

BAB VIII

USE CASE

8.1 Analisa Use Case







Diagram Use Case atau Use Case Diagram adalah pemodelan untuk menggambarkan behavior / kelakuan sistem yang akan dibuat. Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

Menurut Rosa dan Salahudin use case digram tidak menjelaskan secara detail tentang penggunaan tiap use case, namun hanya memberi gambaran singkat hubungan antara use case, aktor, dan sistem. Melalui use case diagram kita dapat mengetahui fungsi-fungsi apa saja yang ada pada sistem (Rosa-Salahudin, 2011: 130).

Adapun syarat penamaan pada use case digram sendiri adalah nama didefinisikan sesederhana mungkin sehingga bisa dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

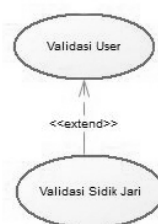
1. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
2. Aktor adalah orang atau system lain yang berinteraksi dengan system yang akan dibuat, jadi meskipun simbol dari aktor adalah gambar orang tapi aktor belum tentu merupakan orang

8.2 Simbol Use Case

Simbol	Deskripsi
Use Case 	Use case adalah fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. biasanya use case diberikan penamaan dengan menggunakan kata kerja di awal frase nama use case
Aktor / actor 	Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi meskipun simbol dari aktor ialah gambar orang, tapi aktor belum tentu merupakan orang. biasanya penamaan aktor dinamakan menggunakan kata benda di awal frase nama aktor
Asosiasi / association 	Asosiasi adalah komunikasi antara aktor dan use case yang berpartisipasi pada use case diagram atau use case yang memiliki interaksi dengan aktor. Asosiasi merupakan simbol yang digunakan untuk menghubungkan link antar element.
Ekstend / extend 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri meski tanpa use case tambahan itu arah panah mengarah pada use case yang ditambahkan
Include 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan membutuhkan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini arah panah include mengarah pada use case yang dipakai (dibutuhkan) atau mengarah pada use case tambahan.
Generalisasi / generalization 	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah use case dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya arah panah mengarah pada use case yang menjadi generalisasinya (umum)

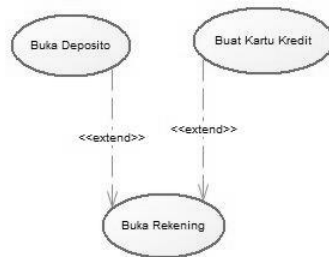
Gambar 8.1 Simbol Use Case Diagram

Penjelasan Simbol Extend



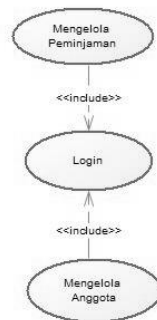
Contoh Simbol Extend

Pada gambar diatas use case Validasi User merupakan use case yang ditambahkan, dimana use case ini dapat berdiri sendiri tanpa use case tambahan (Validasi Sidik Jari). pada contoh diatas setelah pengguna melakukan validasi user, pengguna dapat mengembangkannya (opsional) dengan validasi sidik jari atau tidak



Contoh lainnya adalah seperti pada gambar diatas. use case Buka Rekening merupakan use case yang ditambahkan sehingga use case ini dapat berdiri sendiri sedangkan use case Buka Deposito dan Buat Kartu Kredit merupakan use case tambahan yang berasal dari pengembangan use case extend. pada contoh diatas setelah pengguna melakukan Buka Rekening, pengguna dapat mengembangkannya / melanjutkannya (opsional) dengan Buka Deposito / Buat Kartu Kredit.

