

**RANCANG BANGUN SISTEM KUIS *ONLINE* ADAPTIF
SEBAGAI EKSTENSI *CONTENT MANAGEMENT SYSTEM***

**Tesis
untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-2
Program Studi Magister Sistem Informasi**



**Oleh :
Arief Hidayat
J4F009008**

**PROGRAM PASCASARJANA
UNIVERSITAS DIPONEGORO
SEMARANG
2011**

ABSTRAK

Penelitian ini membahas sebuah sistem kuis online adaptif yang dapat dimanfaatkan sebagai ekstensi *content management system*. Sistem kuis adaptif ini menjadi lebih personal karena model pertanyaan yang disajikan secara khusus dirancang bagi mahasiswa sesuai dengan tingkat kemampuan mereka. Mahasiswa akan lebih mengenal kekuatan dan kelemahan dalam proses belajar mereka karena mereka tidak akan menuju ke tingkat kesulitan yang lebih tinggi jika mereka tidak memenuhi nilai yang dipersyaratkan pada tingkat tertentu. Makalah ini fokus pada komponen utama fitur adaptif dan teknik untuk melaksanakan komponen adaptif tersebut. Sebuah studi perbandingan antara sistem adaptif saat ini dilakukan untuk mengidentifikasi komponen adaptif yang diterapkan dan teknik untuk menerapkan komponen adaptif.

Hasil studi banding menjadi dasar untuk mengembangkan sistem kuis adaptif ini. Sistem kuis adaptif ini terdiri dari tiga komponen utama: *student model*, *domain model* dan *adaptation model*. *Student model* menggambarkan pengetahuan mahasiswa, *domain model* merupakan domain mengajar atau representasi dari *student model*, sedangkan *adaptation model* terdiri dari sekumpulan aturan yang mendefinisikan aksi pengguna. Teknik *stereotype* dan *overlay model* diterapkan untuk *student model*, *semantic network* diterapkan pada *domain model* dan *'IF-THEN' rule* diterapkan pada *adaptation model*. Pengembangan sistem menggunakan metode *system development life cycle* dengan pendekatan berorientasi objek.

Hasil dari penelitian ini adalah sebuah ekstensi *content management system* JOOMLA yang dapat digunakan sebagai sebuah sistem penilaian mahasiswa berdasarkan kemampuan, pengetahuan dan preferensi dari masing-masing mahasiswa.

Kata kunci : Kuis online, adaptif, *student model*, *domain model*, *adaptation model*, ekstensi, *content management system*

ABSTRACT

This research describes an adaptive online quiz system that can be utilized as an extension of content management system. With 'adaptive' capability, quizzes become more personalized as the question presented are specifically model or designed for students according to their level of proficiency. With this, student will get to know their strength and weaknesses in their study as they will not move to a higher level of difficulty if they fail to score a certain rate. A comparative study among the current adaptive systems is done to identify the adaptive components that they applied, the techniques to implement the adaptive components and also the available features.

This Web-based quiz system is an adaptive quiz system for student self-assessment with three main components: Student Model, Domain Model and Adaptation Model. Student model describes the student's knowledge, domain model represents the teaching domain or the representation of the student model, while adaptation model consists of a set of rules that defines the user's action. The Stereotypes and Overlay Model techniques are applied to model the student's knowledge, the Semantic Network is applied to model the Domain Knowledge and the 'IF-THEN' rules is applied to model the Adaptation Model. Development of the system using the method of system development life cycle with object-oriented approach.

The results of this research is an extension for content management system that can be used as a system of student assessment based on skills, knowledge and preferences of each student

Keywords : online quiz, adaptive, student model, domain model, adaptation model, extension, content management system

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penilaian sering digunakan untuk mengukur performa peserta didik (Quinn & Reid, 2003). Alotaiby & Chen (2005) menggambarkan penilaian sebagai salah satu komponen utama yang membantu peserta didik dalam belajar. Kuis, yang berfungsi sebagai jenis penilaian (QuestionMark & League, 2004) adalah yang paling banyak digunakan dan merupakan metode penilaian yang dikembangkan dengan baik dalam pendidikan tinggi (Brusilovsky & Miller, 1999).

Kuis online memberikan banyak keuntungan, salah satunya yaitu memungkinkan peserta didik untuk mengerjakan kuis kapan saja. Penggunaan web sebagai sarana untuk mengerjakan kuis tidak mengatasi isu penting tentang peningkatan proses belajar peserta didik, akan tetapi hanya sebagai media teknologi revolusi kuis. Salah satu permasalahan utama dengan kuis online, adalah kurangnya nilai signifikan yang disebut 'Personalisasi'. Tanpa personalisasi, sistem memperlakukan semua peserta didik dengan cara yang sama. Personalisasi dibutuhkan sistem untuk menyesuaikan dengan kebutuhan peserta didik secara otomatis yang disebut 'Adaptif' (Santally & Senteni, 2005). Kemampuan sistem adaptif dapat memutuskan pilihan mana yang terbaik bagi pengguna berdasarkan *user model* mereka. Sistem adaptif terus melacak aktivitas pola pengguna dan mencoba menyesuaikan antarmuka atau konten yang cocok untuk pengguna yang berbeda dengan keahlian, pengetahuan dan preferensi yang

berbeda (Kules,2000). Istilah adaptif sering membingungkan dengan istilah beradaptasi (*adaptable*). Beradaptasi memungkinkan pengguna untuk mengontrol penyesuaian (Kules, 2000) dengan memungkinkan pengguna untuk menyesuaikan sistem sesuai dengan kebutuhan sendiri (Cheng & Kinshuk, 2004). Adaptif sendiri adalah fitur yang digunakan untuk mengukur tingkat peserta didik saat ini dari kompetensi domain (Cheng & Kinshuk, 2004). Sistem ini disebut adaptif jika dapat mengubah atribut sendiri secara otomatis sesuai dengan kebutuhan pengguna (Weibelzahl, 2002).

Dewasa ini pembuatan dan pengelolaan suatu website akan semakin dimudahkan dengan adanya framework *Content Management System* (CMS). CMS merupakan suatu sistem yang digunakan untuk membuat dan mengelola konten dari suatu website (Luthfie, 2005). Salah satu CMS yang saat ini banyak digunakan adalah *joomla*. *Joomla* merupakan salah satu *opensource* CMS yang memberikan kemudahan dalam membangun dan mengelola website dan aplikasi online yang lain (JoomlaTeam, 2005). *Joomla* sudah digunakan di seluruh dunia dan menjadi basis dari beberapa macam website mulai dari website sederhana sampai yang kompleks. Kemudahan dan kelebihan yang dimiliki *joomla* telah membuat *joomla* menjadi CMS yang banyak digunakan di internet.

Fungsionalitas program inti *joomla* dapat ditambah dengan menginstall aplikasi ekstensi dari CMS yang sesuai dengan keperluan. Aplikasi tersebut bisa berupa komponen, modul atau mambots (Rachdian & Sikumbang, 2006). Komponen merupakan program *add-on* yang biasanya ditulis dalam bahasa

pemrograman PHP. Komponen-komponen yang terdapat pada *joomla* saat ini belum tentu sesuai dengan apa yang dibutuhkan oleh suatu website, oleh karena itulah kemampuan dalam membuat dan menambahkan komponen pada *joomla* sangatlah penting. Contoh komponen ekstensi *joomla* yang sudah ada di internet dan sudah banyak digunakan adalah *phpshop*. *Phpshop* merupakan komponen yang dapat menambahkan kemampuan *e-commerce*. Contoh komponen yang lain adalah *IFPortfolio* yang merupakan komponen yang memiliki kemampuan untuk membuat, meng-*update* atau memodifikasi suatu direktori kecil yang berisi hasil-hasil pekerjaan.

Hal tersebut yang mendasari pentingnya penelitian tentang rancang bangun sistem kuis *online adaptif* untuk mengakomodasi suatu sistem *assessment* berdasarkan kemampuan, pengetahuan dan preferensi dari masing-masing peserta didik, yang nantinya dapat digunakan sebagai ekstensi dari sebuah *Content Management System*.

1.2 Perumusan Masalah

Berdasarkan latar belakang sebelumnya, peneliti merumuskan permasalahan dalam penelitian ini yaitu bagaimana merancang bangun sistem kuis *online adaptif* untuk dapat digunakan dalam proses penilaian peserta didik berdasarkan kemampuan, pengetahuan dan preferensi dari masing-masing peserta didik, yang nantinya dapat digunakan sebagai ekstensi dari sebuah *Content Management System*

1.3 Batasan Masalah

Perumusan masalah yang dikemukakan dalam penelitian ini akan peneliti batasi agar tidak meluas, yaitu

1. Sistem kuis *online* adaptif yang dibangun hanya untuk jenis soal *multiple choice* (pilihan ganda), *true/false* (benar/salah), dan *fill in the blank* (isian singkat).
2. Sistem kuis *online* adaptif yang dibangun hanya mendukung satu jenis teknik penilaian adaptasi yaitu pertanyaan adaptif.
3. Tingkat keahlian peserta didik dari pengetahuan domain ditentukan oleh aturan (*rule*) yang diatur oleh pengajar dalam sistem.
4. Tingkat kesulitan dari masing-masing pertanyaan ditentukan oleh pengajar.
5. Sistem kuis *online* adaptif yang dibangun disesuaikan dengan *framework Content Management System (CMS) JOOMLA* sehingga dapat dimanfaatkan sebagai ekstensi CMS tersebut.

1.4 Keaslian Penelitian

Penelitian terkait dengan perancangan penilaian berbasis web telah banyak dilakukan antara lain Krishna (2001) dalam publikasinya berjudul “*Web-based Adaptive Testing*” mengemukakan tentang sistem penilaian online yang dapat menghasilkan pertanyaan adaptif. Peserta didik akan diberikan *pre test* dengan tingkat kesulitan yang berbeda. Nilai yang diperoleh dari *pre test* ini akan digunakan sebagai nilai estimasi kemampuan awal (*OldValue*). Tingkat kemampuan ini dihitung berdasarkan persamaan berikut.

$$\text{NewValue} = \text{OldValue} + \text{OldValue} * (\text{level of Question} / 10) \dots\dots\dots(1.1)$$

$$\text{NewValue} = \text{OldValue} + \text{OldValue} * (5 - \text{level of Question} / 10) \dots\dots\dots(1.2)$$

Persamaan (1.1) adalah untuk pertanyaan yang dijawab benar dan persamaan (1.2) adalah untuk pertanyaan yang dijawab salah. Ketika peserta didik memulai kuis tersebut, OldValue adalah nilai yang didapatkan oleh peserta didik selama *pre-test*. Setelah itu, nilai ini akan diupdate berdasarkan respon peserta didik untuk setiap pertanyaan. Grade kuis ini dalam batasan 0-1, yang hasil akhirnya laporan akan ditampilkan. Laporan ini akan mencakup hasil serta semua topik yang perlu direvisi oleh peserta didik.

Roberts dan Verbyla (2003) dalam publikasinya yang berjudul “*An Online Programming Assessment Tool*” mengemukakan tentang pengembangan tool yang memberikan kemudahan penggunaan lingkungan pemrograman yang memfasilitasi pengembangan, pengujian, dan menandai tugas pemrograman . Tool ini bertanggungjawab untuk beberapa tugas pengembangan program yang biasanya dibutuhkan sehingga mahasiswa dapat fokus pada tugas menulis *source code* program. Tool ini telah digunakan dalam topik pemrograman Java untuk dua tahun berturut-turut. Pengembangan tool ini menggunakan aplikasi *client/server* dengan klien ‘*fat*’ dan server ‘*thin*’. Pertanyaan yang diujikan dalam tool ini yaitu pertanyaan pilihan ganda dan pertanyaan pemrograman. Pertanyaan pilihan ganda dijawab dengan memilih salah satu jawaban, sedangkan pertanyaan pemrograman mahasiswa diminta mengetikkan kode yang ditentukan dalam baris komentar “*//enter your code here*”. Klien akan menjalankan pada platform apapun yang

mempunyai sebuah *Java Development Kit* dan dapat mengeksekusi script *command*, sedangkan server script adalah Unix.

Cheng dan Kinshuk (2004) dalam publikasinya yang berjudul "*Application of Adaptivity in Quiz Systems*" mengemukakan tentang sistem kuis adaptif yang dirancang dan diterapkan untuk program tahun pertama di Massey University. Sistem ini memanfaatkan arsitektur sistem pendidikan cerdas berbasis web sebagai struktur dasar dan menambahkan teknik adaptif sebagai fitur baru untuk membangun sistem kuis adaptif. Sistem hanya mendukung pertanyaan *multiple-choice*. Ada dua pilihan bagi peserta didik untuk memilih setelah mereka login ke sistem, yaitu

1. area latihan (*exercise*) yang secara acak mengambil tingkat pertanyaan yang dipilih untuk peserta didik, dan
2. area *Self-assessment* yang memberikan pertanyaan yang sesuai berdasarkan tingkat performa peserta didik.

Dengan menggunakan sistem kuis ini, peserta didik mampu menjalani penilaian secara individual, dan mengetahui tingkat kompetensi dan kemajuan belajar mereka sendiri. Selanjutnya, sistem ini dapat memberikan umpan balik yang fokus dan tepat waktu kepada peserta didik. Umpan balik dapat digunakan untuk mengarahkan pembelajaran berikutnya dan memotivasi peserta didik untuk mempelajari *resource* lainnya.

Maryanto (2008) dalam penelitiannya yang berjudul "*Pembangunan Perangkat Lunak Pengelola Ujian Berbasis Web*" menghasilkan sebuah Perangkat Lunak Pengelola Ujian (PLPU) yang bernama *eTest (Electronic Test*

Managemen System) . PLPU ini menangani proses persiapan soal tes, pelaksanaan tesnya, pemeriksaan hasil tes sampai pada penyusunan laporan nilai tes. Sistem ini dibangun oleh Maryanto untuk menguji seberapa baik peserta ujian dapat menyelesaikan ujiannya, hasil rancangan sistemnya mempunyai kelemahan yaitu tidak dapat menyimpan jawaban hasil ujian setiap peserta ujian dalam database storage, sehingga setiap peserta-uji dan para pengajar tidak dapat mengevaluasi kembali titik-titik kelemahan jawaban pada materi-ajar yang diujikan, selain itu sistem tersebut tidak bersifat adaptif.

Penelitian yang peneliti lakukan adalah merancang dan membangun sistem kuis *online* adaptif dengan pendekatan berorientasi objek. Implementasi menggunakan *script PHP* dan menggunakan database *mySQL* serta *Apache* sebagai *web server*. Sistem ini digunakan untuk mengelola tahap-tahap penilaian mulai dari pembuatan soal, pelaksanaan, proses koreksi sampai pada penyampaian hasil kuis dengan menambahkan fungsi penyimpanan setiap jawaban peserta yang nantinya dapat dievaluasi kembali sehingga terdapat umpan balik baik bagi peserta maupun pengajar. Sistem kuis online adaptif ini merupakan bentuk penilaian (*assessment*) terhadap peserta didik yang menerapkan komponen-komponen sistem adaptif seperti *user model*, *domain model*, dan *adaptation model*. *User model* dalam hal ini peserta didik atau yang lebih dikenal dengan *student model* menjelaskan pengetahuan peserta didik, *domain model* menyatakan area pengajaran atau pernyataan dari *student model*, sementara *adaptation model* berisi sekumpulan aturan yang mendefinisikan aksi pengguna. Perbedaan penelitian yang dilakukan oleh peneliti dengan penelitian-penelitian sebelumnya

yaitu sistem kuis *online* adaptif ini mempunyai jenis soal *multiple choice* (pilihan ganda), *true/false* (benar/salah), dan *fill in the blank* (isian singkat), selain itu rancang bangun sistem ini disesuaikan dengan *framework* dari *Content Management System* (CMS) *JOOMLA* sehingga *nantinya* akan dapat dimanfaatkan sebagai ekstensi dari CMS tersebut.

1.5 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. menghasilkan sebuah sistem yang dapat menangani tahapan penilaian melalui kuis *online* secara adaptif serta mampu memberikan umpan balik kepada peserta didik (mahasiswa) setelah mengikuti penilaian tersebut,
2. mengidentifikasi komponen adaptif yang utama dari sebuah sistem kuis adaptif,
3. mengidentifikasi teknik implementasi komponen dari sebuah sistem adaptif.
4. mengidentifikasi teknik pembuatan ekstensi untuk *content management system JOOMLA*.

1.6 Manfaat Hasil Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi institusi pendidikan baik peserta didik, pengajar maupun institusi pendidikan itu sendiri.

1. Bagi peserta didik
 - a. memotivasi peserta didik untuk meningkatkan pembelajaran mereka ketika pertanyaan dan feedback kuis diberikan berdasarkan tingkat kemampuan peserta didik,

b. membantu peserta didik untuk mengetahui kekuatan dan kelemahan dalam pembelajaran mereka dan juga membantu mereka memperbaikinya.

2. Bagi pengajar

Membantu untuk mengetahui tingkat kemampuan peserta didik mereka.

3. Bagi institusi pendidikan

Sistem kuis *online* adaptif ini dapat dimanfaatkan sebagai ekstensi dalam *content management system* yang sudah mereka bangun sebagai salah satu proses penilaian.

