

# PEMODELAN SISTEM BERORIENTASI OBJEK

## ***UML (Unified Modeling Language)***

### ***Pengertian UML***

UML adalah bahasa pemodelan visual yang digunakan untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak.

### ***Pengertian Class dan Objek***

Class adalah cetakan, template atau blueprint

Objek adalah bentuk nyata dari class.

### ***Sejarah UML***

Pada Oktober 1994, Dr. James Rumbaugh bergabung dengan Perusahaan Rational software, dimana Grady Booch sudah bekerja disana sebelumnya. Grady Booch mengembangkan Object Oriented Design (OOD) dan Dr. James Rumbaugh mengembangkan Object Modeling Technique (OMT). Duet Mereka pada Oktober 1995 menghasilkan Unified Method versi 0.8

Musim gugur 1995 Dr. Ivar Jacobson ikut pula bergabung dengan duet RumbaughBooch, dengan memperkenalkan tool use case. Trio tersebut pada bulan Juni 1996 menghasilkan Unified Modeling Language (UML) versi 0.9.

Sebelumnya Dr. Ivar Jacobson mengembangkan Object Oriented Software Engineering (OOSE)

Banyak perusahaan software merasakan bagaimana pentingnya UML dalam tujuan strategis mereka, sehingga beberapa perusahaan membentuk sebuah konsorsium yang terdiri dari perusahaan-perusahaan seperti Microsoft, Oracle, IBM, Hewlett-Packard, Intellicorp, I-Logix, DEC, Digital Equipment Corp. texas instrument

### ***UML Tools***

***Microsoft Visio draw.io***

***Enterprise Architect***

***Star UML***

***Netbeans UML Plugin***

***Rational Rose***

***Visual Paradigm***

### ***Diagram UML***

UML versi 2.4 memiliki 14 diagram yang dibagi ke dalam 2 grup utama yaitu Structure Diagram dan Behavior Diagram

### ***Structure Diagram***

Structure Diagram merepresentasikan data dan hubungan statis di dalam suatu sistem informasi.

### ***Structure Diagram terdiri dari:***

#### **Class Diagram**

Mewakili sesuatu, contoh: pegawai, gaji, dst.  
Menunjukkan relasi antar class

#### **Object Diagram**

Mirip dengan class diagram  
Menunjukkan relasi antar objek

#### **Package Diagram**

Mengelompokkan elemen-elemen UML untuk membentuk tingkat konstruksi yang lebih tinggi

#### **Deployment Diagram**

Menunjukkan arsitektur fisik dan komponen perangkat lunak dari sistem –  
Contoh: simpul jaringan (network nodes)

#### **Component Diagram**

Relasi fisik diantara komponen perangkat lunak  
Contoh: Client/Server o Composite Structure Diagram  
Mengilustrasikan struktur internal dari sebuah class yang kompleks

#### **Behavior Diagram**

Behavior Diagram menggambarkan relasi dinamis diantara objek yang representasikan bisnis sistem informasi.

### ***Behavior Diagram terdiri dari:***

#### **o Activity Diagram**

Memodelkan proses dalam suatu sistem informasi  
Contoh: alur kerja bisnis (business workflows), logika bisnis (business logic)

#### **o Sequence Diagram**

Interaction Diagram  
Menunjukkan interaksi diantara objek

#### **o Communication Diagram**

Pengurutan interaksi berdasarkan waktu

#### **o Interaction Diagram**

Komunikasi antara sekumpulan objek yang berkolaborasi dari suatu aktivitas

### **o Timing Diagram**

Menunjukkan bagaimana suatu objek berubah seiring waktu

### **o Behavior State Machine**

Memeriksa perilaku suatu class

Memodelkan keadaan dan transisi keadaan yang berbeda yang dapat dialami suatu objek

### **o Protocol State Machine**

Mengilustrasikan ketergantungan antara berbagai antarmuka dari suatu class

### **o Use Case Diagram**

Menunjukkan interaksi antara sistem dan lingkungannya – Menangkap kebutuhan bisnis

## ***USE CASE DIAGRAM***

Use case adalah sebuah teknik untuk menangkap kebutuhan fungsional sistem.

Use case menggambarkan interaksi antara pengguna dengan sistem.

