

PENGEMBANGAN LEARNING MANAGEMENT SYSTEM¹

Hendrik, Fathul Wahid

Laboratoium Sistem Informasi dan Rekayasa Perangkat Lunak (Sirkel),
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
Jl. Kaliurang Km. 14 Yogyakarta 55501
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148
E-mail: javanehese2002@yahoo.com; fathulwahid@fti.uui.ac.id

ABSTRAK

Perkembangan teknologi informasi (TI) telah menawarkan paradigma baru dalam pembelajaran di perguruan tinggi (PT) yang disebut dengan *e-learning*. Meskipun saat ini telah banyak dikembangkan aplikasi untuk mendukung *e-learning* atau dikenal sebagai *Learning Management System* namun umumnya aplikasi-aplikasi LMS tersebut masih mahal. Oleh karenanya dirasa perlu mengembangkan aplikasi LMS yang *fully-customized* dan sesuai dengan konteks institusional sehingga diharapkan nantinya aplikasi ini dapat membantu dalam meningkatkan kualitas proses belajar mengajar. Aplikasi ini nantinya tidak ditujukan untuk menggantikan aktivitas pembelajaran konvensional melainkan hanya bersifat sebagai komplementer. Adapun penggunaan metode berorientasi objek dalam pengembangan aplikasi ini dimaksudkan untuk mempermudah proses pengembangan sistem secara keseluruhan karena fasilitas-fasilitas yang ada pada aplikasi ini dapat dibentuk sebagai modul-modul yang nantinya dapat ditambahkan ke dalam sistem sesuai dengan kebutuhan. LMS ini telah diuji secara teknis dan memberikan hasil yang baik. Pengujian non-teknis akan dilakukan untuk mendapatkan masukan perbaikan LMS pada masa yang akan datang.

Kata-kunci: e-learning, metode analisis desain berorientasi objek, learning management system

1. PENDAHULUAN

Perkembangan teknologi informasi (TI) telah menawarkan paradigma baru dalam pembelajaran di perguruan tinggi (PT) yang disebut dengan *e-learning*. Konsep *e-learning* juga diadopsi oleh banyak perusahaan komersial untuk memberikan pelatihan kepada karyawannya. Di PT, TI telah memungkinkan interaksi antara dosen dan mahasiswa yang melintasi sekat ruang dan waktu. Interaksi ini dapat terjadi secara sinkron (pada waktu yang sama) maupun asinkron (dalam waktu yang berbeda). Secara umum, *e-learning* dapat

¹ Tulisan ini didasarkan pada Tugas Akhir sarjana yang berjudul "Implementasi Sistem *E-learning* di Lingkungan Universitas Islam Indonesia dengan Metode Berorientasi Objek" yang dilakukan oleh Hendrik di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

didefinisikan sebagai pembelajaran yang disampaikan melalui semua media elektronik termasuk, Internet, intranet, extranet, satelit, audio/video tape, TV interaktif, dan CD ROM (Govindasamy, 2002).

Saat ini, telah banyak dikembangkan aplikasi untuk mendukung *e-learning* seperti WebCT (www.webct.com) atau ClassFronter (fronter.info/com) yang sering disebut dengan *Learning Management Systems* (LMS) (Govindasamy, 2002). LMS ini sudah banyak diadopsi oleh banyak PT di dunia. Sebagai contoh, WebCT telah digunakan lebih dari 2200 PT di seluruh dunia (Pituch dan Lee, 2004). LMS ini mengintegrasikan banyak fungsi yang mendukung proses pembelajaran seperti memfasilitasi berbagai macam bentuk materi instruksional (teks, audio, video), e-mail, chat, diskusi online, forum, kuis, dan penugasan.

LMS telah digunakan dalam tingkatan yang berbeda. Pituch & Lee (2004) membedakan adopsi ini menjadi dua tingkatan: (a) untuk mendukung proses pembelajaran konvensional, dan (b) untuk mendukung pembelajaran jarak jauh yang menggantikan pembelajaran konvensional. Yang pertama, *e-learning* berfungsi sebagai *suplemen*, sedang yang kedua berfungsi sebagai *substitusi*.

Tulisan ini merupakan laporan *research in progress* yang rencananya akan dilakukan selama tiga tahun dan dibagi dalam beberapa tahap:

- a. Pengembangan LMS
- b. Evaluasi tahap awal proyek percobaan (*pilot project*) penggunaan LMS untuk beberapa matakuliah
- c. Perbaikan LMS dan konsep *e-learning* secara keseluruhan berdasar umpan-balik yang didapat

Tulisan ini akan membahas pengembangan sebuah LMS yang dirancang sebagai suplemen pembelajaran konvensional. Karenanya, rancangan awalnya masih sederhana dan belum mendukung semua fungsi untuk pembelajaran jarak jauh yang menggantikan pembelajaran konvensional. Alasan pengembangan LMS yang *fully-customized* ini adalah masih mahal nya LMS komersial yang ada di pasaran. Selain itu, pengembangan LMS yang mempertimbangkan konteks institusional ini diharapkan sesuai dengan kebutuhan yang ada.