Penjelasan Simbol Include

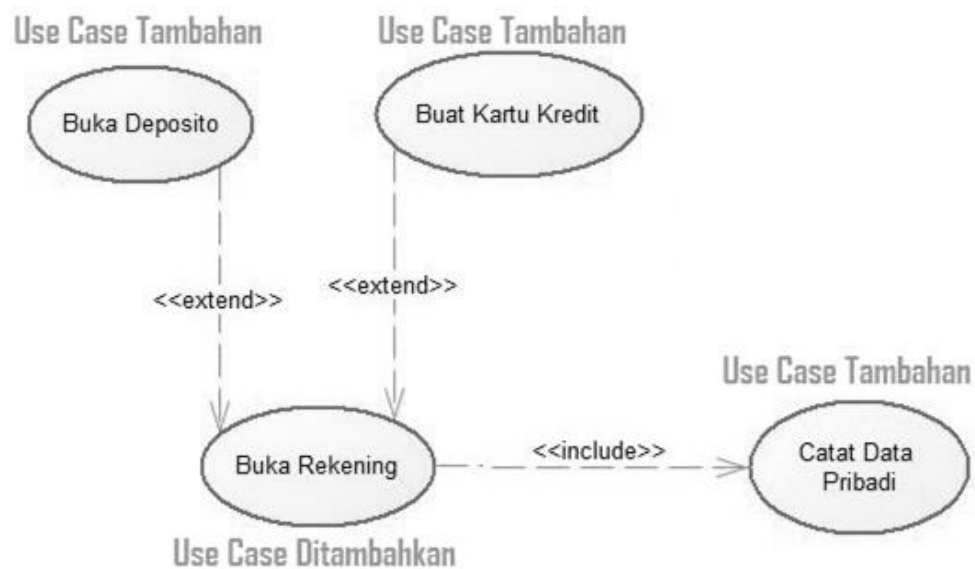


Contoh Simbol Include

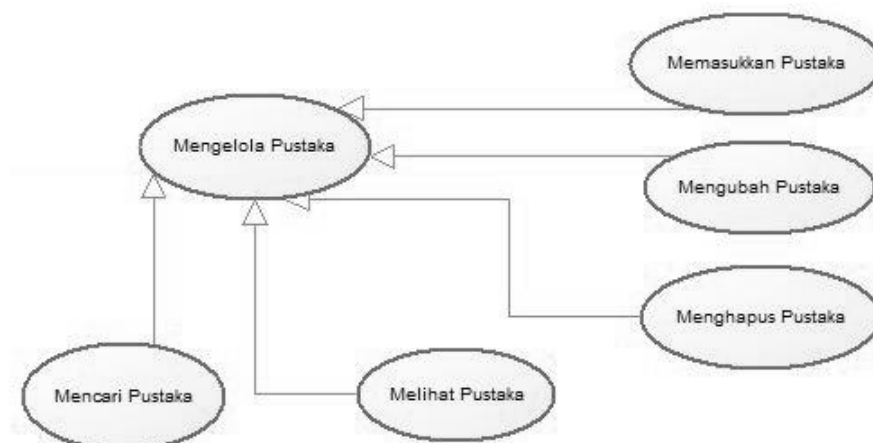
Pada gambar diatas Use Case Login merupakan syarat / selalu dipanggil terlebih dahulu sebelum dijalankannya use case Mengelola Anggota atau use case Mengelola Peminjaman.

Intinya perbedaan mendasar dari use case extend dan use case include adalah : use case extend digunakan untuk mengembangkan sebuah use case (use case inti) misalnya setelah melakukan Buka Rekening selanjutnya bisa melakukan apa lagi ?, dimana pada hubungan extend arah panah mengarah pada use case inti (use case

ditambahkan). sedangkan use case include digunakan untuk menjelaskan bahwa sebuah use case memiliki sebuah syarat agar / ketentuan sebelum bisa dijalankan, misalnya saat kita akan mengelola anggota maka kita diwajibkan login terlebih dahulu. pada hubungan include arah panah mengarah pada use case tambahan (use case yang dipakai / dibutuhkan). Untuk semakin memperjelas, perhatikan contoh dibawah ini:



Penjelasan Simbol Generalisasi



Pada gambar diatas use case Mengelola Pustaka merupakan use case generalisasi / umum. sedangkan use case mencari pustaka, melihat pustaka, memasukkan pustaka, mengubah pustaka dan menghapus pustaka merupakan use case spesialisasi / khusus. hubungan generalisasi ini juga merupakan hubungan yang menggambarkan inheritance baik aktor maupun use case. pada hubungan generalisasi arah panah mengarah pada use case yang menjadi generalisasinya (umum).

Use Case Skenario

Setiap use case diagram dilengkapi dengan skenario, skenario use case / use case skenario adalah alur jalannya proses use case dari sisi aktor dan system. Berikut adalah format tabel skenario use case

Nama Aktor	Reaksi Sistem
Skenario Normal	
Skenario Alternatif	

Skenario use case dibuat per use case terkecil, misalkan untuk generalisasi maka scenario yang dibuat adalah use case yang lebih khusus. Skenario normal adalah scenario bila system berjalan normal tanpa terjadi kesalahan atau error. Sedangkan skenario alternatif adalah scenario bila system tidak berjalan normal atau mengalami error. Skenario normal dan skenario alternatif dapat berjumlah lebih dari satu. Alur skenario inilah yang nantinya menjadi landasan pembuatan sequence diagram / diagram sekuen.

Menentukan Aktor pada Use Case Diagram

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu (Kurt Bittner, Ian Spence. 2002). Cara termudah untuk menemukan aktor adalah dengan bertanya "SIAPA yang akan menggunakan sistem ?"

Namun tidak semua Aktor adalah manusia, aktor juga dapat berupa sistem lain (yang berada diluar sistem yang akan dibuat), ciri system sebagai actor adalah sebagai berikut:

- Jika system yang akan dibuat / dimodelkan bergantung pada sistem lain untuk melakukan sesuatu, maka sistem lain itu adalah aktor.
- Jika sistem yang akan dibuat / dimodelkan meminta (request) informasi dari sistem lain, maka sistem lain itu adalah aktor

Untuk kasus sistem lain yang bertindak sebagai aktor, dapat di ilustrasikan sebagai berikut : misalkan sebuah Sistem Akademik baru dapat menampilkan nilai mahasiswa apabila pembayaran mahasiswa sudah lunas, artinya system akademik memerlukan info dari sistem pembayaran. maka saat kita akan memodelkan use case diagram Sistem Akademik, kita akan memasukkan sistem pembayaran sebagai aktor.