BAB II

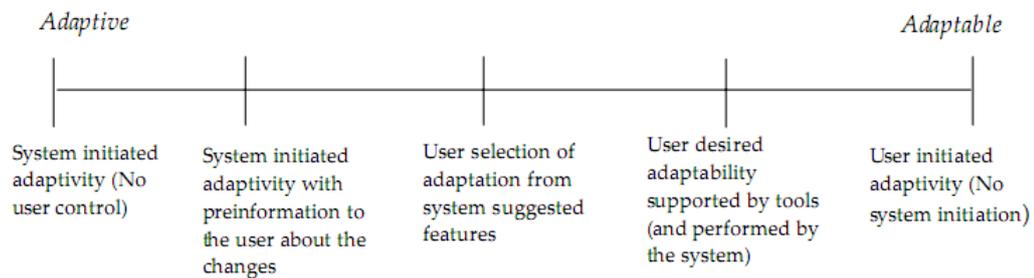
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Menurut Cheng dan Kinshuk (2004) dalam publikasinya yang berjudul ”*Application of Adaptivity in Quiz Systems*” mengemukakan tentang sistem kuis adaptif yang dirancang dan diterapkan untuk program tahun pertama di *Massey University*. Sistem ini memanfaatkan arsitektur sistem pendidikan cerdas berbasis web sebagai struktur dasar dan menambahkan teknik adaptif sebagai fitur baru untuk membangun sistem kuis adaptif. Sistem hanya mendukung pertanyaan *multiple-choice*. Ada dua pilihan bagi peserta didik untuk memilih setelah mereka login ke sistem, yaitu area latihan (*exercise*) yang secara acak mengambil tingkat pertanyaan yang dipilih untuk peserta didik, dan area *Self-assessment* yang memberikan pertanyaan yang sesuai berdasarkan tingkat performa peserta didik

Tingkat *adaptivity* dalam lingkungan pembelajaran cerdas bervariasi dari sistem ke sistem. Di satu sisi, sistem dapat beradaptasi yang memungkinkan pengguna untuk mengubah parameter sistem tertentu dan menyesuaikan perilaku sistem yang sesuai. Di sisi lain, sistem dapat adaptif yaitu mengubah perilaku mereka secara otomatis berdasarkan asumsi sistem tentang kebutuhan pengguna.

Oppermann et al. (1997) merangkum spektrum adaptivity lengkap yang ada dalam sistem pembelajaran, seperti yang ditunjukkan dalam Gambar 2.1



Gambar 2.1 Keseluruhan Spektrum dari Konsep Adaptasi (Chen dan Kinshuk, 2004)

Student adaptivity di sistem kuis ini mampu mengatasi potensi *adaptivity* dalam lingkungan penilaian berbasis Web, antara lain: (Cheng dan Kinshuk, 2004)

1. Monitoring kinerja mahasiswa: ketika seorang mahasiswa menggunakan sistem ini, dia disajikan dengan sekumpulan pertanyaan. Mahasiswa itu bekerja pada sekumpulan pertanyaan, menjawab setiap pertanyaan, dan kemudian menyampaikan jawaban ke sistem. Mesin inferensi dalam sistem mengacu pada *question base* dalam *database backend*. *Question base* terdiri dari serangkaian tingkat pertanyaan yang berbeda untuk menilai, jawaban untuk setiap pertanyaan, dan kriteria penilaian untuk menilai tes mahasiswa. Isi dari *question base* disediakan dan ditentukan oleh guru. Langkah-langkah menyimpulkan meliputi:

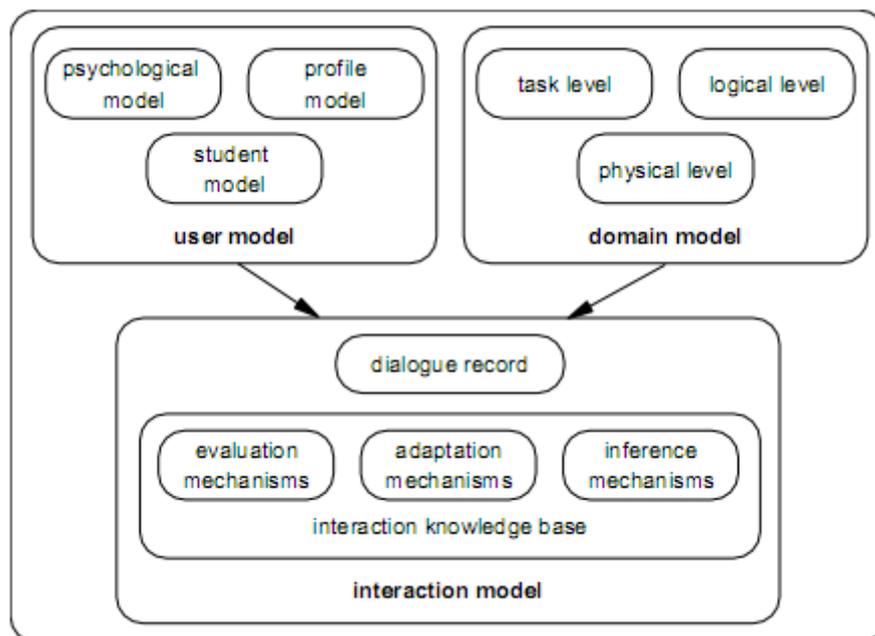
- a. Pencocokan jawaban mahasiswa dengan jawaban standar yang diberikan oleh guru;
- b. Menghitung hasil akhir untuk mahasiswa dan menentukan tingkat mahasiswa dan langkah berikutnya yang tepat.

Kenyataannya, kualitas *question base* adalah inti dari sistem ini, yang sebagian besar bergantung pada pengalaman guru.

2. Menyajikan bimbingan yang tepat: inisialisasi sistem memberikan tingkat pertanyaan yang sesuai kepada mahasiswa, berdasarkan profile-nya. Berdasarkan kinerja mahasiswa, sistem menyesuaikan diri dengan memberikan umpan balik yang tepat untuk mahasiswa. Umpan balik termasuk jawaban yang benar dan langkah yang disarankan berikutnya, misalnya sistem merekomendasikan mahasiswa untuk pergi ke tingkat berikutnya, jika mahasiswa memperoleh hasil memuaskan. Jika tidak, sistem merekomendasikan pengguna untuk mencoba latihan lagi atau meninjau kembali materi pembelajaran yang relevan. Sebagai contoh, dalam sistem ini:
 - a. Jika mahasiswa mempunyai kurang dari 80 poin dalam satu set pertanyaan sebanyak 100, maka sistem menyimpulkan bahwa mahasiswa memerlukan studi lebih lanjut dalam materi yang relevan. Sistem dalam hal ini memberikan bimbingan yang sangat konkret untuk mahasiswa, seperti menampilkan jawaban yang benar sebagai umpan balik kepada mahasiswa, informasi terkait dan link, dan kemudian merekomendasikan mahasiswa untuk memeriksa konten yang relevan dan mencoba penilaian lagi. Berdasarkan umpan balik dan rekomendasi, mahasiswa mampu memutuskan langkah berikutnya dan membawa mereka untuk membuka link yang disediakan oleh sistem atau mengacu pada buku yang relevan.
 - b. Jika mahasiswa mendapatkan 80 poin dalam penilaian, maka sistem menyimpulkan bahwa mahasiswa tersebut telah menguasai pengetahuan yang relevan dan memiliki pengetahuan yang memadai berkaitan dengan

penilaian. Sistem hanya mengirimkan mahasiswa untuk tingkat penilaian berikutnya.

Benyon dan Murray (1993) memperkenalkan arsitektur sistem adaptif yang berfokus pada komponen *adaptivity*. Arsitektur ini dirancang untuk mendukung pengembang dalam memilih teknik representasi yang sesuai. Arsitektur pada dasarnya terdiri dari tiga komponen utama: *user model*, *domain model*, dan *interaction model*, seperti yang ditunjukkan dalam Gambar 2.2



Gambar 2.2 Arsitektur Sistem Adaptif (Benyon dan Murray, 1993)

1. *User model* merupakan sistem keyakinan tentang pengguna. Ini terdiri dari tiga komponen interlinking: Pertama, *student model*, yang berisi asumsi sistem tentang kepercayaan pengguna tentang domain. Sebagai contoh, sistem akan berasumsi bahwa pengguna tahu bagaimana cara membuka file teks, dengan demikian, informasi ini tergantung pada aplikasi dan domain. Komponen kedua model itu adalah *profile model*, yang menyimpan informasi tentang

latar belakang pengguna, minat dan pengetahuan umum. Komponen ketiga, *psychological model*, memiliki domain independen sifat-sifat kognitif dan afektif pengguna.

2. *Domain model* mendefinisikan aspek sistem dan dunia yang penting untuk kesimpulan, misalnya, fungsi yang mungkin berubah. Aspek-aspek tersebut bisa digambarkan pada tingkat yang berbeda, seperti tingkat tugas, tingkat logis, atau tingkat fisik. Model domain adalah dasar untuk semua kesimpulan dan adaptasi.
3. *Interaction model* menangani dialog antara pengguna dan aplikasi. Mungkin mencatat interaksi sebelumnya dan berisi mekanisme untuk kesimpulan sifat pengguna, mekanisme adaptasi sistem untuk sifat-sifat pengguna, dan mekanisme untuk evaluasi adaptasi.

Berdasarkan tinjauan pustaka yang telah dipaparkan sebelumnya, peneliti berkeinginan merancang dan membangun sistem kuis adaptif online yang menerapkan komponen-komponen sistem adaptif yaitu *user model*, *domain model*, dan *adaptation model*. Sistem yang akan peneliti rancang dan bangun disini akan menggunakan *stereotypes* dan *overlay model* untuk menerapkan komponen *user model*, *semantic network* untuk menerapkan komponen *domain model*, dan *IF-THEN rules* untuk menerapkan komponen *adaptation model*. Sistem kuis *online* adaptif yang dihasilkan ini disesuaikan dengan *framework* dari *Content Management System (CMS) JOOMLA* sehingga *nantinya* akan dapat dimanfaatkan sebagai ekstensi dari CMS tersebut.

2.2 Landasan Teori

2.2.1 Kuis Online

Kuis, sebagaimana didefinisikan oleh WordWeb Online (2005) adalah "sebuah ujian yang berisi pertanyaan singkat". Selain itu, Mifflin (1995) mendefinisikan kuis sebagai "serangkaian pertanyaan atau latihan yang dirancang untuk menentukan pengetahuan atau keterampilan". QuestionMark & League (2004) memberikan gambaran lebih detail kuis sebagai "sekumpulan pertanyaan yang digunakan untuk mengukur pengetahuan mahasiswa atau keterampilan untuk tujuan memberikan umpan balik untuk menginformasikan mahasiswa dari tingkat saat ini pengetahuan atau keterampilan". Kuis dapat digunakan untuk menghasilkan beberapa jenis pertanyaan: pertanyaan pilihan ganda, mengisi kekosongan, respon numerik, pencocokan daftar, jawaban singkat, respon hot spot dan benar / salah (Judge, 1999) untuk berbagai tujuan penilaian seperti sebagai penilaian formatif dan sumatif (Quinn dan Reid, 2003).

1. Penilaian Formatif biasanya disebut *self-assessment*, yang memungkinkan peserta didik untuk belajar sendiri, tanpa bimbingan guru (QuestionMark dan League, 2004).
2. Penilaian Sumatif memberikan penilaian kinerja secara keseluruhan dengan memberikan *grading* kuantitatif dan kualitatif dari prestasi belajar peserta didik, yang sering dikenal sebagai "ujian jangka menengah" atau "ujian akhir" (QuestionMark dan League, 2004).

Namun, menurut QuestionMark dan League (2004) kuis biasanya mengacu pada penilaian formatif. Sebelum adanya komputer dan Internet, kuis ini dilakukan

secara manual menggunakan kertas dan pensil. Guru dibutuhkan untuk mempersiapkan pertanyaan-pertanyaan dan kemudian mendistribusikannya secara manual untuk peserta didik mereka. Seiring pertumbuhan komputer dan teknologi, kuis dilakukan melalui Internet. Lopez (2000) kuis online didefinisikan sebagai "sebuah tes yang dilakukan pengguna dengan komputer melalui Internet". Contoh komersial kuis online yang tersedia di pasaran saat ini QuestionMark [<http://www.questionmark.com>] dan Can Studios [<http://www.the-can.com>] (Lalos et al., 2005).

2.2.2 Kuis Adaptif

Pertanyaan dalam kuis dapat disajikan sesuai dengan tingkat kemampuan peserta didik, hal ini dapat dilakukan dengan menambahkan fitur adaptif di kuis,. Perkiraan *performance* peserta didik didasarkan pada dua faktor (Gouli et al., 2001) yaitu

1. nilai jawaban benar yang diberikan oleh peserta didik, dan
2. tingkat kesulitan dari pertanyaan-pertanyaan yang mampu dijawab benar oleh peserta didik.

Gouli et al. (2001) telah menunjukkan tiga pertanyaan penting ketika mengembangkan kuis adaptif yaitu

1. pertanyaan mana untuk diberikan kepada peserta didik ?
2. bagaimana mengukur perkiraan kemampuan para peserta didik dan juga bagaimana mempertahankan atau memperbarui tingkat kemampuan peserta didik? dan
3. apa kriteria untuk menghentikan kuis ?

Berdasarkan pertanyaan di atas, Gouli et al. (2001) telah menggambarkan tentang bagaimana prosedur penilaian adaptif bekerja pada umumnya.

1. Tahap awal sebuah pertanyaan dengan tingkat kesulitan beragam disajikan, karena pada tahap ini, sistem tidak tahu tingkat kemampuan mahasiswa. Sistem ini mengasumsikan bahwa mahasiswa memiliki tingkat pengetahuan beragam untuk suatu pokok bahasan.
2. Sistem akan melakukan pemeriksaan setelah mahasiswa menjawab dan mengirim jawaban,. Sistem akan memperbarui kemampuan peserta didik menjadi lebih tinggi jika jawabannya benar, jika tidak, sistem akan mengatur ke tingkat yang lebih rendah.
3. Sistem kemudian memilih pertanyaan berikutnya yang sesuai untuk peserta didik menurut tingkat mereka.
4. Proses ini berlanjut sampai satu kriteria pengakhiran tercapai seperti peserta didik keluar dari sistem atau peserta didik tersebut telah mencapai pada tingkat tertinggi dari suatu subjek.

Gouli et al. (2001) juga mengeluarkan algoritma sederhana yang dapat digunakan sebagai pedoman dalam mengembangkan kuis adaptif seperti Gambar 2.3

<p><i>Make initial estimation of the learner's proficiency</i></p> <p><i>Repeat</i></p> <p><i> Select questions based on the current estimation and pose it to the learner</i></p> <p><i> Depending on the learner's answer, update the estimation</i></p> <p><i>Until one of the termination criteria is met</i></p>

Gambar 2.3 Algoritma untuk Kuis Adaptif (Gouli et al., 2001)

2.2.3 Pertanyaan Adaptif (adaptive questions)

Pada teknik pertanyaan adaptif, urutan pertanyaan yang dihasilkan secara dinamis tergantung pada jawaban peserta didik (Tzanavari et al., 2004), dengan kata lain, "jawaban yang diberikan untuk pertanyaan-pertanyaan tertentu digunakan untuk menentukan seri berikutnya pertanyaan" (Pitkow, & Recker, 1995). Faktor untuk menentukan pertanyaan yang harus disampaikan kepada peserta didik sangat tergantung pada beberapa aturan yang telah ditetapkan (Gouli et al., 2002), biasanya telah diatur oleh pengajar. Melalui cara ini, peserta didik hanya disajikan dengan pertanyaan-pertanyaan yang relevan, sesuai dengan tingkat keahlian mereka. Manfaat dari jenis adaptasi ini adalah sebagai berikut (Gouli et al., 2002):

1. mengurangi jumlah pertanyaan, dan
2. mengurangi kompleksitas pertanyaan yang disajikan kepada pengguna individu.

Teknik pertanyaan adaptif ini menggunakan konsep kelompok yang sangat terstruktur. Pertanyaan-pertanyaan dikelompokkan dalam kelompok atau dikenal sebagai *class*, sesuai dengan kriteria yang spesifik atau aturan yang ditetapkan oleh tutor. Setiap *class* mungkin berisi beberapa sub dengan karakteristik yang berbeda seperti tingkat kesulitan pertanyaan. *Class* atau *subclass* dipicu dalam urutan, namun *class* yang akan dipicu *class* berikutnya, ditentukan oleh tanggapan pembelajar (Gouli et al., 2002). Teknik ini menerapkan aturan IF <condition> THEN <action> oleh (Lalos et al., 2005):

1. <condition> disebut tingkat pengetahuan para peserta didik tentang topik tertentu, nilai dan tingkat kesulitan penilaian mereka.
2. <action> disebut perubahan yang dihasilkan dalam penilaian ini dan saat ini meliputi perubahan di tingkat pengguna pengetahuan, perubahan di tingkat kesulitan penilaian dan / atau memindahkan pengguna ke bagian yang berbeda.

Pertanyaan berikutnya ditampilkan untuk peserta didik didasarkan pada seperangkat aturan yang telah dibuat oleh penulis.

