

Pembuatan Sistem Informasi Beasiswa Internal Direktorat Jenderal Perbendaharaan Menggunakan PHP dan MySQL

Candra Dwi Aprida dan Febriliyan Samopa

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: iyan@is.its.ac.id

Abstrak—Saat ini pengelolaan beasiswa yang dilakukan bagian pengembangan pegawai pada Direktorat Jenderal Perbendaharaan menggunakan masih manual. Penggunaan aplikasi bertujuan untuk memberikan kemudahan-kemudahan yang antara lain lebih kemudahan dalam management data terkait data yang ada. Dengan adanya management data yang lebih baik tersebut nantinya juga memberikan manfaat apabila data-data tersebut dibutuhkan untuk diolah untuk berbagai keperluan. Dengan beberapa manfaat yang ditawarkan oleh penggunaan aplikasi yang mendukung bisnis proses pengelolaan beasiswa yang dikelola bagian pengembangan pegawai, maka penggunaan aplikasi yang mendukung proses bisnis ini layak untuk dikembangkan. Dalam penelitian ini, penulis akan membuat Sistem Informasi Beasiswa Internal menggunakan PHP (*Hypertext Preprocessor*) dan MySQL. PHP merupakan salah satu bahasa pemrograman yang digunakan dalam pengembangan aplikasi yang berbasis web dan MySQL adalah database management system yang didistribusikan gratis yang berada di bawah *General Public License (GPL)*. Hasil dari penelitian ini adalah sebuah aplikasi pengelolaan beasiswa internal yang ditujukan untuk membantu pengelolaan beasiswa pada bagian pengembangan pegawai Direktorat Jenderal Perbendaharaan.

Kata Kunci—Beasiswa, PHP, MySQL.

I. PENDAHULUAN

SAAT ini pengelolaan beasiswa yang dilakukan bagian pengembangan pegawai Direktorat Jenderal Perbendaharaan belum didukung oleh aplikasi. Dokumentasi terhadap data-data yang ada dalam bentuk hardcopy sehingga ketika data-data tersebut diperlukan maka berkas data tersebut harus dicari terlebih dahulu dalam tepat penyimpanan berkas. Selain memerlukan waktu dengan semakin banyaknya berkas yang ada tentunya juga memerlukan tempat penyimpanan berkas yang semakin banyak. Hal ini menjadi masalah ketika data-data tersebut diperlukan segera. Akan lebih mudah tentunya ketika data-data tersebut dalam bentuk digital sehingga pencarian menjadi lebih cepat.

Dalam penelitian sebelumnya, Sistem Informasi Beasiswa ITS berbasis Sistem Pendukung Keputusan Menggunakan Analytical Hierarchy Process pada Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya menjelaskan bahwa sistem yang dibangun tersebut nantinya didukung sebuah metode pengambilan keputusan yang bernama Analytical Hierarchy Process dimana dengan sistem itu nantinya diharapkan dapat memberikan keluaran yang optimal terhadap seleksi untuk mendapatkan beasiswa dengan memperhatikan beberapa faktor sebagai tolak ukurnya. Namun demikian dalam penelitian tersebut tidak

membahas terlalu detail terhadap keberadaan sistem yang akan dibangun, titik berat dari sistem yang dibangun tersebut berada pada dukungan pengambilan keputusan [1]. Sementara itu, sistem tersebut dibangun menggunakan bahasa pemrograman JSP dengan Database Management System berupa MySQL [2]. Namun demikian dari kedua penelitian sebelumnya belum mengakomodasi kebutuhan terhadap pelaporan hasil prestasi melainkan hanya menggunakan hasil yang dicapai pada proses belajar sebelumnya sebagai acuan terhadap berhak atau tidaknya mahasiswa untuk mendapatkan beasiswa yang ada. Sistem tersebut sangatlah berbeda dengan proses bisnis yang ada dalam pengelolaan beasiswa yang dilakukan oleh bagian pengembangan pegawai dimana penerimaan dilakukan bukan hanya dengan nilai/hasil ujian/tes tapi juga dengan psikotes terhadap pelamar beasiswa. Dengan perbedaan yang mendasar terhadap pengelolaan beasiswa yang dibahas pada kedua penelitian sebelumnya maka belum dapat menjawab kebutuhan

