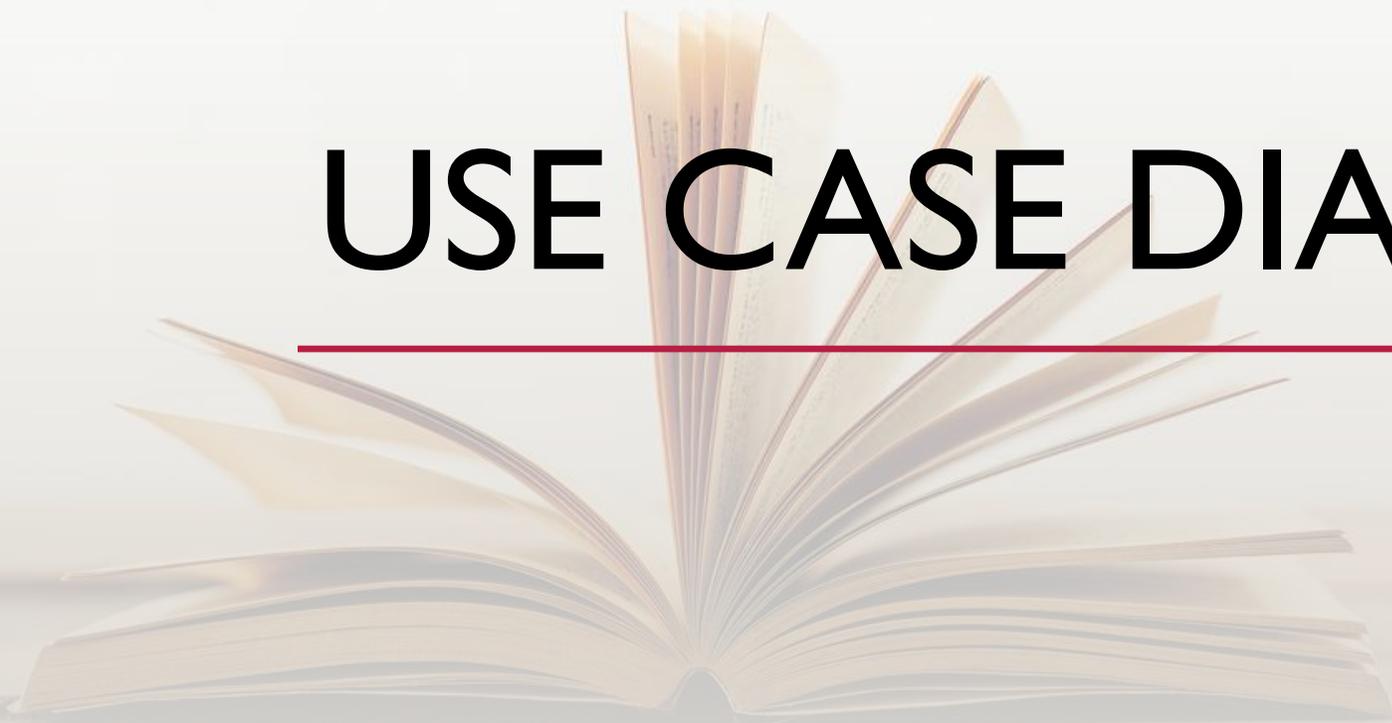


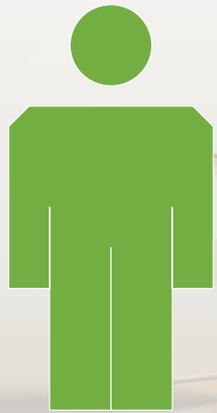
USE CASE DIAGRAM



PENDAHULUAN

- Merupakan pemodelan untuk kelakuan (behavior) system informasi yang dibuat
- Mendeskripsikan interaksi antara satu atau lebih actor dengan system informasi yang dibuat
- Digunakan untuk mengetahui fungsi apa saja dalam sebuah system.
- Untuk mengetahui siapa saja yang berhak menggunakan fungsi-fungsi tersebut.
- Nama Usecase dan Actor didefinisikan sesimpel mungkin dan dapat dipahami

