

MODUL II

USE CASE DIAGRAM

1. Tujuan Praktikum

- Praktikan mengenal tentang Use Case Diagram
- Praktikan dapat membuat Use Case Diagram
- Praktikan mengimplementasikan Use Case Diagram pada kasus pada modul sebelumnya

2. Dasar Teori

2.1. Pengertian Use Case Diagram

Use case Diagram adalah Diagram deskripsi fungsi dari sebuah sistem dari perspektif pengguna. **Use case Diagram** bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antar pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Selain itu Use Case juga diartikan sebagai abstraksi dari interaksi antara system dan actor. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah

interaksi antara actor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya.

2.2. Kebutuhan Dasar / Fungsi Sistem

Kebutuhan dasar /fungsi sistem adalah fungsionalitas apa yang mesti disediakan oleh sistem, yang mana akan didokumentasikan sebagai use case yang menggambarkan fungsi sistem yang diharapkan. Atau yang lebih dikenal dengan Use Case Rule. Use Case Rule ini terdiri dari beberapa aktifitas yang dapat dijalankan sistem. Sebagai contoh proses pemesanan barang oleh pelanggan dalam kasus dimodul sebelumnya, sebelum memesan pelanggan harus sudah terdaftar sebagai anggota.

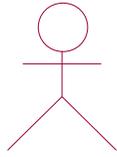
2.3. Tujuan Penggunaan Use Case Diagram

Digunakan sebagai alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang sistem dari sudut pandangnya. Use Case mewakili pandangan di luar sistem. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

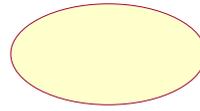
2.4. Komponen-komponen yang terlibat dalam use case diagram

a. Notasi Use Case

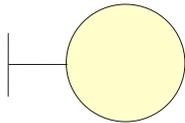
Diagram Use Case menunjukkan 3 aspek dari sistem yaitu: *actor*, *usecase*, dan sistem/sub sistem boundary. Aktor mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.



Notasi Aktor



Notasi Use Case

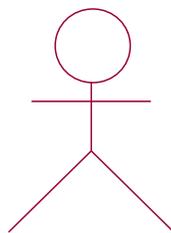


Notasi Boundari

Gambar 2.3. Notasi Use Case

b. Actor

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



Pada dasarnya actor bukanlah bagian dari use case diagram, namun untuk dapat terciptanya suatu use case diagram diperlukan beberapa actor dimana 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 member informasi pada sistem, actor hanya berinteraksi dengan use case tetapi tidak memiliki kontrol atas use case. Actor digambarkan dengan stickman. Actor dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan relationship. Contoh :



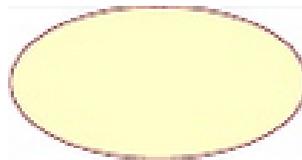
Ada beberapa kemungkinan yang menyebabkan actor tersebut terkait dengan sistem antara lain:

- 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.
- Sistem lain yang berinteraksi dengan sistem yang akan dibuat.

c. 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.).



Sistem Belanja Online

(from Use Case View)

Gambar 2.6. UseCase

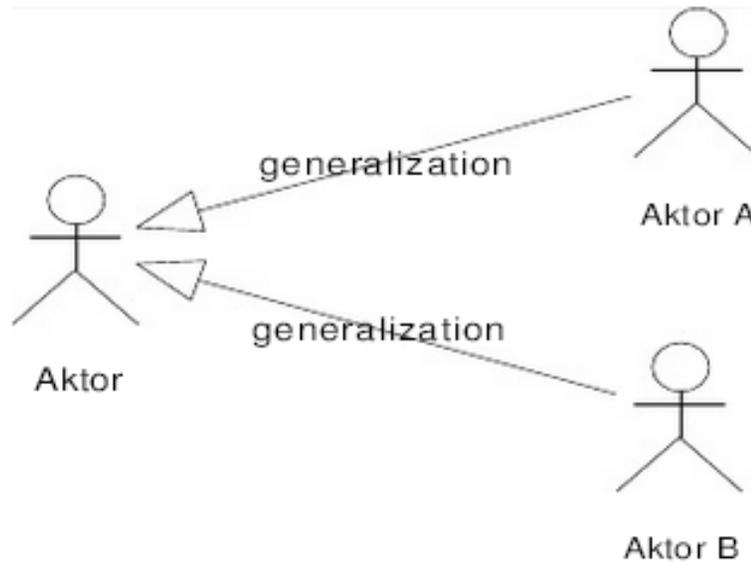
d. Relasi dalam Use Case

Relationship atau relasi merupakan hubungan antar elemen. Ada beberapa relasi yang terdapat pada use case diagram:

1. **Association** , menghubungkan link antar element.



