

PEMODELAN ANALISIS BERORIENTASI OBJEK DENGAN USE CASE

Rini Astuti

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI
Jl. Ir. H. Juanda 96 Bandung 40132

E-mail: riniastuti@likmi.ac.id

Abstrak

Metode berorientasi objek merupakan suatu strategi pembangunan sistem yang mengorganisasikan komponen sistem sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metode berorientasi objek memiliki berbagai aktivitas diantaranya adalah analisis berorientasi objek. Tahap Analisis berorientasi objek melakukan investigasi masalah untuk menemukan (mengidentifikasi) dan mendefinisikan objek-objek atau konsep-konsep yang ada di ruang lingkup masalah, menentukan kebutuhan sistem serta menentukan objek-objek potensial yg ada dalam sistem dan mendeskripsikan karakteristik dan hubungannya dalam sebuah notasi formal.

Kebutuhan sistem menjelaskan kemampuan atau persyaratan fungsional yang harus dipenuhi sistem.

Use case merupakan alat bantu grafis untuk merepresentasikan model kebutuhan sehingga dapat menjelaskan aktivitas-aktivitas apa saja yang harus dikerjakan oleh sistem, dan menjelaskan perilaku dari komponen-komponen sistem.

Kata-kata kunci: *Analisis, Kebutuhan, Metode berorientasi Objek, use case*

1. PENDAHULUAN

Sistem informasi seharusnya dikembangkan dengan memperhatikan berbagai aspek seperti *scalability*, keamanan, dan eksekusi yang *robust* walaupun dalam keadaan sesulit apapun. Selain itu juga kita harus memperhatikan arsitekturnya yang didefinisikan dengan jelas, agar kelemahan-kelemahan atau kekurangan-kekurangan dari sistem yang dirancang mudah ditemukan dan diperbaiki. Oleh karena itu, untuk mendapatkan rancangan suatu sistem yang baik, perlu dilakukannya pemodelan (*modeling*).

Pemodelan (*modeling*) merupakan proses menggambarkan hasil analisis dan perancangan suatu sistem informasi sebelum dilakukan pengkodean (*coding*) dengan menggunakan *tool* tertentu. Pemodelan yang dilakukan untuk membangun system informasi dapat dianalogikan seperti pembuatan *blueprint* dalam membangun sebuah bangunan, sehingga dengan menggunakan model diharapkan pengembangan sistem dapat memenuhi atau sesuai dengan kebutuhan pemakai.

Tahap analisis diawali dengan mengidentifikasi kebutuhan (*requirement*) fungsional dan kebutuhan non fungsional.

Use case merupakan salah satu diagram UML (*Unified Modeling Language*) yang merupakan suatu alat bantu teknis untuk menyelesaikan suatu permasalahan dengan pendekatan berorientasi objek. UML telah menjadi standar dalam dunia industri rekayasa system termasuk perangkat lunak untuk memvisualisasikan, merancang dan mendokumentasikan sistemnya. Secara lebih mendalam, UML lebih dari sekedar sebuah standard dan penemuan dari suatu notasi-notasi yang disatukan, juga berisi konsep-konsep baru yang menarik yang tidak ditemukan secara umum dalam komunitas *object oriented*. [Errikson and Penker, 1998].

Use case dapat digunakan untuk memodelkan kebutuhan fungsional dari sudut pandang bisnis dan produk perangkat lunaknya.

2. PEMBAHASAN

Analisis berorientasi objek adalah proses untuk menentukan objek-objek potensial yang ada dalam sistem dan mendeskripsikan karakteristik serta hubungannya dalam sebuah notasi formal. Kegiatan dalam tahapan analisis diantaranya mempelajari masalah, menentukan kebutuhan dan mendokumentasikan hasil analisis.

Kebutuhan atau *requirement* tidak hanya ditemukan dan ditulis oleh pembangun, tapi sebelumnya justru ditulis oleh pelanggan atau pengguna yang memesan produk sistem informasi. Pelanggan menuliskan kebutuhan sistem dalam bentuk yang masih abstrak tentang berbagai kebutuhan sistemnya. Kemudian kebutuhan tersebut diserahkan kepada tim pembangun. Saat sudah ada persetujuan pembangun pun kemudian menuliskan kemampuan sistem yang bisa dipahami oleh pelanggan atau pengguna.