Pertanyaan dapat terdiri dari pertanyaan-pertanyaan terbuka seperti *multiple-choice*, *true/false*, *fill in the blanks*, *matching list* dan lain-lain. Contoh dari membangun aturan adaptif akan ditampilkan dalam Gambar 2.4 sebagai berikut :

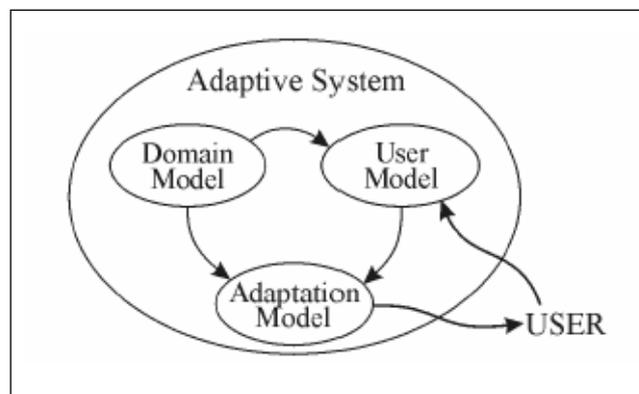
<p>Rule Name: Rule 1</p> <p>Conditions:</p> <ol style="list-style-type: none"> 1. When student is at level ____ 2. When student has score ____ 3. When student has knowledge level ____ at the topic ____ <p>Actions:</p> <ol style="list-style-type: none"> 4. Set the knowledge for section topic's ____ 5. Move to section ____ 6. Move to level ____ <p>Description:</p> <ol style="list-style-type: none"> 7. When a student has score between 80-100 and 8. When a student is at level 3 <p>THEN</p> <ol style="list-style-type: none"> 9. Move to Section XYZ – question <p>Trigger Point</p> <ol style="list-style-type: none"> 10. Trigger point is: Introduction (section) questions no 1

Gambar 2.4 Contoh Aturan Adaptif (Tzanavari et. Al, 2004)

2.2.4 Komponen Sistem Adaptif

Benyon dan Murray's (1993), telah memperkenalkan tiga komponen utama adaptif (berdasarkan arsitektur sistem adaptif) sebagaimana tercantum di bawah:

1. *User model* - merupakan siapa yang dalam proses pembelajaran (misalnya: mahasiswa, pelajar)
2. *Domain model* - merupakan domain yang diajarkan (materi pelajaran)
3. *Adaptation model* - merupakan cara mengajar pengguna



Gambar 2.5 Modul dari Sebuah Sistem Adaptif (Kavcic, 2001)

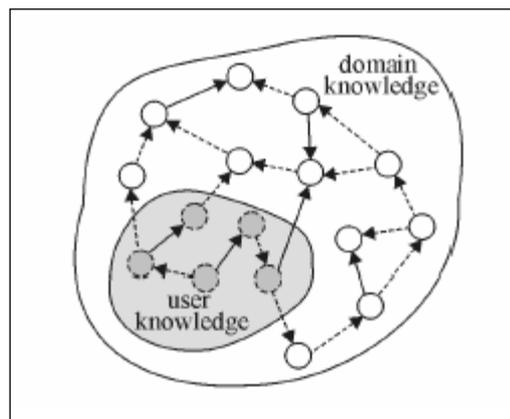
Gambar 2.5 menunjukkan hubungan antara tiga komponen adaptif dan semua adaptasi diselesaikan melalui *user model*. Adaptasi adalah semata-mata tergantung pada kualitas dari informasi yang tersimpan dalam *user model*, oleh karena itu memerlukan informasi pengguna yang diperbarui berdasarkan perintah untuk mendapatkan tingkat pengetahuan pengguna sesungguhnya (Kavcic, 2001).

2.2.4.1 Domain Model

Secara umum, *domain model* dijelaskan sebagai berikut:

1. "kerangka kerja bagi representasi pengetahuan domain pengguna"
(Brusilovsky, 2003)

2. merupakan pengetahuan tentang domain mengajar atau pengetahuan representasi dari *student model* (Kavcic, 2000)
 3. berisi pengetahuan yang akan dikirim kepada peserta didik (Pena et al., 2004).
- Pengetahuan pengguna (*user knowledge*) sering disebut sebagai bagian dari pengetahuan para ahli (*expert knowledge*), sehingga pengetahuan pengguna juga dikatakan sebagai bagian dari pengetahuan domain (*domain knowledge*). Pengetahuan pengguna adalah sebuah lapisan atas (*overlay*) pengetahuan domain (Kavcic, 2000). Gambar 2.6 menunjukkan hubungan antara pengetahuan pengguna dan pengetahuan domain, dan dapat dilihat bahwa pengetahuan pengguna merupakan bagian dari pengetahuan domain.



Gambar 2.6 Hubungan antara Domain Knowledge dan User Knowledge (Kavcic, 2001)

Domain model terdiri dari serangkaian konsep, yang mengacu pada potongan-potongan pengetahuan domain diidentifikasi oleh ahli seperti topik, unsur-unsur pengetahuan, objek dan hasil pembelajaran (Brusilovsky & Anderson, 1998; Brusilovsky et al., 1998). "Untuk setiap konsep *domain model*, sebuah model pengetahuan mahasiswa individu menyimpan beberapa nilai, yang merupakan

perkiraan dari tingkat pengetahuan mahasiswa tentang konsep ini" (Brusilovsky et al, 1998.). Akibatnya, *domain model* memberikan struktur dari domain pengetahuan mahasiswa, sebagai sebuah lapisan atas (*overlay*) untuk *student model* (Brusilovsky & Anderson 1998). Menurut Kavcic (2001), ada ketergantungan antara konsep ketika pengetahuan konsep sebelumnya diperlukan untuk memahami konsep berikutnya (relasi prasyarat).

Struktur termudah dari *domain model* mungkin hanya terdiri dari sekelompok konsep domain. Konsep dalam pengetahuan domain bisa lebih kecil atau lebih besar dalam sebuah potongan. Konsep dapat direpresentasikan sebagai *node* dalam jaringan, namun, dalam cara yang lebih lanjut konsep menjadi struktur dasar dari domain model (Busilovsky et al, 1998.).

Salah satu metode umum yang digunakan untuk merepresentasikan *domain model* yaitu *Semantic Network*. "*Semantic Network* adalah struktur representasi pengetahuan yang sangat baik" (Geller et al, 2002.). Struktur ini terdiri dari sekelompok *node* dan sekelompok *link*. *Node* digunakan untuk mewakili konsep atau benda-benda dunia nyata dan *link* digunakan untuk mewakili hubungan semantik antara konsep dan objek. Menurut Feng et al. (2002), model *semantic network* terdiri dari empat komponen yaitu sekelompok *node*, sekelompok *edge* atau *link* yang berarah, sekelompok label, dan sekelompok *constraint*.

a. Node

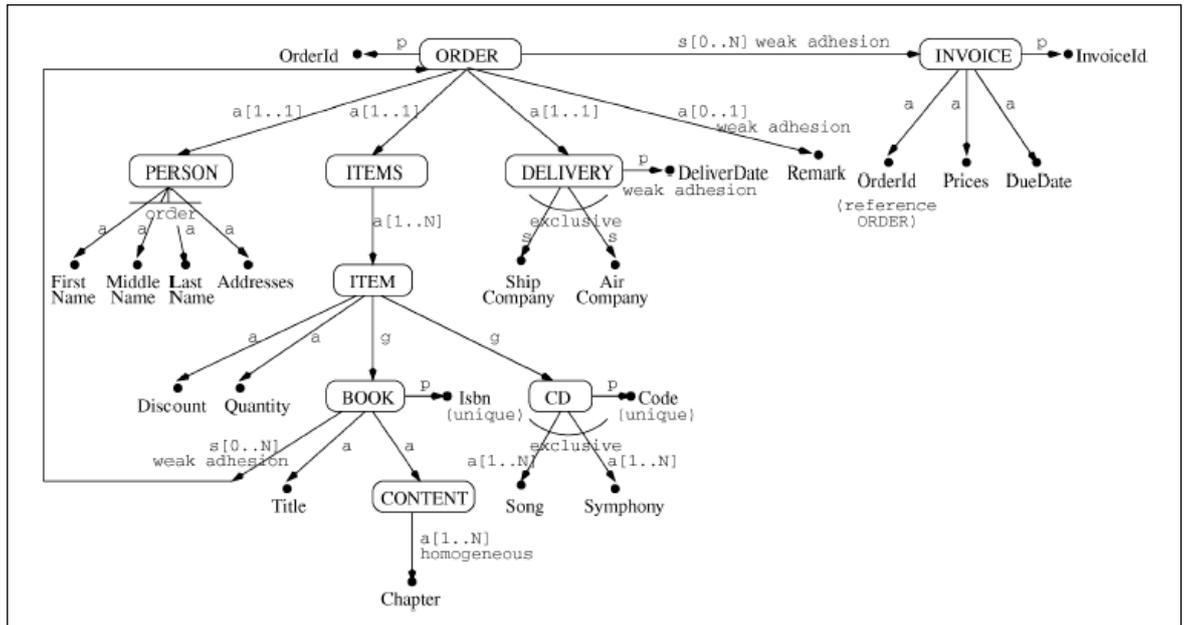
Node dapat dikategorikan sebagai *node* dasar atau *node* kompleks. Sebuah *node* dasar merupakan *node* daun dan tidak memiliki *edge* yang berasal dari *node* tersebut. *Node* kompleks memiliki satu atau lebih *link* yang berasal dari *node* yang

menuju ke *node* lain dan diwakili oleh label. Contoh *node* dasar dan *node* kompleks ditunjukkan pada Gambar 2.7. *Node* dasar dinotasikan dengan *dot* (seperti *First Name*, *Address*, *ISBN*) dan *node* kompleks dinotasikan dengan kotak persegi panjang (seperti *ORDER*, *PERSON*, *BOOK*, *ITEM*).

Setiap *node* dasar terdiri dari sebuah konten yang sederhana baik nilai *atomic* atau nilai *konstruksional*. Nilai *Atomic* terdiri dari tipe data dasar seperti *string*, *integer*, *char* dan sebagainya. Nilai *konstruksional* dapat terdiri dari "*a set value*", "*bag value*", "*list value*" dan "*union value*".

- a. "*a set value*" berarti nilai-nilai dalam sekelompok yang tidak diduplikasi, dengan kata lain, ada yang unik dan tidak urut.
- b. "*bag value*" berarti nilai-nilai di dalam sekelompok yang tidak unik atau dapat diduplikasi, dan tidak urut.
- c. "*list value*" berarti nilai-nilai yang diurutkan dan duplikasi diperbolehkan.
- d. "*union value*" berarti mengembalikan nilai dari anggota koleksi manapun.

Contoh *node* dasar, sebagaimana dimaksud pada Gambar 2.7, adalah "*First Name*", "*Middle Name*" dan "*Last Name*" yang isinya diambil dari tipe data *string atomic*. Contoh lain dari *node* dasar yang nilainya diambil dari nilai yang ditetapkan *konstruksional* adalah "*Addresses*" karena setiap orang dapat memiliki banyak alamat kontak.



Gambar 2.7 Contoh dari Semantic Network (Feng et al, 2002)

Pada *node* yang kompleks, konten disebut konten kompleks jika *edge* menghubungkan antara dua *node*, dan masing-masing *edge* diberi label melalui hubungan mereka seperti *aggregation*, *association*, "of-property" dan *generalization*. Konten *node* kompleks dapat disajikan sebagai berikut (ditunjukkan pada Gambar 2.8).

"A connection of a node $n \in Node$ is an ordered pair $\langle l, n_0 \rangle$, where l is a label in $Lable$ and n_0 is a node in $Node$, representing that node n is connected to node n_0 via relation l "

Gambar 2.8 Konten dari Node Kompleks (Feng et al., 2002)

Contoh konten dari *node* kompleks: $PERSON$ is $\langle "a", \{Nama Depan, Nama Tengah, Nama Belakang, Alamat\} \rangle$.

b. Edge

Edge mewakili hubungan semantik (*semantic relationships*), yang sebenarnya merupakan hubungan antara dua *node*. Ada empat jenis link yang dijelaskan oleh Feng et al. (2002).

- a. *a generalization (g)* yang sering disebut "*is-a*" (adalah sebuah) hubungan yang menunjukkan warisan antara objek.
- b. *An association (s)* menentukan objek yang terhubung ke objek lain dengan memberikan asosiasi.
- c. *An aggregation (a)* merupakan sebuah "*part-of*" (bagian dari) hubungan dimana objek komposit ("*whole*") terdiri dari beberapa bagian objek komponen ("*parts*").
- d. Hubungan-properti (*of-property (p)*) menentukan atribut dari objek.

Contohnya,

Edge "BOOK \xrightarrow{p} *ISBN"* menyatakan BUKU (*BOOK*) memiliki properti *ISBN*.

Edge "ORDER \xrightarrow{s} *INVOICE"* menyatakan setiap pesanan (*ORDER*) dapat memiliki satu atau lebih faktur (*INVOICE*) terkait.

c. Constraint

Ada tiga jenis *constraint* yang sering terjadi dalam hubungan semantik (Feng et al, 2002):

- a. *constraint* melalui sebuah *edge*
- b. *constraint* melalui sekelompok *edge*
- c. *constraint* melalui sebuah *node*

Constraint melalui sebuah *edge* mendefinisikan kardinalitas, komposisi homogen / heterogen dan juga adhesi.

- a. *Constraint* Kardinalitas menentukan jumlah *instance* dari satu *node* yang berhubungan dengan sebuah *instance* dari *node* lain yang berhubungan seperti *node* "one" [1..1], "zero-or-one" [0..1] atau "zero-or-more"[0..N]. Misalnya, sebuah buku dapat berisi lebih dari satu bab, maka kardinalitas adalah [1..N].
- b. *Constraint* Komposisi Homogen terjadi jika "seluruh" objek terdiri dari "bagian" dari objek jenis yang sama seperti konten buku terdiri dari beberapa bab saja. Komposisi heterogen adalah kebalikan dari Komposisi homogen.
- c. *Constraint* Adhesi menentukan jika satu rekan yang ada dalam hubungan tersebut, apakah rekan lain harus hidup berdampingan. Jika keduanya harus hidup berdampingan, *constraint* disebut adhesi kuat, sebaliknya disebut adhesi lemah.

Constraint melalui sekelompok *edge* terdiri dari *ordered composition* dan *exclusive disjunction*

- a. *ordered composition* terjadi ketika sebuah objek "keseluruhan" terdiri dari "bagian" dalam urutan tertentu. Sebagai contoh, informasi *PERSON* mungkin muncul dalam urutan dimulai dengan *First Name*, diikuti oleh *Middle Name* dan berakhir dengan *Last Name*.
- b. *exclusive disjunction* terjadi ketika hanya ada satu sambungan *cluster* yang diperbolehkan pada satu waktu. Sebagai contoh, pengiriman barang yang dipesan hanya dapat dikirim oleh *Ship Company* atau *Air Company*.

Constraint melalui sebuah *node* terdiri dari *uniqueness constraint*, *referential integrity constraint*, dan *domain constraint*.

- a. *uniqueness constraint* diperlukan dalam rangka untuk memiliki nilai unik dari konten *node*. Sebagai contoh, properti ISBN harus memiliki nilai yang berbeda.
- b. *Referential integrity constraint* membutuhkan sebuah *node* referensi pada *node* referensi lain yang terkait seperti properti *OrderID* dari *INVOICE* harus sesuai dengan *OrderID* dari *ORDER*.
- c. *domain constraint* dapat terjadi karena rangr dari nilai (max / min) yang dibatasi, batasan nilai-nilai string atau pra-menentukan jumlah anggota (daftar nilai).

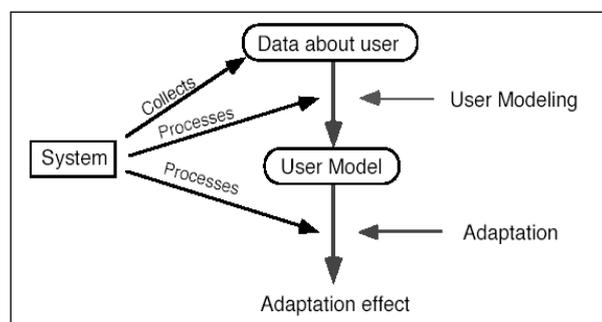
2.2.4.2 User Model

Menurut Koch (2000), *user model* adalah informasi tentang pengguna. "Tanpa *user model*, sistem tidak bisa membedakan antara pengguna yang berbeda dan akan memperlakukan semua pengguna dengan cara yang sama" (Kavcic, 2001). Pengguna berbeda dalam banyak hal seperti memiliki latar belakang yang berbeda, pengalaman, pengetahuan, preferensi, minat dan lain-lain. Kesimpulannya *user model* dibutuhkan untuk sebuah sistem yang berfungsi membuat pilihan respon untuk individualisasi dan personalisasi pengguna (Koch, 2000).