Menentukan Use Case pada Use Case Diagram

Sebuah use case harus mendeskripsikan sebuah pekerjaan dimana pekerjaan tersebut akan memberikan NILAI yang bermanfaat bagi aktor (Kurt Bittner, Ian Spence. 2002).

Untuk menemukan use cases, mulailah dari sudut pandang aktor, misalnya dengan bertanya:

1. Informasi apa sajakah yang akan didapatkan aktor dari sistem ?
2. Apakah ada kejadian dari sistem yang perlu diberitahukan ke aktor ?

Sedangkan dari sudut pandang sistem, misalnya dengan pertanyaan sebagai berikut:

1. Apakah ada informasi yang perlu disimpan atau diambil dari sistem ?
2. Apakah ada informasi yang harus dimasukkan oleh aktor?

Pengertian ini penting untuk diingat, karena dari hal inilah akan menentukan bahwa sebuah use case tidak akan menjadi terlalu kecil. Karena use case yang terlalu kecil tidak akan memberikan nilai bagi aktor

Untuk lebih memahami pembahasan mengenai use case diagram, mari perhatikan studi kasus dibawah ini untuk kemudian kita modelkan menjadi sebuah diagram use case.

Studi Kasus

Studi kasus ini juga akan digunakan pada berbagai macam diagram pemodelan di beberapa artikel lain. Diharapkan dengan adanya studi kasus yang menyatu, sobat dapat lebih memahami proses pemodelan dengan berbagai macam diagram pemodelan pada proses rekayasa perangkat lunak. Berikut studi kasus yang akan kita gunakan:

Nama Aplikasi: Sistem Informasi Manajemen Perpustakaan

Deskripsi:

Sistem informasi manajemen perpustakaan adalah sebuah sistem informasi untuk mengelola informasi yang diperlukan dalam sebuah perpustakaan yang meliputi pengelolaan pustaka, pengelolaan anggota, pengelolaan petugas dan pengelolaan peminjaman pustaka. Aturan perpustakaan yang harus dipenuhi pada sistem informasi manajemen perpustakaan yang akan dimodelkan adalah sebagai berikut:

1. Pustaka dapat memiliki lebih dari satu pengarang
2. Anggota dapat meminjam lebih dari satu buku (pustaka) dalam satu waktu (waktu yang bersamaan)
3. Anggota dapat memiliki lebih dari satu nomor telepon
4. Anggota dapat mengembalikan pustaka yang dipinjam tidak dalam waktu yang bersamaan, meskipun pustaka-pustaka tersebut dipinjam pada waktu yang bersamaan.

5. Pengunjung yang bukan anggota diperbolehkan mencari data pustaka yang ingin dibacanya.
6. Pengunjung yang bukan anggota tidak diperbolehkan meminjam pustaka.
7. Proses pendaftaran pustaka, anggota, dan peminjaman dilakukan oleh petugas perpustakaan.
8. Anggota dan pengunjung dapat melakukan pencarian pustaka.
9. Satu pustaka akan disimpan sebagai satu data dengan id yang unik

Sistem Informasi Manajemen Perpustakaan yang akan dimodelkan memiliki fungsi-fungsi sebagai berikut:

Validasi Petugas

- Login

Mengelola data Pustaka

- Memasukkan data pustaka baru
- Mengubah data pustaka
- Menghapus data pustaka
- Mencari data pustaka
- Melihat data pustaka

Mengelola data petugas

- Memasukkan data petugas baru
- Mengubah data petugas
- Menghapus data petugas
- Mencari data petugas
- Melihat data petugas

Mengelola data Anggota

- Memasukkan data anggota baru
- Mengubah data anggota

- Menghapus data anggota
- Mencari data anggota
- Melihat data anggota

Mengelola data Peminjaman

- Memasukkan data peminjaman
- Mengubah data peminjaman
- Mencari data peminjaman
- Melihat data peminjaman

Penyelesaian Studi Kasus menjadi sebuah Use Case Diagram

Untuk menyelesaikan studi kasus diatas menjadi sebuah use case diagram, umumnya terdapat 4 tahapan yang harus dilalui yaitu :

Pendefinisian Aktor

1. Pendefinisian Use Case
2. Pembuatan Use Case Skenario
3. Menggambarkan Use Case Diagram

Untuk lebih jelasnya, mari langsung saja kita menuju ke tahapan pertama.

1. Pendefinisian Aktor

Berikut adalah hasil pendefinisian aktor pada Sistem Informasi Manajemen Perpustakaan

No	Aktor	Deskripsi
1	Petugas Perpustakaan	Petugas perpustakaan adalah orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pustaka, anggota, dan proses peminjaman pustaka
2	Anggota / Pengunjung Perpustakaan	Anggota adalah orang yang 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.