SIMBOL-SIMBOL DALAM USE CASE



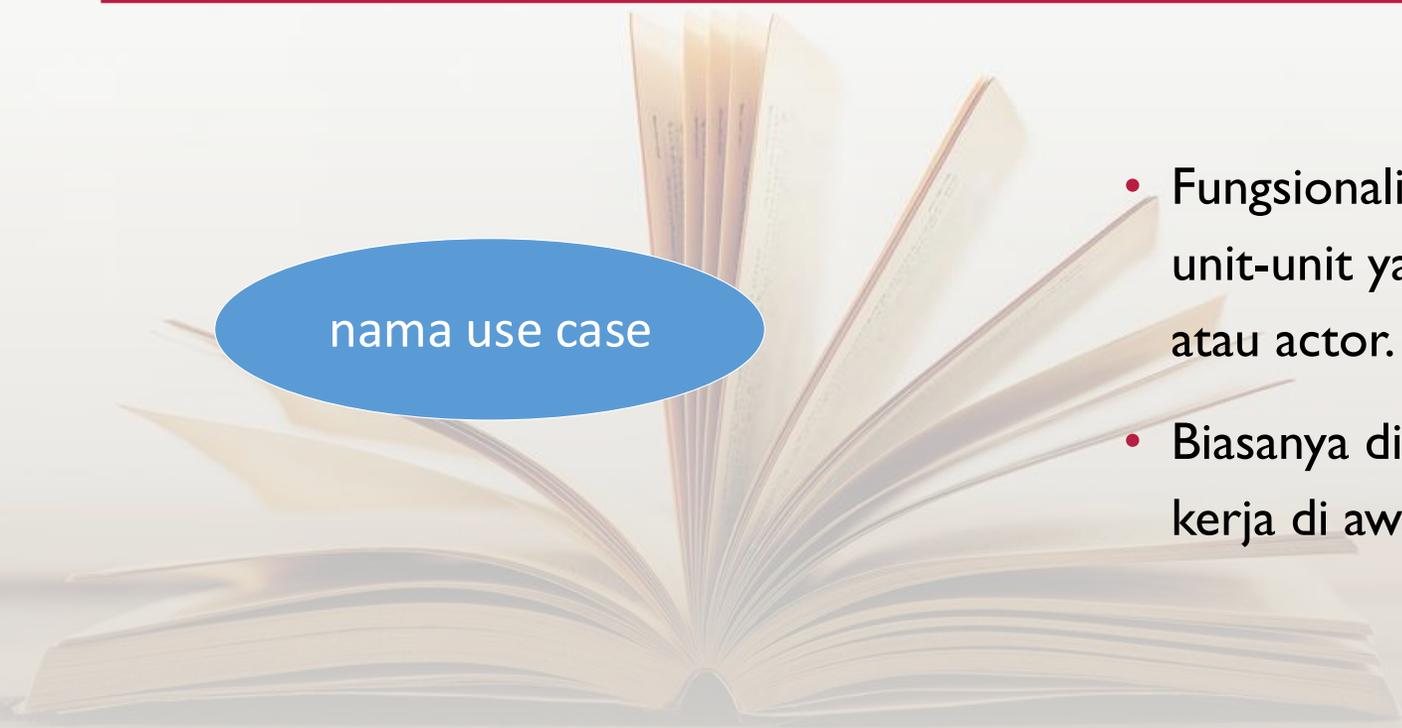
nama aktor



nama aktor

- AKTOR/ ACTOR
- Orang, proses atau system lain yang berinteraksi dengan Sistem informasi yang akan dibuat.

USE CASE



nama use case

- Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor.
- Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case

ASOSIASI/ ASSOCIATION



- Komunikasi antara actor dan usecase yang berpartisipasi pada use case
- Atau Use case memiliki interaksi dengan aktor

EXTEND

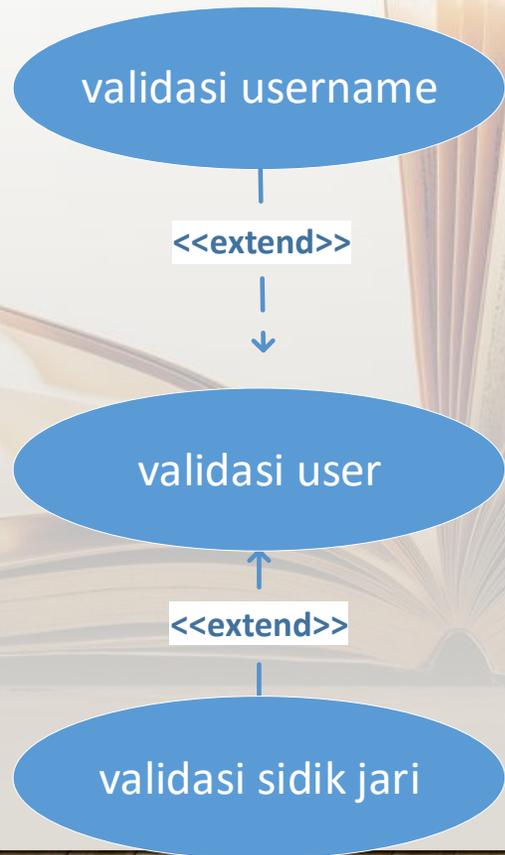


— <<extend>> —>

The diagram shows a red rectangular box containing the text '<<extend>>' in blue. To the left of the box is a blue horizontal line with a small arrowhead pointing right. To the right of the box is another blue horizontal line with a larger arrowhead pointing right, indicating the direction of the relationship.

- Relasi Use Case tambahan ke sebuah use case
- Use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu
- Mirip dengan prinsip inheritance pada OOP
- Biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan

CONTOH EXTEND



- Arah panah mengarah pada use case yang ditambahkan
- Use case yang menjadi extend –nya merupakan jenis yang sama dengan use case yang menjadi induknya

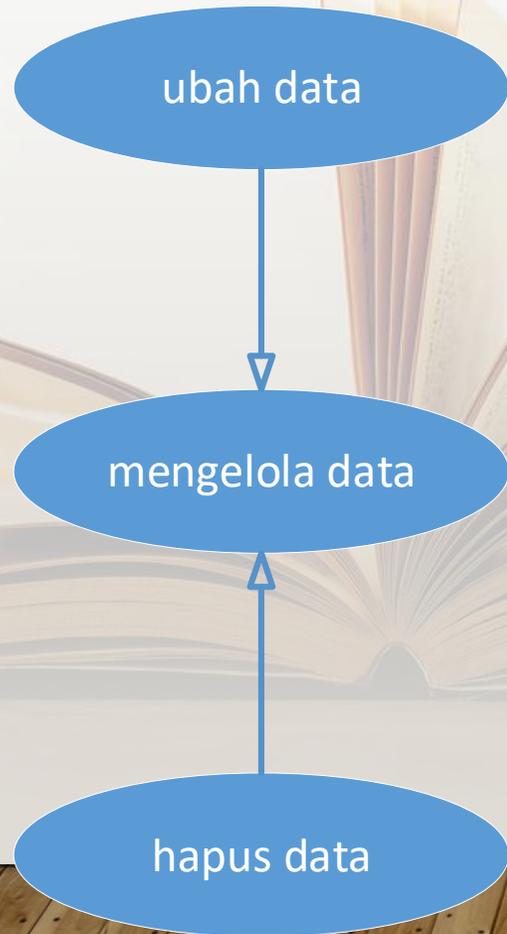
EKSTEND

```
Class ValidasiUser {  
//atribut  
.....  
  
procedure  
validasiUsername() {  
//proses  
.....  
}  
  
procedure  
validasiSidikJari() {  
//proses  
.....  
}  
}
```



- Pada hubungan ekstensi maka dapat hanya diambil use case induknya yang dijadikan kelas dengan metode berupa use case ekstensinya

CONTOH GENERALIZATION



- Arah panah mengarah pada use case yang menjadi generalisasinya (umum)

