

UNIFIED MODELLING LANGUAGE

Dr. Muhammad Said Hasibuan



+

•

○

Definisi

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997

tujuan dan fungsi perlu adanya UML yaitu sebagai berikut:

Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.

Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.

Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.

Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.

Mempermudah pengguna untuk membaca suatu sistem.

Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program.






Fungsi dari Use Case

- **Memperlihatkan Urutan Aktivitas Proses dalam Sebuah Sistem**

Fungsi yang pertama, mampu memperkenalkan fase awal setiap kegiatan proses dalam sistem yang dikembangkan. Hal tersebut dapat memudahkan pengembang dalam menentukan kebutuhan yang sesuai dengan perangkat lunak dan pengguna.

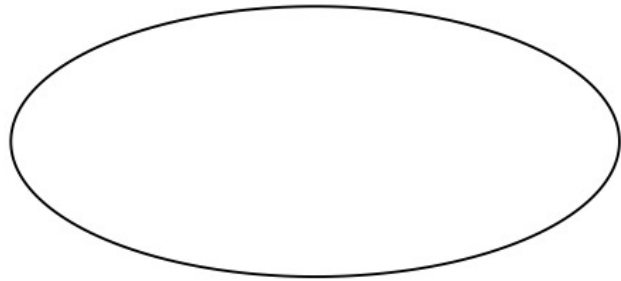
- **Menggambarkan *Business Process* dalam Sistem**

Kedua, *use case* mampu menggambarkan urutan proses bisnis secara lebih jelas dan transparan untuk mencegah terjadinya kesalahan pada sistem yang akan dibangun.



Simbol - simbol pada diagram use case

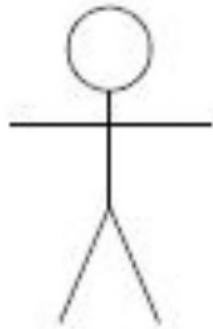
Use Case



- Fungsionalitas yang disediakan sistem sebagai unit - unit yang saling bertukar pesan antar unit atau aktor
- Biasanya dinyatakan dengan menggunakan kata kerja awal di awal frase nama *use case*
- Use case merepresentasikan fungsi, kebutuhan dari prespektif user.

Simbol - simbol pada diagram use case

Actor/Aktor



- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
- Aktor adalah orang atau sistem yang menerima atau memberikan informasi dari sistem.

Karakteristik Aktor

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
- Simbol aktor adalah orang, tapi aktor belum tentu merupakan orang.
- Aktor adalah eksternal terhadap sistem.
- Aktor berinteraksi dengan sistem.
- Aktor memanfaatkan fungsi yang disediakan sistem, termasuk fungsi aplikasi dan pemeliharaan.
- Aktor bisa menerima dan menyediakan informasi yang pada sistem.
- Aktor class memiliki objek aktor yang menyatakan aktor tertentu.

Relasi dalam diagram use case

Asosiasi/ Association

- Menghubungkan link antar element.
- Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case yang memiliki interaksi dengan aktor.

Relasi dalam diagram use case

Generalisasi/ *generalization*



- Generalisasi disebut juga inheritance (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
- Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)

Relasi dalam diagram use case

Include/uses

`<<include>>`
.....➤

- Perilaku yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya
- Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.
- Include berarti use case yang ditambahkan akan selalu dipanggil saat usecase tambahan dijalankan.
- Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan.

Relasi dalam diagram use case

Ekstensi/Extend



- Perilaku yang hanya berjalan di bawah kondisi tertentu.
- Mirip dengan relasi generalisasi tetapi dengan aturan tambahan di dalamnya.
- Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri, walau tanpa use case tambahan.

Langkah- langkah



Pertama, identifikasi terlebih dahulu siapa saja yang akan menggunakan *website* tersebut.

Pilih salah satu *user* dari situs *web* yang akan diimplementasikan.

Tentukan juga apa saja yang akan dilakukan oleh pengguna di dalam situs tersebut. Setiap hal yang dilakukan *user* di dalam *website* akan menjadi *use case*.

Untuk setiap *use case* -nya, anda perlu untuk memutuskan rangkaian setiap kejadian (aktivitas) secara normal saat *user* menggunakan *website*.

Menjelaskan tahapan dasar dalam mendeskripsikan sebuah *use case diagram*. Anda dapat mencoba menjelaskan hal apa saja yang dilakukan pengguna dan apa yang bisa dilakukan sistem, sebagai bentuk respons yang perlu untuk diketahui.

Saat tahapan dasar dilakukan, anda dapat mempertimbangkan rangkaian acara (*timeline*) alternatif dan tambahkan "*extend*" pada diagram.

Selanjutnya, anda dapat mencari kesamaan di antara *use case*, kemudian ekstrak dan catat hasilnya.