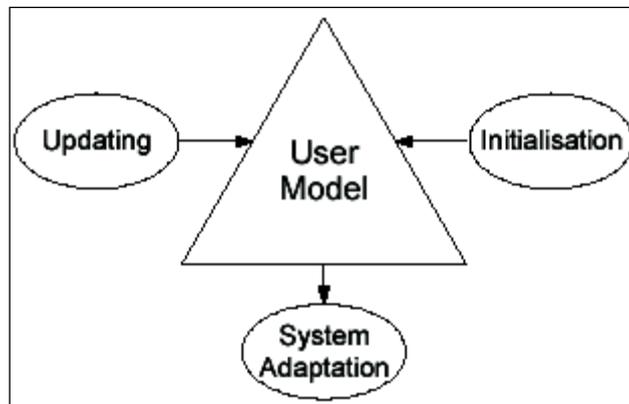
User Model (UM) biasanya dikenal sebagai personalisasi, rekomendasi, penyaringan atau sistem adaptif, yang memberikan pengalaman berbeda untuk pengguna individu (Flor, 2004). *User model* meningkatkan kontrol pengguna dan

pengalaman pengguna dengan menyediakan kemampuan untuk menyaring konten didasarkan pada kepentingan pribadi, menyesuaikan konten dan dalam beberapa kasus navigasi (Flor, 2004). Pentingnya menambahkan kemampuan ini ke sistem informasi ini dibuktikan dalam berbagai bidang, pemodelan pengguna telah diterapkan seperti: pencarian informasi, penyaringan dan sistem ekstraksi, antarmuka pengguna adaptif, dan perangkat lunak pendidikan (Papatheodorou, 2001).

User model berisi semua informasi tentang pengguna yang digunakan untuk menentukan daya adaptasi sistem (Kavcic, 2000). Proses "yang mencakup seluruh siklus hidup *user model* termasuk akuisisi pengetahuan tentang pengguna, konstruksi, pembaruan, pemeliharaan dan eksploitasi pengguna" dikenal sebagai pemodelan pengguna (*user modelling*) (Koch, 2000). Pemodelan pengguna memerlukan teknik untuk membangun suatu *user model*. Gambar 2.9 menunjukkan perulangan klasik "adaptasi pengguna-model" dalam sistem adaptif, sedangkan Gambar 2.10 menunjukkan tiga proses yang terlibat dalam pengembangan *user model* - inialisasi *user model*, memperbarui *user model* dan adaptasi sistem.



Gambar 2.9 Perulangan klasik “*user modelling-adaptation*” dalam Sistem Adaptif (Brusilovsky, 1996)



Gambar 2.10 Tiga Proses yang Terlibat dengan User Model (De Vrieze et al, 2004)

Menurut Kavcic (2000), ada tiga aspek utama yang perlu dipertimbangkan untuk mengembangkan atau membangun *user model*:

1. Informasi yang diperlukan untuk mengembangkan *user model* dan bagaimana informasi tersebut diperoleh.
2. Representasi informasi tersebut dalam sistem.
3. Proses pembentukan dan memperbarui *user model* atau dengan kata lain untuk memelihara *user model*.

Menurut Kavcic (2000), informasi tentang pengguna dapat bersifat statis dan dinamis. Informasi statis biasanya dikumpulkan pada awal proses pembelajaran, sedangkan informasi dinamis biasanya dikumpulkan langsung dari tindakan pengguna (misalnya penggunaan fitur antarmuka dan membaca petunjuk tentang mereka, aspek-aspek perilaku yang mencerminkan kapasitas memori kerja yang tersedia), tanggapan untuk item ujian atau praktek (misalnya jawaban untuk pertanyaan ujian ditangani secara individual atau kolaborasi, jawaban untuk

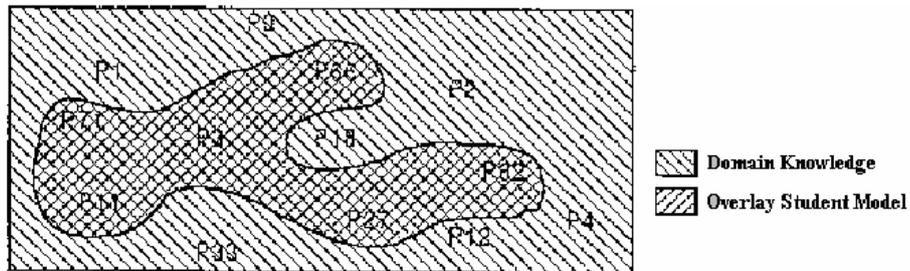
pertanyaan ujian, jawaban untuk item ujian dalam sistem pengajaran) dan sebagainya.

1. informasi statis dapat terdiri dari preferensi pengguna, minat, sikap, karakteristik pribadi pengguna (umur, jenis kelamin, kelas, dan lain-lain), kemampuan pengguna (latar belakang pengetahuan, keahlian) dan sebagainya (Kavcic, 2000).
2. informasi dinamis dapat terdiri dari informasi tentang interaksi pengguna dengan sistem seperti pengetahuan pengguna, konsep, keterampilan, gaya belajar, tujuan, keyakinan, tujuan yang telah dicapai dan sebagainya (Kavcic, 2000).

Salah satu metode umum untuk mengembangkan *user model* atau *student model* yaitu: *overlay model* dan *stereotypes*. *Overlay model* menyajikan pengetahuan pengguna sebagai bagian dari pengetahuan sistem seperti yang ditunjukkan pada Gambar 2.11. *Overlay* adalah teknik yang paling sering digunakan untuk memodelkan mahasiswa (Jeremic et al, 2004) dan sering digunakan untuk mengukur tingkat pengetahuan mahasiswa (Ozdemir & Alpaslan, 2000).

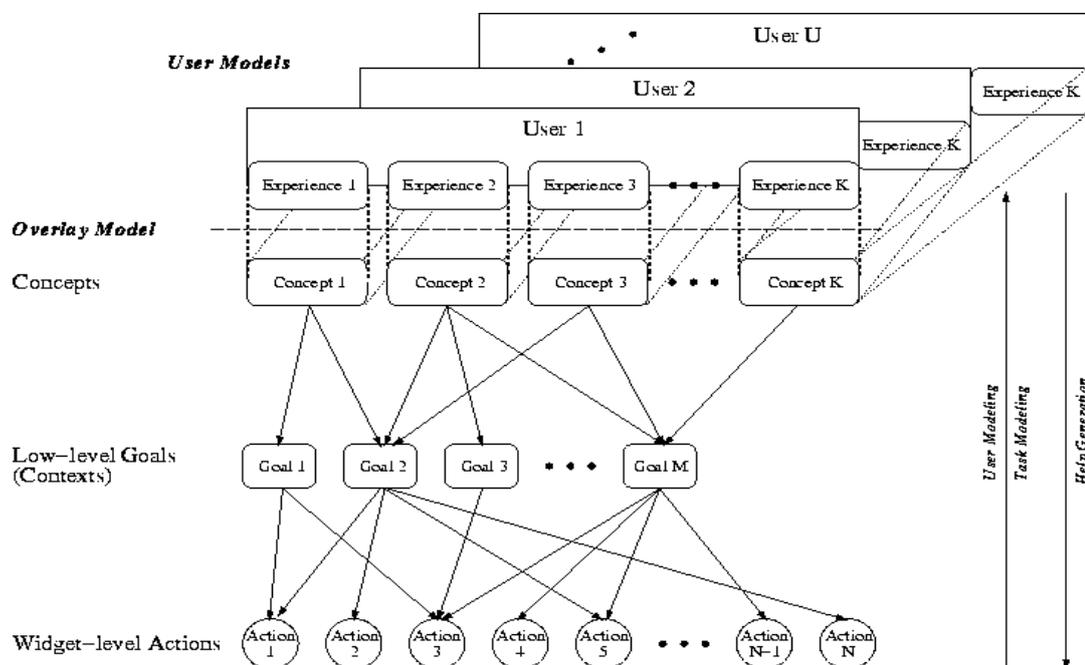
Menurut Abdullah (2003) dalam *overlay model*, representasi pengetahuan benar-benar terstruktur baik untuk *student model* maupun *domain model*. Teknik representasi pengetahuan dapat mencakup representasi berbasis aturan (*rule based representation*) dan jaringan semantik (*semantic network*). Albi (2001) berpendapat bahwa dalam *overlay model*, subjek dapat direpresentasikan sebagai hirarki prasyarat, yang dapat terdiri dari sejumlah konsep dan dalam konsep juga

dapat terdiri dari prasyarat lain. *Overlay model* menunjukkan kemajuan mahasiswa terhadap pengetahuan subjek.



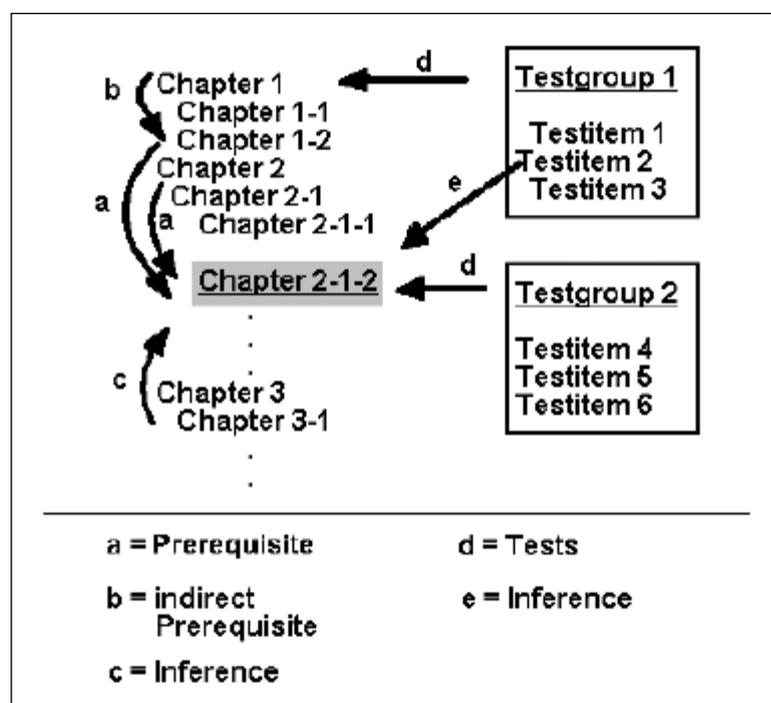
Gambar 2.11 *Overlay Student Model* (Albi, 2001)

Gambar 2.12 menunjukkan tingkatan tertinggi dari konsep *overlay*. "Prinsip utama dari *overlay model* adalah bahwa untuk setiap konsep domain, pengetahuan pengguna individu menyimpan beberapa data, yang merupakan estimasi tingkat pengetahuan pengguna dari setiap konsep domain" (Brusilovsky, 2003).



Gambar 2.12 *Overlay Model dan Hirarki dasar dari Konsep, Tujuan Tingkat Bawah dan Aksi* (Encarnacao, 1996)

Kebanyakan sistem menggunakan pengukuran kualitatif, yang membedakan tingkat pengetahuan pengguna berdasarkan bobot. Pengetahuan pengguna dari *overlay model* yang terukur dapat dilihat sebagai pasangan dari "nilai-konsep", yang diwakili oleh satu pasang untuk setiap konsep domain (Brusilovsky, 2003) seperti ditunjukkan pada Gambar 2.13.



Gambar 2.13 Contoh hubungan dari konsep (Weber et al., 2001)

Overlay model, Domain pengetahuan dibagi menjadi topik-topik tertentu atau dikenal sebagai 'konsep', sedangkan pengetahuan mahasiswa dibangun berdasarkan konsep-per-konsep. Konsep-per-konsep biasanya yang dipetakan ke topik atau konsep domain tertentu. Setelah mahasiswa melalui sistem tersebut, hasilnya akan diperbarui dengan sistem (Conlan, 2000).

Permasalahan untuk menginisialisasi *user model* baik secara individual maupun kelompok sering diselesaikan dengan menerapkan teknik *stereotype* (Tsiriga & Virvou, 2002). *Stereotype* sebenarnya adalah sebuah teknik yang *powerful* (Tsiriga & Virvou, 2002) dan banyak digunakan dalam pemodelan pengguna (Elsevier, 2004). *Stereotype* terdiri dari atribut umum dari kelompok pengguna yang biasanya menggambarkan kepentingan pengguna (Kavcic, 2000). stereotip Masing-masing memiliki properti standar sendiri, dan pengguna yang dikenakan *stereotype* mewarisi properti tersebut (Kavcic, 2000).

Stereotype biasanya menjaga atau menyimpan karakteristik pengguna umum (Elsevier, 2004) atau pengetahuan umum tentang sekelompok pengguna (Elsevier, 2004). Pengguna individu diberikan satu atau lebih *stereotype* (seperti pemula, menengah, lanjut atau ahli). Setiap pengguna dapat diberikan ke lebih dari satu kelompok *stereotype* (Kavcic, 2000). Ada dua jenis informasi yang perlu diambil dengan menggunakan teknik *stereotype*:

1. sistem harus mengetahui informasi, atribut atau properti apa untuk menangkap *stereotype*,
2. skenario, peristiwa atau perilaku apa yang mencerminkan atau menyiratkan sebuah *stereotype* tertentu (Sollenborn, 2004).

2.2.4.3 Adaptation Model (model adaptasi)

Adaptation Model (model adaptasi) mendefinisikan dalam sistem sebagai berikut (Paramythis et al., 2004):

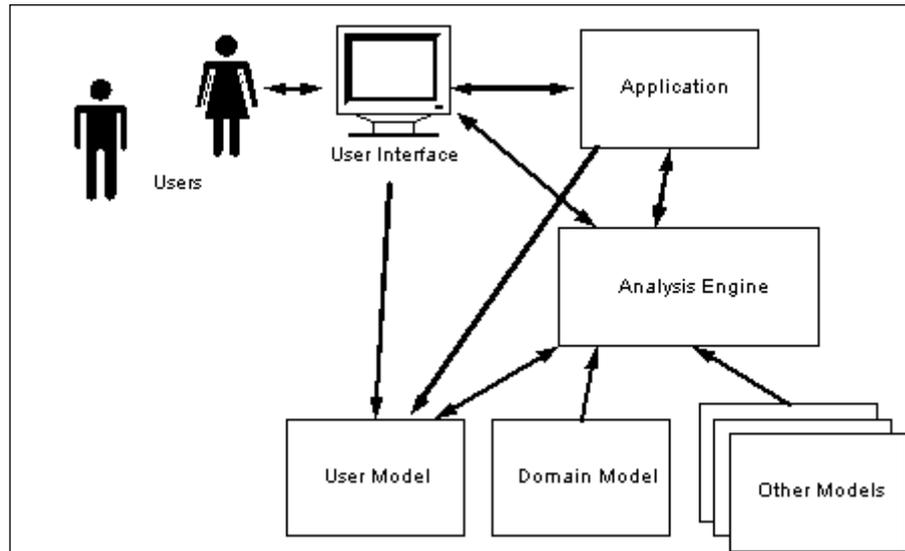
1. Apa yang dapat diadaptasi ?
2. Kapan diadaptasi ?

3. Bagaimana hal tersebut diadaptasi ?

Adaptation model dapat terdiri dari informasi untuk menjawab "Apa yang dapat diadaptasi" seperti di bawah ini (Lalos et al., 2005):

1. Pertanyaan-pertanyaan yang harus dipertimbangkan mudah atau sulit,
2. Grade untuk menafsirkan pengetahuan peserta didik,
3. Berapa banyak pertanyaan yang diperlukan untuk memperkirakan pengetahuan peserta didik dengan keyakinan, dan
4. Bagaimana kinerja peserta didik akan mempengaruhi model pembelajar seperti mencoba menjawab lagi, dan permintaan peserta didik untuk sebuah petunjuk.

Pertanyaan "Kapan diadaptasi" dan "Bagaimana hal tersebut diadaptasi" dapat dijawab yaitu membutuhkan adaptasi untuk melakukan proses pengolahan tambahan, dengan kata lain, adaptasi tidak dapat dilakukan, dengan menyimpan dan mengambil kejadian atau data secara langsung dari database (De Vrieze et al., 2004). Berdasarkan hal tersebut model adaptasi terdiri dari serangkaian aturan yang mendefinisikan tindakan pengguna dan juga memicu tindakan yang menentukan hasil dari aksi pengguna, yang kemudian perlu diperbarui dalam model pengguna (De Bra, 2001). Gambar 2.14 menunjukkan salah satu contoh sistem adaptif yang disebut "sistem pola dasar" (*archetypal system*) yang menggunakan mesin analisis untuk melaksanakan adaptasi yang menggabungkan profil pengguna dengan model lain dari sistem untuk menurunkan "fakta" baru tentang pengguna (Kules, 2000).



Gambar 2.14 Sebuah Sistem Pola Dasar (*archetypal system*) memanfaatkan sebuah user model (Kules, 2000)

2.2.5 Kuis Adaptif

Kuis berbasis web berfungsi sebagai alat utama untuk penilaian diri mahasiswa (Brusilovsky & Sosnovsky, 2005), dengan menambahkan fitur *adaptif*, kuis yang disajikan kepada mahasiswa menjadi lebih personal karena ia akan menyesuaikan tingkat kesulitan pertanyaan berdasarkan keterampilan mahasiswa. Pertanyaan yang diberikan kepada mahasiswa lebih sulit atau mudah tergantung pada jawaban yang diberikan oleh mahasiswa. Melalui pendekatan ini, pertanyaan yang disajikan kepada mahasiswa disesuaikan dengan tingkat performa mereka. Melalui cara ini, tidak ada waktu yang terbuang untuk seorang mahasiswa tingkat lanjut jika pertanyaan yang terlalu dasar diberikan, sedangkan untuk pemula, pertanyaan-pertanyaan yang disajikan membantu mereka untuk meningkatkan dalam pembelajaran mereka.