GENERALISASI/ GENERALIZATION



- Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

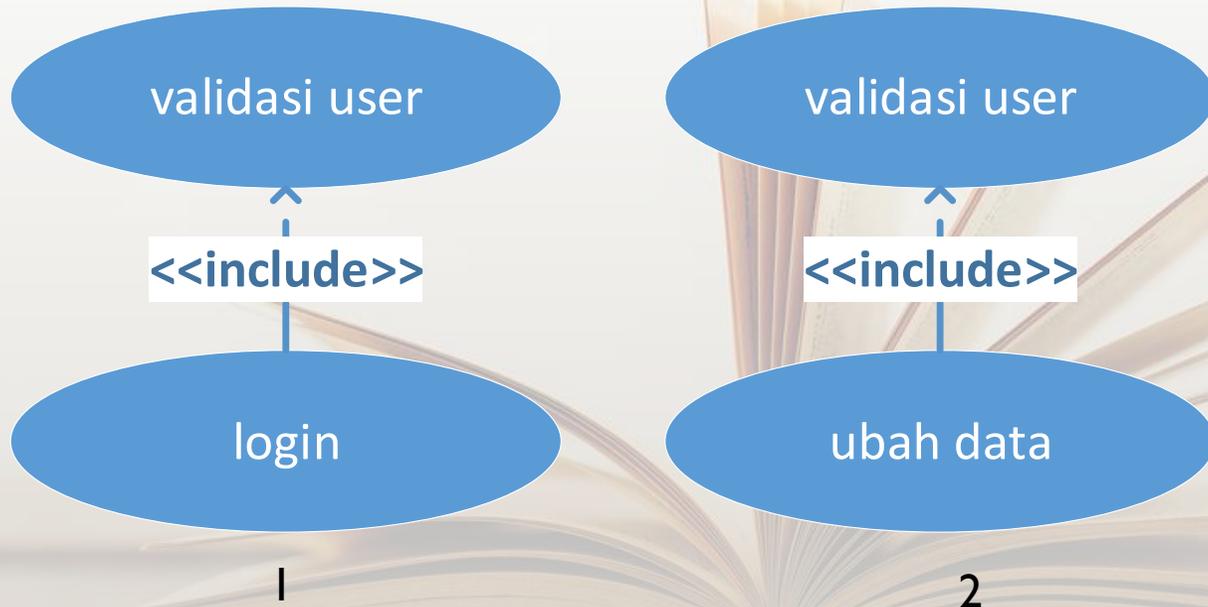
INCLUDE



```
— <<include>> —>
```

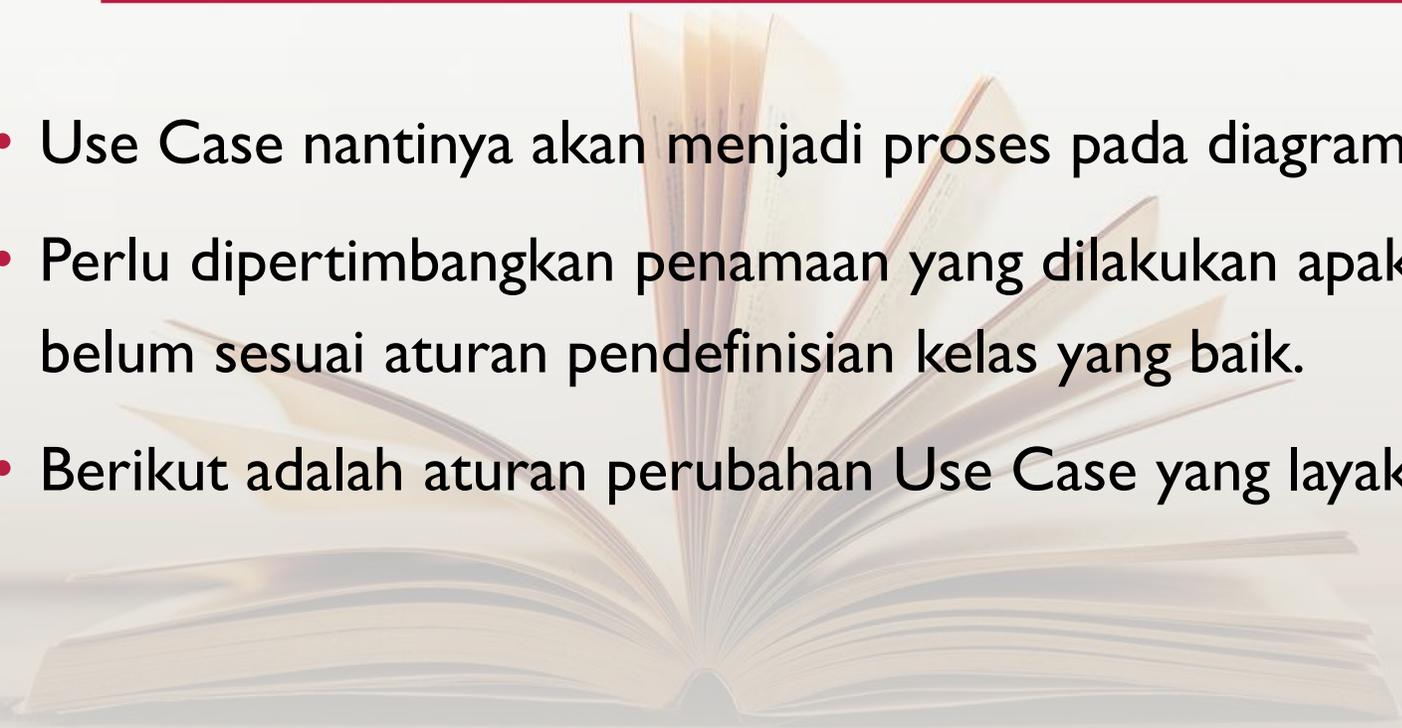
- INCLUDE/ USES/ MENGGUNAKAN
- Relasi use case tambahan ke sebuah use case
- Use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat

CONTOH INCLUDE



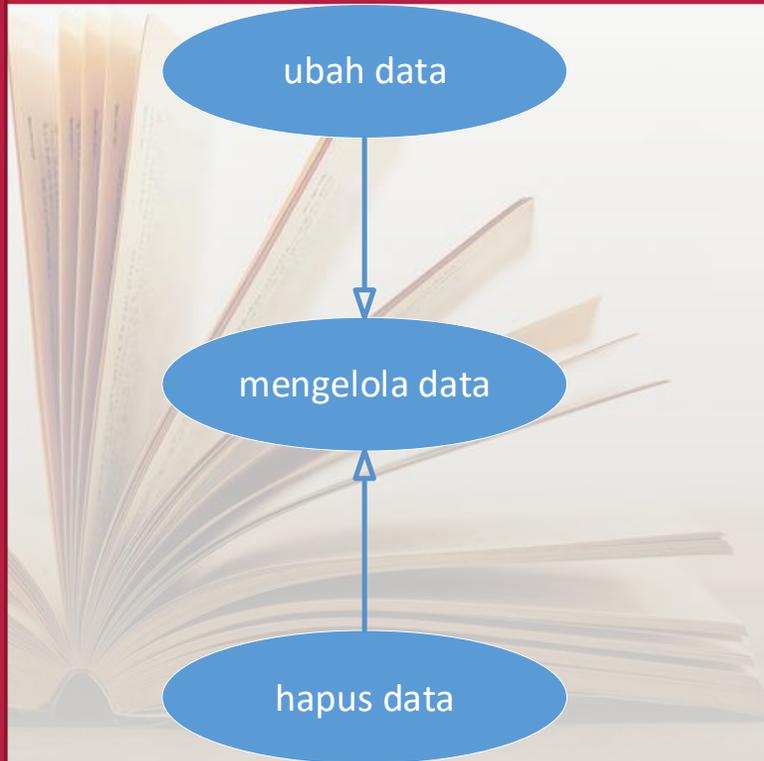
- Ada dua sudut pandang mengenali include:
 1. Include berarti use case akan selalu dipanggil saat use case tambahan dijalankan
 2. Include berarti use case akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan.

Kedua interpretasi tersebut dapat dianut salah satu atau keduanya tergantung kebutuhan.

-
- 
- Use Case nantinya akan menjadi proses pada diagram kelas
 - Perlu dipertimbangkan penamaan yang dilakukan apakah layak menjadi kelas atau belum sesuai aturan pendefinisian kelas yang baik.
 - Berikut adalah aturan perubahan Use Case yang layak menjadi kelas proses.

GENERALIZATION

```
Class MengolahData {  
//atribut  
.....  
  
procedure ubahData () {  
//proses  
.....  
}  
  
procedure hapusData() {  
//proses  
.....  
}  
}
```



- Pada hubungan generalisasi maka dapat hanya diambil use case umumnya yang dijadikan kelas dengan metode berupa use case khususnya

USE CASE BERDIRI SENDIRI

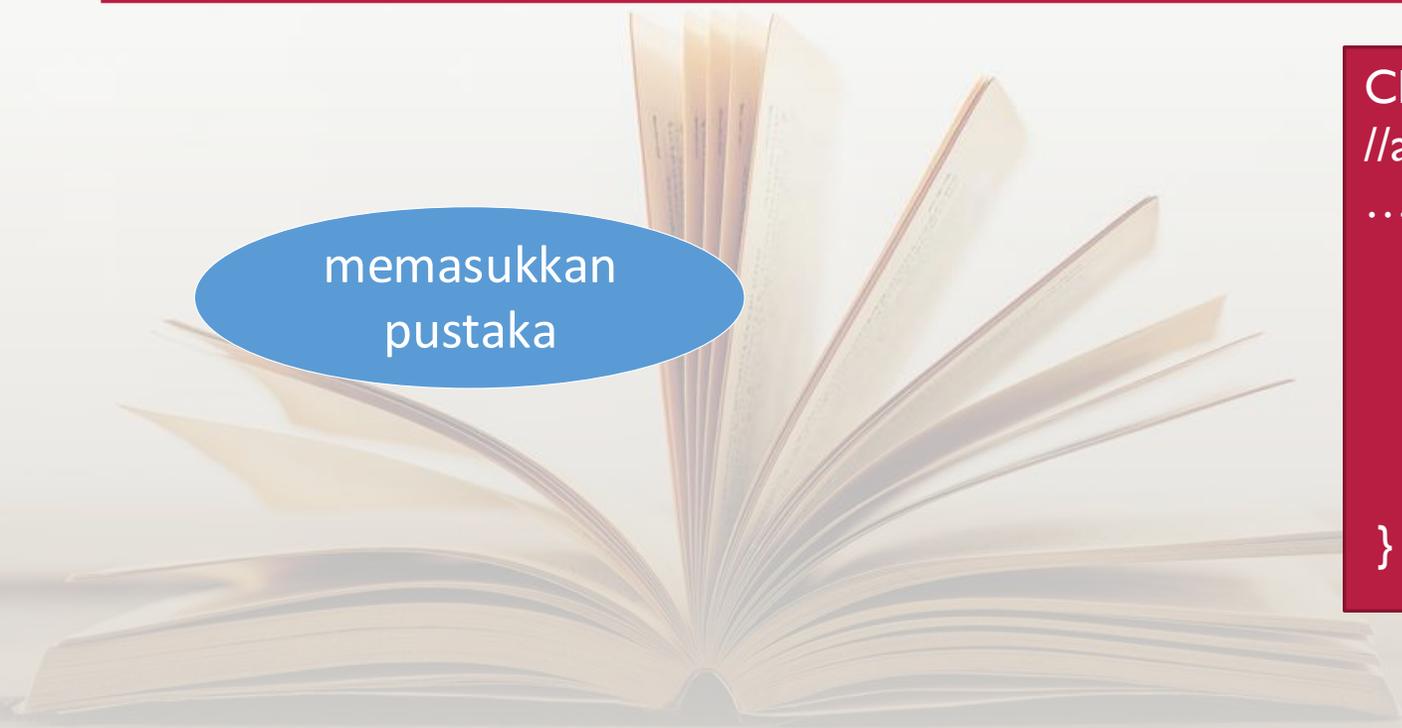
- Metode yang mungkin biasa ada dalam kelas login adalah sbb:

```
Class Login {  
  //atribut  
  .....  
  
  procedure login () {  
    //proses  
    .....  
  }  
  
  procedure logout () {  
    //proses  
    .....  
  }  
}
```



