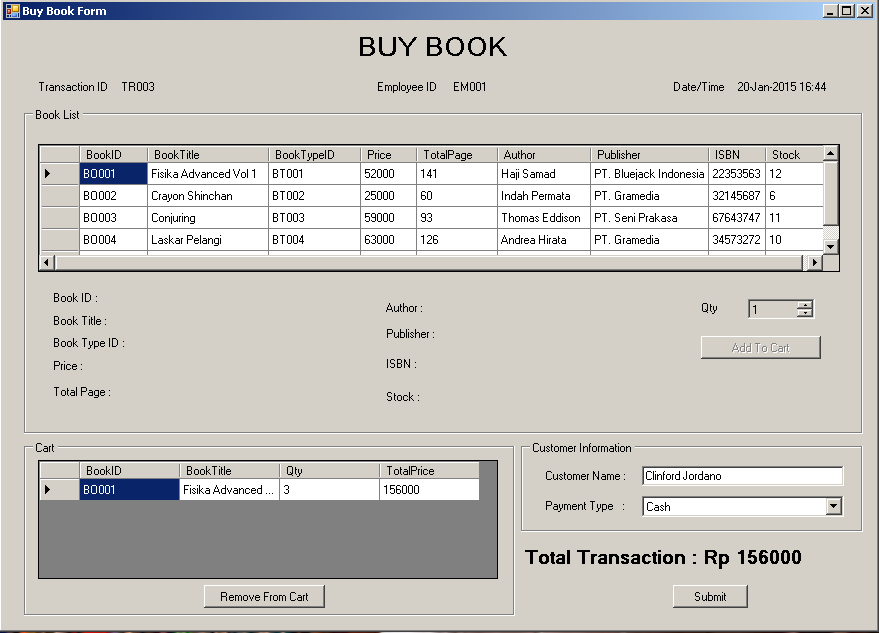
**Description**:

* The **initial state** of form:
  + Show all data from **MsBook** in **DataGridView**.
  + **DataGridView** is **enabled**.
  + Set **Total Page’s NumericUpDown** to **read-only** and set **minimum** value (1) and **maximum** value (500).
  + Set **Stock’s NumericUpDown** to **read-only** and set **minimum** value (0) and **maximum** value (50).
  + **Book Type ID ComboBox** is filled with available **BookTypeID** from **MsBookType** table.
  + Set **BookTypeID’s ComboBox** to **DropDownList**.
  + All fields to input **MsBook** data are **disabled**.
  + Insert, Update, and Delete Buttons are **enabled**.
  + Save and Cancel Buttons are **disabled**.
* If **DataGridView** is clicked, then show all information from selected row to the existing fields.
* If **Insert Button** is clicked, then:
* **DataGridView** will be **disabled**.
* All fields to input **MsBook** data except **BookID** are **enabled** and **empty**.
* **BookID** TextBox is automatically generated from **MsBook** table with format **[BOxxx] plus by 1**. For example, if the last **BookID** is “**BO003**”, then the program will generate it into “**BO004**”. If it is the first book, then the program will generate it into “**BO001**”.
* Insert, Update, and Delete Button will be **disabled**.
* Save and Cancel Buttons will be **enabled**.
* If **Update Button** is clicked, then:
  + **DataGridView** will be **disabled**.
  + If there is no data chosen, then show error message “**Data must be chosen before update** !”.
  + All fields to input **MsBook** data except **BookID** are **enabled** and **contain the last selected data**.
  + Insert, Update, and Delete Button will be **disabled**.
  + Save and Cancel Buttons will be **enabled**.
* If **Delete Button** is clicked, then validate:
  + If there is no data chosen, then show error message “**Data must be chosen before delete** !”.
  + Otherwise, show confirmation message “**Are you sure to delete [BookID] ?**”. If the user chooses ‘**Yes**’, the data will be deleted and if the user chooses ‘**No**’, the confirmation dialog will be closed.

****

* If **Save Button** is clicked, then validate:
* If **Book Title** is empty, then show error message “**Book Title must be filled !**”.
* Iffirst letter in each word of **Book Title** isn’t capital, then show error message “**First letter in each word of Book Title must be capital !**”.
* If **Book Type ID** isn’t chosen, then show error message “**Book Type ID must be chosen !**”.
* If **Price** is empty, then show error message “**Price must be filled !**”.
* If **Price** isn’t numeric, then show error message “**Price must be numeric !**”.
* If **Price** isn’tbetween **5000** and **200000**, then show error message “**Price must between 5000 and 200000!**”.
* If **Author** is empty, then show error message “**Author must be filled !**”.
* If **Publisher** is empty, then show error message “**Publisher must be filled !**”.
* If **Publisher** isn’t started with ‘PT. ‘ , then show error message “**Publisher must be started with ‘PT. ’ !**”.
* If **ISBN** is empty, then show error message “**ISBN must be filled !**”.
* If **ISBN** isn’t13 characters, then show error message “**ISBN must be 13 characters !**”.
* If **ISBN** isn’t numeric, then show error message “**ISBN must be numeric !**”.
* If all data have been inputted correctly, then the application will insert or update into **MsBook** and will show message“**Data succesfuly inserted !**” or “**Data succesfuly updated !**” according to the selected button before and the form will return to initial state.
* If **Cancel Button** is clicked, then the form will **return to initial state**.