2.2.6 Content Management System

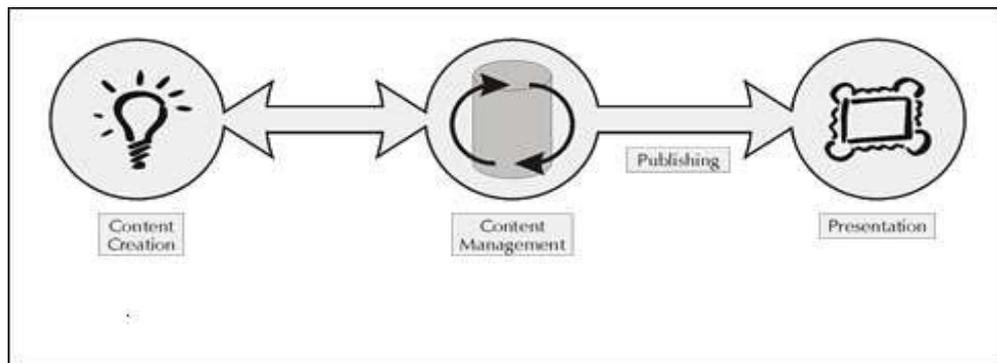
Content Management System (CMS) adalah sebuah perangkat lunak yang menyediakan *tools* / perangkat untuk pembuatan, penerbitan/*publishing* maupun pengelolaan *website* (Robertson, 2003) Sebuah *website* di dalamnya terdapat konten/isi yang berisi informasi atau pengetahuan yang disajikan untuk pengguna. Konten merupakan suatu unit informasi digital yang digunakan untuk mengisi sebuah webpage (TreeWorks, 2005). Konten yang dimaksud bisa meliputi file-file elektronik, gambar-gambar, file-file audio, video atau dokumen-dokumen elektronik.

CMS banyak digunakan di internet , baik untuk *website* pribadi maupun umum. Beberapa contoh penggunaan CMS untuk pribadi adalah untuk personal *website* dan *blog*. Contoh lain penggunaan CMS adalah untuk website perusahaan, organisasi, institusi pendidikan atau komunitas tertentu, portal, galeri foto, dan aplikasi e-commerce serta aplikasi e-learning.

Saat ini, terdapat beberapa CMS yang bisa digunakan. Beberapa diantaranya ada yang bersifat komersial seperti *stellent*, *vignette*, *documentum* dan lainnya. Sebagian pengguna memanfaatkan CMS yang bersifat *open source* karena keuntungan – keuntungan yang dimiliki oleh perangkat lunak *open source*. Beberapa diantara software CMS tersebut adalah *joomla*, *mambo*, *drupal*, *wordpress*, *phpnuke*, *impresscms*, *typo3*, *zope*, *wordpress* dan *xoops*.

2.2.6.1 Anatomi Content Management System

Anatomi *content management system* (CMS) dapat menggambarkan aktivitas–aktivitas yang bisa dilakukan berkaitan dengan fungsi yang disediakan oleh suatu CMS. Anatomi CMS yang dapat dilihat pada Gambar 2.15 di bawah ini.



Gambar 2.15 Anatomi Content Management System (Robertson, 2003)

Pada Gambar 2.15, terdapat 3 lingkaran yang masing-masing lingkaran mewakili penciptaan konten, manajemen konten dan presentasi konten. CMS menyediakan lingkungan pembuatan konten memudahkan pengguna sehingga dapat digunakan oleh seseorang yang bahkan tidak memiliki keahlian dalam pemrograman web. Konten yang telah dibuat akan disimpan di tempat penyimpanan tertentu dalam basis data CMS dan CMS menyediakan fasilitas manajemen konten untuk mengelola. Presentasi merupakan bagian terakhir dari anatomi CMS, dimana bagian ini bertugas mengatur format tampilan konten.

2.2.6.2 Ekstensi Content Management System

Salah satu keuntungan dari *content management system* (CMS) adalah adanya fasilitas untuk menambah fungsionalitas pada CMS.

Kemampuan atau fungsionalitas tambahan pada CMS ini dikenal dengan istilah ekstensi (extension) (Compass Design Team, 2005). Ekstensi merupakan suatu perangkat lunak tambahan pada CMS yang di-*install* pada CMS dan berguna untuk menambah fungsionalitas pada CMS tersebut

Ekstensi merupakan perangkat lunak yang tidak bisa berdiri sendiri. Ekstensi harus di-*install* pada perangkat lunak yang akan ditambah fungsionalitasnya, dalam hal ini pada aplikasi web yang menggunakan perangkat lunak CMS yang sesuai dengan ekstensi yang akan ditambahkan. Ekstensi akan berintegrasi dengan perangkat lunak CMS dan memberikan layanan atau fungsi yang bisa dinikmati oleh pengguna atau pengunjung website tempat dimana ekstensi tersebut di- *install*.

Salah satu bentuk ekstensi pada CMS adalah komponen. Komponen memungkinkan pengembang CMS menambahkan fungsionalitas pada CMS dengan cara yang lebih mudah, diantaranya karena sudah adanya file-file library yang biasa digunakan untuk aplikasi web, sudah tersedianya basis data yang bisa dimanfaatkan serta toolbar-toolbar yang bisa digunakan.

2.2.7 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem informasi berarti suatu metode yang digunakan untuk melakukan pengembangan sistem informasi berbasis komputer (Supriyanto, 2005). Metode yang paling umum digunakan adalah siklus pengembangan sistem (*system development life cycle – SDLC*).

Tahap-tahap dalam pengembangan sistem sesuai dengan SDLC meliputi tahapan sebagai berikut (Supriyanto, 2005) :

1. Perencanaan sistem (*system planning*) / Pengumpulan kebutuhan

Tahap perencanaan adalah tahap awal pengembangan sistem yang mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi), dan anggaran yang sifatnya masih umum (belum detail/rinci).

2. Analisis sistem (*system analyst*)

Tahap analisis sistem adalah tahap penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui. Tahap ini merupakan tahap kritis dan sangat penting, karena akan dapat menentukan berhasil tidaknya sistem yang akan dibangun atau dikembangkan. Kesalahan pada tahap analisis akan menjadikan kesalahan pada tahap-tahap yang selanjutnya.

3. Desain/perancangan sistem (*system design*)

Tahap desain sistem adalah tahap setelah analisis sistem yang menentukan proses dan data yang diperlukan oleh sistem baru. Beberapa pakar berpendapat bahwa desain sistem dibedakan menjadi dua macam, yaitu desain sistem umum dan desain sistem terinci. Desain sistem umum berupa desain konseptual atau desain logis, sedangkan desain sistem terinci berupa desain fisik atau desain internal.

4. Penerapan/implementasi sistem (*system implementation*)

Tahap implementasi atau penerapan adalah tahap desain sistem dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.

5. Perawatan sistem (*system maintenance*)

Tahap pemeliharaan merupakan tahap yang dilakukan setelah tahap implementasi, yang meliputi pemakaian atau penggunaan, audit sistem, pemeliharaan, perbaikan, dan peningkatan sistem.

Penelitian ini hanya sampai pada tahap keempat yaitu tahap penerapan atau implementasi, untuk tahap perawatan tidak dilakukan dikarenakan tahap ini membutuhkan waktu yang cukup lama.

BAB III

CARA PENELITIAN

3.1 Bahan Penelitian

Bahan penelitian yang digunakan dalam proses penelitian ini antara lain sebagai berikut :

3.1.1 Data Sekunder

Data yang diperoleh melalui studi pustaka, jurnal, proceeding, artikel, buku, dan website yang terkait dengan permasalahan kuis (termasuk kuis tradisional, kuis online dan kuis adaptif), penilaian adaptif (*adaptive assessment*), komponen sistem adaptif, ekstensi *content management system*, analisis, pemrograman, dan teknologi yang digunakan.

3.1.2 Knowledge

Knowledge merupakan suatu persepsi yang jelas terhadap sesuatu hal yang diperoleh melalui dokumentasi-dokumentasi pengetahuan (*knowledge*) tentang komponen sistem adaptif yaitu *domain model*, *student model*, dan *adaptation model*.

1. Domain Model

Domain model digunakan untuk memberikan struktur pengetahuan domain, selain itu, digunakan untuk memberikan catatan contoh atau acuan bagi peserta didik untuk melakukan revisi tanpa perlu bagi mereka untuk merujuk ke buku secara manual. Teknik yang digunakan adalah dengan menerapkan *Semantic*

Network (Jaringan Semantik) dengan membagi pengetahuan domain menjadi topik atau yang dikenal sebagai "konsep".

2. Student Model

User model merupakan sistem keyakinan tentang pengguna yang menggambarkan pengetahuan pengguna. *User model* dalam penelitian ini mengacu pada *student model*. *Student model* menggambarkan pengetahuan peserta didik dari *domain model*. Melalui *student model*, sistem memperlakukan cara yang berbeda untuk setiap peserta didik. *Student model* berisi semua informasi yang sistem ketahui tentang peserta didik. Ada dua jenis informasi yang diperlukan yaitu informasi statis dan dinamis. Informasi statis mungkin berisi profil peserta didik seperti nama peserta didik, alamat, nomor identifikasi, jenis kelamin, dan nomor kontak. Informasi yang dinamis mungkin berisi informasi tentang interaksi pengguna dengan sistem seperti tingkat kesulitan pengetahuan peserta didik saat ini dan sejarah tingkat kesulitan pengetahuan peserta didik, tanggal kuis dikerjakan, nilai kuis dan subyek kuis yang dikerjakan. Informasi tersebut diperoleh dengan menggunakan *overlay model* dan teknik *stereotype*.

3. Adaptation Model

Adaptation model yang digunakan dalam penelitian ini, menggunakan aturan IF-THEN. Aturan tersebut diterapkan untuk menangkap informasi mengenai:

- a. berapa banyak pertanyaan yang akan disajikan dalam kuis,
- b. setup awal untuk akses peserta didik pertama kali,

- c. saran atau informasi umpan balik, jawaban yang benar, jawaban yang salah, bimbingan,
- d. skor per kuis,
- e. grade untuk menafsirkan pengetahuan peserta didik, dan
- f. pertanyaan mana yang dianggap mudah atau sulit.

3.2 Alat Penelitian

Alat penelitian yang digunakan dalam proses penelitian ini antara lain sebagai berikut :

1. Perangkat Keras berupa satu unit komputer dengan sistem operasi Windows
2. Perangkat Lunak berupa Rational Rose, XAMPP (web server Apache, Script PHP, mySQL), dan web browser.

3.3 Jalan Penelitian

Penelitian ini menerapkan metodologi *SDLC (System Development Life Cycle)* dengan menggunakan pendekatan berorientasi objek. Pengembangan yang dilakukan yaitu *iterative* dan *incremental*. Tahapan-tahapan penelitian adalah sebagai berikut :

3.3.1 Pengumpulan Kebutuhan

Tahap ini mendefinisikan kebutuhan sistem apa yang harus dilakukan atau bagaimana seharusnya melakukan. Ada dua jenis kebutuhan yang akan diidentifikasi. Jenis pertama adalah kebutuhan fungsional yang menangkap apa yang sistem lakukan. Jenis kedua adalah kebutuhan non-fungsional yang

menangkap kendala pada sistem itu sendiri, serta kendala yang dihadapi sistem dalam tahap pengembangan.

3.3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem kuis *online adaptif* ini berasal dari kerangka penelitian yang sedang dijelaskan dan disesuaikan untuk memenuhi target pengguna yaitu institusi pendidikan. Berikut ini adalah kebutuhan fungsional yang dibutuhkan sistem kuis *online adaptif*:

1. Sistem ini akan mendukung tiga jenis pengguna - Administrator, pengajar dan mahasiswa.
2. Sistem harus mendukung tiga model adaptif utama – *student model*, *domain model* dan *adaptation model*. Teknik yang diterapkan adalah teknik *Overlay* dan *Stereotype* untuk *student model*, sedangkan untuk *domain model*, sistem menerapkan teknik *Semantic Network* dan untuk *adaptation model*, sistem menerapkan *IF-THEN rule*.
3. Sistem harus mendukung tiga jenis pertanyaan yaitu pertanyaan pilihan ganda dengan jawaban tunggal (*multiple choice*), pertanyaan benar atau salah (*true/false*), dan pertanyaan mengisi bagian yang kosong (*fill in the blank*).
4. Sistem ini harus menyediakan editor yang akan memudahkan pengajar untuk membuat pertanyaan.
5. Sistem ini akan memungkinkan pengajar untuk mengelompokkan tingkat kesulitan domain pengetahuan yang sama di bagian yang sama.
6. Sistem ini akan memungkinkan pengajar untuk menetapkan sejumlah pertanyaan yang harus disajikan dalam satu sesi kuis.

7. Pertanyaan yang diajukan kepada mahasiswa akan berasal dari tingkat kesulitan domain pengetahuan yang sama.
8. Sistem ini dikembangkan untuk mahasiswa perguruan tinggi.
9. Sistem hanya akan memungkinkan pengguna untuk mengakses melalui PC, bukan dari perangkat lain.
10. Sistem harus menyediakan informasi umpan balik dimana oleh sistem merekomendasikan bahan yang relevan bagi mahasiswa untuk merevisi jika mereka memberikan jawaban yang salah, hasil kuis dan tingkat kesulitan berikutnya direkomendasikan bagi para mahasiswa.
11. Sistem harus mendukung fungsi pendaftaran untuk pengajar dan mahasiswa.
12. Sistem hanya akan memungkinkan pengguna terdaftar untuk menggunakan sistem.
13. Sistem ini akan memungkinkan mahasiswa untuk melihat catatan hasil mereka.
14. Sistem ini akan memungkinkan pengajar untuk melihat catatan hasil mahasiswa.
15. Sistem ini akan memungkinkan pengajar untuk membuat konten domain pengetahuan.
16. Sistem ini akan memungkinkan pengajar untuk mengkonfigurasi kuis seperti jumlah pertanyaan yang disajikan per kuis, tingkat awal waktu akses pertama.
17. Pertanyaan harus disajikan secara acak diambil dari bank pertanyaan.
18. Pertanyaan yang sama tidak akan ditampilkan dua kali dalam kuis sesi yang sama.

19. Sekelompok pertanyaan yang tidak sama disajikan untuk masing-masing mahasiswa yang mengambil kuis pada waktu dan tingkat kesulitan domain pengetahuan yang sama.
20. Sistem ini akan memungkinkan pengajar untuk mengelola aturan untuk kuis.
21. Sistem ini akan memungkinkan pengajar untuk mengelola informasi subjek dalam sistem.

3.3.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional atau batasan sistem adalah sebagai berikut:

Browser

Sistem kuis *online adaptif* ini berbasis web, oleh karena itu, klien diperlukan untuk mengakses sistem ini melalui web browser seperti *Microsoft Internet Explorer* atau *Mozilla Firefox*.

Pemeliharaan dan Perluasan

Struktur untuk mengembangkan sistem harus mudah dipelihara dan dapat diperluas setiap saat tanpa mempengaruhi struktur saat ini dan yang paling penting adalah data atau informasi yang tersimpan.

Kegunaan

Sistem seharusnya memahami jenis pengguna yang akan menggunakan sistem. Seperti yang telah disebutkan sebelumnya, ada tiga jenis pengguna (administrator, pengajar dan mahasiswa). Dengan demikian, sistem akan mempertimbangkan tingkat yang berbeda dari pengguna dan menyediakan *tool* untuk mendukung pengguna yang bersangkutan.

Skalabilitas

Sistem harus mendukung skalabilitas yaitu harus mudah untuk memodifikasi komponen atau sistem sesuai dengan masalahnya.

Efisiensi

Menjadi aplikasi berbasis web, sistem harus dapat memberikan respon yang baik paling sedikitnya 3 detik.

3.3.2 Analisis Sistem

Pada tahap analisis sistem, masing-masing kebutuhan fungsional dan non-fungsional dianalisis secara rinci. Teknik analisis berorientasi objek telah dipilih untuk mengidentifikasi lebih lanjut kebutuhan sistem yang di rinci dengan menggunakan UML (*Unified Modeling Language*). Kebutuhan sistem dijabarkan dalam bentuk notasi grafis dengan menggunakan UML. Setiap *use case* dikaitkan dengan aktor, sehingga setiap *use case* menyatakan tujuan aktor dalam menggunakan sistem Hal ini juga menunjukkan interaksi antara aktor dan sistem.

Untuk memudahkan pemahaman dan kesederhanaan, hanya *use case* utama untuk membangun fungsionalitas sistem yang akan dijelaskan, *use case* yang lain dijelaskan secara lengkap dalam Lampiran A.

3.3.2.1 Identifikasi Aktor

Aktor adalah orang atau pihak eksternal yang berinteraksi dan memiliki peran dalam sistem. Seperti disebutkan dalam kebutuhan sistem, sistem harus mendukung tiga jenis pengguna - Administrator, Pengajar dan Mahasiswa, ketiga pengguna sesungguhnya adalah aktor sistem kuis *online adaptif* . Bagian-bagian

di bawah ini mengidentifikasi aktor untuk sistem kuis *online adaptif*.

1. Mahasiswa

Seorang mahasiswa mengacu pada seseorang yang belajar di institusi pendidikan yang mengimplementasikan sistem kuis *online adaptif* dan memiliki hak untuk mengakses sistem dalam rangka untuk mengambil kuis adaptif., dalam penelitian ini yang dimaksud mahasiswa adalah mahasiswa sebuah perguruan tinggi.