login

USE CASE YANG KURANG TEPAT SEBAGAI USE CASE YANG BERDIRI SENDIRI



memasukkan
pustaka

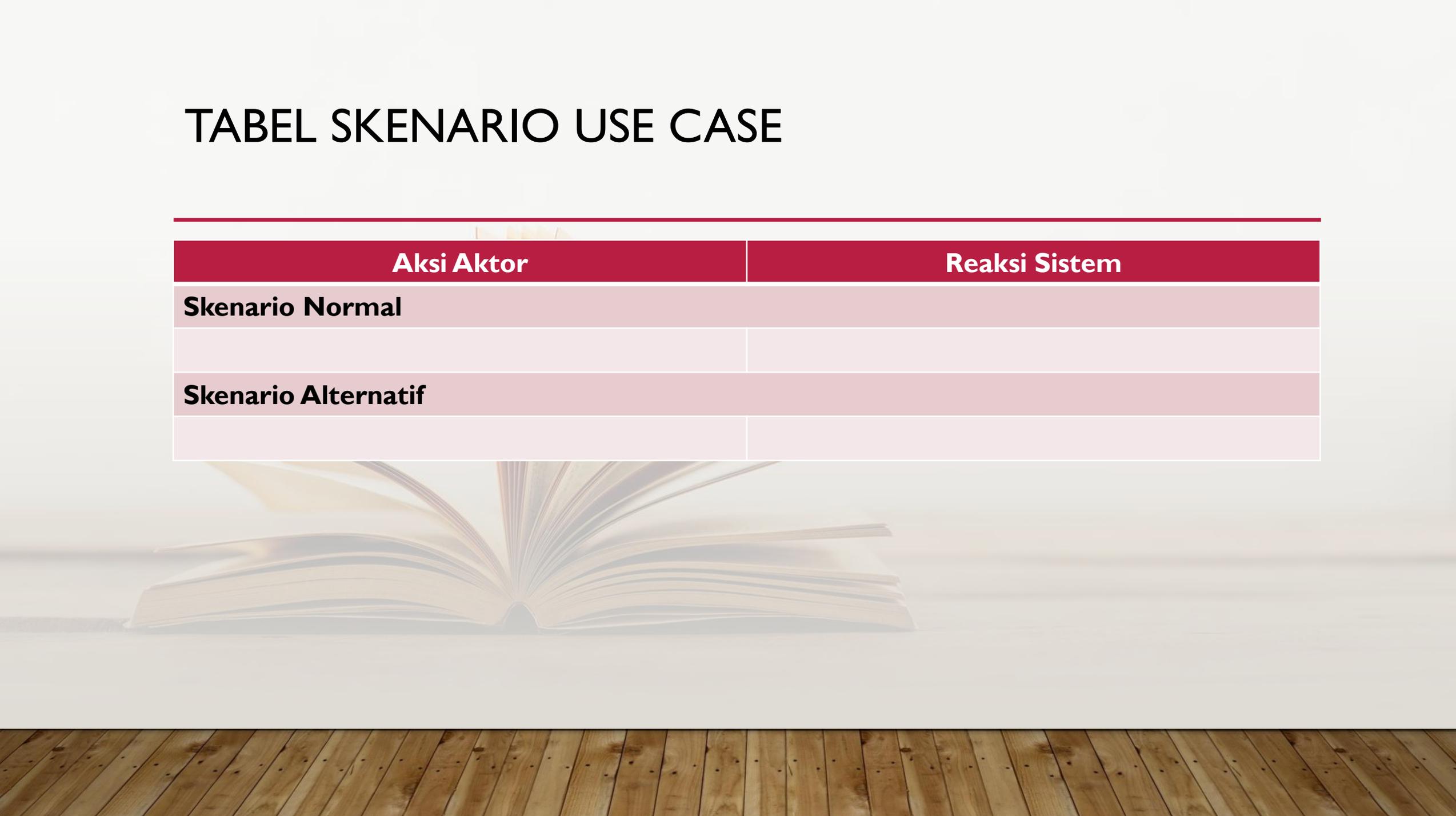
```
Class MemasukkanPustaka {  
  //atribut  
  .....  
  
  procedure MemasukkanPustaka () {  
    //proses  
    .....  
  }  
}
```

- Kelas yang hanya terdiri dari satu metode sebenarnya kurang efisien.

SKENARIO USE CASE

- Alur jalannya proses use case dari sisi actor dan system
- Skenario use case dibuat per use case terkecil
- Contoh untuk generalisasi maka yang dibuat adalah use case yang lebih khusus
- Skenario normal adalah scenario bila system berjalan normal tanpa ada error
- Skenario alternative adalah bila system tidak berjalan normal atau mengalami error
- Alur scenario ini yang nantinya menjadi dasar pembuatan Sequence Diagram.