Sistem Informasi Beasiswa yang dibutuhkan oleh bagian pengembangan pegawai pada Direktorat Jenderal Perbendaharaan dimana sistem yang dibangun nantinya diharapkan sesuai dengan SOP yang ada dalam pengelolaan beasiswa yang ada di lingkungan Direktorat Jenderal Perbendaharaan. Untuk mengatasi permasalahan tersebut, perlu dibuat sebuah aplikasi sistem pengelolaan beasiswa yang sesuai dengan SOP pengelolaan beasiswa yang berada pada Direktorat Jenderal Perbendaharaan.

II. METODE PENELITIAN

A. Studi Literatur

Pada studi literatur ini, penulis melakukan pembelajaran dan pemahaman pada literatur-literatur yang berkaitan dengan permasalahan. Literatur ini dapat berupa peraturan perundang-undangan, buku referensi dan halaman web.

B. Analisis Kebutuhan

Pada tahap ini akan dilakukan identifikasi kebutuhan Sistem Beasiswa Internal Direktorat Jenderal Perbendaharaan melalui proses bisnis yang ada.

C. Desain

Pada tahap ini akan dilakukan desain aplikasi yang akan dibuat yang terdiri dari desain tampilan antar muka pengguna. Dari tampilan desain antar muka inilah nantinya dapat disusun sebuah usecase skenario yang kemudian pada tahapan selanjutnya dirinci lagi menggunakan robustness diagram dan juga sequence diagram. Hasil dari robustness

dan *sequence* diagram nantinya digunakan dalam menyempurnakan domain model dan juga class diagram.

D. Pengkodean

Setelah proses desain selesai, akan dilanjutkan dengan pembuatan kode aplikasi menggunakan PHP dengan MySQL sebagai manajemen basis data. Dalam proses pengkodean akan dilakukan dengan menggunakan konsep pemrograman berorientasi objek. Selain itu untuk memudahkan nantinya dalam pengelolaan di masa yang akan datang maka pola dari pengkodean menggunakan pola MVC yang memisahkan antara logic dari aplikasi dengan tampilan/view nya.

E. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah dibangun untuk memastikan kesesuaian aplikasi dengan desain yang telah dibuat sehingga fungsionalitas sistem yang dibutuhkan dapat terpenuhi.

III. PERANCANGAN APLIKASI

Aplikasi dirancang menggunakan *Iconix Process* yang dimulai dengan penentuan kebutuhan fungsional dari sistem yang dilanjutkan dengan pembuatan *domain model*, *Graphical User Interface (GUI) Storyboard*, *use case*, *robustness analysis*, *sequence diagram* dan *class diagram* [3].

A. Kebutuhan Fungsional

Kebutuhan fungsional yang telah diidentifikasi dari aplikasi adalah sebagai berikut:

1. Mencatat/menginput nilai perkembangan tugas belajar
2. Mengupload berkas softcopy
3. Mencetak laporan

B. Kebutuhan Fungsional

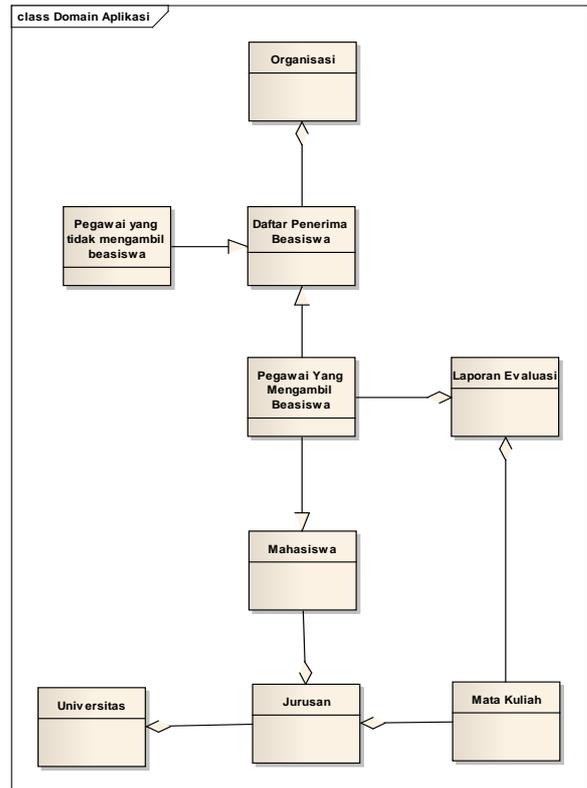
Sedangkan kebutuhan Non Fungsional dari aplikasi ini adalah kemandirian aplikasi. Namun implementasi terhadap keamanan hanya terbatas pada enkripsi password menggunakan SHA.

