

**LAPORAN PRAKTIKUM**  
**ANALISIS DESAIN SISTEM**  
**PERTEMUAN KE – 8**



**Disusun Oleh :**

**NAMA : TARISA DWI SEPTIA**  
**NIM : 205410126**  
**JURUSAN : TEKNIK INFORMATIKA**  
**JENJANG : S1**

**Sekolah Tinggi Management Informatika Komputer**

**AKAKOM**

**YOGYAKARTA**

**2020**

## **Modul 8**

### **DASAR UNIFIED MODELING LANGUAGE (UML)**

#### **A. Tujuan**

- Mahasiswa dapat menggunakan UML untuk desain sistem

#### **B. Dasar Teori**

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya [HAN98]. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch , James Rumbaugh , dan Ivar Jacobson di bawah bendera Rational Software Corp [HAN98]. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

##### **Bagian-bagian Dari UML**

Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism.

##### **1. View**

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antara lain:

- Use case view,
- Logical view,
- Component view,
- Concurrency view,
- Deployment view.

##### **2. Diagram**

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

- Use Case Diagram
- Class Diagram
- State Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

##### **Tujuan UML**

- 1.** Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object
- 2.** Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Komponen-komponen yang terlibat dalam use case diagram :

- Actor

Actor digambarkan dengan stick man, tujuan dari actor sendiri adalah menciptakan suatu case diagram yang digunakan untuk tersebut mempresentasikan seseorang atau sesuatu yang berinteraksi dengan sistem.



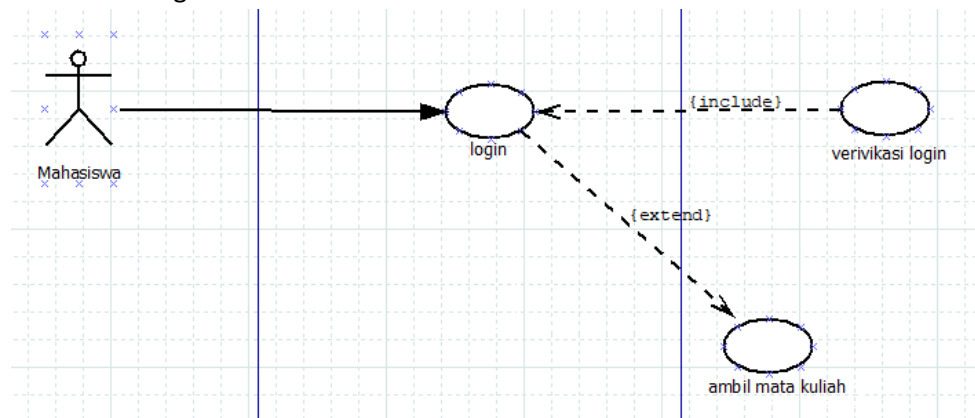
- Use Case

Use case ini merupakan bentuk fungsionalitas dari suatu sistem. Use case juga merupakan dialog antara actor dan sistem. Notasi use case di gambarkan seperti dibawah ini



### C. Praktik

Membuat Diagram Use Case



Membuat actor pada DIA

1. Klik gambar Actor pada toolbar, lalu letakkan pada documentation window
2. Klik gambar Actor tersebut, dan namakan Mahasiswa.
3. Ulangi langkah 2 untuk menambah actor yang diperlukan pada diagram.

Membuat Use Case

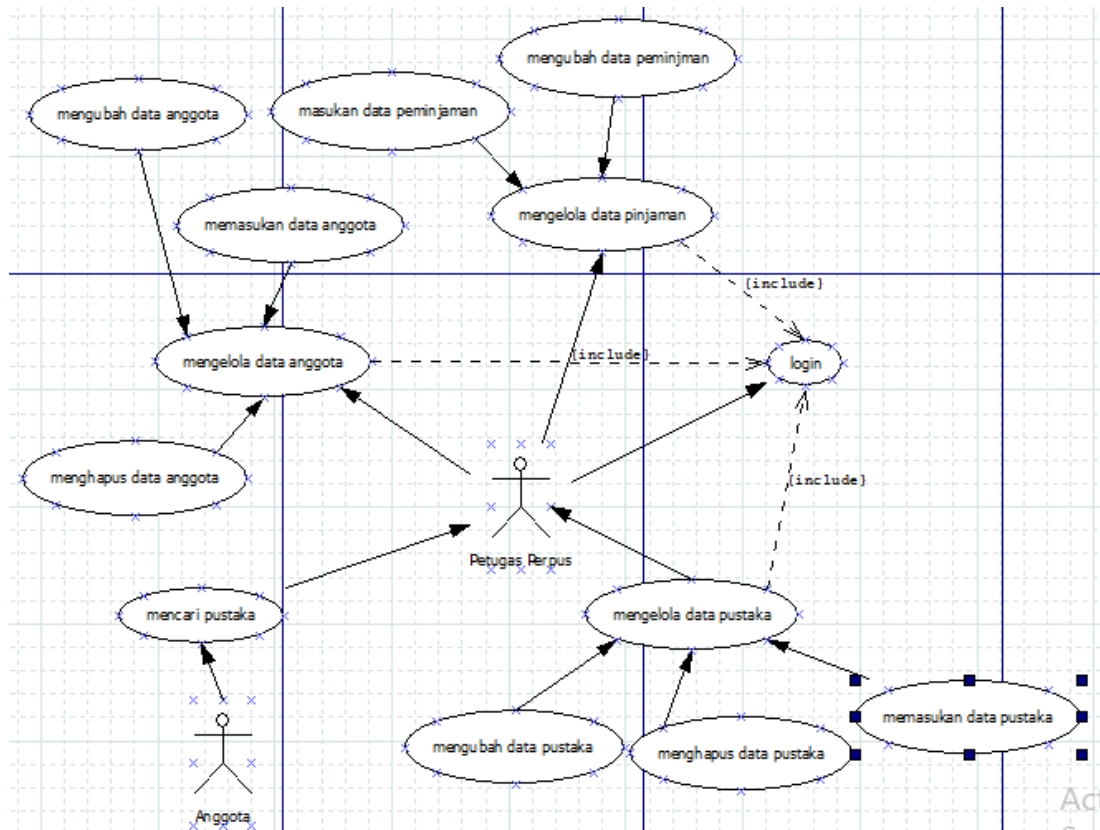
1. Klik gambar Use Case , pada toolbar, lalu letakkan pada documentation window
2. Klik gambar Use Case tersebut, dan namakan Login
3. Ulangi langkah 2 untuk menambah use case pada diagram

Membuat Relasi

1. Klik gambar , untuk membuat relasi dependency dari actor ke use case.
2. Ulangi no. 1 untuk menambahkan dependency dari actor ke use case lainnya
- Ulangi no. 1 untuk menambahkan dependency dari actor ke use case lainnya
1. Klik gambar constraint, untuk membuat hubungan dari use case ke use case.
2. Ulangi no. 3 untuk menambahkan dependency dari use case ke use case lainnya

3. Double click pada relasi dari Use Case : Verifikasi Login ke Use Case : Login, lalu pada Stereotype, pilih <include>
4. Ulangi langkah 5, untuk membuat relasi <>.dari Login ke Ambil Mata Kuliah, pada Stereotype, pilih <extend>

#### D. Latihan



#### E. Kesimpulan

Setelah melakukan praktik diatas dapat disimpulkan bahwa dapat menggunakan UML untuk desain sistem.