TABEL SKENARIO USE CASE



Aksi Aktor	Reaksi Sistem
Skenario Normal	
Skenario Alternatif	

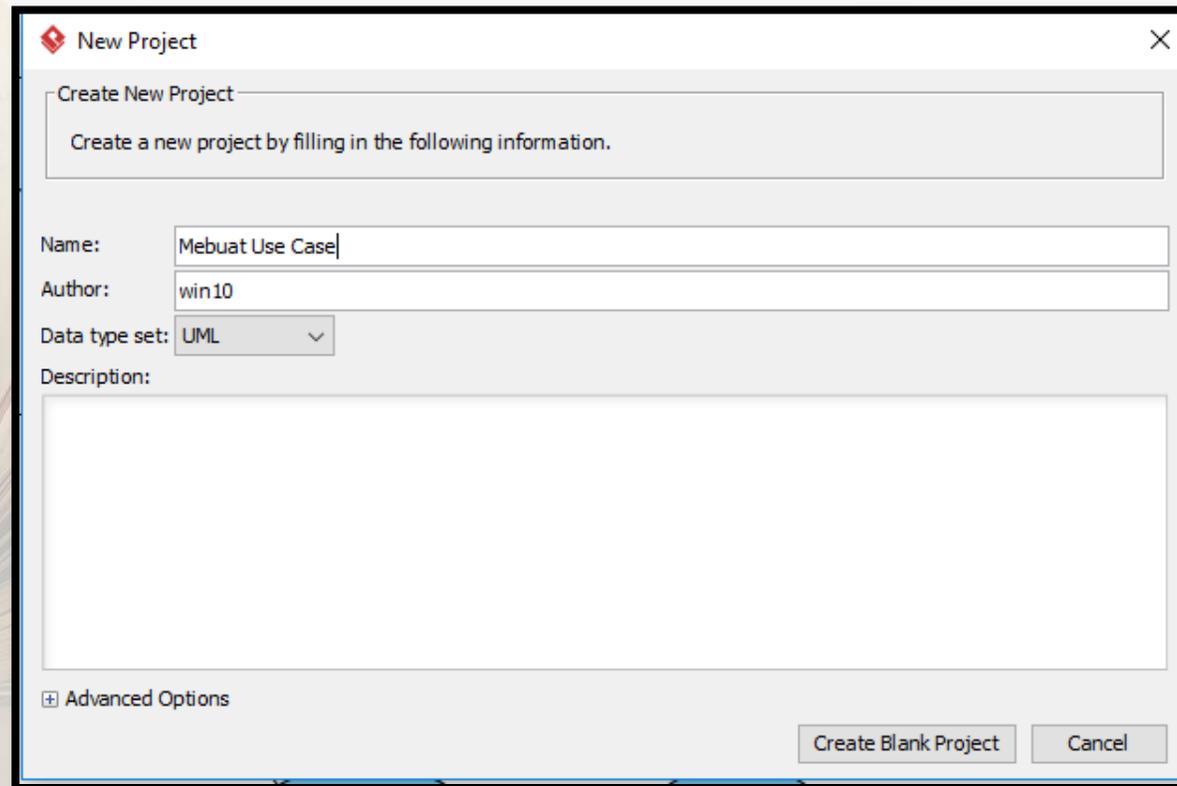
CONTOH SKENARIO USE CASE

- Nama Use Case: Login

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan username dan password	
	2. Memeriksa valid tidaknya data masukan
	3. Masuk ke Dashboard Aplikasi
Skenario Alternatif	
1. Memasukkan username dan password	
	2. Memeriksa valid tidaknya data masukan
	3. Menampilkan pesan <i>Login Tidak Valid</i>
4. Memasukkan username dan password yang valid	
	5. Memasukkan username dan password
	6. Masuk ke Dashboard Aplikasi

MEMBUAT USE CASE DENGAN VISUAL PARADIGM

- Buka Visual Paradigm
- Klik New Project
- Isikan Nama Project
- Klik Create Blank Project



New Project

Create New Project

Create a new project by filling in the following information.

Name: Mebuat Use Case|

Author: win10

Data type set: UML

Description:

Advanced Options

Create Blank Project Cancel

- Klik Tab Menu Diagram

Project Kuliah ANSI - Visual Paradigm Community Edition [Perlambang] (not for commercial use)

Dash Project ITSM Agile **Diagram** View Team Tools Modeling Window Help

New Undo Redo Copy Paste Select All Format Copier Search Group Ungroup Alignment Distribution Layout Link Jump Bookmarks

Welcome to Visual Paradigm Community Edition

System Modeling Cloud Arch. Design Web Diagram Team More Features

UML +

Design software system by drawing UML diagrams

- Use Case
- Class
- Sequence
- Deployment
- Package
- Object

ERD +

Create conceptual, logical or physical data model that aids the communication and implementation of database design.

SysML +

Create SysML diagrams for system engineering applications.