2. Pengajar

Seorang pengajar mengacu pada orang yang mengajar di institusi pendidikan yang mengimplementasikan sistem kuis *online adaptif* dan diberi ijin untuk mengakses sistem. pengajar bertanggung jawab untuk membuat, update dan menghapus pertanyaan dalam sistem kuis *online adaptif*. Pengajar juga bertanggung jawab untuk memberikan struktur dan isi pengetahuan domain. Selain itu, ia juga bertanggung jawab untuk menyediakan aturan untuk tujuan *adaptation*, dalam penelitian ini yang dimaksud pengajar adalah dosen yang mengampu sebuah mata kuliah.

3. Administrator

Administrator mengacu pada orang yang ditugaskan atau diberi tanggung jawab untuk memantau dan mengelola sistem kuis *online adaptif*.

3.3.2.2 Identifikasi Use Case

Kebutuhan sistem dalam sistem kuis *online adaptif* dalam sub bab sebelumnya akan diwujudkan menjadi *use case* untuk melihat sistem dari sudut pandang pengguna. Pada tahap ini, faktor yang paling penting adalah untuk

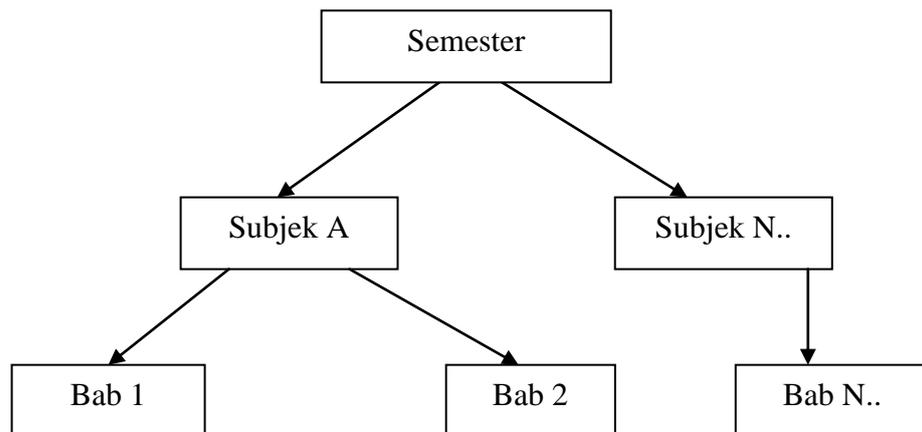
mengidentifikasi *use case* untuk komponen *adaptif* sistem kuis *online adaptif*, sebagai komponen (*student model*, *domain model* dan *adaptation model*) yang merupakan komponen utama dari sistem kuis *online adaptif* ini. Tahap selanjutnya, kebutuhan sistem yang ada digambarkan sebagai *use case*. Seperti yang disebutkan sebelumnya, ada tiga model adaptif utama yang digunakan dalam sistem kuis *online adaptif* yaitu - *domain model*, *student model*, dan *adaptation model*.

3.3.2.2.1 Domain Model dari Sistem Kuis Online Adaptif

Seperti yang disebutkan dalam bab sebelumnya, *domain model* merupakan subjek yang diajarkan. *Domain model* yang digunakan untuk menyediakan struktur pelaksanaan sistem kuis *online adaptif*. Pengetahuan domain dikembangkan berdasarkan subjek yang diajarkan kepada mahasiswa seperti Algoritma dan Pemrograman, Struktur Data, Sistem Operasi dan mata kuliah yang lain. Sistem kuis *online adaptif* ini mendukung semua jenis subjek yang menyesuaikan dengan Semester dari subjek tersebut. Masing-masing mata kuliah untuk semester tertentu seperti Algoritma dan Pemrograman untuk semester ganjil, Struktur Data untuk semester genap, dan mata kuliah yang lainnya dianggap sebagai pengetahuan domain yang berbeda. Dengan demikian, sistem kuis *online adaptif* ini memungkinkan banyak pengetahuan domain untuk dibuat dalam sistem.

Struktur pengetahuan domain dibentuk dalam rangka untuk mendukung semua jenis struktur pengetahuan domain, maka peneliti mendefinisikan struktur umum sebagai berikut seperti yang ditunjukkan pada Gambar 3.1:

- a. Semester pada sebuah institusi pendidikan, mengacu pada Semester Gasal dan Semester Genap.
- b. Masing-masing semester, terdiri dari setidaknya satu atau banyak mata kuliah (subjek).
- c. Setiap subjek dapat terdiri dari setidaknya satu atau banyak bab.



Gambar 3.1 Struktur Umum dari Pengetahuan Domain Sistem Kuis Online Adaptif

Untuk pemahaman yang lebih baik dan gambaran yang jelas penyajian *domain model* sistem kuis *online adaptif* ini, *Semantic Network* dipilih untuk mengimplementasikannya.

Seperti yang telah dijelaskan sebelumnya dalam Bab 2, *Semantic Network* terdiri dari sekumpulan *node* dan sekumpulan *link*. *Node* menyatakan konsep atau objek dunia nyata dan *link* digunakan untuk menyatakan hubungan semantik antara objek dan konsep.

Node

Sistem kuis *online adaptif* ini terdiri dari *node* dasar dan *node* kompleks. Gambar 3.2 menunjukkan *Semantic Network* dari sistem kuis *online adaptif*

a. Node Dasar

Node dasar terdiri dari ‘Tingkat Semester’, ‘Kode Mata Kuliah’, ‘Nama Mata Kuliah’, ‘Deskripsi Mata Kuliah’, ‘Nomor Bab’, ‘Nama Bab’, ‘Deskripsi Bab’. Konten dari *node* dasar sistem kuis *online adaptif* (‘Kode Mata Kuliah’, ‘Nama Mata Kuliah’, ‘Deskripsi Mata Kuliah’, ‘Nama Bab’, ‘Deskripsi Bab’) adalah tipe data *string*. Hanya konten *node* dasar ‘Nomor Bab’ saja yang bertipe data *integer*. ‘Kode Mata Kuliah’ adalah *node* dasar yang kontennya bersifat *constructional set value*.

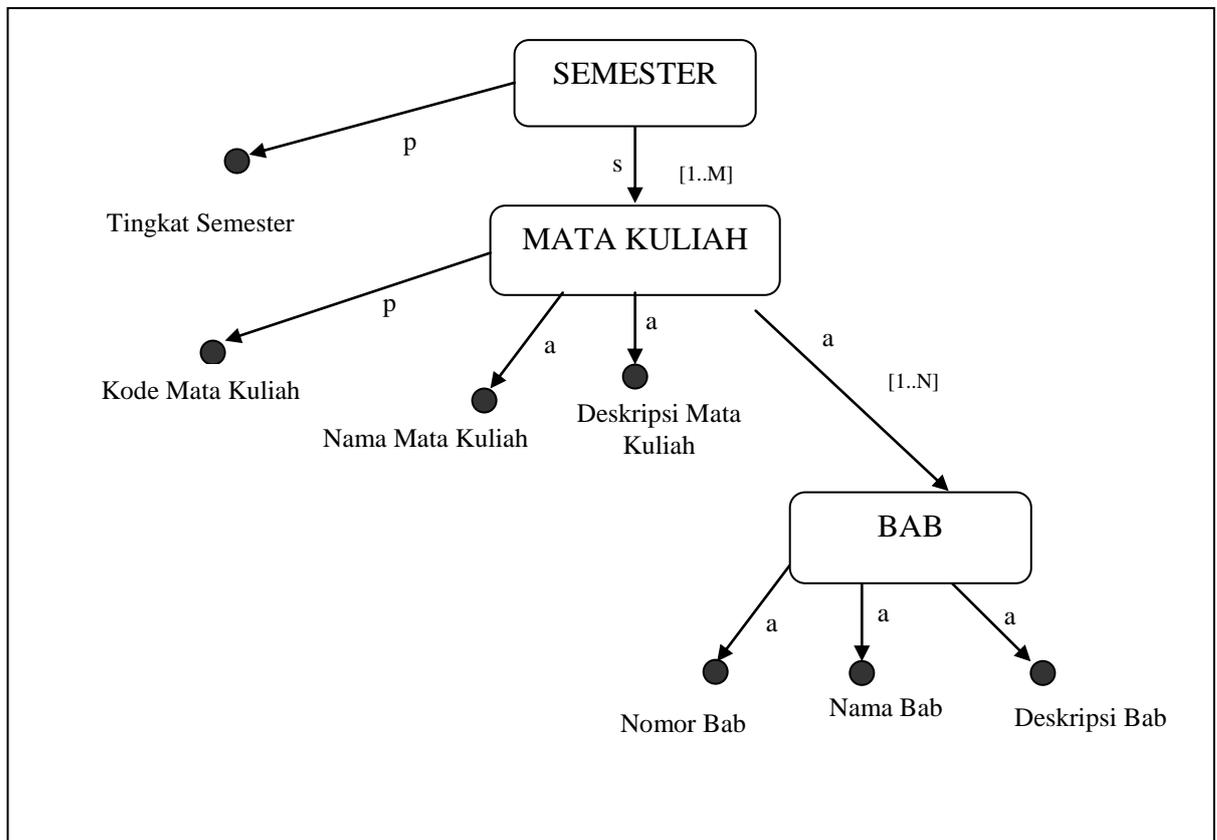
b. Node Kompleks

Node kompleks untuk sistem kuis *online adaptif* terdiri dari ‘Semester’, ‘Mata Kuliah’, dan ‘Bab’ dengan masing-masing *node* mempunyai satu atau lebih *edge* yang keluar.

1. Sebuah *node* kompleks yaitu SEMESTER terhubung ke *node* {MATA KULIAH} dengan *edge* keluar yang diberi label hubungan “s” (menunjukkan *association*) dan terhubung ke *node* {Tingkat Semester} dengan *edge* keluar yang diberi label hubungan “p” (menunjukkan *of-property*). Dengan demikian, konten dari *node* SEMESTER adalah {<“s”, MATA KULIAH>, <“p”, {Tingkat Semester}>}.
2. Sebuah *node* kompleks yaitu MATA KULIAH terhubung ke sekumpulan *node* {Nama Mata Kuliah, Deskripsi Mata Kuliah} dengan hubungan “a” (menunjukkan *aggregation*), dan terhubung ke *node* {Kode Mata Kuliah} dengan hubungan “p” (menunjukkan *of-property*). Dengan demikian,

konten dari *node* MATA KULIAH adalah {< “a”, { Nama Mata Kuliah, Deskripsi Mata Kuliah }>, < “p”, { Kode Mata Kuliah }>}

- Sebuah *node* kompleks yaitu BAB terhubung ke sekumpulan *node* {Nomor Bab, Nama Bab, dan Deskripsi Bab} dengan hubungan “a”. Oleh karena itu konten dari *node* BAB adalah {< “a”, { Nomor Bab, Nama Bab, Deskripsi Bab }>}



Gambar 3.2 Semantic Network dari Sistem Kuis Online Adaptif

Link

Tabel berikut ini akan menjelaskan *link* (hubungan antara *node*) untuk sistem kuis *online adaptif*

Tabel 3.1 Hubungan antara Node dari Sistem Kuis Online Adaptif

No	Edge	Deskripsi
1.	<i>edge</i> "SEMESTER" \xrightarrow{s} "MATAKULIAH"	Menentukan bahwa masing-masing SEMESTER mempunyai satu atau lebih MATA KULIAH
2.	<i>edge</i> "SEMESTER" \xrightarrow{p} "Tingkat Semester"	Menentukan bahwa SEMESTER mempunyai properti Tingkat Semester
3.	<i>edge</i> "MATAKULIAH" \xrightarrow{p} "Kode Mata Kuliah"	Menentukan bahwa MATA KULIAH mempunyai properti Kode Mata Kuliah
4.	<i>edge</i> "MATAKULIAH" \xrightarrow{a} "Nama Mata Kuliah"	Menandakan bahwa MATA KULIAH adalah sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah Nama Mata Kuliah
5.	<i>edge</i> "MATAKULIAH" \xrightarrow{a} "Deskripsi Mata Kuliah"	Menandakan bahwa MATA KULIAH adalah sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah Deskripsi Mata Kuliah
6.	<i>edge</i> "MATAKULIAH" \xrightarrow{a} "BAB"	Menandakan bahwa MATA KULIAH adalah

Lanjutan Tabel 3.2 **Hubungan antara Node dari Sistem Kuis Online Adaptif**

No	Edge	Deskripsi
		sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah BAB
7.	$edge \text{ "BAB" } \xrightarrow{a} \text{ "Nomor Bab" }$	Menandakan bahwa BAB adalah sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah Nomor Bab
8.	$edge \text{ "BAB" } \xrightarrow{a} \text{ "Nama Bab" }$	Menandakan bahwa BAB adalah sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah Nama Bab
9.	$edge \text{ "BAB" } \xrightarrow{a} \text{ "Deskripsi Bab" }$	Menandakan bahwa BAB adalah sebuah objek utuh (<i>whole</i>) dan salah satu bagian (<i>part</i>) objek tersebut adalah Deskripsi Bab

Constraint (Batasan)

Constraint berikut ini adalah *constraint* yang telah diidentifikasi untuk untuk Semantic Network sistem kuis *online adaptif*.

- SEMESTER dan MATA KULIAH memiliki hubungan adhesi yang kuat, karena MATA KULIAH mahasiswa hanya dapat dibedakan oleh SEMESTER dan tidak dapat ada tanpa SEMESTER tersebut.

7. Masing-masing SEMESTER harus mempunyai minimal satu atau lebih MATA KULIAH yang akan dipelajari oleh mahasiswa. Jadi kardinalitas [1 .. N] diterapkan.
8. Masing-masing MATA KULIAH, harus memiliki minimal satu atau lebih BAB. Jadi kardinalitas [1 .. N] diterapkan. Karena BAB dihasilkan dari subjek yang sama (jenis objek), maka hubungan ini disebut Komposisi Homogen (*Homogeneous Composition*).
9. Masing-masing MATA KULIAH dibedakan dengan Kode Mata Kuliah (*uniqueness*).

Berdasarkan implementasi *domain model* dengan *semantic network* tersebut, untuk mengembangkan *domain model* sistem kuis *online adaptif* ini peneliti mentransformasikan ke dalam bentuk *use case*. Terdapat tiga *use case* utama yang dapat diidentifikasi untuk mengembangkan *domain model* sistem kuis *online adaptif* ini, yaitu:

1. *Use Case* Mengelola Semester, digunakan untuk menangkap informasi semester mahasiswa.
2. *Use Case* Mengelola Mata Kuliah, digunakan untuk menambah, memperbarui, menghapus, dan melihat tiap mata kuliah berdasarkan semester mahasiswa tersebut. Pengguna diijinkan untuk membuat setidaknya satu atau lebih mata kuliah untuk setiap semester.
3. *Use Case* Mengelola Bab, digunakan untuk menambah, memperbarui, menghapus, dan melihat bab untuk masing-masing mata kuliah yang telah

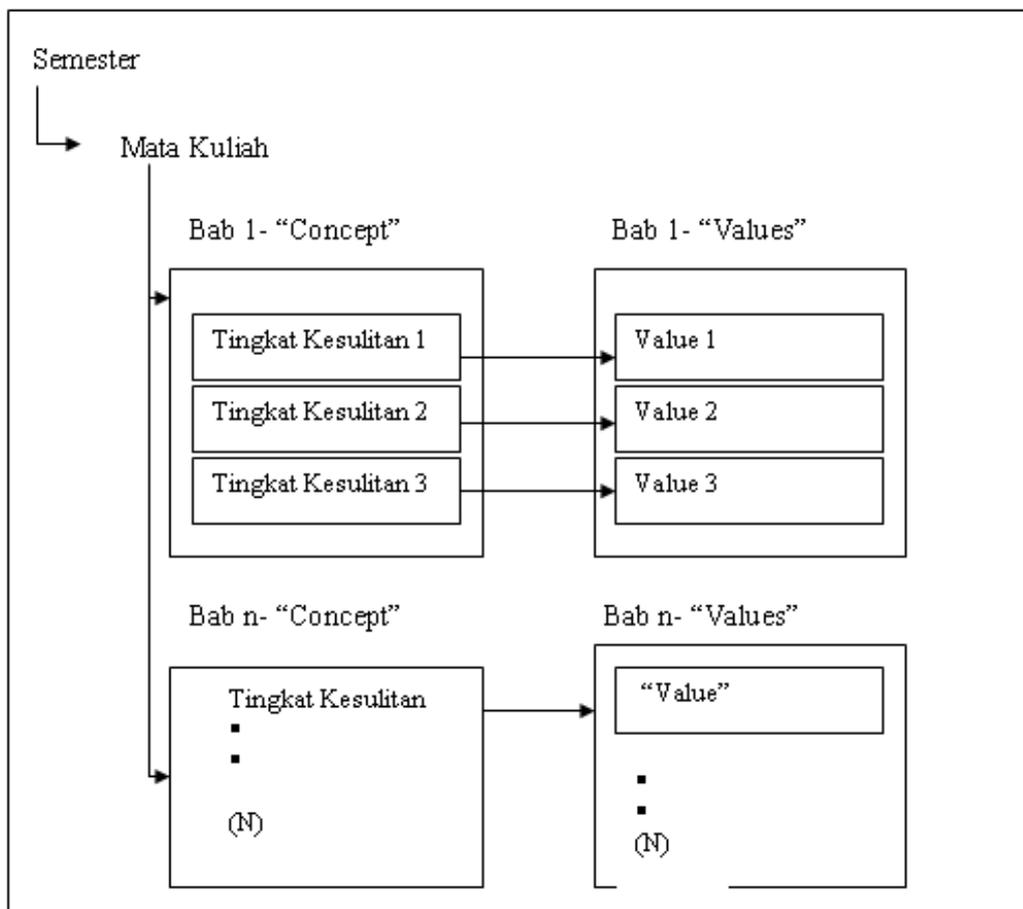
dibuat dalam *Use Case* Mengelola Mata Kuliah. Pengguna diijinkan untuk membuat setidaknya satu atau lebih bab dalam satu mata kuliah.

