

MODUL 1

USE CASE DIAGRAM

Tujuan Praktikum

1. Mahasiswa mampu membuat sebuah skenario suatu sistem yang nantinya dapat diimplementasikan menjadi sebuah perangkat lunak.
2. Mahasiswa bisa memahami alur dari setiap tahap yang digunakan dalam perancangan perangkat lunak menggunakan UML.
3. Mahasiswa dapat memahami hubungan antara *actor* dengan *use case diagram*.
4. Mahasiswa mampu membuat *use case diagram* dari skenario yang telah ada.

1.1 Pengertian Use Case Diagram

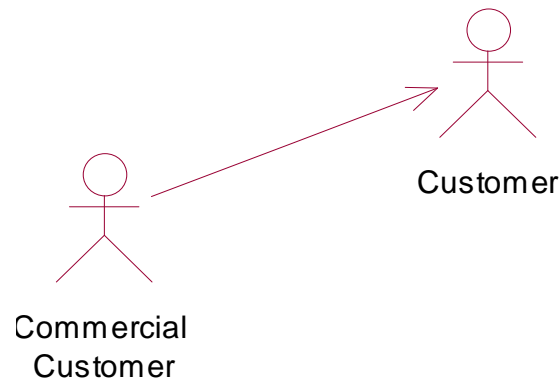
Use case diagram adalah gambaran graphical dari beberapa atau semua *actor*, *usecase*, dan *interaksi* diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Use case diagram menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, use case diagram berperan untuk menetapkan perilaku (behavior) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa use case diagram. Kebutuhan atau *requirements* sistem adalah fungsionalitas apa yang mesti disediakan oleh sistem kemudian didokumentasikan pada model *use case* yang menggambarkan fungsi sistem yang diharapkan (*use case*), dan yang mengelilinginya (*actor*), serta hubungan antara *actor* dengan *use case* (*use case diagram*) itu sendiri.

1.2 Komponen yang terlibat dalam Use Case Diagram :

1.2.1 Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.



Gambar 1. 1 Actor

Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain:

- Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang dia inputkan ke sistem.
- Orang ataupun pihak yang akan mengelola sistem tersebut.
- *External resource* yang digunakan oleh sistem.
- Sistem lain yang berinteraksi dengan sistem yang akan dibuat.

Membuat actor pada Rasional Rose 2000

- a. Klik pada *Use Case View package* di *browser*.
- b. Pilih *New:Actor* pada menu *option* . Sebuah *actor* baru bernama *New class* ditempatkan di *browser*.
- c. Pilih *actor New class*, lalu masukkan nama yang diinginkan untuk *actor* tersebut

Mendokumentasikan actors

- a. Jika *documentation window* belum terlihat, buka dengan memilih *Documentation*
- b. *menu* dari *View menu*.
- c. Klik untuk memilih *Actor* di *browser*.
- d. Tempatkan *cursor* di *documentation window*, lalu ketikkan dokumentasi yang
- e. diinginkan.

1.2.2 Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. *Catatan* : *Use case diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

Cara menentukan Use Case dalam suatu sistem:

- Pola perilaku perangkat lunak aplikasi.
- Gambaran tugas dari sebuah *actor*.
- Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.
- Apa yang dikerjakan oleh suatu perangkat lunak (*bukan bagaimana cara mengerjakannya).



Use Case

Gambar 1. 2 Use Case

Membuat Use Cases

- Klik kanan *Use Cases View* pada browser.
- Pada menu *option* pilih *New:Use Case*. Sebuah *Use Case* ditempatkan pada *browser*.
- Klik *Use Case* tersebut, lalu masukkan nama yang diinginkan.

Membuat Use Case Diagram Utama

- Klik kanan *Main diagram* pada *Use Case View* di *browser* untuk membuka diagram.
- Klik *actor* di *browser* dan tarik *actor* ke dalam *diagram*.
- Ulangi langkah 2 untuk menambahkan *actor* yang diperlukan dalam diagram.
- Klik untuk memilih sebuah *use case* di *browser* dan tarik *use case* ke dalam diagram.
- Ulangi langkah 4 untuk menambahkan *use case* yang diperlukan dalam diagram.

Catatan :

actor dan *use cases* dapat juga langsung diciptakan dalam sebuah *use casediagram* dengan menggunakan *toolbar*.

1.2.3 Relasi dalam Use Case

Ada beberapa relasi yang terdapat pada *use case diagram*:

- Association*, menghubungkan link antar element.
- Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
- Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi/ *stereotype* yang mungkin terjadi pada *use case diagram*:

- <<include>>** , yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
- <<extends>>**, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.

3. <<**communicates**>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*.

1.2.4 Use Case Diagram

Use Case Diagram adalah gambaran graphical dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem.