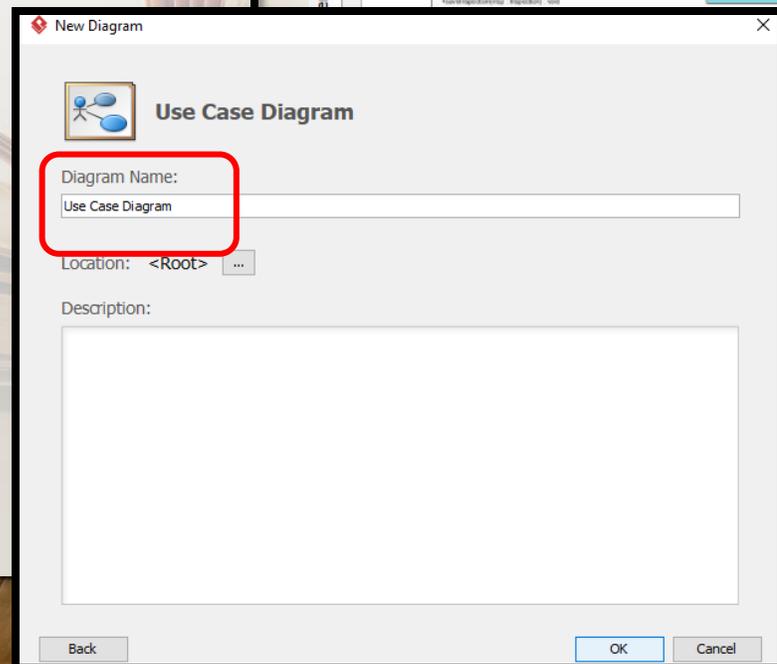
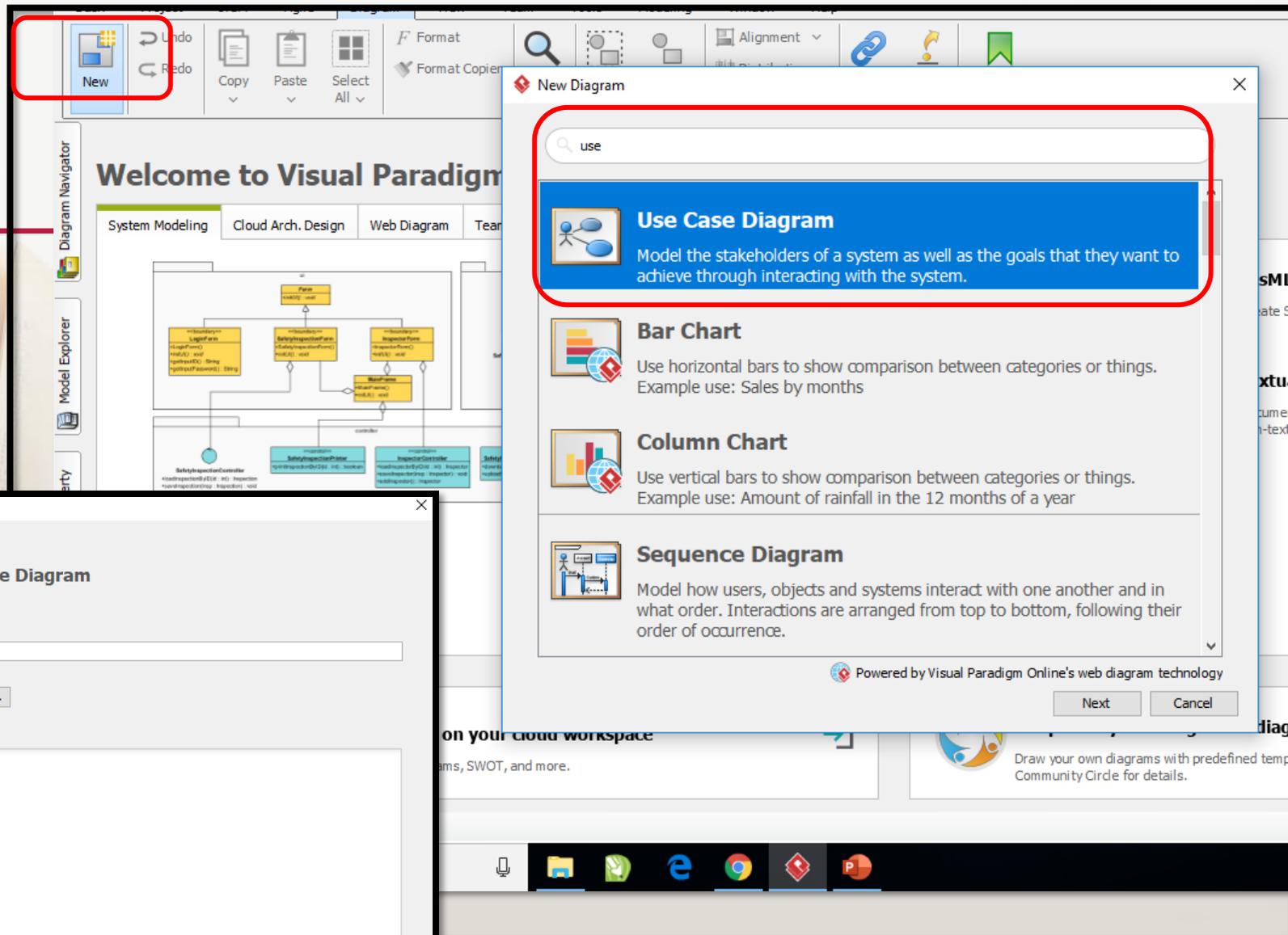
Textual Analysis +

Document project, user problem or other textual information in a rich-text editor and extract model elements from it.

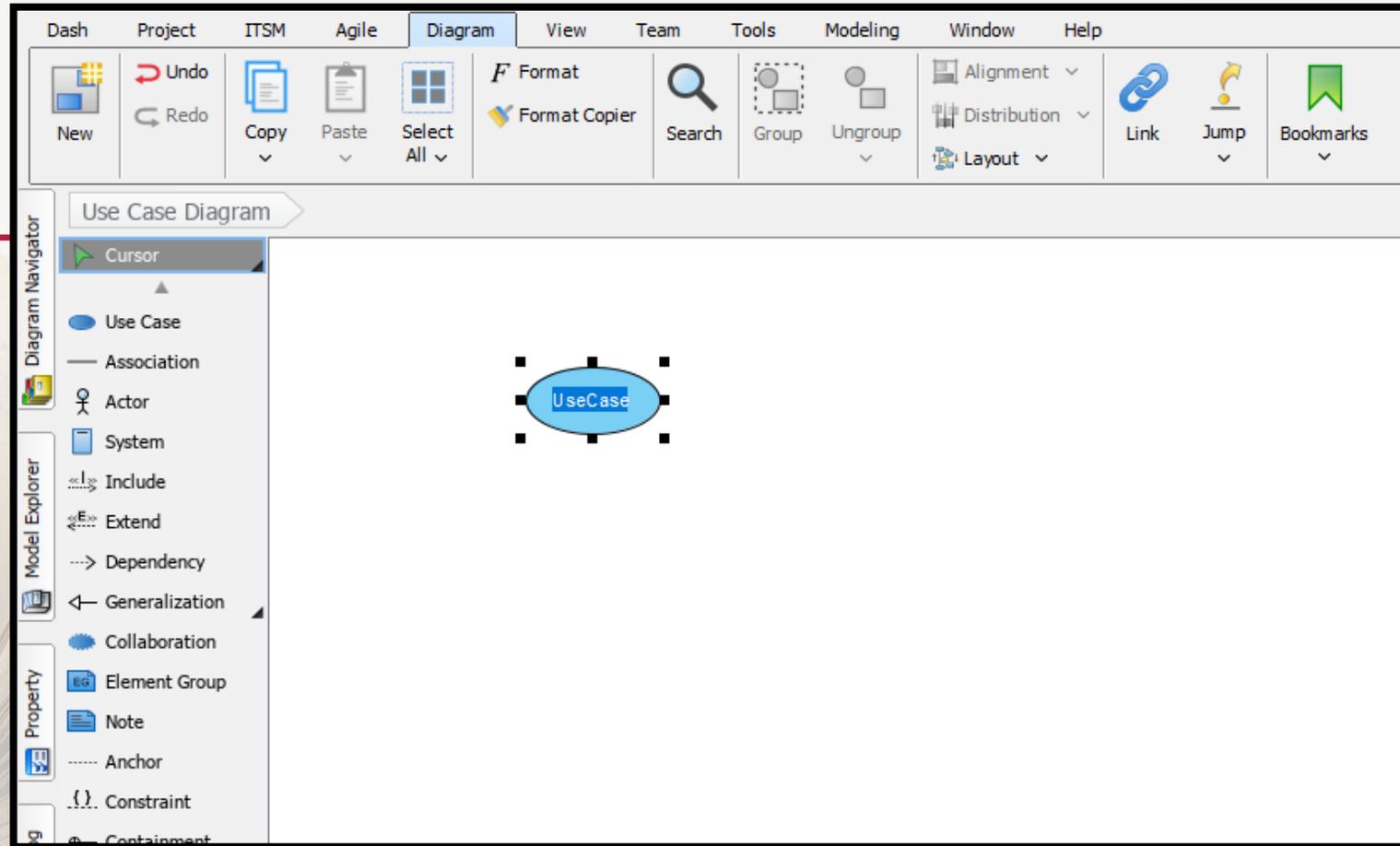
Store and share your design on your cloud workspace
Enjoy additional features like web diagrams, SWOT, and more.

