


Project – External Documentation	
COMP6114001 Pattern Software Design	
<Even> Semester Year <2022/2023>	

Dokumentasi Project PSD LAB

Kelas: BN01

ID Kelompok: f9adb3c5-e1ce-ed11-ae31-d8d385fce79e

Anggota:

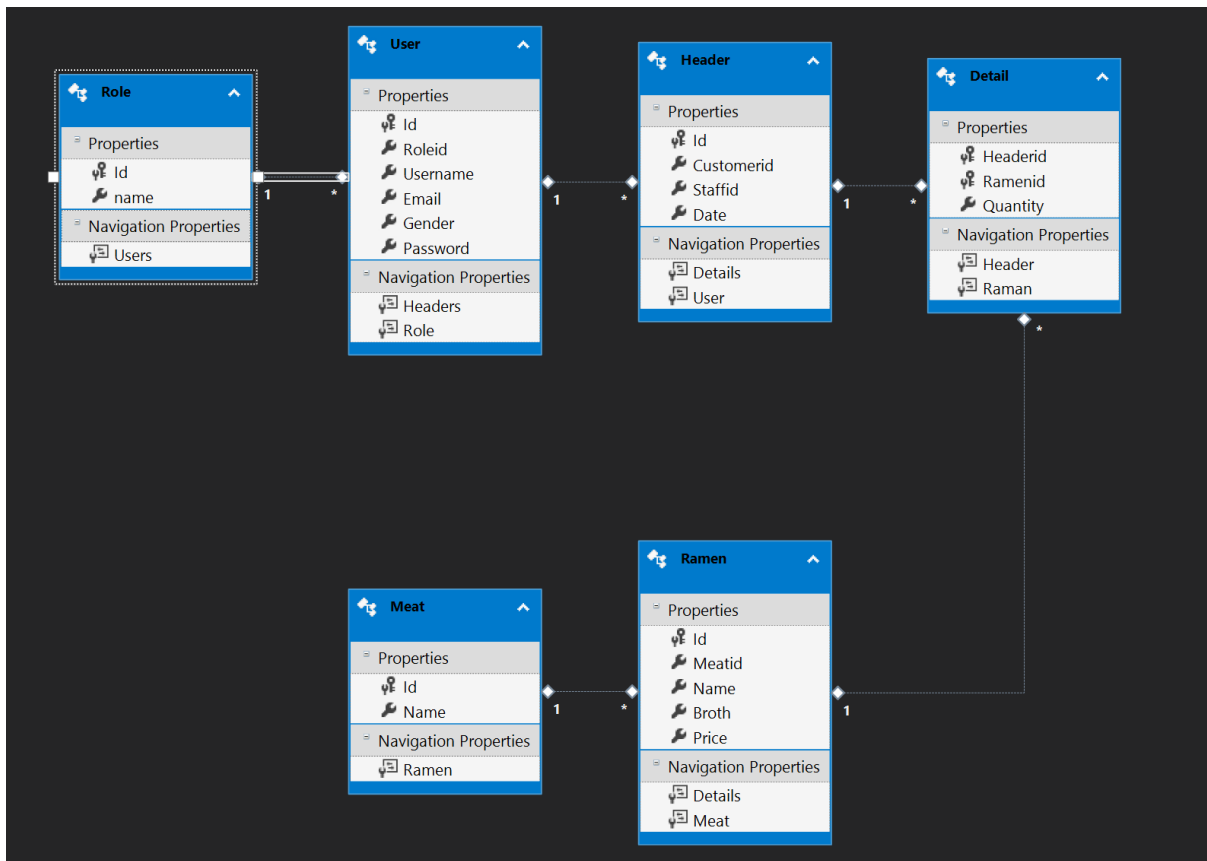
2502041975 - STEVEN FARRELIO JORGENSEN

2502024382 - BUNGA AURA PRAMESWARI

2502021853 - NICHOLAS HERNAN GONZALES


2502043886 - MORENO ADDAM RODERICK

Model



Database

1. Role

	Id	name
	1	Admin
	2	Staff
	3	User
	NULL	NULL

2. User

	Id	Roleid	Username	Email	Gender	Password
	1	1	Admin	admin@yahoo....	Male	admin123
	3	2	Staff	staff@yahoo.co...	Female	staff123
	4	3	baba	baba@yahoo.c...	Male	baba123
	5	3	hailey	hailey@yahoo.c...	Female	hailey123
🔍	NULL	NULL	NULL	NULL	NULL	NULL

3. Meat

	Id	Name
	1	Chicken
	2	Fish
	3	Pork
	4	Beef
🔍	NULL	NULL



4. Ramen

	Id	Meatid	Name	Broth	Price
	1	1	Chicken Ramen	Chicken	50000
	2	2	Beef Ramen	Beef	60000
🔍	NULL	NULL	NULL	NULL	NULL

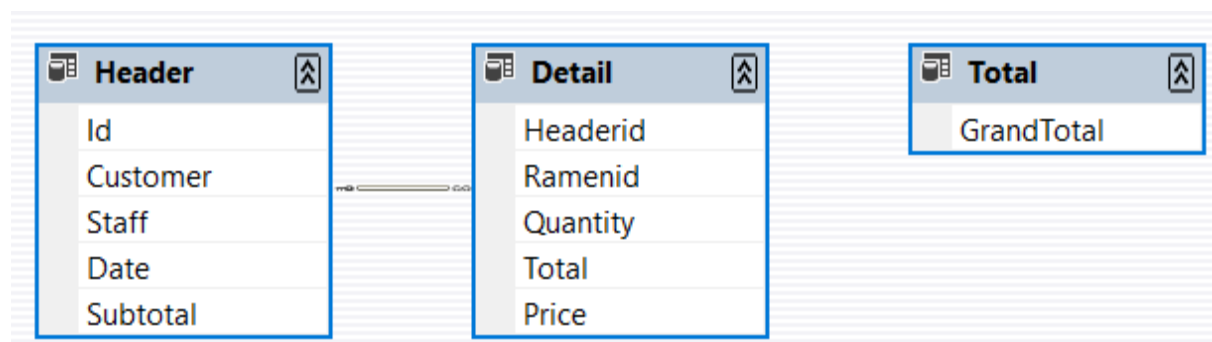
5. Header

	Id	Customerid	Staffid	Date
	1	4	3	NULL
	2	4	3	NULL
	3	5	3	NULL
	4	5	3	NULL
	5	5	3	NULL
🔍	NULL	NULL	NULL	NULL

6. Detail

	Headerid	Ramenid	Quantity
	1	1	2
	1	2	4
	2	1	3
	2	2	6
	3	1	4
	3	2	5
	4	1	2
	4	2	3
	5	1	4
	5	2	1
	NULL	NULL	NULL

Dataset



Crystal Report

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Section1 (Report Header)																			
Section2 (Page Header)																			
Print Date																			
Id	Customer	Staff	Date	Subtotal															
Section3 (Details)																			
Id	Customer	Staff	Date	Subtotal															
Detail																			
Section4 (Report Footer)																			
Section5 (Page Footer)																			
Grand Total																			
																		Subreport:Grand Total	

Crystal Report - Detail

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ReportHeaderSection1 (Report Header a)																			
ReportHeaderSection2 (Report Header b)																			
Print Date																			
Headerid	Ramenid	Quantity	Total	Price															
DetailSection1 (Details)																			
Headerid	Ramenid	Quantity	Total	Price															
ReportFooterSection1 (Report Footer a)																			
ReportFooterSection2 (Report Footer b)																			

Folder

Controller:

- UserController.cs:
 - memvalidasi ketika createmember dan update member
 - mengupdate session setelah dilakukan update member
 - memvalidasi cek alfabet
 - memvalidasi login user
 - menghubungkan antara method dan views dari handler
- RamenController.cs:
 - memvalidasi sebelum melakukan insert dan update ramen

- menghubungkan antara method dan views dari handler
- 3. TransactionController.cs:
 - Menghubungkan method ke views dari handler
- 4. CartController.cs:
 - Menghubungkan method ke views dari handler

Handler:

1. CartHandler.cs:
 - Menghubungkan ke CartRepository
2. TransactionHandler.cs:
 - Menghubungkan ke TransactionRepository
3. RamenHandler.cs:
 - Menghubungkan ke RamenRepository
4. UserHandler.cs:
 - Menghubungkan ke UserRepository

Factory:

1. UserFactory.cs:
 - Membuat database User baru
2. CartFactory.cs:
 - Membuat database Cart baru
3. TransactionFactory.cs:
 - Membuat transaction baru
4. RamenFactory.cs:
 - Membuat database Ramen baru

Repository:

1. CartRepository:
 - Mengatur database cart
 - Mendapatkan list cart, menambahkan cart baru, dan menghubungkan ke factory

- Menghapus cart
- 2. TransactionRepository:
 - Mengatur transaction
 - Mengambil list transaction, dan menghubungkan ke factory
 - Menghandle transaction
- 3. UserRepository:
 - Mengatur user, cek email yang unique, cek credential login, menghubungkan ke factory
 - Mengambil list user, update db User
- 4. RamenRepository:
 - Mengatur ramen, menghubungkan ke factory untuk create ramen.
 - Delete ramen, dan untuk update db ramen, dan mendapatkan list ramen atau ramen sendiri, juga untuk mendapatkan list meat.

Views:

1. Login:
 - Untuk login ke website
 - Cookie jika klik remember me akan tersimpan,
 - LoginButton untuk login dan memvalidasi ke UserController
 - Melakukan validasi di controller
2. Register:
 - Untuk register akun,
 - RegisterBtn untuk memvalidasi register dan register account.
 - Setelah validasi, create user dilakukan melalui UserController, UserHandler, UserRepository dan UserFactory.
 - Melakukan validasi di controller
3. Site1.Master:
 - Untuk master page,
 - Berisikan navbar sesuai requirement Role
 - Semua user memiliki menu Profile, Home, dan logout button
 - User punya OrderRamen dan History

- Staff punya ManageRamendan OrderQueue
- Admin punya ManageRamen, OrderQueue, History, dan Report.
- Master page digunakan semua webform dalam views.

4. Home:

- Home berisikan data-data user bagi Staff
- Untuk admin berisikan data staff.
- Data ditampilkan dalam gridview yang didapatkan dari list db user sesuai rolenya.

5. Profile:

- Dapat melakukan update profile dengan validasi ke UserController. User dapat mengupdate username, email, dan gender.
- Password untuk validasi harus sama dengan password sebelumnya.

6. History:

- Berisi history transaction
- Role user hanya menampilkan yang telah dilakukan user
- Sementara admin menampilkan semuanya
- Untuk melihat detailnya dapat mengklik detail lalu akan redirect ke TransactionDetail menggunakan query string.

7. TranscationDetail:

- Ambil id transaksi dari string link query. dan mengambil semua informasi dengan ID yang sama dengan header..
- Memiliki tombol back untuk Kembali ke history

8. OrderRamen:

- Berisikan data ramen. Untuk order klik kotak ramendan akan masuk ke cart. Jika admin atau staff yang mengkliknya maka tidak akan terjadi apa-apa.
- Di cart berisikan data ramen yang belum dibuy.
- Ada 2 tombol di bawah cart yaitu buy untuk membeli ramen dan membuat unhandled transaction baru.
- Tombol clear akan menghapus cart yang ada.