2. Pendefinisian Use Case

Berikut adalah hasil pendefinisian use case pada Sistem Informasi Manajemen Perpustakaan:

No	Use Case	Deskripsi
1	Login	Merupakan proses untuk melakukan login petugas perpustakaan
2	Mengelola Pustaka	Mengelola Pustaka merupakan proses pengelolaan data pustaka yang meliputi memasukkan pustaka, melihat pustaka, mengubah pustaka, menghapus pustaka dan mencari pustaka.
3	Memasukkan Pustaka	Merupakan proses memasukkan data pustaka ke dalam basis data
4	Melihat Pustaka	Merupakan proses menampilkan data pustaka yang ada di dalam basis data
5	Mengubah Pustaka	Merupakan proses mengubah data pustaka yang ada di dalam basis data
6	Menghapus Pustaka	Merupakan proses menghapus data pustaka yang ada di dalam basis data
7	Mencari Pustaka	Merupakan proses mencari data pustaka yang ada di dalam basis data
8	Mengelola Anggota	Mengelola Anggota merupakan proses pengelolaan data anggota yang meliputi memasukkan anggota, melihat anggota, mengubah anggota, menghapus anggota dan mencari anggota.
9	Memasukkan Anggota	Merupakan proses memasukkan data anggota ke dalam basis data
10	Melihat Anggota	Merupakan proses menampilkan data anggota yang ada di dalam basis data
11	Mengubah Anggota	Merupakan proses mengubah data anggota yang ada di dalam basis data
12	Menghapus Anggota	Merupakan proses menghapus data anggota yang ada di dalam basis data
13	Mencari Anggota	Merupakan proses mencari data anggota yang ada di dalam basis data
14	Mengelola Peminjaman	Mengelola Peminjaman merupakan proses pengelolaan data peminjaman yang meliputi memasukkan peminjaman, melihat peminjaman, mengubah peminjaman, menghapus peminjaman dan mencari peminjaman.
15	Memasukkan Peminjaman	Merupakan proses memasukkan data peminjaman ketika ada anggota yang meminjam pustaka
16	Melihat Peminjaman	Merupakan proses menampilkan / melihat data peminjaman yang ada di dalam basis data
17	Mengubah Peminjaman	Merupakan proses mengubah data peminjaman yang dapat dilakukan untuk mengubah status peminjaman begitu pustaka dikembalikan
18	Menghapus Peminjaman	Merupakan proses menghapus data peminjaman jika ternyata peminjaman tidak jadi dilakukan atau data sudah terlalu banyak dan data sudah di backup terlebih dahulu
19	Mencari Peminjaman	Merupakan proses mencari data peminjaman yang ada di dalam basis data
20	Mengelola Petugas	Mengelola Petugas merupakan proses pengelolaan data petugas yang meliputi memasukkan petugas, melihat petugas, mengubah petugas, menghapus petugas dan mencari petugas.
21	Memasukkan Petugas	Merupakan proses memasukkan data petugas ke dalam basis data
22	Melihat Petugas	Merupakan proses menampilkan data petugas yang ada di dalam basis data
23	Mengubah Petugas	Merupakan proses mengubah data petugas yang ada di dalam basis data
24	Menghapus Petugas	Merupakan proses menghapus data petugas yang ada di dalam basis data
25	Mencari Petugas	Merupakan proses mencari data petugas yang ada di basis data

3. Pembuatan Use Case Skenario

Berikut adalah hasil pendefinisian beberapa use case skenario (tidak kami sertakan semua) dari masing-masing use case yang telah didefinisikan sebelumnya:

Nama Use Case : Login

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan username dan password	2. Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel petugas
	3. Masuk ke aplikasi pengelolaan data perpustakaan
Skenario Alternatif	
1. Memasukkan username dan password	2. Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel petugas
	3. Menampilkan pesan login tidak valid
4. Memasukkan username dan password yang valid	5. Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel petugas
	6. Masuk ke aplikasi pengelolaan data perpustakaan

Nama Use Case : Memasukkan Pustaka

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukkan data pustaka sesuai kolom yang ada	3. Memeriksa valid tidaknya data masukan
	4. Menyimpan data pustaka ke basis data
	5. Menampilkan pesan sukses disimpan
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukkan data pustaka sesuai kolom yang ada	3. Memeriksa valid tidaknya data masukan
	4. Mengeluarkan pesan bahwa data masukan tidak valid
5. Memperbaiki data masukan yang tidak valid	6. Memeriksa valid tidaknya data masukan
	7. Menyimpan data pustaka ke basis data
	8. Menampilkan pesan sukses disimpan

Nama Use Case : Melihat Pustaka

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
	2. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
3. Memilih pustaka yang dicari	
	4. Menampilkan data pustaka (detail sebuah data pustaka) dari pustaka yang dipilih

Nama Use Case : Mengubah Pustaka

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukkan kata kunci dan kategori pencarian	
	3. Mencari data pustaka yang akan diubah
	4. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
5. Memilih data pustaka yang akan diubah	
	6. Menampilkan data pustaka (detail sebuah data pustaka) dari pustaka yang akan diubah
7. Mengubah data pustaka	
	8. Memeriksa valid tidaknya data masukan
	9. Menyimpan data yang telah diubah ke basis data
	10. Menampilkan pesan bahwa data sukses disimpan
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukkan kata kunci dan kategori pencarian	
	3. Mencari data pustaka yang akan diubah
	4. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
5. Memilih data pustaka yang akan diubah	
	6. Menampilkan data pustaka (detail sebuah data pustaka) dari pustaka yang akan diubah
7. Mengubah data pustaka	
	8. Memeriksa valid tidaknya data masukan
	9. Menampilkan pesan bahwa data masukan tidak valid
10. Memperbaiki data masukan yang diubah dan tidak valid	
	11. Memeriksa valid tidaknya data masukan
	12. Menyimpan data yang telah diubah ke basis data
	13. Menampilkan pesan bahwa data sukses disimpan