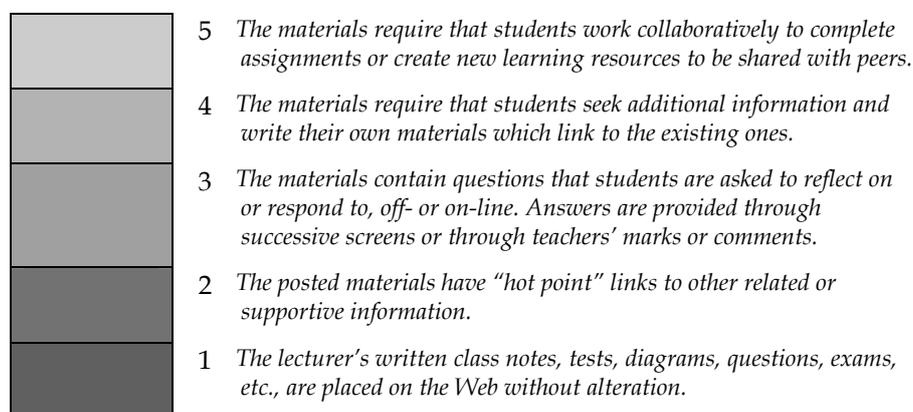
2. LANDASAN TEORI

2.1 LMS dan e-Learning

Banyak kritik dialamatkan kepada para vendor yang tidak mempertimbangkan aspek pedagogis dalam membangun LMS untuk mendukung *e-learning*. Menurut Institute for Higher Education Policy, Amerika (dalam Govindasamy, 2002) terdapat tujuh parameter yang perlu diperhatikan dalam menerapkan *e-learning* yang mempertimbangkan prinsip-prinsip pedagogis, yaitu: (1) institutional support; (2) course development; (3) teaching and learning; (4) course structure; (5) student support; (6) faculty support; dan (7) evaluation and assessment.

Terkait dengan proses, pembelajaran tidak akan terjadi tanpa peran aktif dari mahasiswa dalam menyerap materi yang disampaikan. Karenanya, teknologi yang digunakan bukan merupakan kondisi terpenting dalam proses pembelajaran.

Tingkat *interactiveness* ini dapat digambarkan dalam kontinum seperti ditunjukkan pada Gambar 1.



Gambar 1. Kontinum interaktivitas
Sumber: Hunt (1999)

Dalam penelitian ini, baru interaktivitas tingkat ketiga yang bisa diakomodasi. Meskipun demikian, interaktivitas tingkat keempat dan kelima sebagian telah diakomodasi dengan memungkinkan mahasiswa mengupload materi-materi pembelajaran dan berperan aktif dalam forum diskusi.

2.2 Metode Berorientasi Obyek

Metode berorientasi obyek (*object oriented method*) merupakan suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi obyek-obyek yang ada di dunia nyata. Brooks (1987) - dalam Nugroho (2002) -- menyatakan bahwa bagian tersulit dari pengembangan perangkat lunak dan/atau sistem informasi adalah tahap analisis, yang di dalamnya abstraksi dilakukan. Dalam metode berorientasi obyek kejadian atau hubungan antar entitas dalam dunia nyata direpresentasikan dalam obyek-obyek.

Metode ini dipercaya memungkinkan para pengembang sistem informasi untuk menyederhanakan dan menyingkat waktu yang diperlukan untuk pengembangan sistem informasi, terutama dalam pemakaian kembali komponen-komponen perangkat lunak. Ada beberapa tema yang mendasari metode berorientasi obyek, antara lain abstraksi, pembungkusan (*encapsulation*), *sharing*, dan sinergi.

Abstraksi dilakukan untuk menemukan hal-hal yang esensial atau yang paling penting pada suatu obyek dan mengabaikan hal-hal yang bersifat insidental. Penggunaan konsep abstraksi selama analisis berarti perancangan dan implementasi tidak dapat dilakukan sebelum persoalan ditemukan dan benar-benar dipahami.

Pembungkusan (*encapsulation*) berarti meninggalkan aspek eksternal dari obyek yang dapat diakses oleh obyek lain dan memfokuskan diri pada

implementasi internal dari suatu obyek tersembunyi dari obyek-obyek lain dan terpisah dari implementasi eksternal, yaitu *antarmuka* satu obyek dengan obyek yang lainnya.

Teknik pengembangan sistem informasi berorientasi obyek mengizinkan berbagi data (*sharing*) atau informasi serta berbagi penggunaan suatu operasi pada beberapa tingkatan. Terakhir, identitas, klasifikasi, polimorfisme, serta pewarisan adalah karakter utama dari bahasa pemrograman berorientasi obyek. Setiap komponen digunakan secara terpisah, tetapi dapat saling melengkapi satu sama lainnya secara sinergis.