2. **Generalization**, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.



3. **Dependency**, sebuah element bergantung dalam beberapa cara ke element lainnya.



4. **Aggregation**, bentuk association dimana sebuah elemen berisi elemen lainnya.

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

1. <<**include**>>, yaitu kelakuan yang harus terpenuhi agar sebuah hevent dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya.
2. <<**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.

3. Langkah Praktikum

3.1. Menganalisa kasus yang ada

Dari kasus yang ada dapat disimpulkan beberapa kebutuhan dasar/fungsi sistem:

1. Pelanggan melakukan registrasi yang mengisikan profil serta alamat penagihan, yang kemudian akan mendapatkan **kode member**.
2. Pelanggan harus melakukan proses login terlebih dahulu untuk masuk dalam halaman pemesanan, setelah masuk pelanggan memilih buku yang akan dipesan. Setelah yakin dengan pesanannya, pelanggan akan melakukan proses persetujuan pembelian yang berisi batas pembayaran(1x24 jam) serta info pembayaran dan

nomor order pemesanannya dan sistem akan mengurangi stok barang yang dipesan dan status pemesanan menjadi 'PESAN'.

3. Sistem secara otomatis mengirimkan email kepada pelanggan yang berisi daftar detail barang yang dipesan, harga yang harus dibayar info pembayaran dan nomor rekening untuk pembayaran serta batas pembayaran.
 4. Dalam batas pembayaran (1x24 jam) pelanggan harus transfer uang ke rekening toko sesuai pesanan setiap nomor order dan mengirim SMS dengan format "NO_ORDER NO_SLIP_TRANSFER". Jika dalam batas pembayaran belum transfer maka pesanan akan dibatalkan dan stok dikembalikan.
 5. Setelah transfer dan konfirmasi sampai maka toko akan mengirimkan barang dan sistem secara otomatis mengirimkan email yang berisi data pengiriman serta mengubah status pemesanan menjadi 'KIRIM'.
- Dari hasil analisa diatas dapat disimpulkan bahwa kasus tersebut terdiri dari **dua (2) actor**, yaitu:
1. Pelanggan
 2. Pemilik (Toko)

- Dari hasil analisa diatas dapat disimpulkan bahwa kasus tersebut ter diri dari **use case**, yaitu:

No	Requirement	Actor	Use Case
1	Pelanggan melakukan registrasi yang mengisikan profil serta alamat penagihan, yang kemudian akan mendapatkan kode member .	Pelanggan	Registrasi Member
2	Pelanggan harus melakukan proses login terlebih dahulu untuk masuk dalam halaman pemesanan, setelah masuk pelanggan memilih buku yang akan dipesan. Setelah yakin dengan pesanannya, pelanggan akan melakukan proses persetujuan pembelian yang berisi batas pembayaran (1x24 jam) serta info pembayaran dan nomor order pemesanannya dan sistem akan mengurangi stok barang yang dipesan dan status pemesanan menjadi 'PESAN'.	Pelanggan	Pemesanan, Login
3	Sistem secara otomatis mengirimkan	Pelanggan	Pemesanan,

	email kepada pelanggan yang berisi daftar detail barang yang dipesan, harga yang harus dibayar info pembayaran dan nomor rekening untuk pembayaran serta batas pembayaran.		Pengiriman Informasi
4	Dalam batas pembayaran (1x24 jam) pelanggan harus transfer uang ke rekening toko sesuai pesanan setiap nomor order dan mengirim SMS dengan format "NO_ORDER NO_SLIP_TRANSFER". Jika dalam batas pembayaran belum transfer maka pesanan akan dibatalkan dan stok dikembalikan.	Pelanggan, Toko	Pembayaran, Transfer
5	Setelah transfer dan konfirmasi sampai maka toko akan mengirimkan barang dan sistem secara otomatis mengirimkan email yang berisi data pengiriman serta mengubah status pemesanan menjadi 'KIRIM'.	Toko, Pelanggan	Pembayaran, Pengiriman Barang

Beberapa **Scenario** terkait use case diatas, yaitu:

1. Use Case : Registrasi member

Actor : Pelanggan

Tujuan : Pelanggan mendaftar menjadi member

ACTOR	SISTEM
1. Pelanggan membuka website. 2. Pelanggan memasukkan data pribadi dan alamat penagihan	
	3. Sistem mengecek keabsahan data pelanggan. 4. Sistem memberikan kode member