Nama Use Case : Menghapus Pustaka

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukkan kata kunci dan kategori pencarian	
	3. Mencari data pustaka yang akan dihapus
	4. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
5. Memilih data pustaka yang akan dihapus	
	6. Menampilkan pesan konfirmasi apakah data akan benar-benar dihapus
7. Mengklik pilihan setuju data dihapus	
	8. Menghapus data pustaka dari basis data
	9. Menampilkan pesan bahwa data sukses dihapus
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukkan kata kunci dan kategori pencarian	
	3. Mencari data pustaka yang akan dihapus
	4. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
5. Memilih data pustaka yang akan dihapus	
	6. Menampilkan pesan konfirmasi apakah data akan benar-benar dihapus
7. Mengklik pilihan tidak setuju data dihapus	
	8. Kembali ke form pencarian pustaka

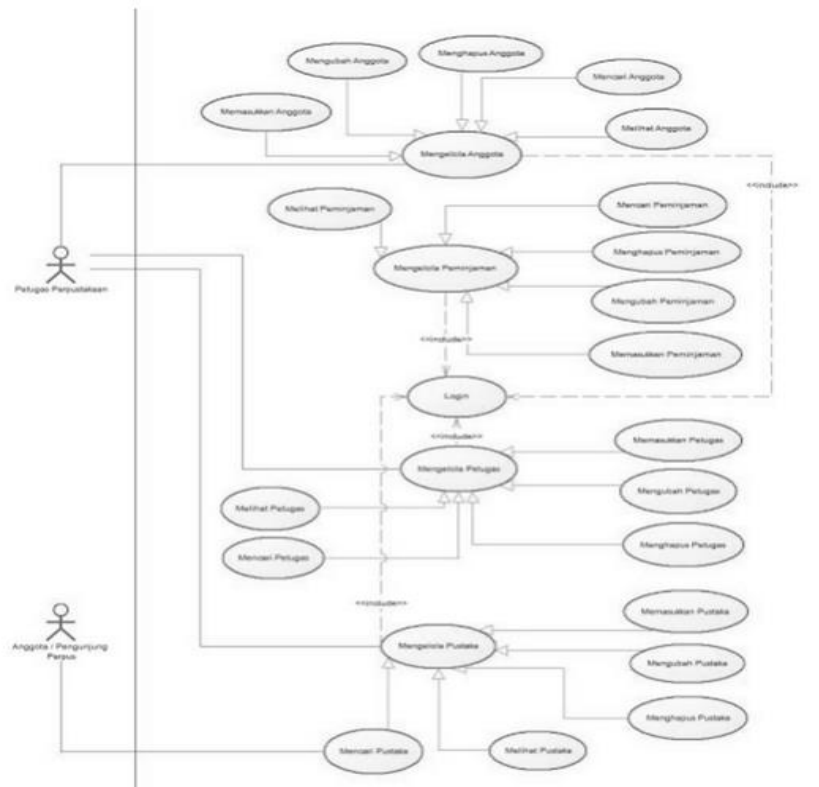
Nama Use Case : Mencari Pustaka

Skenario:

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan kata kunci dan kategori pencarian	
	2. Mencari data pustaka yang dicari
	3. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
4. Memilih data pustaka yang akan dicari	
	5. Menampilkan data pustaka (detail sebuah data pustaka) dari pustaka yang dipilih
Skenario Alternatif	
1. Memasukkan kata kunci dan kategori pencarian	
	2. Mencari data pustaka yang dicari
	3. Menampilkan pesan data pustaka tidak ada
4. Memasukkan kata kunci dan kategori pencarian	
	5. Mencari data pustaka yang dicari
	6. Menampilkan data pustaka yang dicari (belum detail, missal hanya judulnya saja dan tampil dalam bentuk list)
7. Memilih data pustaka yang akan dicari	
	8. Menampilkan data pustaka (detail sebuah data pustaka) dari pustaka yang dipilih

4. Menggambarkan Use Case Diagram

Berikut adalah use case diagram / diagram use case dari Sistem Informasi Manajemen Perpustakaan:



8.3 Peranan Use Case

Semua kemungkinan tanggapan terhadap suatu kejadian didokumentasikan. Use case sangat berguna ketika situasi yang dianalisa sangat kompleks. Sebuah use terdiri dari elemenelemen sebagai berikut:

1. Informasi Dasar

- a. Nama, jumlah dan deskripsi singkat
- b. Trigger-kejadian yang menyebabkab adanya usecase
- c. Eksternal trigger: trigger yang berasal dari luar sistem
- d. Temporal Trigger: kejadian yang berbasis waktu
- e. Sudut pandang use case harus konsisten

2. Input-output utama

- a. Asal dan tujuan
- b. Tujuan harus lengkap dan komprehensif

3. Detail

- a. Harus ada detail dari langkah-langkah yang harus dilakukan berikut data masukan dan keluarannya.