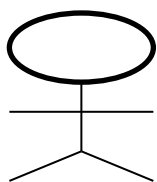
Use case menyediakan sebuah narasi bagaimana sebuah sistem digunakan.

Sebuah use case menggambarkan bagaimana pengguna berinteraksi dengan sistem untuk melakukan beberapa aktivitas, seperti menempatkan pesanan, membuat reservasi, atau mencari informasi.

Use case digunakan untuk mengidentifikasi dan mengkomunikasikan persyaratan (requirement) untuk sistem ke programmer yang harus menulis sistem.

### **Elemen-Elemen Use Case Diagram**

#### ***Actor***



*Orang atau sistem lain yang berinteraksi dengan sistem saat ini.  
Sebuah peran, bukan pengguna spesifik.  
Memberikan input, menerima output, ataupun keduanya.*

#### ***Use case***



*Bagian utama dari fungsionalitas sistem.  
Bisa extend (memperluas) use case lainnya.  
Ditempatkan di dalam system boundary (batasan sistem).  
Dilabeli dengan kata kerja – frase kata benda.*

Boundary

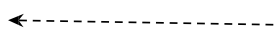
*Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas boundary.  
Mewakili ruang lingkup sistem.  
Actor berada di luar ruang lingkup sistem.*

#### **Association Relationship**

*Menghubungkan actor dengan use case.  
Menunjukkan komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah).  
Tanda \* untuk keragaman dari asosiasi (multiplicity of the association).  
Namun umumnya hanya digambarkan garis saja.*

#### **Include Relationship**

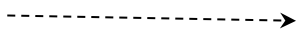
**<<include>>**



*Memasukkan satu use case dalam use case lainnya.  
Perilaku (behavior) yang harus terpenuhi agar  
Sebuah event dapat terjadi, di mana kondisi ini  
Sebuah use case adalah bagian dari use case lainnya.  
Tanda panah mengarah dari base use case (pusat)  
menuju ke use case yang di-include*

#### **Extend Relationship**

**<<extend>>**



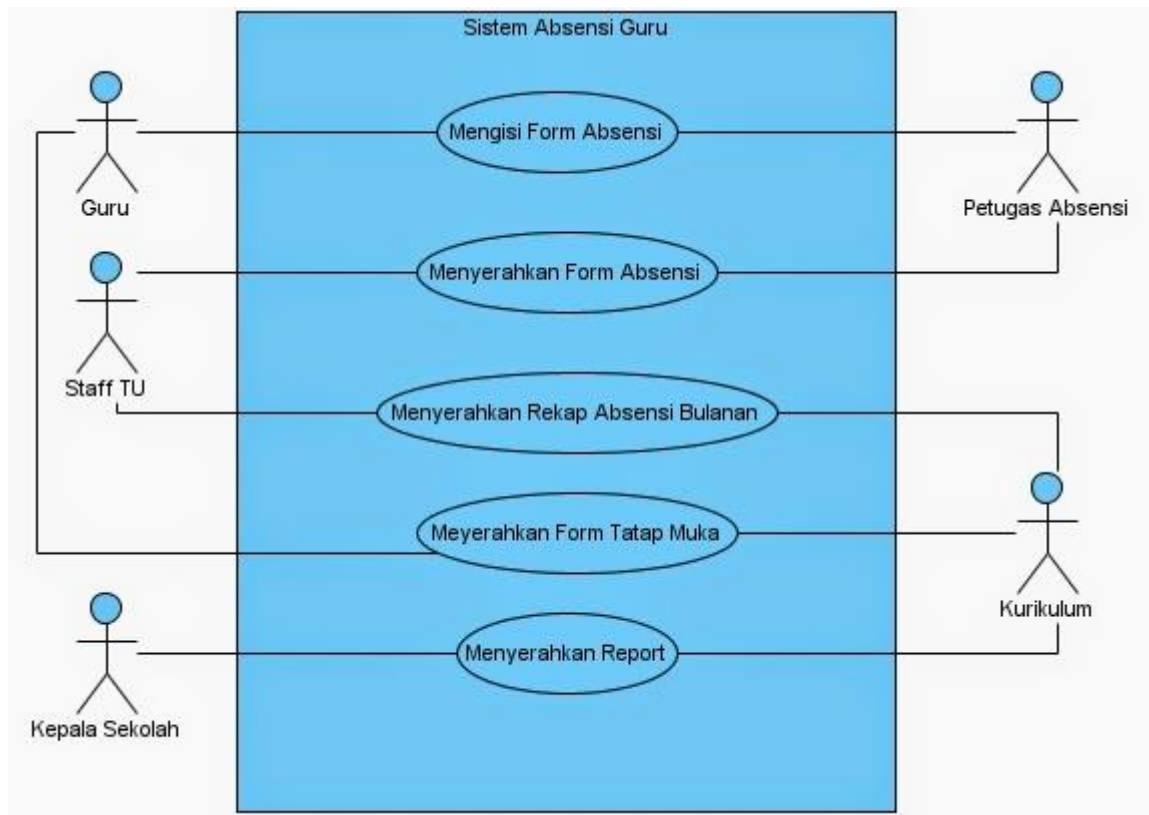
*Memperluas use case untuk memasukkan perilaku opsional.  
Tanda panah mengarah dari use case tambahan ke base use case (pusat)*