2.3 Unified Modeling Language

Banyaknya alat bantu yang dapat digunakan dalam analisis dan desain berorientasi obyek menyebabkan munculnya ketidakjelasan alat bantu apa yang paling unggul. Para pengguna harus memilih diantara bahasa pemodelan dan alat bantu pemodelan yang beragam. Untuk mengatasi masalah tersebut maka *Object Management Group* (OMG) kemudian mengeluarkan UML, dimana dengan adanya UML ini diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri perangkat lunak (Suhendar, 2002).

UML merupakan gabungan dari Booch, OMT, OOSE dengan membuang elemen-elemen yang tidak praktis, ditambah dengan elemen-elemen lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu, sehingga UML lebih ekspresif dan seragam daripada metode lainnya (Suhendar, 2002).

Menurut Suhendar (2002) untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

a. Diagram use case

Diagram *use case* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem 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 menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa use case diagram.

Dalam diagram *use-case* terdapat beberapa hubungan antar *use-case* antara lain: *Extend* dan *Include*. Hubungan *include* menunjukkan bahwa sebuah *business use-case* selalu termasuk dalam alur kerja dari *business use-case* lainnya. Hubungan *extend* menunjukkan bahwa sebuah *use-case* merupakan kelanjutan dari *business use-case* lainnya jika suatu syarat tertentu telah dipenuhi.

b. Diagram class

Diagram *class* menggambarkan seluruh *class* yang terlibat dalam sebuah perangkat lunak. Diagram ini menggambarkan seluruh atribut dan operasi suatu *class* beserta hubungannya dengan *class* lainnya.

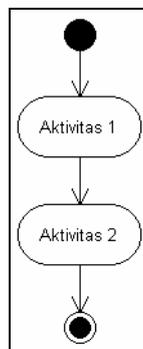
c. Diagram aktivitas

Diagram aktivitas adalah diagram flowchart yang disempurnakan. Diagram aktivitas menggambarkan operasi pada suatu Obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram flowchart adalah adanya dukungan konkurensi (pelaksanaan aktivitas secara bersamaan), pengiriman pesan dan *swimlane* (pelaku/penanggung jawab aktivitas).

Diagram aktivitas sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai use case berinteraksi.

Diagram aktivitas diawali dengan lingkaran hitam, dan diakhiri dengan lingkaran hitam bertepi putih. Aktivitas digambarkan dengan kotak persegi panjang bersudut lengkung. Setiap aktivitas dihubungkan dengan panah dari awal hingga akhir diagram aktivitas.

Sama halnya dengan diagram flowchart, diagram aktivitas pun memiliki simbol yang sama untuk menggambarkan keputusan. Keputusan digambarkan dengan intan, namun deskripsi kondisi yang menyertai keputusan diletakkan diluar simbol intan. Diagram aktivitas dapat menggambarkan konkurensi, yaitu satu atau lebih aktivitas yang berjalan secara bersamaan. Konkurensi diawali dengan sebuah garis tebal horizontal yang menjadi tempat keluarnya garis aktivitas. Konkurensi juga diakhiri dengan garis tebal horizontal. Diagram aktivitas adalah sebagaimana pada Gambar 2.



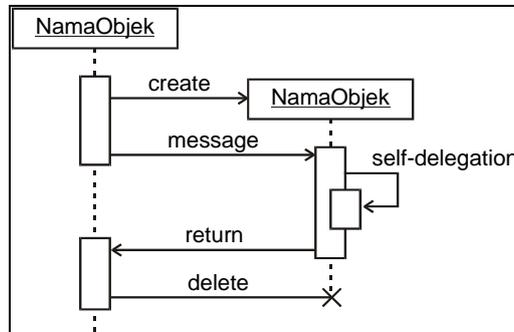
Gambar 2. Contoh diagram aktivitas

d. Diagram interaksi

Diagram interaksi menggambarkan interaksi antar Obyek dalam sebuah urutan waktu. Nama asli diagram interaksi dalam UML adalah *sequence diagram*, namun diterjemahkan sebagai diagram interaksi agar lebih tepat.

Diagram interaksi dapat digunakan untuk memperjelas setiap use case. Sebuah use case dapat melibatkan lebih dari satu Obyek, dan interaksi antar

Obyek akan mudah diamati dengan diagram interaksi. Contoh diagram interaksi ditunjukkan pada Gambar 3.



Gambar 3. Diagram interaksi (*sequence*)

e. Diagram kolaborasi

Diagram kolaborasi merupakan sebuah cara alternatif dalam menggambarkan interaksi antar obyek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar obyek, sedangkan *sequence* diagram lebih menekankan pada urutan kejadian.

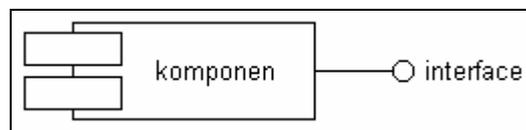
Dalam satu diagram kolaborasi terdapat beberapa obyek, *link*, dan *message*. Diagram kolaborasi digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

f. Diagram keadaan

Diagram ini memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah obyek, kejadian yang menyebabkan transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas.

g. Diagram komponen

Diagram komponen menggambarkan alokasi semua kelas dan obyek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code*, *binary code* dan komponen tereksekusi. Gambar 4 menunjukkan diagram komponen.