2. Use Case : Pemesanan

Actors : Pelanggan

Tujuan : Pemesanan barang yang diinginkan

ACTOR	SISTEM
1. Pelanggan login menggunakan kode member yang telah dimiliki	
	2. Sistem memverifikasi proses login pelanggan. 3. Sistem memperbolehkan pelanggan memasuki halaman pemesanan.
4. Setelah login pelanggan dapat memilih barang yang akan dipesan.	
	5. Sistem mengecek jumlah stok barang yang akan dipesan. 6. Sistem menghitung total harga pembelian.
7. Pelanggan melakukan proses pemesanan dengan menyetujui persetujuan pembelian	
	8. Sistem memberikan nomor order kepada pelanggan dan mengubah status jadi 'PESAN' 9. Sistem mengurangi stok barang yang telah dipesan pelanggan 10. Sistem secara otomatis mengirim email sebagai informasi pemesanan kepada pelanggan

3. Use Case : Pembayaran

Actors : Pelanggan, Toko

Tujuan : Pelanggan mentransfer uang untuk pemesanan

ACTOR	SISTEM
1. Pelanggan melakukan transfer uang ke rekening toko 2. Pelanggan mengirim SMS sebagai konfirmasi transfer uang dengan format "NO_ORDER NO SLIP TRANSFER".	
	3. Sistem akan menerima konfirmasi dan melakukan pengecekan tentang transfer dengan toko. 4. Sistem merubah status pemesanan menjadi 'KIRIM' dan toko akan mengirimkan barang.

3.2. Membuat Use Case Diagram dari kasus yang ada

3.2.1. Membuat actor pada Rational Rose 2000 dan Menghapusnya

1. **Klik kanan** pada Use Case View package di browser.
2. Pilih **New** \Rightarrow **Actor** pada menu option . Sebuah actor baru bernama Newclass ditempatkan di browser.
3. Pilih actor New class, lalu masukkan nama yg diinginkan untuk actor tersebut

Misal :



Pelanggan



Toko



Pengunjung

4. Untuk menghapus Actor **klik kanan** pada actor yang akan dihapus pilih **Delete**

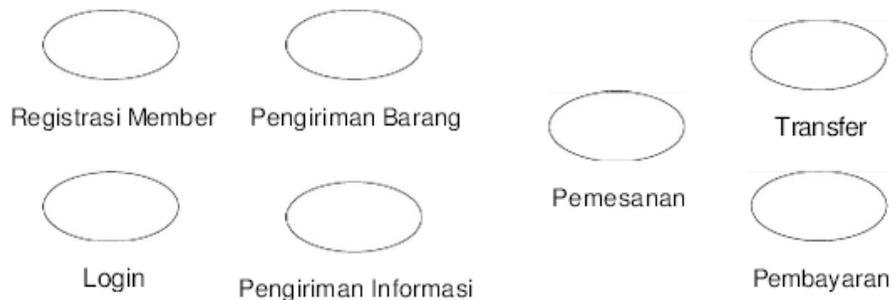
3.2.2. Mendokumentasikan actors

1. Jika documentation window belum terlihat, buka dengan memilih Documentation menu dari View menu.
2. **Klik** untuk memilih Actor di browser.
3. Tempatkan cursor di documentation window, lalu ketikkan dokumentasi yang diinginkan.