3.3.2.2.2 Student Model dari Sistem Kuis Online Adaptif

Seperti yang disebutkan dalam bab sebelumnya, *user model* mewakili sistem keyakinan tentang pengguna yang menggambarkan pengetahuan pengguna. Sistem kuis *online adaptif* ini mengacu ke *student model* untuk *user modelnya*. *Student model* menggambarkan pengetahuan mahasiswa dari *domain model*. Berdasarkan *student model*, sistem memperlakukan cara yang berbeda untuk setiap mahasiswa. *Student model* berisi semua informasi yang sistem ketahui tentang mahasiswa. Terdapat dua jenis informasi yang diperlukan dalam rangka pengembangan *student model*, yaitu informasi statis dan dinamis. Informasi statis berisi profil mahasiswa seperti nama mahasiswa, alamat, nomor induk mahasiswa, jenis kelamin, semester dan nomor telepon. Informasi dinamis berisi informasi tentang interaksi pengguna dengan sistem seperti tingkat kesulitan pengetahuan mahasiswa saat ini, sejarah tingkat kesulitan pengetahuan mahasiswa, tanggal kuis diambil, dan mata kuliah kuis yang diambil. *Overlay model* dan teknik *stereotype* dipilih untuk menangkap informasi tersebut baik statis maupun dinamis.

Overlay model digunakan untuk menjelaskan pengetahuan mahasiswa melalui pengetahuan domain. Pengukuran kualitatif dari *overlay model* menunjukkan bobot pengetahuan pengguna diukur berdasarkan pasangan *concept-value*. Konten atau topik dari pengetahuan domain dianggap sebagai '*concept*' dan hasil prestasi masing-masing mahasiswa dari perbab dianggap sebagai

'value'. Untuk setiap 'concept', penelitian ini mengijinkan tingkat kesulitan lebih dari satu yang akan dibuat. Masing-masing dari tingkat kesulitan ini kemudian dipetakan ke sebuah 'value' dari prestasi mahasiswa. Peneliti hanya menggunakan tingkat kesulitan 1 s.d. 3, semakin tinggi tingkat kesulitannya, semakin sulit pertanyaannya dan semuanya itu ditentukan oleh pengajar. Gambar 3.3 menunjukkan *student overlay model* dari sistem kuis *online adaptif*.



Gambar 3.3 *Student Overlay Model Sistem Kuis Online Adaptif*

Peneliti menggunakan *stereotype* untuk mengelompokkan mahasiswa berdasarkan kriteria umum dari model mahasiswa. Ada dua jenis informasi yang diperlukan untuk diambil dalam rangka menerapkan teknik *stereotype*:

1. sistem harus mengetahui informasi, atribut atau properti mahasiswa untuk menangkap *stereotype*,
2. skenario, peristiwa atau perilaku apa yang mencerminkan *stereotype*.

Dalam rangka menerapkan teknik *stereotype* pada sistem kuis *online adaptif*, terdapat beberapa atribut yang dibutuhkan antara lain:

1. semester yang ditempuh mahasiswa,
2. mata kuliah yang diambil oleh mahasiswa,
3. tingkat kesulitan pengetahuan domain terakhir (termasuk bab dan tingkat kesulitannya),
 - a. jika mahasiswa adalah pengguna kuis pertama kali, tingkat pengetahuan domain terakhir ditentukan oleh *pretest*,
 - b. jika mahasiswa bukan pengguna kuis pertama kali, tingkat pengetahuan mahasiswa terakhir mengacu pada kuis yang terakhir diambil.
4. nilai mahasiswa dalam tingkat kesulitan pengetahuan domain terakhir.

Berdasarkan atribut yang dibutuhkan, pengajar diwajibkan untuk menetapkan aturan kondisi yang mencerminkan *stereotype* tertentu. Kelompok *stereotype* bagi mahasiswa berubah tergantung pada nilai yang mereka dapatkan. Mereka mungkin termasuk dalam kelompok mahasiswa pemula yang mengakses kuis pertama kali dan kemudian pindah ke kelompok mahasiswa tingkat lanjut setelah mengambil kuis dan memenuhi nilai yang ditentukan. Misalnya, pada semester 1 mahasiswa mengambil kuis mata kuliah algoritma pemrograman dan mendapatkan nilai 70, maka sistem mencari aturan yang sesuai kondisi yaitu,

- a. jika nilai mahasiswa jatuh dalam kisaran 1 sampai 49, maka mahasiswa dianggap sebagai tingkat pemula,
- b. jika nilai mahasiswa jatuh pada kisaran 50 sampai 74, mahasiswa dianggap sebagai mahasiswa tingkat menengah, dan
- c. jika nilai mahasiswa di atas 74, mahasiswa dianggap sebagai mahasiswa tingkat lanjut.

Berdasarkan representasi dari *student model* melalui model *overlay* dan teknik *stereotype* tersebut, untuk mengembangkan *student model* sistem kuis *online adaptif* ini peneliti mentransformasikan ke dalam bentuk *use case*. Terdapat tiga *use case* utama yang dapat diidentifikasi dalam rangka pengembangan *student model* sistem kuis *online adaptif* ini. Dalam rangka untuk menangkap informasi statis, peneliti telah mengidentifikasi satu *use case* utama yang disebut "Mengelola Profil Mahasiswa". *Use case* "Mengelola Profil Mahasiswa" ini digunakan untuk menyimpan semua informasi statis yang merupakan profil mahasiswa seperti yang disebutkan di atas dalam sistem. *Use case* ini memungkinkan administrator untuk menambah, memperbarui, menghapus, dan melihat profil mahasiswa dalam database. Dalam rangka untuk menangkap informasi dinamis, peneliti telah mengidentifikasi dua *use case* utama: "Mahasiswa Mengambil Kuis" dan "Histori Hasil Kuis Mahasiswa". *Use case* "Mahasiswa Mengambil Kuis" memungkinkan mahasiswa untuk mengambil kuis melalui sistem yang mengakibatkan tingkat pengetahuan mahasiswa saat ini, tanggal kuis, nilai dan mata kuliah kuis direkam. *Use case* "Histori Hasil Kuis

Mahasiswa” digunakan untuk menangkap informasi sejarah tingkat pengetahuan mahasiswa.

3.3.2.2.3 Adaptation Model dari Sistem Kuis Online Adaptif

Adaptation model mendefinisikan apa yang dapat diadaptasi, kapan harus diadaptasi dan bagaimana itu harus diadaptasi dalam sistem. Apa yang harus diperbarui yaitu mendefinisikan informasi seperti:

1. pertanyaan-pertanyaan mana yang seharusnya dianggap mudah atau sulit,
2. grade untuk menafsirkan pengetahuan mahasiswa,
3. berapa banyak pertanyaan yang akan disajikan per kuis untuk memperkirakan pengetahuan peserta didik ,
4. bagaimana kinerja peserta didik akan mempengaruhi model pembelajar

Faktor yang menentukan informasi yang akan disesuaikan dengan *adaptation model* sistem kuis *online adaptif* adalah sebagai berikut:

Tabel 3.3 Faktor-faktor yang menentukan Adaptasi Model Sistem Kuis Online Adaptif

Apa yang diadaptasi	Faktor yang menentukan
<ul style="list-style-type: none"> ▪ Pertanyaan-pertanyaan mana yang seharusnya dianggap mudah atau sulit 	<ul style="list-style-type: none"> ▪ pertanyaan itu termasuk mudah atau sulit ditentukan oleh pengajar
<ul style="list-style-type: none"> ▪ Grade untuk menafsirkan pengetahuan mahasiswa 	<ul style="list-style-type: none"> ▪ Pengetahuan mahasiswa diukur berdasarkan jawaban benar yang diberikan oleh mahasiswa terhadap jumlah pertanyaan yang diberikan kepada mahasiswa per kuis dalam rangka untuk menyatakan grade.

Lanjutan Tabel 3.4 Faktor-faktor yang menentukan Adaptasi Model Sistem Kuis Online Adaptif

Apa yang diadaptasi	Faktor yang menentukan
<ul style="list-style-type: none"> ▪ Berapa banyak pertanyaan yang akan disajikan per kuis 	<ul style="list-style-type: none"> ▪ Pengajar bertanggung jawab untuk menentukan jumlah pertanyaan yang disajikan per kuis
<ul style="list-style-type: none"> ▪ Bagaimana kinerja peserta didik akan mempengaruhi model pembelajar 	<ul style="list-style-type: none"> ▪ Faktor yang mempengaruhi <i>student model</i> adalah kinerja mahasiswa. Kinerja mahasiswa kemudian diperbarui dalam <i>student model</i>

Informasi tambahan lain yang mendefinisikan *adaptation model* sistem kuis *online adaptif* adalah:

1. tingkat kesulitan pengetahuan domain berikutnya yang direkomendasikan,
2. bagaimana pertanyaan-pertanyaan tersebut dipilih,
3. informasi umpan balik,
4. item kuis apa yang akan disajikan untuk mahasiswa pada saat akses pertama kali.

Tabel berikut ini menggambarkan faktor tambahan untuk mendefinisikan *adaptation model* dari sistem kuis *online adaptif*

Tabel 3.5 Faktor Tambahan untuk Mendefinisikan Adaptation Model

Apa yang diadaptasi	Faktor yang menentukan
<ul style="list-style-type: none"> ▪ tingkat kesulitan pengetahuan domain berikutnya yang direkomendasikan 	<ul style="list-style-type: none"> ▪ Faktor yang menentukan tingkat kesulitan berikutnya adalah berdasarkan hasil mahasiswa dan aturan yang didefinisikan pengajar
<ul style="list-style-type: none"> ▪ bagaimana pertanyaan-pertanyaan tersebut dipilih 	<ul style="list-style-type: none"> ▪ Karena penelitian ini menggunakan teknik <i>adaptive question</i> yaitu jawaban mahasiswa menentukan serangkaian

Lanjutan Tabel 3.6 Faktor Tambahan untuk Mendefinisikan Adaptation

Apa yang diadaptasi	Faktor yang menentukan
	<p>pertanyaan berikutnya, jadi untuk sistem kuis <i>online adaptif</i> ini, serangkaian pertanyaan berikutnya adalah merujuk ke sekumpulan pertanyaan yang memiliki tingkat kesulitan yang sama.</p> <ul style="list-style-type: none"> ▪ Tingkat kesulitan untuk setiap pertanyaan ditentukan oleh pengajar. ▪ Pertanyaan-pertanyaan dipilih untuk kuis berdasarkan pada teknik pengacakan. Berikut ini membahas langkah tentang bagaimana teknik pengacakan dilakukan: <ul style="list-style-type: none"> ○ Berdasarkan tingkat kesulitan pengetahuan domain mahasiswa yang direkomendasikan, sistem kuis <i>online adaptif</i> mendapatkan seluruh id pertanyaan untuk tingkat kesulitan pengetahuan domain yang direkomendasikan. ○ Sistem kuis <i>online adaptif</i> melakukan proses pengecekan pada berapa banyak pertanyaan yang diperlukan untuk ditampilkan per kuis. Hal ini mengacu pada konfigurasi kuis. Sebagai contoh, jika konfigurasi kuis menunjukkan bahwa sistem yang mendefinisikan 10 pertanyaan per kuis, maka sistem diharuskan untuk menampilkan

Lanjutan Tabel 3.7 Faktor Tambahan untuk Mendefinisikan Adaptation

Apa yang diadaptasi	Faktor yang menentukan
	<p>10 pertanyaan.</p> <ul style="list-style-type: none"> ○ Setelah mendapatkan jumlah pertanyaan yang akan ditampilkan, sistem kuis <i>online adaptif</i> mengacak seluruh id pertanyaan dan mengambil sebagian. ○ Jika id pertanyaan berikutnya yang dipilih sama dengan id pertanyaan sebelumnya, sistem kuis <i>online adaptif</i> akan melakukan pengacakan lagi sampai mendapat id pertanyaan yang unik ○ Berdasarkan id pertanyaan unik yang dipilih, sistem kuis online adaptif mencari deskripsi pertanyaan dan menampilkannya kepada mahasiswa. ○ Hasilnya pertanyaan-pertanyaan yang disajikan kepada mahasiswa tidak akan sama.
<ul style="list-style-type: none"> ▪ informasi umpan balik 	<ul style="list-style-type: none"> ▪ sistem menampilkan hasilnya, dan juga tingkat kesulitan berikutnya yang direkomendasikan.
<ul style="list-style-type: none"> ▪ tingkat kesulitan pengetahuan domain yang mana untuk disampaikan kepada mahasiswa pada saat pertama kali mengakses 	<ul style="list-style-type: none"> ▪ Pengajar diperbolehkan untuk menentukan tingkat kesulitan pengetahuan domain mana yang akan disajikan, untuk pertama kali mahasiswa mengakses sistem melalui <i>pre test</i>.

Penelitian ini menerapkan aturan *IF-THEN* untuk *adaptation model* terutama pada bagian "kapan dan bagaimana diadaptasi". Bentuk aturan dari IF <kondisi> THEN <aksi> adalah sebagai berikut:

- <kondisi> menunjuk pada tingkat pengetahuan mahasiswa dari *student model*
- <aksi> menunjuk pada perubahan hasil tingkat pengetahuan mahasiswa yang mencerminkan ke tingkat kesulitan pengetahuan domain berikutnya. Apakah ke tingkat yang lebih tinggi, tetap di tingkat yang sama atau pindah ke tingkat yang lebih rendah dari pengetahuan domain

Aturan <kondisi> dan <aksi> adalah pasangan aturan yang ditetapkan oleh pengajar dengan ketentuan untuk setiap <kondisi>, hanya ada satu <aksi>. Berdasarkan <kondisi>, sistem kuis *online adaptif* akan mencari <aksi> yang sesuai. Untuk sistem kuis *online adaptif* ini <kondisi> ditentukan oleh hasil nilai mahasiswa terakhir, bersamaan dengan bab dan tingkat kesulitan pengetahuan domain terakhir, sedangkan <aksi> digunakan untuk menentukan tingkat kesulitan berikutnya yang direkomendasikan dan bab dari pengetahuan domain berdasarkan rentang skor tertentu yang didefinisikan oleh pengajar.

Berdasarkan implementasi *adaptation model* tersebut, untuk menangkap "apa, kapan dan bagaimana harus diadaptasi" dalam sistem kuis *online adaptif* ini, peneliti mentransformasikan ke dalam delapan bentuk *use case* :

1. *Use Case* Konfigurasi Kuis digunakan untuk menangkap informasi seperti berapa banyak pertanyaan yang harus ditetapkan untuk setiap kuis, di mana tingkat kesulitan pengetahuan yang mahasiswa harus mulai (akses pertama kali) dan skor untuk setiap pertanyaan.
2. *Use Case* Mengelola Tingkat Kesulitan digunakan untuk mengukur pemahaman mahasiswa terhadap mata kuliah dengan memungkinkan pengajar untuk membuat, memperbarui, menghapus dan melihat tingkat kesulitan pengetahuan yang berbeda dari pengetahuan domain seperti level 1 untuk mahasiswa pemula, level 2 untuk mahasiswa menengah dan level 3 untuk mahasiswa tingkat lanjut.
3. *Use Case* Mengelola Pertanyaan digunakan untuk membuat, memperbarui, menghapus, dan melihat pertanyaan untuk kuis adaptif. Pertanyaan dikelompokkan di tempat yang berbeda berdasarkan pada tingkat kesulitannya. Artinya pertanyaan dengan tingkat kesulitan, bab, dan mata kuliah yang sama akan diletakkan di tempat yang sama. Tingkat kesulitan pertanyaan ditentukan oleh pengajar.
4. *Use Case* Menampilkan Tingkat Kesulitan saat ini dan berikutnya digunakan untuk menampilkan tingkat kesulitan saat ini dan berikutnya dari pengetahuan domain mahasiswa.
5. *Use Case* Melaksanakan Kuis digunakan untuk mengizinkan mahasiswa mengambil kuis, setelah mengirim jawabannya, sistem melakukan perhitungan dalam rangka untuk mendapatkan hasil.

6. *Use Case* Memberikan Umpan Balik digunakan untuk menampilkan hasil nilai mahasiswa.
7. *Use Case* Memilih Pertanyaan digunakan untuk mengacak dan menampilkan pertanyaan berdasarkan tingkat kesulitan yang pengetahuan mahasiswa direkomendasikan.
8. *Use Case* Mengelola Aturan digunakan untuk memungkinkan pengajar mendefinisikan aturan untuk adaptasi. Kondisi yang harus disesuaikan ditetapkan berdasarkan aturan dan tindakan yang akan dipicu didasarkan pada kondisi yang ditetapkan. Agar aturan dapat dieksekusi, perlu informasi dari "*Use Case* Histori Hasil Kuis Mahasiswa " yang ditangkap dalam *student model*..