Gambar 4. Diagram komponen

h. Deployment diagram

Diagram deployment menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Diagram ini memberi simbol untuk segala bentuk perangkat keras, seperti: komputer, printer, router, hub, jaringan, dan sebagainya.

Diagram deployment menunjukkan tata-letak perangkat keras secara fisik, dan komponen yang terdapat didalamnya. Diagram ini menunjukkan secara lengkap arsitektur dari sistem dan perangkat kerasnya. Diagram ini akan memperjelas pengguna bagaimana hasil akhir dari sistem yang dikembangkan, dan mempermudah teknisi dalam pemeliharaan sistem.

3. PENGEMBANGAN LMS

3.1 Analisis Kebutuhan

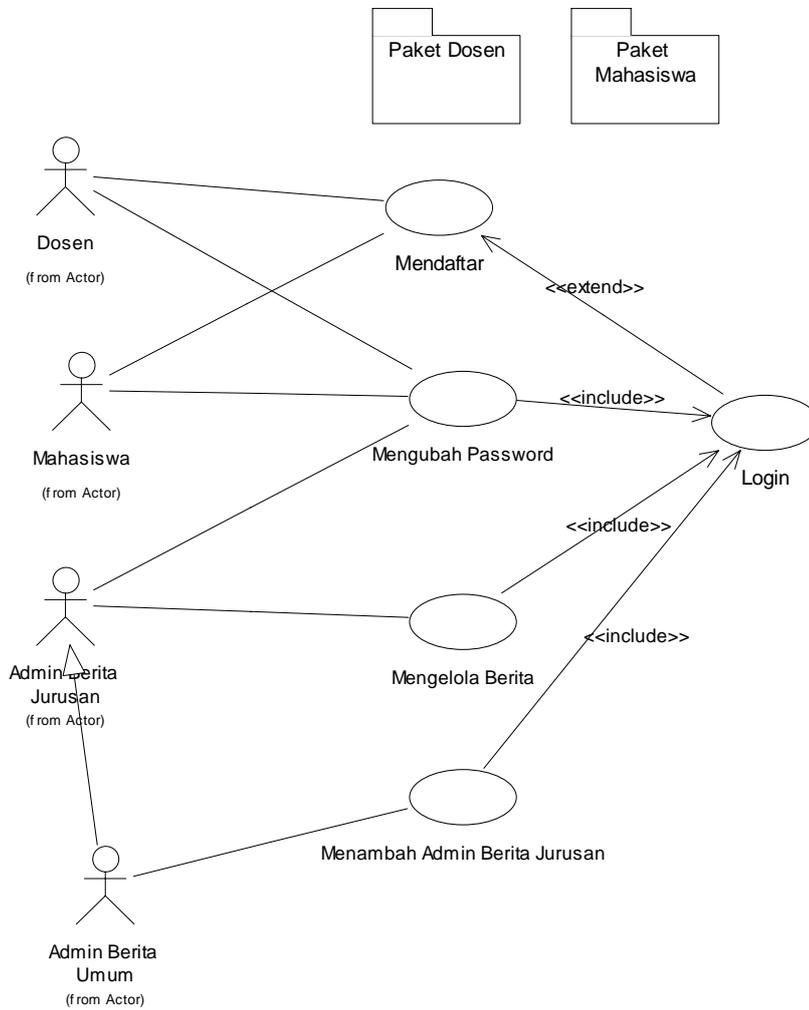
Metode analisis yang digunakan dalam mengembangkan perangkat lunak ini menggunakan metode analisis berorientasi objek dengan menggunakan UML sebagai alat bantu pemodelan.

Bahwa dari analisis yang dilakukan terdapat beberapa hal yang menjadi permasalahan dalam proses belajar mengajar konvensional, yaitu :