**Generalization**

*Mewakili use case khusus untuk use case yang lebih umum.  
Tanda panah mengarah dari use case khusus (specialized) ke use case yang lebih umum*

## CONTOH USE CASE DIAGRAM



## ACTIVITY DIAGRAM

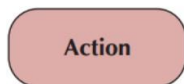
### Pengertian Activity Diagram

Activity diagram digunakan untuk memodelkan proses dalam suatu sistem informasi. Activity diagram menggambarkan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses.

Setiap use case dapat membuat satu diagram aktivitas.

### Elemen-Elemen Activity Diagram

#### Action

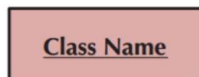


*Perilaku yang sederhana dan tidak dapat diuraikan.  
Dilabeli dengan namanya.*

#### Activity



*Digunakan untuk mewakili sekumpulan tindakan (action).  
Dilabeli dengan namanya.*



#### Object Node

*Digunakan untuk mewakili sebuah objek yang terhubung ke  
sekumpulan arus objek.*

*Dilabeli dengan nama classnya.*

#### Control Flow

*Menunjukkan urutan eksekusi*



#### Object Flow



*Menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke  
aktivitas (atau tindakan) lain.*

#### Initial Node



*Menggambarkan awal dari serangkaian tindakan atau kegiatan.*

### **Final-Activity Node**



**Digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).**

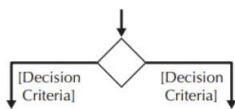
### **Final-Flow Node**



**Digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.**

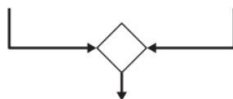
### **Decision Node**

**Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Dilabeli dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.**



### **Merge Node**

**Digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.**



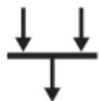
### **Fork Node**

**Digunakan untuk membagi perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).**



### **Join Node**

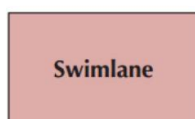
**Digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan**