Contoh sebuah use case:

Scenario Name: Patient makes, cancels, or changes an appointment		ID number: <u>1</u>																													
Short description: This describes how we make a new appointment as well as changing or canceling an appointment																															
Trigger: <u>Patient calls and asks for an appointment or asks to cancel an existing appointment</u>																															
Type: <u>External</u> Temporal																															
Major Inputs		Major Outputs																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Description</th> <th style="text-align: left;">Source</th> </tr> </thead> <tbody> <tr> <td><u>Patient name</u></td> <td><u>Patient</u></td> </tr> <tr> <td><u>Desired appointment</u></td> <td><u>Patient</u></td> </tr> <tr> <td><u>Appointment to change/cancel</u></td> <td><u>Patient</u></td> </tr> <tr> <td><u>Patient information</u></td> <td><u>Patient's DB</u></td> </tr> <tr> <td><u>Available appointments</u></td> <td><u>Appointment's DB</u></td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Description	Source	<u>Patient name</u>	<u>Patient</u>	<u>Desired appointment</u>	<u>Patient</u>	<u>Appointment to change/cancel</u>	<u>Patient</u>	<u>Patient information</u>	<u>Patient's DB</u>	<u>Available appointments</u>	<u>Appointment's DB</u>					<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Description</th> <th style="text-align: left;">Destination</th> </tr> </thead> <tbody> <tr> <td><u>Appointment</u></td> <td><u>Appointment</u></td> </tr> <tr> <td><u>Appointment</u></td> <td><u>Patient</u></td> </tr> <tr> <td><u>Possible appointments</u></td> <td><u>Patient</u></td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Description	Destination	<u>Appointment</u>	<u>Appointment</u>	<u>Appointment</u>	<u>Patient</u>	<u>Possible appointments</u>	<u>Patient</u>						
Description	Source																														
<u>Patient name</u>	<u>Patient</u>																														
<u>Desired appointment</u>	<u>Patient</u>																														
<u>Appointment to change/cancel</u>	<u>Patient</u>																														
<u>Patient information</u>	<u>Patient's DB</u>																														
<u>Available appointments</u>	<u>Appointment's DB</u>																														
Description	Destination																														
<u>Appointment</u>	<u>Appointment</u>																														
<u>Appointment</u>	<u>Patient</u>																														
<u>Possible appointments</u>	<u>Patient</u>																														
Major Steps Performed		Information for Steps																													
<p>1. If this is a change or cancellation, then find current appointment in the Appointment File and cancel it.</p> <p>2. Check to make sure the patient is a current patient and has no unpaid bills. If this is a new patient, perform the "Add New Patient" use case before continuing. If the patient has unpaid bills, then transfer the call to the business office.</p> <p>3. Find the available times for an appointment and select ones to propose to the patient (some appointment times will be too short or too long for the patient's problem). This may be repeated several times until a good time is found.</p> <p>4. The patient picks an appointment time to be scheduled.</p>		<table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td><u>Patient name</u></td> </tr> <tr> <td><u>Appointment to change/cancel</u></td> </tr> <tr> <td><u>Revised appointment</u></td> </tr> <tr> <td><u>Patient name</u></td> </tr> <tr> <td><u>Available appointments</u></td> </tr> <tr> <td><u>Possible appointments</u></td> </tr> <tr> <td><u>Desired appointment</u></td> </tr> <tr> <td><u>Appointment</u></td> </tr> </tbody> </table>		<u>Patient name</u>	<u>Appointment to change/cancel</u>	<u>Revised appointment</u>	<u>Patient name</u>	<u>Available appointments</u>	<u>Possible appointments</u>	<u>Desired appointment</u>	<u>Appointment</u>																				
<u>Patient name</u>																															
<u>Appointment to change/cancel</u>																															
<u>Revised appointment</u>																															
<u>Patient name</u>																															
<u>Available appointments</u>																															
<u>Possible appointments</u>																															
<u>Desired appointment</u>																															
<u>Appointment</u>																															

Proses pengembangan Use Case

1. Identifikasi use case utama
2. Identifikasi setiap langkah dalam setiap use case

3. Identifikasi elemen-elemen dalam setiap langkah
4. Konfirmasikan use case
5. Ulangi langkah-langkah diatas secara iteratif

Langkah 1: Identifikasi use case utama

Aktifitas	Pertanyaan tipikal yang diajukan
Mulailah sebuah form use case dari setiap use case Jika lebih dari 9 kelompokkan dalam paket-paket	<p>Tanyakan who, what dan where tentang tugas dan input/output mereka:</p> <ul style="list-style-type: none"> ▪ Tugas utama apa yang dilakukan? ▪ Trigger apa yang memicu tugas ini? ▪ Informasi/form/laporan apa yang diperlukan untuk melakukan tugas ini? ▪ Siapa yang memberi informasi/form/laporan ini? ▪ Informasi apa yang dihasilkan dan pergi ke mana informasi tersebut?