2.1 Diagram Use Case

Use-case diagram merupakan suatu bentuk diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dikembangkan. Dalam *Use-case diagram* penekanannya adalah “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Sebuah *use-case* akan merepresentasikan sebuah interaksi antara pelaku atau *actor* dengan sistem.

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Use-case diagram yang digunakan dapat sangat membantu pada saat kita menyusun *requirement* sebuah sistem, mengomunikasikannya dengan klien, dan merancang pengujian untuk semua fitur yang terdapat dalam sistem.

Dalam suatu sistem aplikasi berbasis komputer, *use-case diagram* sangat membantu menggambarkan *requierement* apa saja sesuai proses bisnis dan cakupan permasalahannya.

2.2 Simbol pada Diagram Use Case

Komponen diagram *use case* yang utama adalah aktor, *use case*, asosiasi dan *stereotype*. *Stereotype* merupakan

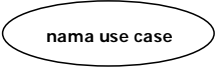


Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

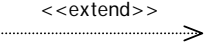
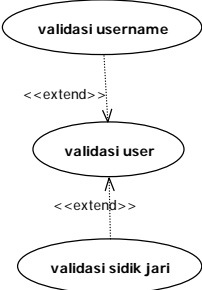
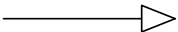
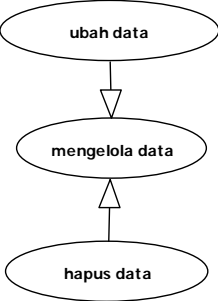
- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri,

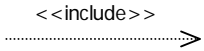
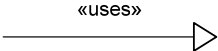
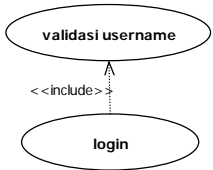
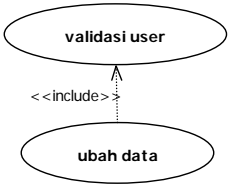
jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i></p>
<p>Aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p> <p>Aktor merupakan peran yang dimainkan oleh pemakai ketika berinteraksi dengan sistem.</p>
<p>Asosiasi / <i>association</i></p> 	<p>komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p>	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan</p>

Simbol	Deskripsi
	<p>dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>

Simbol	Deskripsi
<p data-bbox="391 289 756 321">Menggunakan / <i>include</i> / <i>uses</i></p> <div data-bbox="428 428 630 478">  </div> <div data-bbox="423 590 639 640">  </div>	<p data-bbox="797 289 1315 531">relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p data-bbox="797 600 1315 684">ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul data-bbox="846 705 1315 940" style="list-style-type: none"> • <i>include</i> atau <i>uses</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <div data-bbox="971 982 1182 1150">  </div> <ul data-bbox="846 1199 1315 1539" style="list-style-type: none"> • <i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <div data-bbox="964 1583 1192 1766">  </div> <p data-bbox="846 1814 1315 1898">Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung</p>

Simbol	Deskripsi
	<p>pada pertimbangan dan interpretasi yang dibutuhkan.</p> <p>arah panah include mengarah pada <i>use case</i> yang dipakai</p>