C. Domain Model

Domain model tersebut merupakan bentuk sederhana dari *class diagram* dengan garis relasi dari beberapa *class* yang berbeda (*domain object*) [3]. Domain model pada sistem aplikasi ini dapat dilihat pada Gambar 1.

D. Graphical User Interface (GUI)

Kebutuhan fungsional sistem terhadap pengelolaan beasiswa ini nantinya tentunya melibatkan tampilan grafis sebagai sarana komunikasi antara user dengan system (GUI) yang antara lain berupa form dan tabel. Berikut adalah GUI design dari form yang digunakan untuk merekam data nilai dan table data nilai yang telah direkam oleh pengguna:

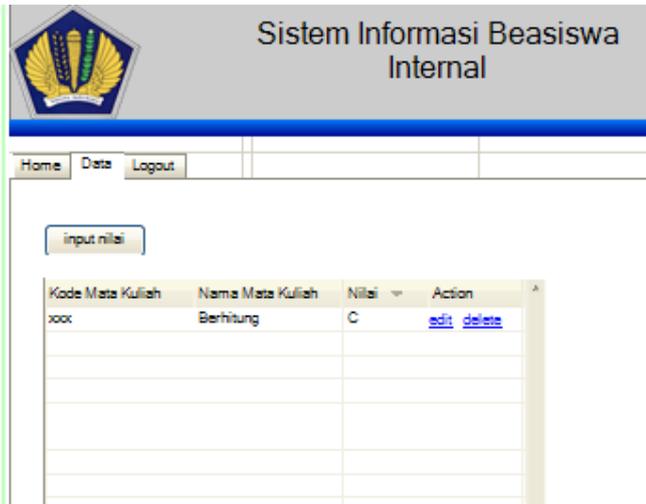


Gambar. 1. Domain model.

Gambar. 2. Form rekam data nilai.

Dalam form ini nantinya pengguna dapat merekam data nilai yang diperoleh pada setiap akhir periode belajar. Setelah memilih mata kuliah yang diikuti pada semester yang bersangkutan, pengguna memilih nilai sesuai dengan hasil yang diperoleh tersebut dan kemudian dapat menekan tombol simpan untuk menyimpan data nilai pada form rekam data nilai tersebut.

Pada tabel data nilai ini nantinya pengguna dapat melihat daftar data nilai yang telah direkam oleh pengguna sebelumnya. Pada tabel ini nantinya disediakan link untuk mengedit data nilai yang telah dimasukkan atau menghapus data nilai tersebut.



Gambar. 3 Tabel data nilai.

E. Use Case

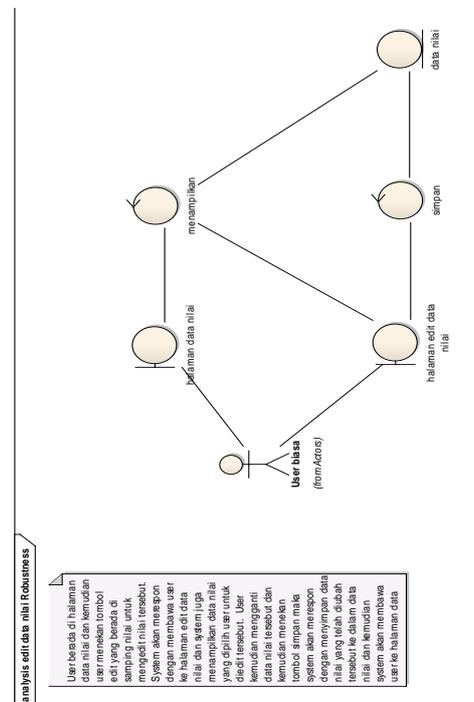
Dari GUI design yang telah dibuat sebelumnya maka sebuah Use Casediagram dapat disusun. Dalam Use Casediagram ini nantinya berisi aksi yang pengguna lakukan pada tampilan yang ada dan juga reaksi dari system terhadap aksi pengguna. Berikut adalah daftar use case yang ada pada aplikasi ini:

1. Login
2. Rekam user
3. Edit user
4. Hapus user
5. Hapus mata kuliah
6. Rekam mata kuliah
7. Edit mata kuliah
8. Rekam jurusan
9. Edit jurusan
10. Hapus jurusan
11. Rekam universitas
12. Edit universitas
13. Hapus universitas
14. Verifikasi
15. Add detail topsis
16. Do topsis
17. Cetak laporan
18. Edit password
19. Rekam data nilai
20. Ubah data nilai
21. Delete data nilai
22. Membaca file excel
23. Add personal info
24. Edit personal info
25. Delete personal info
26. Logout

Gambar 4 adalah salah satu contoh dari use case diagram pada aplikasi ini:

Use case name	: Edit data nilai
Actor	: User
Basic course	: User berada di halaman data nilai dan kemudian user menekan tombol edit yang berada di samping nilai untuk mengedit nilai tersebut. System akan merespon dengan membawa user ke halaman edit dan system juga menampilkan data nilai yang dipilih user untuk diedit tersebut. User kemudian mengganti data nilai tersebut dan kemudian menekan tombol simpan maka system akan merespon dengan menyimpan data nilai yang telah diubah tersebut ke dalam data nilai dan kemudian system akan membawa user ke halaman data
Alternative course	: Apabila user langsung menekan tanda simpan maka system akan menyimpan data sesuai dengan data yang tertampil di dalam form tersebut walaupun user belum mengubahnya

Gambar. 4. Deskripsi use case edit data nilai.



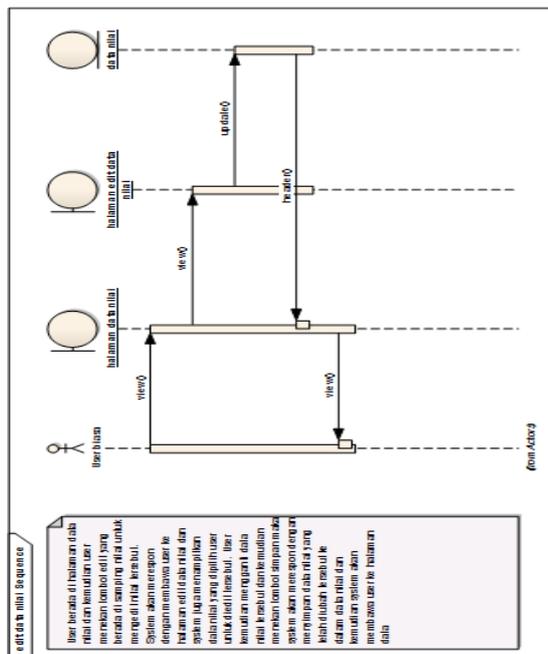
Gambar. 5. Robustness diagram edit data nilai.

F. Robustness Analysis

Proses berikutnya adalah melakukan robustness analisis terhadap object yang ada pada use case tersebut. Object-object tersebut adalah *boundary*, *entity*, dan *controller*. Dari robustness analisis ini nantinya akan menghasilkan robust diagram. Gambar 5 adalah salah satu robustness diagram

G. Sequence Diagram

Setelah melakukan robustness diagram, untuk dapat lebih melihat detail operasi dari use case tersebut maka diperlukan *sequence diagram*. Dalam *sequence diagram* ini nantinya gambaran dari robustness analisis akan terlihat lebih mendetail dengan operasi yang diperlihatkan dengan tanda anak panah. Operasi ini biasanya akan menjadi *method*.



Gambar. 6. Sequence diagram edit data nilai.

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	User mengisi data nilai mata kuliah	Menekan Tombol simpan	Data nilai mata kuliah tersebut tersimpan ke dalam database	sukses

Gambar. 7. Test case edit data nilai.

Tabel 1.