Langkah 2: Identifikasi setiap langkah dalam setiap use case

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap use case, masukkan langkah-langkah utama untuk memproses input dan menghasilkan output	<p>Menanyakan "how" tentang setiap use case:</p> <ul style="list-style-type: none"> ▪ Bagaimana anda memproduksi laporan?

	<ul style="list-style-type: none"> ▪ Bagaimana informasi berubah pada laporan? o you process forms? ▪ Tool apa yang dilakukan pada langkah ini?
--	---

Langkah 3: Identifikasi elemen-elemen dalam setiap langkah

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap langkah, identifikasi setiap langkah, input dan outputnya	<p>Tanyakan How tentang setiap langkah ini:</p> <ul style="list-style-type: none"> ▪ Bagaimana orang mengetahui kapan melakukan langkah ini? ▪ Laporan/form apa yang dihasilkan dari langkah ini? ▪ Laporan/form apa yang dibutuhkan langkah ini? ▪ Apa yang terjadi jika laporan/form ini tidak ada?

Langkah 4: Konfirmasikan use case

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap langkah, validasi bahwa semuanya lengkap dan benar	Minta user untuk mengeksekusi prosesproses yang tertulis dalam use case.

Berikut ini contoh use case yang diterapkan pada proses pemilihan CD pada sebuah rental CD:

Use case name: Take requests for CDs		ID number: <u>1</u>	
Short description: This describes how customers can search the Web site and place requests to hold CD's in stock or place special orders			
Trigger: Customer searches Web and places request to hold a CD or to special order it			
Type: <input checked="" type="radio"/> External <input type="radio"/> Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Search request	Customer	Special order	Special order DBs
CDs selected for request	Customer	Hold for in-stock CD	In-store hold DB
Customer information	Customer		
Marketing materials	Marketing DB		

Use case name: Maintain marketing materials		ID number: <u>2</u>	
Short description: This adds, deletes, and modifies the additional marketing material from vendors (e.g, reviews, musics clips)			
Trigger: Materials from vendors, distributors, wholesalers, record companies, and articles in trade magazines			
Type: <input checked="" type="radio"/> External <input type="radio"/> Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Marketing materials	Vendor	Marketing materials	Marketing DB

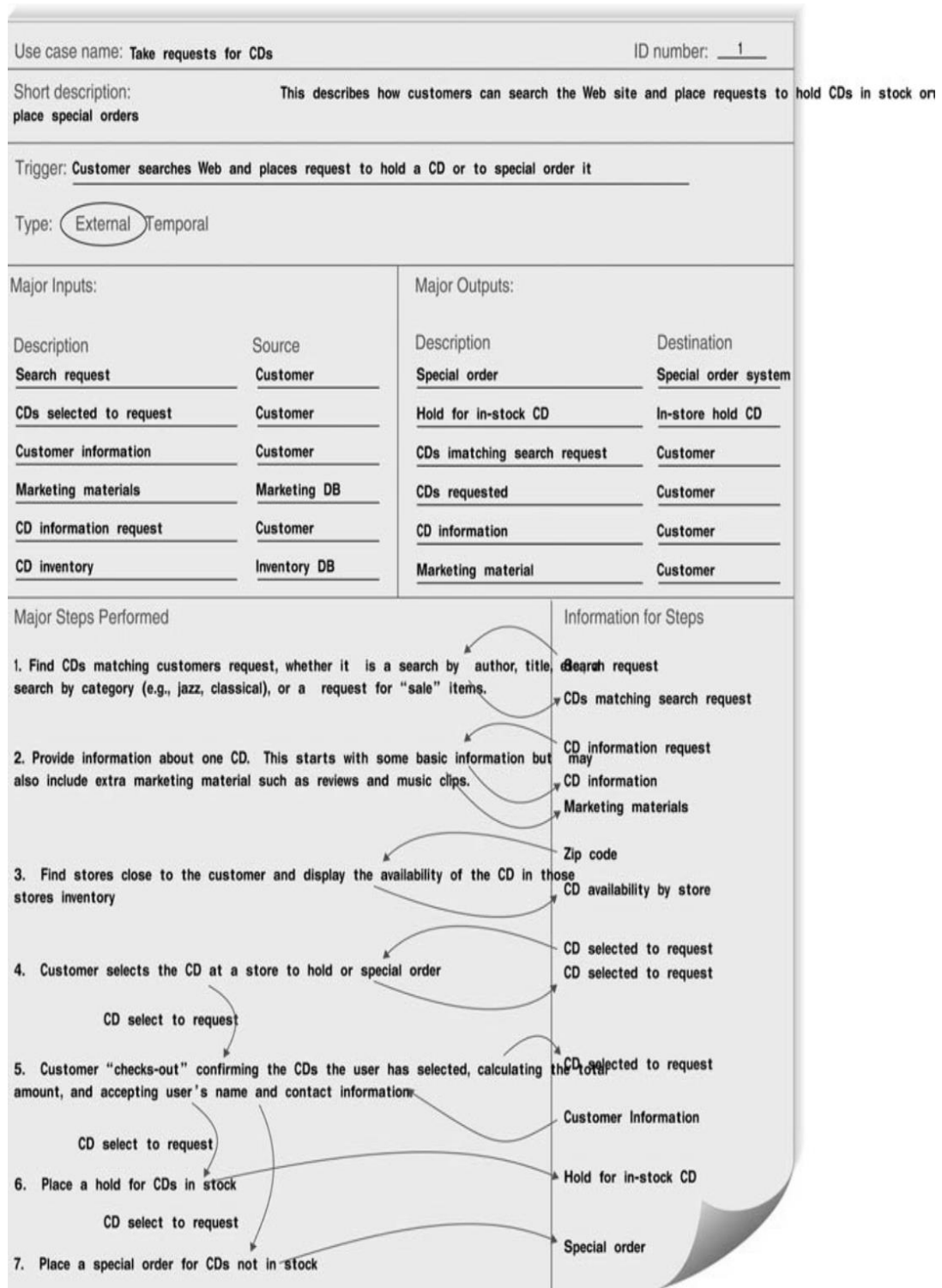
Use case name: Process in-store holds		ID number: <u>3</u>	
Short description: This alerts the store staff to pull a requested CD from the shelves the special order section			
Trigger: Hold request from take request use case			
Type: <input checked="" type="radio"/> External <input type="radio"/> Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Hold request	Take request use case	Hold label	Store staff