9. TransactionQueue:

- Untuk dapat mengakses page ini dapat menggunakan tombol OrderQueue
- Berisikan data unhandled transaction, ini akan ditampilkan hanya untuk staff dan admin. Staff atau admin dapat handle yang dipilih saja atau handle semua unhandled dengan tombol Handle All

10. ManageRamen:

- Menu ini berisi list ramen yang ada. user tidak bisa access menu ini, sedangkan staff atau admin dapat melakukan edit atau delete ramen.
- jika mengklik delete ramen maka akan menghapus ramen yang dipilih.
- Jika mengklik edit maka akan diredirectkan ke UpdateRamen dengan query string id.

11. InsertRamen:

- Untuk membuat ramen baru, dengan validasi RamenController.
- Jika insertRamen maka akan create ramen yang baru.

12. UpdateRamen:

- Data-data ramen dapat diupdate disini dan sesuai dengan validasi dari RamenController.
- Jika sudah sesuai maka database nya akan terupdate.

13. TransactionReport:

- Berisikan report transaction yang ada. Page ini hanya bisa dibuka admin.
- Semua detail ramen diperlihatkan di sini, beserta dengan total harga ramen yang dibeli, subtotal, dan grandtotal.
- Subtotal untuk setiap header dan grandtotal untuk semua transaksi.

Asset:



Logo.png

(dibuat di Figma, gambar dari plugin, font Lexend Peta - Bold, warna #BC4749)

Link File:

https://binusianorg-my.sharepoint.com/personal/bunga_prameswari_binus_ac_id/_layouts/15/guestaccess.aspx?guestaccesstoken=fatw%2F3MHCflYxtqHp%2BKsELXBh7%2FZDuhKfuWmA29HXC%3D&folderid=2_0b213148c205d4f32887469926dee5387&rev=1&e=ETn8Z6

DatabaseQuery :

```
CREATE TABLE [dbo].[Role] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [name] VARCHAR (50) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

```
CREATE TABLE [dbo].[User] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Roleid] INT NOT NULL,  
    [Username] VARCHAR (50) NOT NULL,  
    [Email] VARCHAR (50) NOT NULL,  
    [Gender] VARCHAR (50) NOT NULL,  
    [Password] VARCHAR (50) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC),  
    CONSTRAINT [FK_User_ToRole] FOREIGN KEY ([Roleid])  
REFERENCES [dbo].[Role] ([Id])  
);
```

```
CREATE TABLE [dbo].[Meat] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

```

CREATE TABLE [dbo].[Ramen] (
    [Id]          INT          IDENTITY (1, 1) NOT NULL,
    [Meatid] INT          NOT NULL,
    [Name]       VARCHAR (50) NOT NULL,
    [Broth]      VARCHAR (50) NOT NULL,
    [Price]      VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_Ramen_ToMeat] FOREIGN KEY ([Meatid])
REFERENCES [dbo].[Meat] ([Id])
);

```

```

CREATE TABLE [dbo].[Header] (
    [Id]          INT          IDENTITY (1, 1) NOT NULL,
    [Customerid] INT          NOT NULL,
    [Staffid]     INT          NOT NULL,
    [Date]        DATETIME NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_Header_ToUser] FOREIGN KEY ([Customerid])
REFERENCES [dbo].[User] ([Id])
);

```

```

CREATE TABLE [dbo].[Detail] (
    [Headerid] INT NOT NULL,
    [Ramenid]  INT NOT NULL,
    [Quantity] INT NOT NULL,
    CONSTRAINT [Primary Key] PRIMARY KEY CLUSTERED ([Headerid]
ASC, [Ramenid] ASC),
    CONSTRAINT [FK_Detail_ToHeader] FOREIGN KEY ([Headerid])
REFERENCES [dbo].[Header] ([Id]),
    CONSTRAINT [FK_Detail_ToRamen] FOREIGN KEY ([Ramenid])
REFERENCES [dbo].[Ramen] ([Id]) ON DELETE CASCADE ON UPDATE
CASCADE
);

```