3.2.3. Membuat Use Case dan Menghapus Use Case

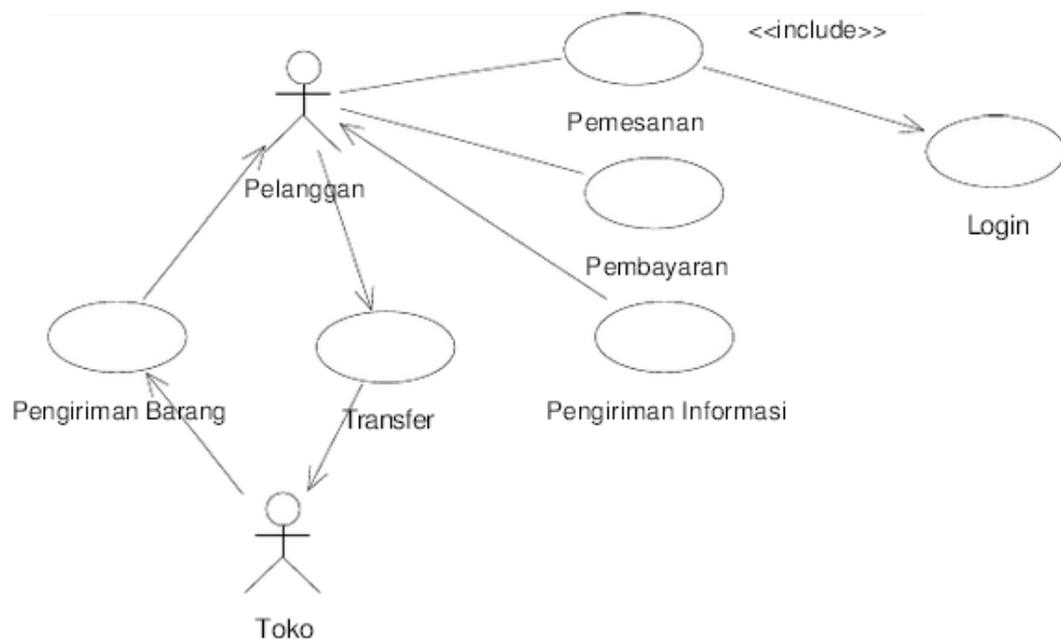
1. **Klik kanan** Use Case View pada browser.
2. Pada menu option pilih **New** \Rightarrow **Use Case**. Sebuah Use Case ditempatkan pada browser.
3. **Klik** Use Case tersebut, lalu masukkan nama yang diinginkan.



4. Untuk menghapus Use Case, **klik kanan** pada Use Case yang akan dihapus lalu pilih **Delete**.

3.2.4. 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.



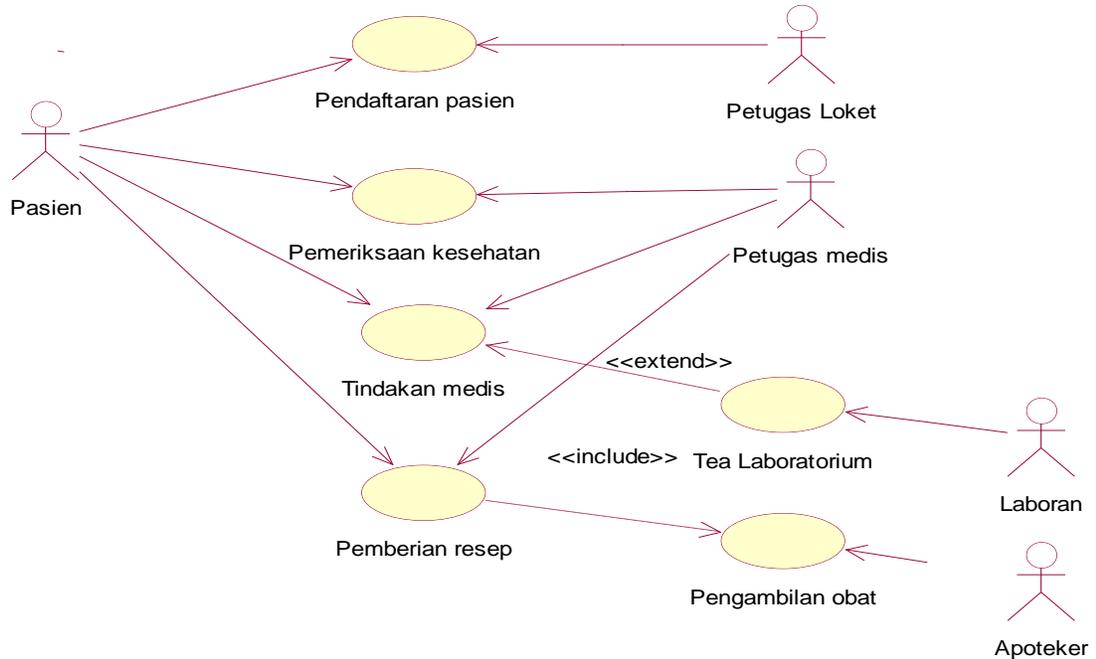
6. Jangan lupa simpan file ini dalam directory anda

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



Latihan

1. Buatlah diagram use case seperti di bawah ini!



2. Buatlah diagram use case untuk (pilih salah satu) :

- Rental VCD/DVD
- Penjualan barang pada supermarket
- Perpustakaan Universitas Bina Darma

- Sistem informasi akademik Bina Darma
- Penerimaan mahasiswa baru Bina Darma
- Apotik
- Administrasi kepegawaian
- Inventori laboratorium komputer
- Sistem parkir Bina Darma

dengan asumsi dan batasan yang anda buat sendiri!