Spesifikasi Hardware

Perangkat	Spesifikasi
Personal Computer	Prosesor : Intel Pentium Q6600 2.4 Mhz
	Memori : 4 GB RAM
	Sistem Operasi : Windows 7

Tabel 2.

Spesifikasi Software

Tools	
Webserver	Xampp
Bahasa Pemrograman	PHP
Database	MySql
Database Management	PhpmyAdmin 3.3.9
Development Kit	JDK 1.6.0
Editor	Netbeans IDE 7.1.2
Browser	Mozilla Firefox 21.0
Library tampilan pendukung	
Tampilan utama	Bootstrap tweeter 2.2
Tampilan table	DT Bootstrap
Validasi input	Jquery validate
Librarv untuk membaca excel	PHP excel

H. Class Diagram

Dari apa yang dilakukan terhadap sequence diagram nantinya mungkin terdapat perubahan-perubahan yang ada pada domain model sehingga perlu adanya update domain model. Semakin mendetail gambaran proses yang ada pada robustness analisis dan sequence diagram maka akan semakin memudahkan nantinya membuat update dari domain model yang nantinya menjadi class diagram dimana dalam class diagram ini telah ada atribut-atribut dari object yang ada dan juga operasi yang bisa dilakukan pada object tersebut.

I. Database

Dari kelas-kelas yang telah terbentuk beserta attributnya tadi maka nantinya ketika dibutuhkan sebuah media penyimpanan terhadap data yang ada proses ini menjadi lebih mudah. Akan tetapi tidak semua kelas yang ada tersebut akan menjadi sebuah tabel dalam database.

J. Code

Proses code dilakukan untuk mengimplementasikan design yang telah disusun sebelumnya. Untuk mengecek apakah code yang dibuat telah sesuai dengan design yang telah disusun sebelumnya maka dilakukan test case terhadap code tersebut.

K. Test Case

Test case merupakan suatu rangkaian masukan, kondisi eksekusi dan hasil yang diharapkan yang dikembangkan untuk tujuan tertentu, seperti menguji jalannya program atau memverifikasi kepatuhan dengan kebutuhan yang spesifik [5]. Gambar 7 adalah contoh test case dari aplikasi ini.

IV. IMPLEMENTASI

A. Penggunaan Teknologi

Pada tahapan implementasi ini dibutuhkan beberapa komponen pendukung yang terlibat diantaranya berupa hardware dan software. Hardware berperan dalam menciptakan lingkungan operasi dari system yang dibangun dan software berperan dalam menyusun aplikasi tersebut. Tabel 1 dan 2 adalah daftar hardware dan software yang dipakan dalam pengembangan aplikasi ini.

B. Pembuatan struktur Model View Controller (MVC)

Aplikasi ini menggunakan framework MVC dalam proses pemngembangannya. Menggunakan pola ini mempunyai tujuan agar logic terpisah dari view. Dengan pola ini maka akan memberikan kemudahan dalam pengelolaan file yang ada (blendwork 2013) [4]. Tahapan pembuatan sturktur MVC pada aplikasi ini adalah sebagai berikut :

1. Konfigurasi file

Tahapan konfigurasi berikutnya dilakukan adalah melakukan setting terhadap file utama yang nantinya akan dibaca ketika pengguna mengakses website adalah file index.php. File ini nantinya akan dibaca pertama kali secara default ketika link dari website kita diakses. Gambar 8-11 adalah setting yang digunakan.

2. Pembuatan Kelas Utama

Kelas-kelas utama ini digunakan sebagai kerangka pendahuluan sebelum penulisan terhadap kelas-kelas yang nanti kelas-kelas utama tersebut ditempatkan pada sebuah direktory khusus yakni pada direktory "libs". Kelas-kelas utama ini nantinya sebagian diturunkan dan dipakai oleh kelas-kelas turunannya. Setelah kerangka MVC telah siap, penulisan terhadap code dapat dilakukan.