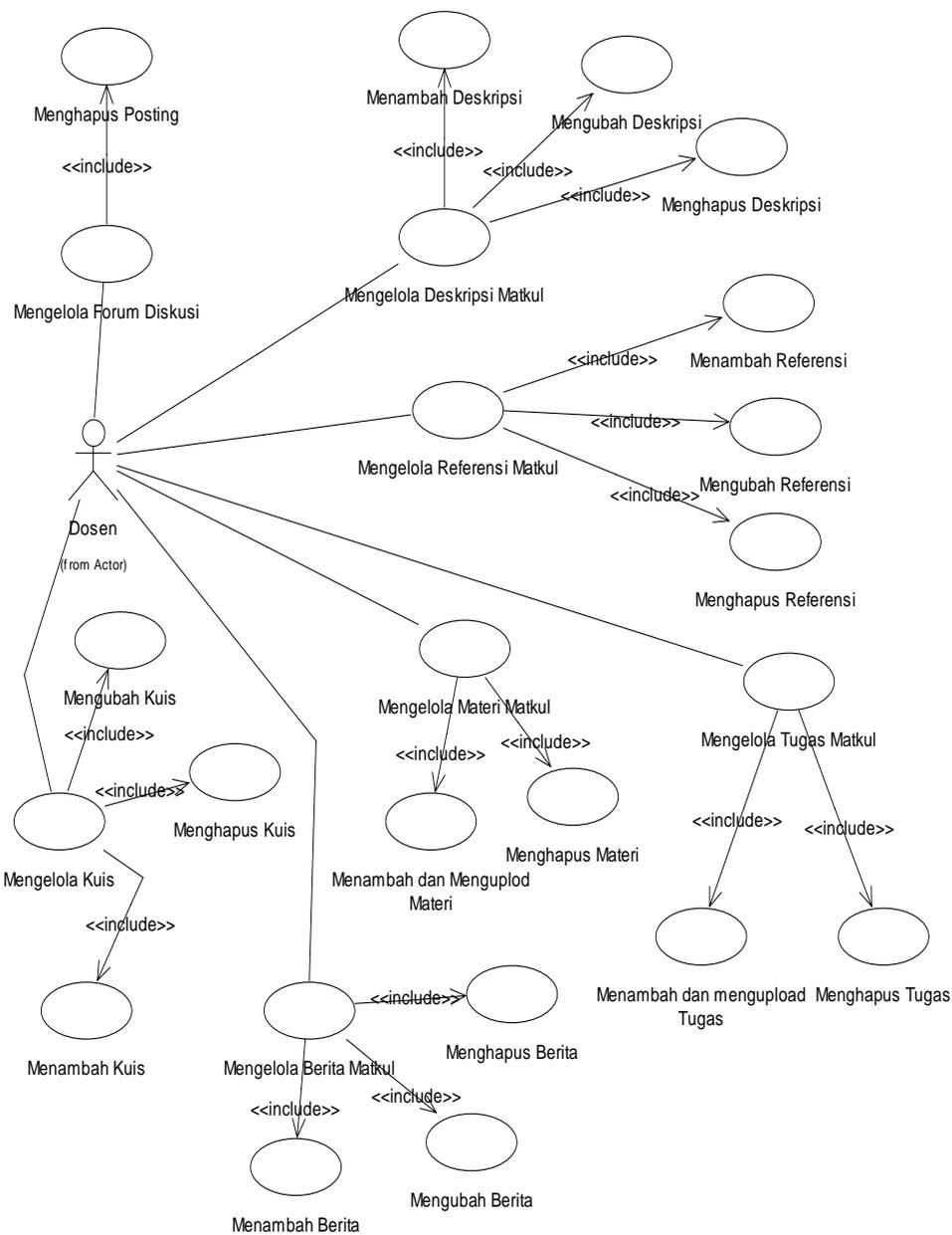
- a. Komunikasi antara mahasiswa dan dosen yang tidak selalu lancar mengakibatkan kurangnya waktu untuk berdiskusi antara dosen dan mahasiswa.
- b. Kurangnya interaksi yang memungkinkan mahasiswa untuk lebih kreatif dan aktif dalam proses pembelajaran sehingga diperlukan media yg dapat mempermudah proses interaksi ini.
- c. Kurangnya referensi maupun acuan yang digunakan sehingga umumnya mahasiswa hanya memperoleh materi yang disampaikan dosen di kelas.

Dari analisis permasalahan tersebut kemudian diidentifikasi objek-objek apa saja yang berperan dalam sistem tersebut dan untuk selanjutnya dapat dikembangkan dan dirinci lebih dalam hasil analisis tersebut. Dalam tahapan ini digunakan diagram *use case* untuk menggambarkan fungsionalitas sistem dari perspektif pengguna sehingga dapat diketahui dengan jelas cakupan dari sistem, aktor-aktor yang berperan dalam sistem, serta interaksi antara aktor dengan sistem. Gambar 5 menunjukkan diagram *use case* untuk keseluruhan aplikasi *Learning Management System*. Adapun isi paket Dosen ditunjukkan pada Gambar 6 dan paket Mahasiswa adalah Gambar 7.