2.3 Membuat Diagram Use Case

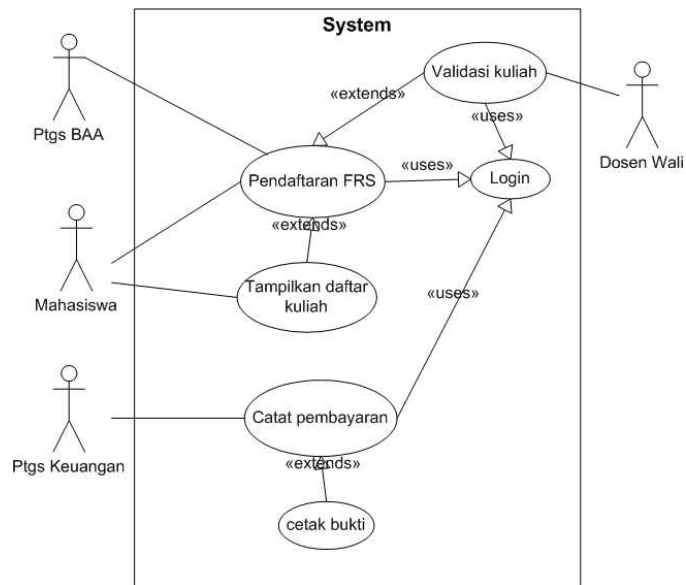
Diagram *use case* dapat dibuat dengan sederhana sesuai dengan kebutuhan sistem. Misalkan akan dibuat diagram use case untuk kasus pendaftaran ulang yang memiliki kebutuhan sistem sebagai berikut :

1. dapat mencatat pendaftaran dari mahasiswa secara langsung setelah melakukan login
2. dapat mencatat pendaftaran oleh petugas Bag Administrasi Administrasi (BAA) yang berhak
3. dosen wali harus melakukan validasi terhadap mata kuliah yang sudah diambil mahasiswa
4. sistem harus dapat mencatat setiap pembayaran pendaftaran ulang dari mahasiswa oleh petugas keuangan dan membuat bukti pembayaran untuk mahasiswa.

Tabel 1. Deskripsi use case

NO	Uraian	Aktor
1	Melakukan login dengan memasukkan nama dan password	Mahasiswa, petgs BAA, Dosen Wali, Ptgs Keuangan
2	Melakukan pendaftaran FRS, extend tampilkan daftar mata kuliah	Mahasiswa
3	Menampilkan daftar mata kuliah	Ptgs BAA, Mahasiswa
4	Melakukan validasi, extend pendaftaran FRS	Dosen Wali
5	Mencatat pembayaran, extend cetak bukti	Ptgs Keuangan

Di bawah ini, terlihat pada Gambar 1. model Diagram Use case dari Sistem Pendaftaran ulang di atas.



Gambar 1. Diagram use case Sistem pendaftaran ulang

3. KESIMPULAN

Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara pemakai sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pemakai dengan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan mungkin diinisialisasi oleh orang, sistem lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat, use case adalah serangkaian skenario yang digabungkan bersama-sama oleh urutan pemakai.

Model diagram use case adalah bagian dari model kebutuhan (*requirement*), yang meliputi spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

Use case adalah salah satu alat bantu guna menstimulasi pemakai potensial untuk mengatakan tentang suatu sistem dan sudut pandangnya. Tidak mudah bagi pemakai untuk mengungkapkan bagaimana kebutuhan atau tuntutan sebuah sistem yang kelak akan digunakan. Diharapkan dengan adanya alternatif pemodelan ini,

pemakai dapat dengan tepat menjelaskan dan mengungkapkan keinginan dan kebutuhan yang berkaitan dengan sistem yang akan dibangunnya.

4. DAFTAR PUSTAKA

- [1]. Erriksson, Hans-Erik and Penker, Magnus, 1998,*UML Toolkit*, Wiley Computer Publishing, Canada.
- [2]. Jeffrey L Whitten, Lonnie D. Bentley(2007), *System Analysis & Design Method*,.7th Ed,McGraw-Hill/Irwin.
- [3]. Joey F. George, Dinesh Batra, Joseph S. Valacich, Jeffrey A. Hoffer (2004), *Object-Oriented Systems Analysis and Design*, Prentice Hall.
- [4]. Martin Fowler, Kendall Scott (2000), *UML Distilled* Addison-Wesley
- [5]. Doug Rosenberg and Matt Stephens(2007), *Use Case Driven Object Modeling with UML: Theory and Practice*, Apress
- [6]. Markus Muetschard, Byron Munday, Matthias Nieder, Wolfgang Sanyer,*Using VisualAge UML Designer*, IBM-International Technical Support Organization