Major Steps Performed		Information for Steps	

Hasil Langkah 2:

Use case name: Take requests for CDs		ID number: <u>1</u>	
Short description: place special orders.		This describes how customers can search the Web site and place requests to hold CDs in stock or	
Trigger: <u>Customer searches Web and places requests to hold a CD or to special order it</u>			
Type: <u>External</u> Temporal			
Major Inputs		Major Outputs	
Description	Source	Description	Destination
<u>Search request</u>	<u>Customer</u>	<u>Special order</u>	<u>Special order DB</u>
<u>CDs selected to request</u>	<u>Customer</u>	<u>Hold for CDs in stock</u>	<u>Hold DB</u>
<u>Customer information</u>	<u>Customer</u>		
<u>Marketing materials</u>	<u>Marketing DB</u>		
Major Steps Performed		Information for Steps	
1. Find CDs matching customer's request, whether it is a search by author, title, etc., a search by category (e.g., jazz, classical), or a request for "sale" items.			
2. Provide information about one CD. This starts with some basic information but may also include extra marketing material such as reviews and music clips.			
3. Find stores close to the customer and display the availability of the CD in those stores' inventory.			
4. Customer selects the CD at a store to hold or special order.			
5. Customer "checks out " confirming the CDs the user has selected, calculating the total amount, and accepting user's name and contact information.			
6. Place hold(s) for CD(s) in stock.			
7 Place special order for CDs not in stock.			

Hasil langkah 3:



	use case name: Process in-store holds	ID number: <u>3</u>	material from vendors (e.g., reviews,																												
	Short description: the special order section	This alerts the store staff to pull a request CD from the shelves and place it in																													
Major Steps	Trigger: Hold request from take request use case																														
	Type: <u>(External)</u> Temporal																														
	Major Inputs: <table border="0"> <thead> <tr> <th>Description</th><th>Source</th></tr> </thead> <tbody> <tr> <td><u>Hold request</u></td><td><u>In-store hold DB</u></td></tr> <tr> <td><u>Hold confirmation</u></td><td><u>In-store staff</u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> </tbody> </table>	Description	Source	<u>Hold request</u>	<u>In-store hold DB</u>	<u>Hold confirmation</u>	<u>In-store staff</u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	Major Outputs: <table border="0"> <thead> <tr> <th>Description</th><th>Destination</th></tr> </thead> <tbody> <tr> <td><u>Hold Label</u></td><td><u>In-store staff</u></td></tr> <tr> <td><u>Hold request alert</u></td><td><u>In-store staff</u></td></tr> <tr> <td><u>Hold confirmation</u></td><td><u>In-store hold DB</u></td></tr> <tr> <td><u>Inventory adjustment</u></td><td><u>Inventory DB</u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> <tr> <td><u> </u></td><td><u> </u></td></tr> </tbody> </table>	Description	Destination	<u>Hold Label</u>	<u>In-store staff</u>	<u>Hold request alert</u>	<u>In-store staff</u>	<u>Hold confirmation</u>	<u>In-store hold DB</u>	<u>Inventory adjustment</u>	<u>Inventory DB</u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>	
Description	Source																														
<u>Hold request</u>	<u>In-store hold DB</u>																														
<u>Hold confirmation</u>	<u>In-store staff</u>																														
<u> </u>	<u> </u>																														
<u> </u>	<u> </u>																														
<u> </u>	<u> </u>																														
<u> </u>	<u> </u>																														
Description	Destination																														
<u>Hold Label</u>	<u>In-store staff</u>																														
<u>Hold request alert</u>	<u>In-store staff</u>																														
<u>Hold confirmation</u>	<u>In-store hold DB</u>																														
<u>Inventory adjustment</u>	<u>Inventory DB</u>																														
<u> </u>	<u> </u>																														
<u> </u>	<u> </u>																														
	Major Steps Performed	Information for Steps																													