C. Penulisan Kode Program

Setelah framework MVC tersebut telah siap maka penulisan code program dapat dilakukan. Kode program yang ditulis nantinya akan menggunakan kelas-kelas utama yang telah dibangun sebagai framework tersebut dan kemudian untuk menulis code program maka dibuat kelas-kelas baru yang merupakan turunan dari kelas utama.

```

1 <?php
2
3 require 'configs/path.php';
4 require 'configs/database.php';
5
6 require 'libs/Bootstrap.php';
7 require 'libs/Controller.php';
8 require 'libs/View.php';
9 require 'libs/Model.php';
10 require 'libs/Database.php';
11 require 'libs/Session.php';
12 require 'libs/Hash.php';
13
14
15
16
17 //require 'public/Excel/Reader.php';
18
19 $app = new Bootstrap();
20 ?>
    
```

Gambar. 8. Konfigurasi file.

```

] class User_Model extends Model {
]     public function __construct() {
]         parent::__construct();
]     }
]
]     public function userList() {
]
]         $sth = $this->db->prepare("select user_id,nip,username,role f:
]
]         $sth->execute();
]
]         $data = $sth->fetchAll();
]
]         return $data;
]
]         //     print_r($data);
]     }
    
```

Gambar. 9. Model.

```

<form class="form-horizontal" id="register" method="POST" action="addSave">
<fieldset>
<legend>Rekam Data User Baru</legend>
<div class="control-group">
<label class="control-label"> Nomor Induk Pegawai (NIP)</label>
<div class="controls">
<input type="text" name="nip" id="nip" placeholder="nip pegawai" />
</div>
</div>
<div class="control-group">
<label class="control-label"> Nomor Registrasi Pokok (NRP)</label>
<div class="controls">
<input type="text" name="nrp" id="nrp" placeholder="nrp pegawai" />
</div>
</div>
<div class="control-group">
<label class="control-label"> Nama Universitas</label>
<div class="controls">
<select name="nama_univ" class="selectpicker" data-style="">
<?php
$options = $this->univList;
foreach ($options as $key => $value) {
    echo '<option value="' . $value[1] . '>';
    echo $value[2];
}
    
```

Gambar. 10. View.

Kelas utama yang sering diturunkan untuk dipakai dalam aplikasi ini adalah kelas model yang digunakan untuk menggambarkan model, kelas view yang digunakan untuk menampilkan hasil query terhadap model tersebut dan kelas controller, kelas yang digunakan untuk menghubungkan kelas model dan kelas viewnya.

```

- * @author bancan
- */
] class User extends Controller {
]     public function __construct() {
]         parent::__construct();
]
]         Session::init();
]
]         $logged = Session::get('loggedIn');
]         $role = Session::get('role');
]
]         if ($logged == false || $role != 'Admin') {
]             Session::Destroy();
]             header('location: ' . URL . 'error');
]             exit;
]         }
]         print_r($_SESSION['role']);
]     }
- }
]     public function index() {
]         //     $item = 10;
]         //
]         //     $start = ($x - 1) * 10;
    
```

Gambar. 11. Controller.

V. KESIMPULAN

Berdasarkan hasil penelitian tugas akhir ini, maka dapat disimpulkan aplikasi Sistem Informasi Beasiswa Internal Direktorat Jenderal Perbendaharaan dapat dikembangkan dengan menggunakan pola Model View Controller.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Direktorat Jenderal Perbendaharaan, Kementerian Keuangan, yang telah memberikan kesempatan kepada penulis untuk mengikuti program beasiswa pada Jurusan Sistem Informasi, ITS Surabaya.

DAFTAR PUSTAKA

- [1] Kirom, Dalu Nuzlul. *Sistem Informasi Manajemen ITS Berbasis Sistem Pendukung Keputusan Menggunakan Analytical Hierarchy Process*. Paper, Surabaya: ITS, 2012.
- [2] Hidayat, Remil. "Aplikasi Pengelolaan Beasiswa Divisi Pendidikan Di Rumah Zakat Indonesia DSUQ Berbasis Web." Paper, Bandung, 2006.
- [3] Rosenberg, Dough, dan Matt Stephens. *Use Case Driven Modelling with UML: Theory and Practice*. Newyork: Apress, 2007.
- [4] blendwork. <http://www.blendwork.biz>. 7 1, 2013. <http://www.blendwork.biz/kelebihan-menggunakan-web-application-framework/> (accessed 7 1, 2013).
- [5] Heumann, Jim. "The Rational Edge: Gerating Test Cases From Use Cases." [www.ibm.com](http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf). 2001. <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf> (diakses Juni 2013).