Membuat Use Case Diagram Utama

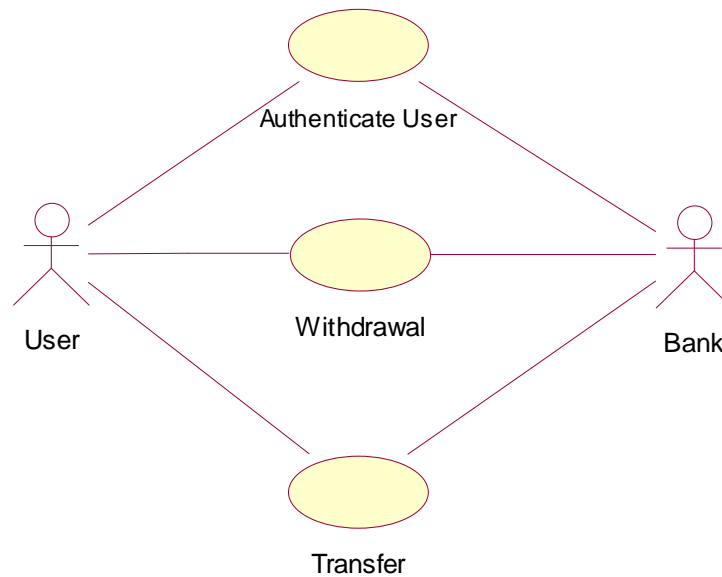
1. Klik kanan *Main diagram* pada *Use Case View* di *browser* untuk membuka diagram.
2. Klik *actor* di *browser* dan tarik *actor* ke dalam *diagram*.
3. Ulangi langkah 2 untuk menambahkan *actor* yang diperlukan dalam diagram.
4. Klik untuk memilih sebuah *use case* di *browser* dan tarik *use case* ke dalam diagram.
5. Ulangi langkah 4 untuk menambahkan *use case* yang diperlukan dalam diagram.

Catatan :

actor dan *use case* dapat juga langsung diciptakan dalam sebuah *use case diagram* dengan menggunakan *toolbar*.

Berikut ini adalah contoh dari sebuah studi kasus yang menangani Aplikasi pada sebuah ATM dengan skenario sbb:

Sebuah bank mengoperasikan ATM dan mengelola banyak tabungan. Setiap nasabah memiliki setidaknya satu rekening tabungan pada satu bank tertentu. Setiap tabungan dapat diakses melalui kartu debit. Proses utama sistem ATM berkomunikasi dengan pusat komputer dan didesain untuk menangani beberapa transaksi. Setiap transaksi menunjuk sebuah tabungan tertentu. Suatu transaksi akan menghasilkan satu dari dua hal berikut: transaksi diterima atau mengeluarkan pesan penolakan transaksi. Untuk melakukan sebuah transaksi akan melalui dua tahap: pengecekan tabungan dan pemrosesan transaksi. Proses pengecekan tabungan akan menetapkan persetujuan untuk proses transaksi. Jika persetujuan ditolak, ATM akan mengeluarkan pesan penolakan. Namun jika diterima, transaksi akan diproses dengan menggunakan nomor rekening tabungan dan ATM membaca dari kartu debit. Pengecekan tabungan dilakukan bersamaan pada saat ATM memvalidasi kartu debit dari bank yang bersangkutan. Jika kartu valid, password akan dicek dengan data nasabah.



Gambar 1. 3 Use Case Diagram Studi Kasus ATM

Untuk memudahkan kita dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya, maka kita dapat melakukan pemilahan terhadap skenario tersebut, antara lain:

Skenario use case

Nama use case : Authenticate user

Actor : User, bank

Type : Primary

Tujuan : verifikasi user

Deskripsi : User datang ke ATM dengan kartu debit untuk melakukan penarikan tunai. User memasukkan kartu ke ATM. ATM meminta user untuk memasukkan PIN.

ACTOR	SISTEM
1. User Memasukkan kartu debit	
	2. ATM meminta PIN dari user
3. User memasukkan PIN dan menekan OK	
	4. ATM memverifikasi dengan Bank bahwa kartu dan PIN adalah legal dari rekening yang benar
	5. ATM meminta jenis transaksi

Nama use case : Withdrawal
 Actors : User, bank
 Type : Primary
 Tujuan : Penarikan uang secara cash
 Deskripsi : User datang ke ATM dengan kartu debit untuk melakukan penarikan tunai. User memasukkan kartu ke ATM. ATM meminta user untuk memasukkan PIN. User memasukkan PIN dan sistem mengotorisasi penarikan tunai. ATM mengeluarkan uang dan mengeluarkan nota. ATM mengirim transaction record ke bank untuk meng-update saldo tabungan. Setelah selesai, user meninggalkan ATM dengan membawa uang dan nota tadi.

ACTOR	SISTEM
1. User memilih menu withdrawal	
	2. ATM meminta jumlah uang yang akan di tarik
3. User memasukkan jumlah uang yang akan ditarik	
	4. ATM mengecek jumlah uang yang akan ditarik dengan saldo minimal yang diperbolehkan pada bank tersebut
	5. Update saldo
	6. ATM mengeluarkan uang
	7. ATM mencetak nota dan mengeluarkan kartu