1. **Buy Book Form**

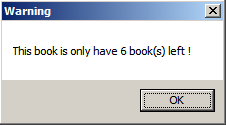
****

**Components**:

* **DataGridView** to show data from **MsBook** table and Cart.
* **Label** for Buy Book, Transaction ID, Employee ID, Date/Time, Book List (Book ID, Book Title, Book Type ID, Price, Total Page, Author, Publisher, ISBN, Stock and Qty), Customer Information (Customer Name and Payment Type) and Total Transaction.
* **TextBox** for Customer Name.
* **Button** Add to Cart, Remove from Cart and Submit.
* **DateTimePicker** for Date.
* **ComboBox** for Payment Type.
* **NumericUpDown** for Qty.
* **GroupBox for** Book List, Cart and Customer Information.

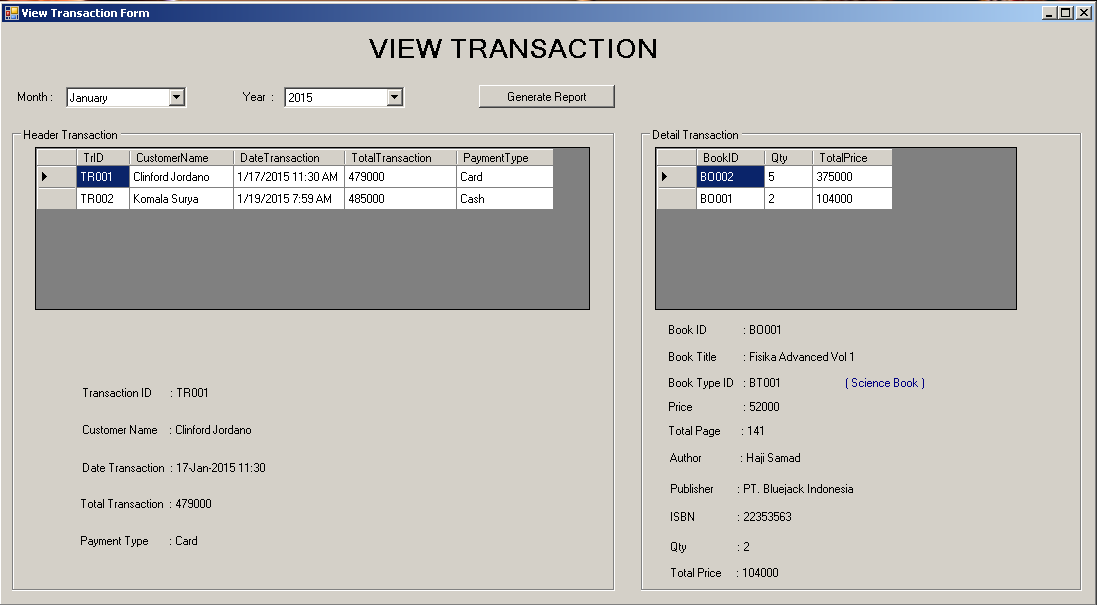
**Description**:

* The **initial state** of form:
* **Transaction ID Label**, **Employee ID Label** and **Date Label** will show the present condition.
  + **Transaction ID Label** is automatically generated from **HeaderTransaction** table with format **[TRxxx] plus by 1**. For example, if the last Transaction ID is “**TR003**”, then the program will generate it into “**TR004**”. If it is the first transaction, then the program will generate it into “**TR001**”.
  + **Employee ID** is filled with the **EmployeeID** of the logged in user.
  + **Payment Type’s ComboBox** is filled ‘—Choose—‘, ‘Cash’ and ‘Card’.
  + Set **PaymentType’s ComboBox** to **DropDownList**.
  + Set **Qty’s NumericUpDown** to **read-only** and set **minimum** value (1) and **maximum** value (20).
  + **Add To Cart** are **enabled**.
  + **Customer Name, Payment Type, Remove From Cart** and **Submit Button** are disabled.
  + **Cart DataGridView** is empty.
* If **Book List DataGridView** is clicked, then show book information from the selected row.
* If **Add To Cart Button** is clicked, then validate:
  + If the selected book already exists on the **Cart** **DataGridView**, then show error message “**Book is already in Cart !**”.
  + If selected Book’s **Stock** is 0, then show error message “**This book is out of stock !**”.
  + If **Qty NumericUpDown** is more than selected Book’s Stock, then show error message "**This book is only have [Stock] book(s) left !**”.