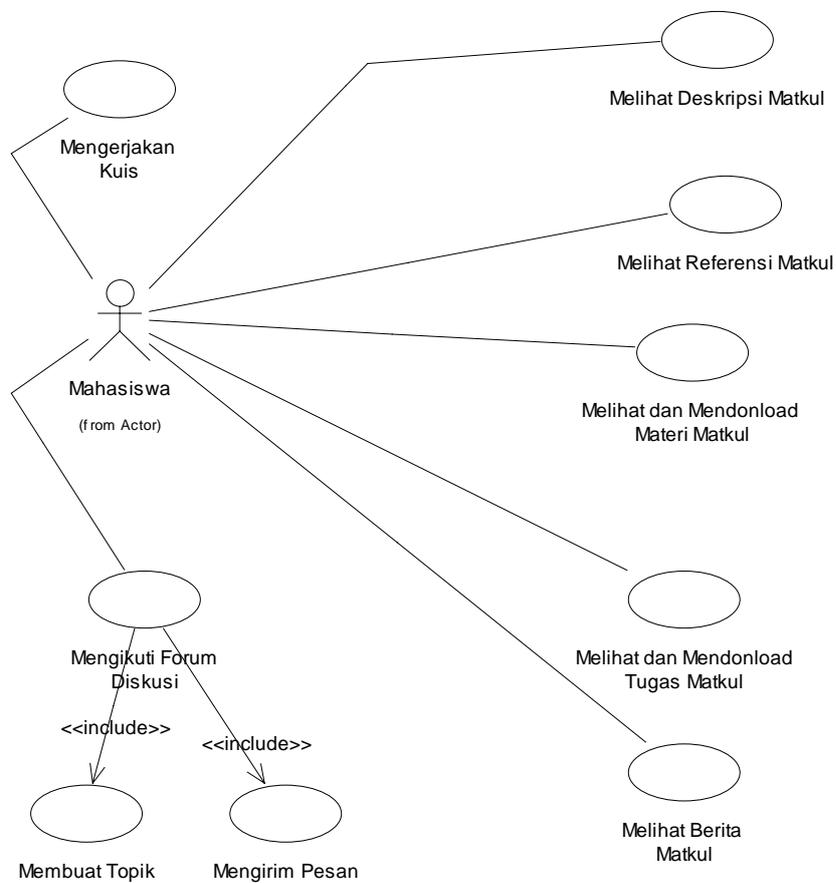
Berdasarkan analisis tersebut ada beberapa fungsionalitas yang harus tersedia dalam sistem yakni kemudahan mahasiswa untuk memperoleh informasi berkaitan dengan mata kuliah yang diambilnya baik berupa deskripsi mata kuliah, referensi yang digunakan dosen dalam mengajar serta kemungkinan mahasiswa untuk *men-download* materi yang diberikan dosen di dalam kelas sehingga dimungkinkan bagi mahasiswa yang tidak mengikuti suatu pertemuan tetap dapat mengikuti materi yang diberikan di samping juga untuk mempermudah bagi dosen dan mahasiswa dalam proses tukar menukar data.



Gambar 5. Diagram *use case* secara keseluruhan untuk aplikasi *Learning Management System*



Gambar 6. Diagram *use case* isi dari paket Dosen



Gambar 7. Diagram *use case* isi dari paket Mahasiswa

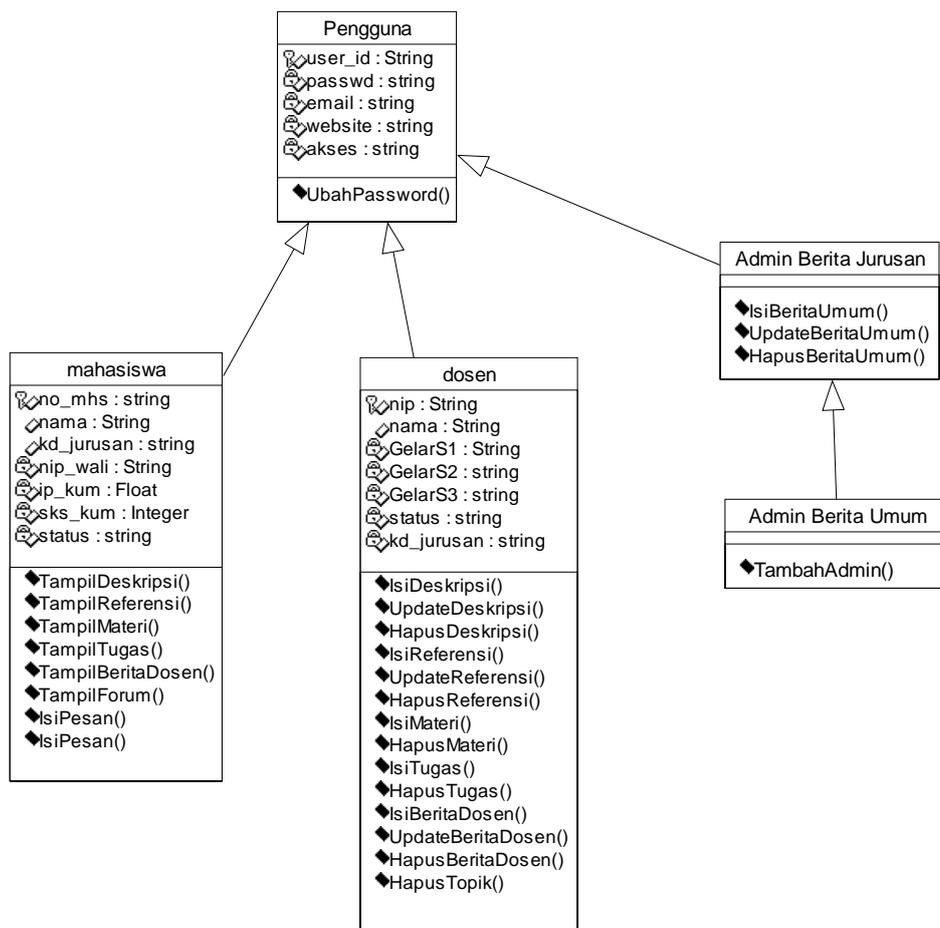
Di samping kemudahan dalam memperoleh informasi seputar mata kuliah, sistem ini juga diharapkan mampu menjembatani komunikasi antara dosen dan mahasiswa yang dirasa masih belum mencukupi melalui kelas konvensional. Diharapkan nantinya mahasiswa dan dosen dapat saling berdiskusi mengenai mata kuliah yang mereka ambil dan mereka ampu. Terakhir, sistem diharapkan mampu mempermudah bagi dosen dan mahasiswa berkaitan dengan tugas-tugas yang diberikan. Di sini diharapkan mahasiswa dapat langsung meng-*upload* tugas mereka dan dosen pun dapat langsung men-*download*-nya.