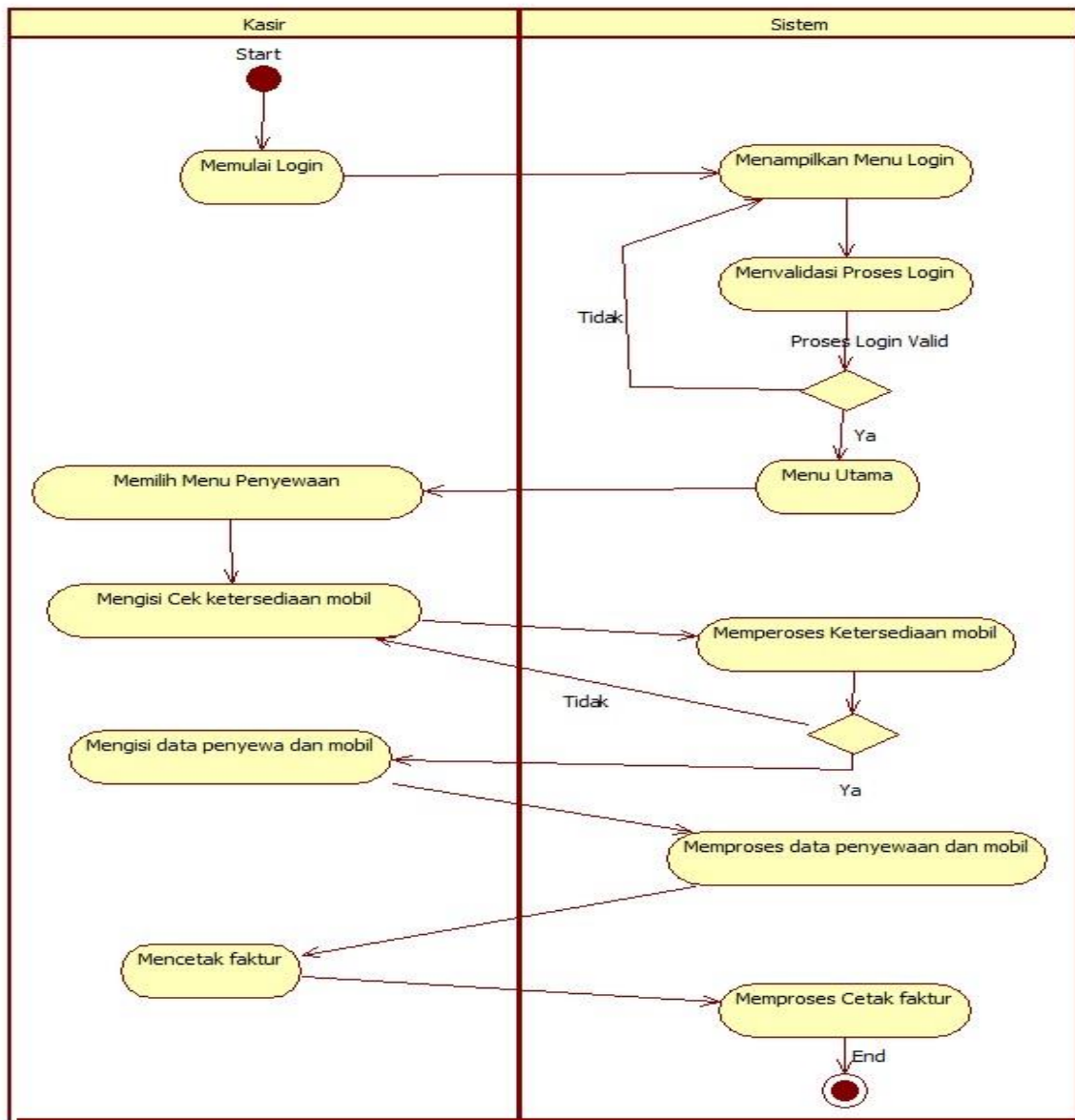
### **Swimlane**

**Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan).**

**Dilabeli dengan nama individu atau objek yang bertanggung jawab.**



## Contoh gambaran Activity Diagram





## ***Latihan***

### **Proses Bisnis Sistem**

Staff perpustakaan memberikan formulir pendaftaran kepada siswa, lalu siswa tersebut mengisi formulir, melengkapi persyaratan dan melakukan pembayaran.

Kemudian staff perpustakaan menerima formulir yang telah diisi tersebut beserta persyaratan dan pembayaran biaya pendaftaran anggota sebesar Rp 5.000,00.

Lalu staff perpustakaan akan mencatat data siswa dan membuatkan kartu anggota.

Siswa yang sudah mendaftar menjadi anggota perpustakaan dan sudah mempunyai kartu anggota dapat langsung meminjam buku ke perpustakaan.

Siswa memilih buku yang diinginkannya lalu menyerahkan buku tersebut dan kartu anggotanya ke staff perpustakaan.

Kemudian staff perpustakaan akan mencatat ke dalam buku peminjaman data siswa tersebut dan buku yg dipinjam serta lama peminjaman.

Siswa yang ingin mengembalikan buku yang telah dipinjam dapat langsung menyerahkan buku tersebut kepada staff perpustakaan beserta kartu anggotanya.

Kemudian staff perpustakaan akan memeriksa buku tersebut apakah ada kerusakan atau tidak dan ada keterlambatan dalam pengembalian buku atau tidak, jika ada keterlambatan atau kerusakan pada buku tersebut, maka siswa yang bersangkutan diharuskan untuk membayar.

Kemudian staff perpustakaan akan mencatat pengembalian buku tersebut di buku pengembalian beserta denda jika ada.

***Buatlah activity diagram dari proses bisnis diatas ?***