****

* + Otherwise, add the data from **Book List** **DataGridView** to the **Cart** **DataGridView** and **Total Transaction** will **increase** depends onTotal Price in **Cart DataGridView**.
* If **Remove From Cart Button** is clicked, then:
* Remove the selected data and **Total Transaction** will **decrease** depends onTotal Price in **Cart DataGridView**.
* **Validate** if there is no data on the **Cart DataGridView**, then **Remove From Cart and Submit Button** are **disabled**.
* If **Submit Button** is clicked, then validate:
* If **Customer Name** is empty, then show error message “**Customer** **Name must be filled !**”.
* If **Customer Name** is **less** than 2 wourrds, then show error message “**Customer** **Name must 2 words minimum !**”.
* If **Payment Type** isn’t chosen, then show error message “**Payment Type must be chosen !**”.
* If all data have been inputted correctly, then the application will **remove** stock on **MsBook** table with each qty of book that have been bought and insert the data to **HeaderTransaction** and **DetailTransaction** table and show message “**Data successfully inserted !**”. After that, **Book List DataGridView** will be **refreshed**.

1. **View Transaction Form**

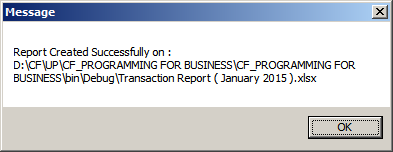


**Components**:

* **Label** for View Transaction, Header Transaction(Transaction ID value, Customer Name value, Date Transaction value, Total Transaction value and Payment Type value) and Detail Transaction (Book ID value, Book Title value, Book Type ID value, Detail Book Type ID value, Price value, Total Page value, Author value, Publisher value, ISBN value, Qty value and Total Price value).
* **DataGridView** for showing HeaderTransaction and DetailTransaction data.
* **GroupBox** for Header Transaction and Detail Transaction.

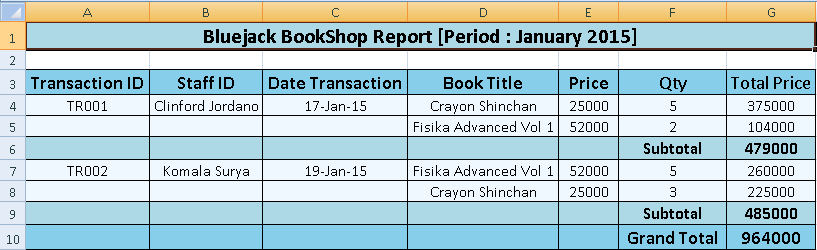
**Description**:

* The **initial state** of form:
  + Show all data from **HeaderTransaction** in **DataGridView**.
  + **DataGridView** is **enabled**.
  + **Month ComboBox** is filled with (—Choose—, January, February, March, April, May, June, July, August, September, October, November, December)
  + **Year ComboBox** is filled with (—Choose—, 2013, 2014, 2015)
  + Set **Month and Year’s ComboBox** to **DropDownList**.
* When the form is opened, DataGridView will show the transaction data from **HeaderTransaction** table.
* If **HeaderTransaction DataGridView** is clicked, then show all information from selected row to the existing fields and fill the **Detail Transaction DataGridView** from **DetailTransaction** table with the **Transaction ID** of the selected ID in the header table.
* If **Detail Transaction DataGridView** is clicked, then show all information from selected row to existing fields.
* If **Generate Button** is clicked, then validate:
  + If **Month** isn’t chosen, then show error message “**Month must be chosen !**”.
  + If **Year** isn’t chosen, then show error message “**Year must be chosen !**”.
  + If all validations are fulfilled, then show message “**Report Created Successfully on : [StartUpPath]\Transaction Report ( [month] [year] ).xlsx**” and the program will create the report depends on selected month and year in Excel file.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bluejack Bookshop Report [Period : <<Month>> <<Year>>]** | | | | | | |
|  |  |  |  |  |  |  |
| **Transaction ID** | **Staff ID** | **Date Transaction** | **Book Title** | **Price** | **Qty** | **Total Price** |
| <<TrID>> | <<EmployeeID>> | <<DateTransaction>> | <<BookTitle>> | <<Price>> | <<Qty>> | [<<TotalPrice>>] |
|  |  |  | [<<BookTitle>>] | [<<Price>>] | [<<Qty>>] | <<TotalPrice>> |
|  |  |  |  |  | Subtotal | <<Subtotal>> |
|  |  |  |  |  | Grand Total |  |

<Refer to screenshot below for more information>

****

**References**:

* <http://www.stanley-livingstone.eu/media/Reisboekwinkel%20Stanley%20&%20Livingstone.jpg>

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:
   * **LINQ** operation in database is optional (students are allowed to use other techniques for database that taught during practicum).
   * 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 the reference links of additional files (image, audio, video, etc.) used in your project