3.2 Perancangan

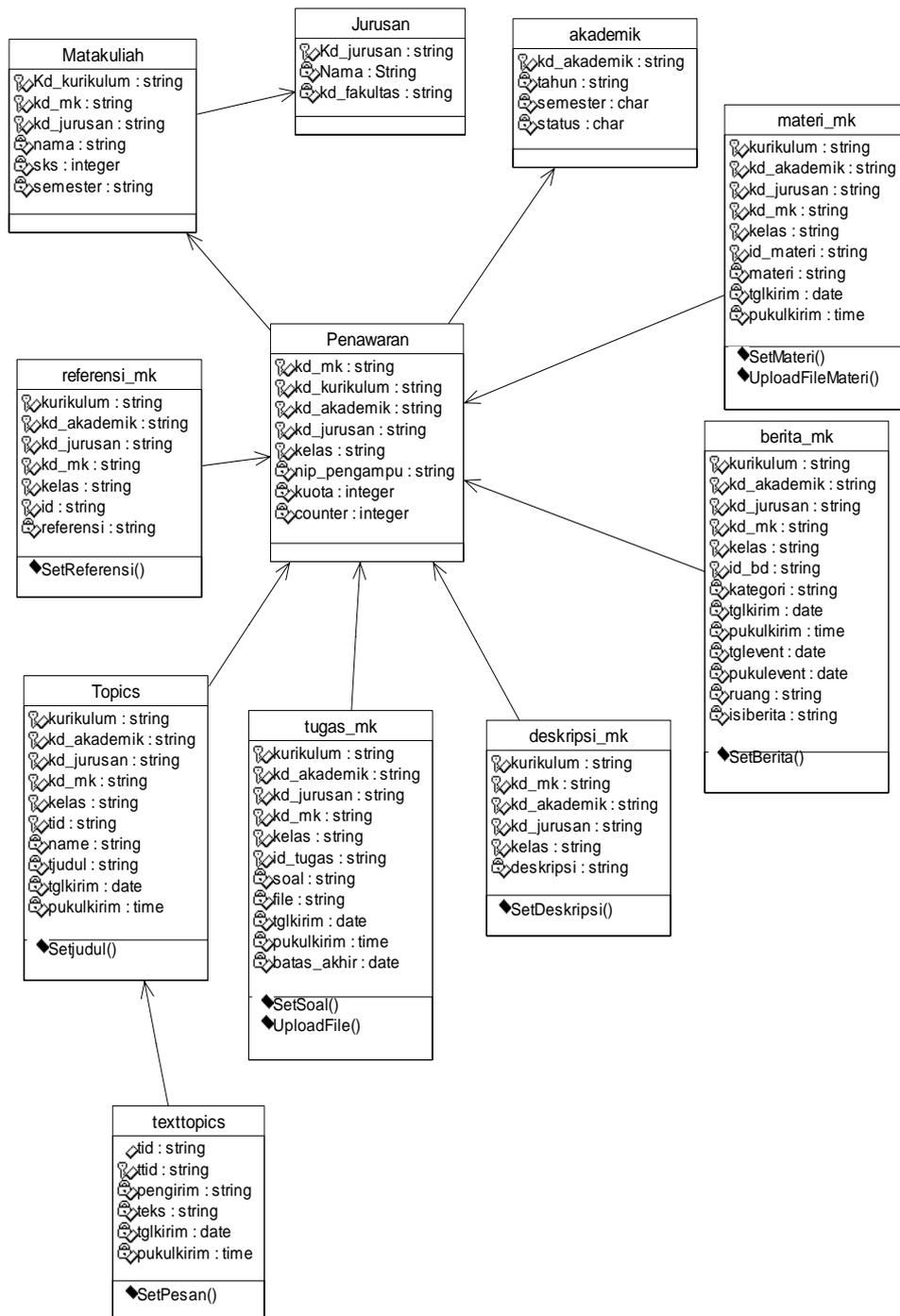
Metode perancangan yang dilakukan juga menggunakan metode berorientasi objek. Berdasarkan hasil analisis kebutuhan sebelumnya kemudian ditentukan alur kerja dari sistem. Hal ini digambarkan melalui diagram *sequence* untuk mengetahui urutan-urutan proses dari masing-masing *use case*.