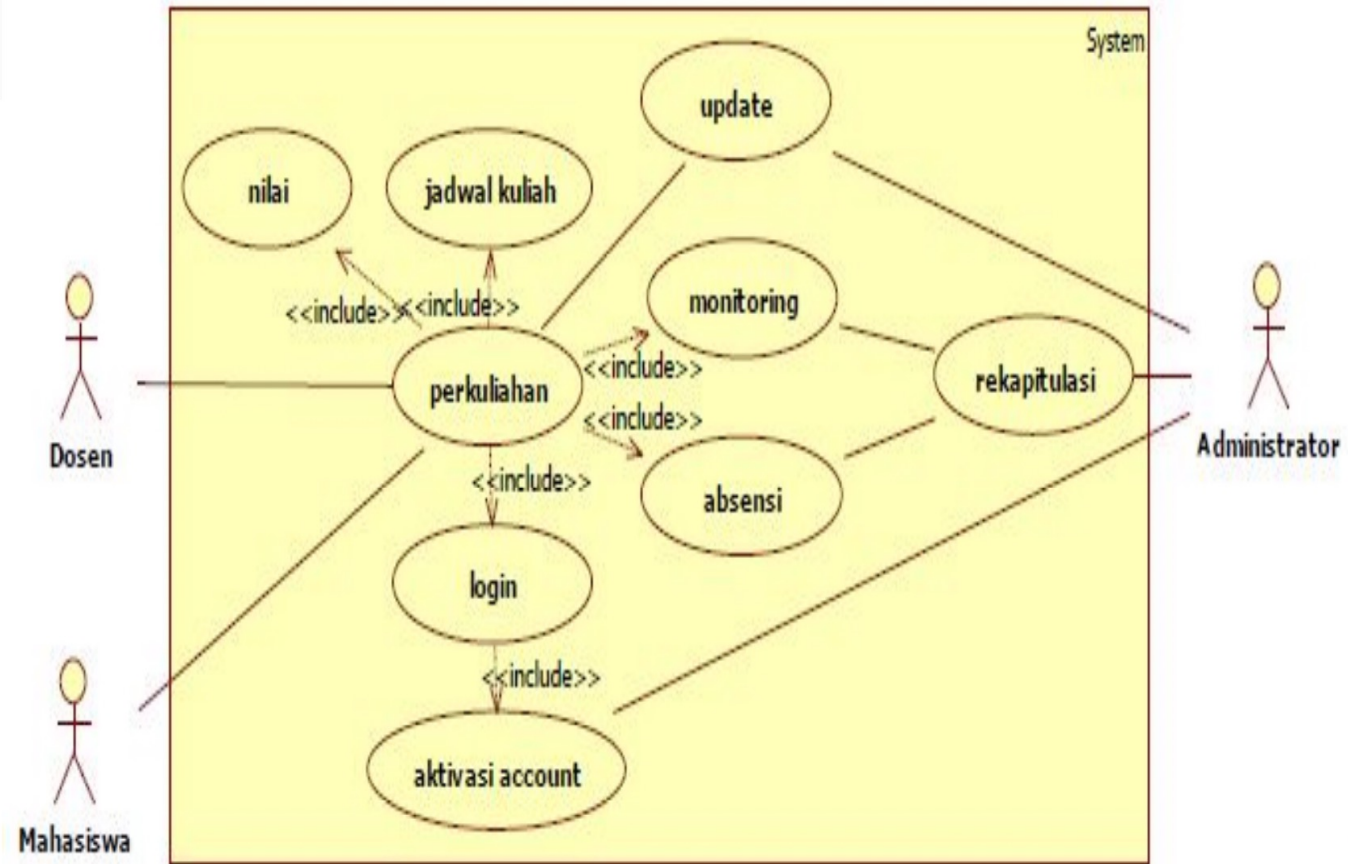
Ulangi langkah kedua hingga ketujuh untuk *user* yang lainnya.

Deskripsi User

NO	User	Deskripsi
1	Petugas Perpustakaan	Orang yg bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pustaka, anggota dan proses peminjaman pustaka
2	Anggota	Orang yg diperbolehkan meminjam pustaka sesuai dengan hak aksesnya, sedangkan pengunjung hanya memiliki hak akses melihat pustaka dan membaca di perpustakaan tanpa memiliki hak untuk meminjam pustaka

Skenario

- Setiap use case dilengkapi dengan skenario. skenario Use case adalah alur jalannya proses use case dari sisi aktor dan sistem.
- Skenario normal adalah skenario bila sistem berjalan normal tanpa terjadi kesalahan atau error.
- Sedangkan skenario alternatif adalah skenario bila sistem tidak berjalan normal, atau mengalami error



Skenario use case bertujuan untuk mendeskripsikan atau menjelaskan diagram use case.

Nama Use Case : Absensi.

Aktor : Mahasiswa.

Deskripsi : Sebagai dokumentasi untuk mahasiswa yang sudah mengikuti atau memasuki perkuliahan.

Skenario Utama

Kondisi Awal : user (mahasiswa) sudah login.

No	Aksi Aktor	No	Reaksi Sistem
1	Memilih menu absensi.	2	Menampilkan form absensi.
3	Input data absensi.	4	Menyimpan data absensi ke dalam database.

Kondisi Akhir : data absensi tersimpan ke dalam database.

Terima kasih