Jump start your design with diagram templates
Draw your own diagrams with predefined templates. Hundreds of templates are available. Visit our Community Circle for details.

- Klik New
- Cari Use Case
- Klik Next
- Pilih Blank
- Klik Next
- Beri Nama Use Case
- Klik Oke

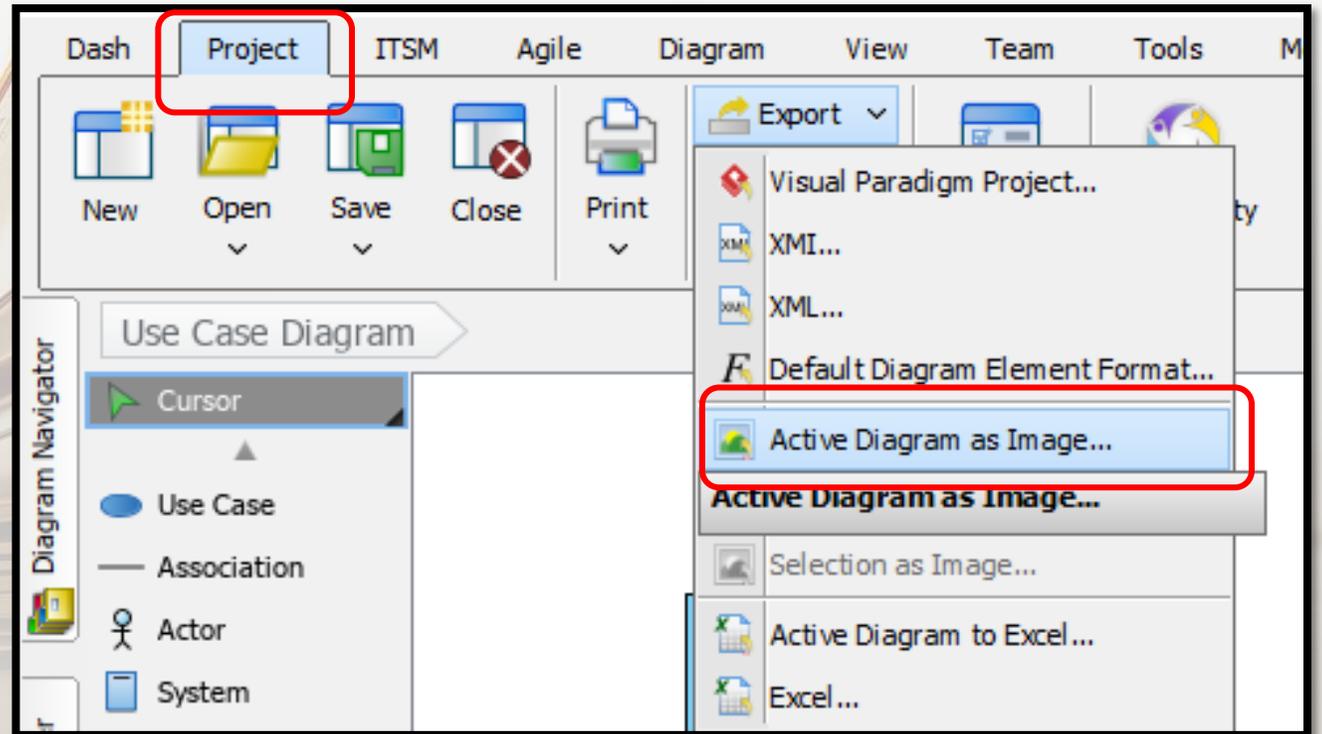


- Drag and Drop Komponen Use Case
- Berikan Nama dengan cara ketik langsung
- Untuk Edit klik 2x
- Lakukan hal yg sama untuk komponen use case lainnya



EXPORT DIAGRAM KE JPG

- Klik Tab Project
- Klik Export
- Pilih Active Diagram As Image
- Simpan Pada Lokasi yang dikehendaki
- Berikan Nama Gambar
- OK



CONTOH SOAL

- Pelanggan datang ke minimarket kemudian memilih barang yang ingin dibeli. Setelah pelanggan memilih barang yang ingin di beli, pelanggan memberikan barang tersebut ke kasir yang kemudian kasir akan melakukan scan barcode dari masing - masing barang. Setelah barang barang discan akan muncul total harga yang harus dibayar oleh pelanggan. Setelah pelanggan melakukan pembayaran, struk belanja akan otomatis tercetak. Jika pelanggan ingin menjadi member pada minimarket tersebut, pelanggan dapat mendaftarkan diri kepada kasir dengan mengisi data diri seperti nama, alamat dan nomor telepon.

TUGAS

- Kirim File hasil membuat diagram use case dengan format *.jpg untuk contoh soal tersebut ke SIP