Selanjutnya ditentukan pula kelas-kelas yang terlibat di dalam sistem dan atribut-atribut dari masing-masing kelas tersebut. Hal ini dilakukan dengan melakukan analisis *use case* dan kemudian digambarkan dalam suatu diagram *class*. Berdasarkan analisis ini didapatkan kelas-kelas untuk aktor yang berperan dalam sistem dan kelas-kelas entitas yang digunakan dalam sistem.

Gambar 8 adalah kelas-kelas untuk aktor yang berperan dalam sistem, sedangkan Gambar 9 adalah kelas-kelas entitas yang digunakan dalam sistem.



Gambar 8. Diagram *class* untuk aktor-aktor yang berperan dalam sistem



Gambar 9. Diagram *class* untuk entitas-entitas yang berperan dalam sistem

3.3 Implementasi

Dalam tahap implementasi ini ada beberapa hal yang harus dipersiapkan untuk pengembangan sistem; yakni perangkat keras, perangkat lunak pengembangan, dan rancangan antar muka sistem.

Adapun perangkat lunak yang digunakan untuk pengembangan sistem adalah:

- a. Desain antarmuka
 1. Microsoft Windows Xp
 2. Macromedia Dreamweaver MX sebagai *tool* untuk mendesain halaman *web*
- b. *Scripting* dan penyimpanan data
 1. Linux Mandrake 9.2
 2. Quanta+ sebagai editor untuk penulisan *script*
 3. PHP 4.3, sebagai bahasa *scripting* yang digunakan untuk pembuatan dan pengembangan sistem berbasis web
 4. PostgreSQL 7.4 sebagai *database server* untuk penyimpanan data yang digunakan.
- c. Internet Explorer 5.0, Mozilla, dan FireFox sebagai *web browser*
- d. Rational Rose 2000 sebagai alat bantu pemodelan UML dalam proses analisis dan desain sistem.

4. PENGUJIAN

4.1 Pengujian teknis

Agar sistem yang dibuat dapat ditentukan telah sesuai dengan hasil analisis dan perancangan, maka dilakukan pengujian terhadap sistem. Pengujian dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi dikarenakan kesalahan sintaks, fungsi maupun implementasi lainnya, sehingga dapat sedini mungkin diantisipasi dengan melakukan perbaikan-perbaikan dan kesalahan-kesalahan yang terjadi menjadi minimal. Pengujian dilakukan baik di komputer lokal maupun terhubung dengan jaringan dan menggunakan beberapa *browser* untuk menentukan kompatibilitas sistem yang dibuat.

4.2 Pengujian non-teknis

Pengujian non-teknis *akan* dilakukan dengan menggunakan LMS yang telah dibuat pada dunia nyata. LMS ini akan diujikan pada kelas online yang diselenggarakan oleh Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia pada semester genap tahun akademik 2005/2006. Hasil pengujian ini akan dianalisis sebagai bahan untuk perbaikan LMS pada masa yang akan datang.

5. PENUTUP

Setelah dilakukan serangkaian pengujian teknis maka dapat disimpulkan bahwa aplikasi *learning management system* ini sudah dapat berjalan dengan baik dan diharapkan dapat memberikan manfaat antara lain:

- a. Dapat menjadi media untuk memperlancar komunikasi antara dosen dan mahasiswa dalam proses belajar mengajar di luar kelas
- b. Mempermudah mahasiswa yang ketinggalan mata kuliah sehingga bisa mendapatkan informasi mengenai materi terakhir maupun tugas yang diberikan.
- c. Memudahkan dosen untuk menyampaikan materi-materi tambahan di luar kelas, serta tidak direpotkan dengan adanya mahasiswa yang silih berganti mengkopi materi yang disampaikan.
- d. Membantu proses penyerapan materi dengan adanya forum diskusi yang memungkinkan untuk berdiskusi secara intens mengenai suatu permasalahan tertentu berkaitan dengan materi yang diberikan.
- e. Membantu mahasiswa untuk mengirimkan tugas secara langsung sesuai tugas-tugas yang diberikan oleh dosen.

PUSTAKA

- Govindasamy, T. (2002). Successful Implementation of e-Learning: Pedagogical Considerations. *Internet and Higher Education*, 4, 287-299.
- Hunt, N. P. (1999). PROJECT LEARN: Supporting On-Campus Learning with On-Line Technologies. *Interactive Learning Environments*, 7(2-3), 269-282.
- Nigroho, A. (2002). *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Obyek*. Bandung: Penerbit Informatika.
- Pituch, K. A., dan Lee, Y.-k. (2004). The Influence of System Characteristics on e-Learning Use. *Computers & Education*.
- Suhendar, A. et al. (2002). *Visual Modeling Menggunakan UML dan Rational Rose*. Bandung: Penerbit Informatika.