3.3.2.3 Identifikasi Use Case Berdasarkan Modul

Semua *use case* yang telah diidentifikasi sebelumnya merupakan *use case* utama, selain itu terdapat *use case* lain yang diidentifikasi untuk mendukung dan menyelesaikan pengembangan sistem kuis *online adaptif* ini sebagai salah satu keseluruhan sistem (sistem lengkap) seperti *use case* mengelola profil pengajar, *use case* mengelola profil mahasiswa, *use case* reset password, dan *use case* mengelola kelas mahasiswa. *Use case* yang telah diidentifikasi dapat dikelompokkan sesuai dengan peran mereka untuk kemudahan aksesibilitas, sehingga *use case* dapat dibagi menjadi tiga modul, yaitu: modul Administrator, modul pengajar, dan modul mahasiswa

3.3.2.3.1 Modul Administrator

Modul Administrator dapat diakses hanya untuk administrator terdaftar dan bertanggung jawab untuk menangani semua pekerjaan administrasi. Berikut ini adalah daftar *use case* utama dari modul Administrator untuk sistem kuis *online adaptif*.

a. *Use Case* Login Administrator

Use case ini memungkinkan administrator untuk login ke sistem

b. *Use Case* Mengelola Profil Pengajar

Use case ini memungkinkan administrator untuk menambah, mengupdate, melihat dan menghapus profil pengajar dalam sistem.

c. *Use Case* Mengelola Profil Mahasiswa

Use case ini memungkinkan administrator untuk menambah, mengupdate, melihat dan menghapus profil mahasiswa dalam sistem.

d. *Use Case* Mengelola Mata Kuliah

Use case ini memungkinkan administrator untuk menambah, mengupdate, melihat dan menghapus profil mata kuliah dalam sistem.

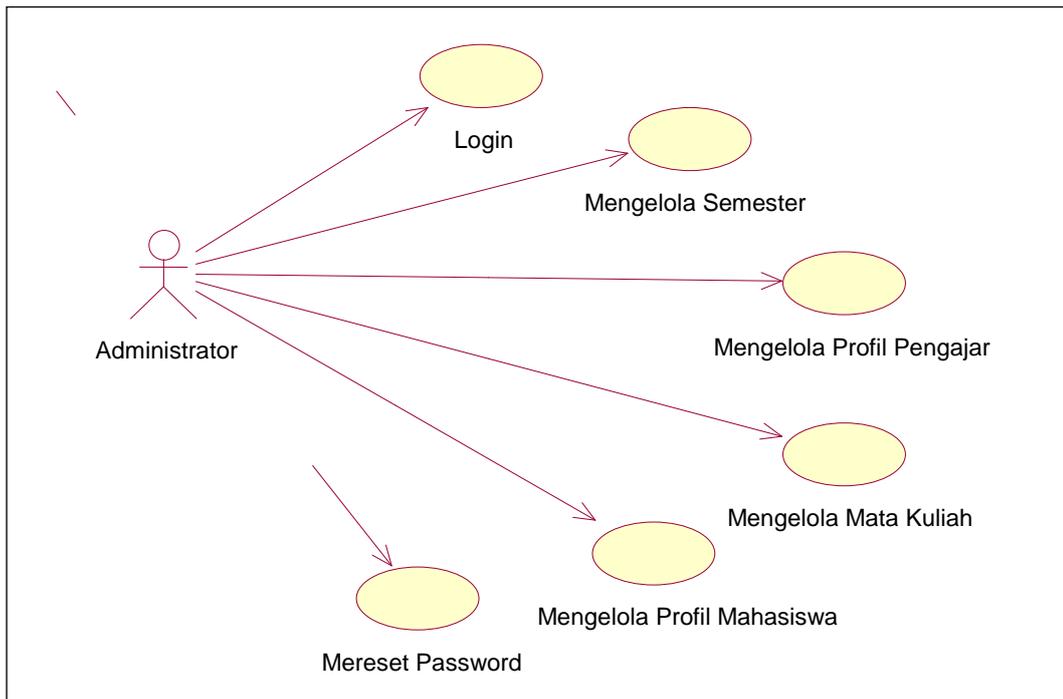
e. *Use Case* Mengelola Semester

Use case ini memungkinkan administrator untuk menambah, mengupdate, melihat dan menghapus semester dalam sistem.

f. *Use Case* Reset Password

Use case ini memungkinkan administrator untuk mereset password pengajar dan mahasiswa.

Gambar 3.4 menunjukkan diagram *use case* utama untuk modul Administrator. *Use case* lengkap dan deskripsi *use case* untuk modul Administrator disajikan dalam Lampiran A. Semua *use case* utama tersebut kemudian diterjemahkan sebagai fungsionalitas sistem kuis online adaptif.



Gambar 3.4 Diagram Use Case Utama untuk Modul Administrator

Tabel 3.8 menampilkan semua fungsi yang diidentifikasi dan didukung oleh *use case* untuk modul Administrator.

Tabel 3.8 Fungsi yang diidentifikasi dan didukung oleh *use case* untuk Modul Administrator

Fungsi Sistem	Use Case
Login Admin	<ul style="list-style-type: none"> ▪ Use Case Login Admin (Lampiran A untuk diagram dan deskripsi use case Login Admin)
Mengelola Semester	<ul style="list-style-type: none"> ▪ Use Case Mengelola Semester

Lanjutan Tabel 3.9 Fungsi yang diidentifikasi dan didukung oleh use case untuk Modul Administrator

Fungsi Sistem	Use Case
	(Lampiran A untuk diagram use case dan deskripsi use case Mengelola Semester)
Mengelola Profil Pengajar	<ul style="list-style-type: none"> ▪ Use Case Menambah Profil Pengajar ▪ Use Case Memperbarui Profil Pengajar ▪ Use Case Menampilkan Profil Pengajar ▪ Use Case Menghapus Profil Pengajar <p style="text-align: center;"><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Include Menampilkan Profil Pengajar <p>(Lampiran A untuk diagram use case dan deskripsi use case Mengelola Pengajar)</p>
Mengelola Profil Mahasiswa	<ul style="list-style-type: none"> ▪ Use Case Menambah Profil Mahasiswa ▪ Use Case Memperbarui Profil Mahasiswa ▪ Use Case Menampilkan Profil Mahasiswa ▪ Use Case Menghapus Profil Mahasiswa <p style="text-align: center;"><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Include Menampilkan Kelas berdasarkan Semester ▪ Include Menampilkan Profil Mahasiswa <p>(Lampiran A untuk diagram use case dan deskripsi use case Mengelola Mahasiswa)</p>
Mengelola Mata Kuliah	<ul style="list-style-type: none"> ▪ Use Case Menambah Mata Kuliah ▪ Use Case Memperbarui Mata Kuliah ▪ Use Case Menampilkan Mata Kuliah ▪ Use Case Menghapus Mata Kuliah <p style="text-align: center;"><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Include Use Case Menampilkan Mata Kuliah <p>(Lampiran A untuk diagram use case dan deskripsi use case Mengelola Mata Kuliah)</p>
Merest Password	<ul style="list-style-type: none"> • Use Case Reset Password untuk Mahasiswa • Use Case Reset Password untuk Pengajar <p style="text-align: center;"><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Include Use Case Menampilkan Profil

Lanjutan Tabel 3.10 Fungsi yang diidentifikasi dan didukung oleh use case untuk Modul Administrator

Fungsi Sistem	Use Case
	<p>Mahasiswa</p> <ul style="list-style-type: none"> ▪ Include Use Case Menampilkan Profil Pengajar <p>(Lampiran A untuk diagram use case dan deskripsi use case Mengelola Reset Password)</p>

3.3.2.3.2 Modul Pengajar

Modul pengajar dapat diakses hanya untuk pengajar yang terdaftar dan mereka bertanggung jawab untuk menangani modul pengajar. Berikut ini adalah *use case* utama dari modul pengajar untuk sistem kuis *online adaptif*.

a. Use Case Login Pengajar

Use case ini memungkinkan pengajar untuk login ke sistem

b. Use Case Mengelola Bab

Use case ini memungkinkan pengajar untuk menambah, memperbarui, melihat dan menghapus profil bab untuk setiap mata kuliah yang terdaftar dalam sistem.

c. Use Case Mengelola Tingkat Kesulitan dari Pengetahuan Domain

Use case ini digunakan untuk mengukur pemahaman mahasiswa terhadap mata kuliah dengan memungkinkan pengajar untuk membuat tingkat kesulitan pengetahuan domain yang berbeda seperti level 1 untuk mahasiswa pemula, tingkat 2 untuk mahasiswa menengah dan level 3 untuk mahasiswa tingkat lanjut. Pengajar juga diperbolehkan untuk memperbarui, menghapus, dan melihat tingkat kesulitan, selain itu, sistem yang memungkinkan pengajar untuk membuat tingkat kesulitan lebih dari satu dalam satu bab (tidak terbatas).

d. Use Case Mengelola Aturan (*rule*) dari Pengetahuan Domain

Use case ini digunakan untuk mendefinisikan aturan pengetahuan domain. Aturan ini digunakan untuk menentukan tingkat pengetahuan domain berikutnya yang harus diambil mahasiswa. Pengajar diharuskan untuk mengisi atau mengatur kondisi dari aturan yang harus dicapai mahasiswa (nilai dan tingkat pengetahuan saat ini) dan juga menetapkan tindakan bahwa sistem seharusnya dilaksanakan atas pencapaian target tersebut.

e. Use Case Mengelola Pertanyaan

Use case ini digunakan untuk membuat, memperbarui, menghapus, dan melihat pertanyaan untuk kuis adaptif. Sistem ini hanya mendukung pertanyaan pilihan ganda (*multiple choice*), benar atau salah (*true false*), dan isian singkat (*fill in the blank*) . Pertanyaan dikelompokkan di kelompok pertanyaan yang berbeda berdasarkan tingkat kesulitannya. Ini berarti, pertanyaan dengan tingkat kesulitan, bab, dan mata kuliah yang sama akan diletakkan di kelompok pertanyaan yang sama. Namun, pengajar diwajibkan untuk menentukan tingkat kesulitan dari setiap pertanyaan.

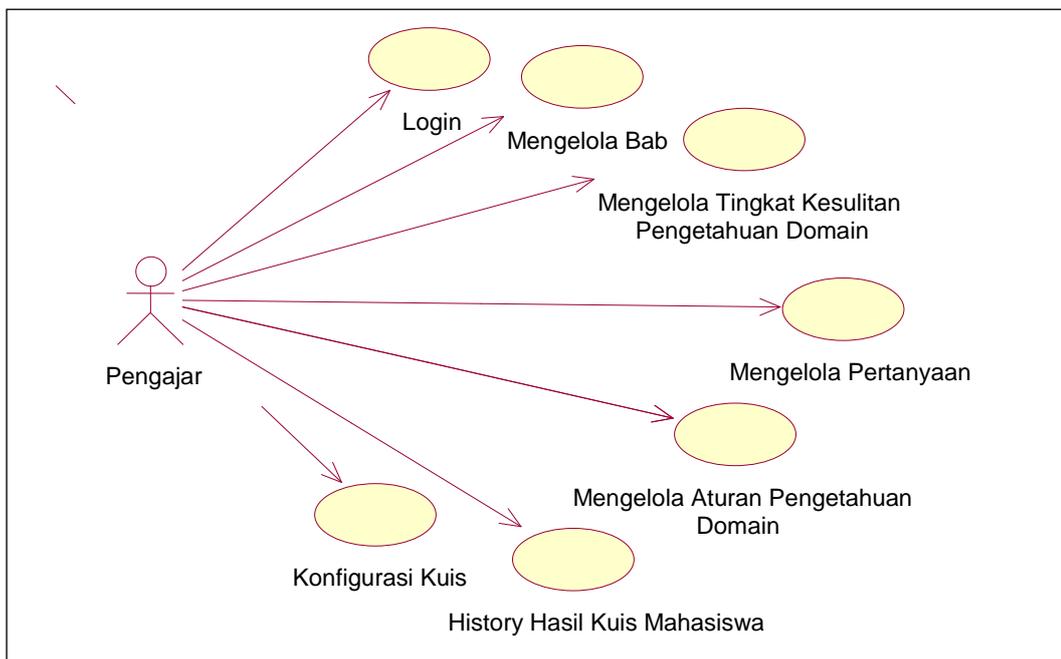
f. Use Case Konfigurasi Kuis

Use case ini digunakan untuk mengatur konfigurasi kuis. Hal ini memungkinkan pengajar untuk menentukan berapa banyak pertanyaan yang harus ditetapkan dalam sebuah kuis.

g. Use Case History Hasil Kuis Mahasiswa

Use case ini mengizinkan pengajar untuk melihat kinerja mahasiswa mereka. Pada saat yang sama pengajar akan mengetahui kekuatan dan kelemahan yang dihadapi mahasiswa dalam mata kuliah berdasarkan nilai mereka.

Gambar 3.5 menunjukkan diagram *use case* utama untuk modul Pengajar. *Use case* dan deskripsi lengkapnya untuk modul Pengajar disajikan pada Lampiran A. Semua *use case* utama tersebut kemudian diterjemahkan sebagai fungsionalitas sistem kuis *online adaptif*. Tabel 4.5 menampilkan daftar semua fungsi yang diidentifikasi dan didukung oleh *use case* untuk modul Pengajar.



Gambar 3.5 Diagram Use Case Utama untuk Modul Pengajar

Tabel 3.11 Fungsi yang diidentifikasi dan didukung oleh Use Case untuk Modul Pengajar

Fungsi	Use Case
Login Pengajar	<ul style="list-style-type: none"> ▪ Use Case Login Pengajar (Lampiran A untuk diagram dan deskripsi use case Login Pengajar)
Mengelola Bab	<ul style="list-style-type: none"> ▪ Use Case Menambah Bab ▪ Use Case Memperbarui Bab ▪ Use Case Menampilkan Bab ▪ Use Case Menghapus Bab

Lanjutan Tabel 3.12 Fungsi yang diidentifikasi dan didukung oleh Use Case untuk Modul Pengajar

Fungsi	Use Case
	<p><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Use Case Include Menampilkan Bab ▪ Use Case Include Menampilkan Mata Kuliah berdasarkan Semester <p>(Lampiran A untuk diagram dan deskripsi use case Mengelola Bab)</p>
<p>Mengelola Tingkat Kesulitan Pengetahuan Domain</p>	<ul style="list-style-type: none"> ▪ Use Case Menambah Tingkat Kesulitan ▪ Use Case Memperbarui Tingkat Kesulitan ▪ Use Case Menampilkan Tingkat Kesulitan ▪ Use Case Menghapus Tingkat Kesulitan <p><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Use Case Include Menampilkan Mata Kuliah berdasarkan Semester ▪ Use Case Include Menampilkan Bab berdasarkan Mata Kuliah ▪ Use Case Include Menampilkan Tingkat Kesulitan <p>(Lampiran A untuk diagram dan deskripsi use case Mengelola Tingkat Kesulitan Pengetahuan Domain)</p>
<p>Mengelola Aturan Pengetahuan Domain</p>	<ul style="list-style-type: none"> ▪ Use Case Menambah Aturan ▪ Use Case Memperbarui Aturan ▪ Use Case Menampilkan Aturan ▪ Use Case Menghapus Aturan <p><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Use Case Include Menampilkan Mata Kuliah berdasarkan Semester ▪ Use Case Include Menampilkan Bab berdasarkan Mata Kuliah ▪ Use Case Include Menampilkan Tingkat Kesulitan berdasarkan Bab ▪ Use Case Include Menampilkan Aturan <p>(Lampiran A untuk diagram dan deskripsi use case Mengelola Aturan)</p>
<p>Mengelola</p>	<ul style="list-style-type: none"> ▪ Use Case Menambah Pertanyaan

Lanjutan Tabel 3.13 Fungsi yang diidentifikasi dan didukung oleh Use Case untuk Modul Pengajar

Fungsi	Use Case
Pertanyaan	<ul style="list-style-type: none"> ▪ Use Case Memperbarui Pertanyaan ▪ Use Case Menghapus Pertanyaan <p><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Use Case Include Menampilkan Mata Kuliah berdasarkan Semester ▪ Use Case Include Menampilkan Bab berdasarkan Mata Kuliah <p>(Lampiran A untuk diagram dan deskripsi use case Mengelola Pertanyaan)</p>
Konfigurasi Kuis	<ul style="list-style-type: none"> ▪ Use Case Menambah Konfigurasi Kuis ▪ Use Case Memperbarui Konfigurasi Kuis ▪ Use Case Menampilkan Konfigurasi Kuis ▪ Use Case Menghapus Konfigurasi Kuis <p><u>Use Case Include</u></p> <ul style="list-style-type: none"> ▪ Use Case Include Menampilkan Mata Kuliah berdasarkan Semester ▪ Use Case Include Menampilkan Bab berdasarkan Mata Kuliah ▪ Use Case Include Menampilkan Tingkat Kesulitan berdasarkan Bab ▪ Use Case Include Menampilkan Konfigurasi Kuis <p>(Lampiran A untuk diagram dan deskripsi use case Mengelola Konfigurasi Kuis)</p>
History Hasil Kuis Mahasiswa	<ul style="list-style-type: none"> ▪ Use Case Menampilkan History Hasil Kuis Mahasiswa <p>(Lampiran A untuk diagram dan deskripsi use case History Hasil Kuis Mahasiswa)</p>

3.3.2.3.3 Modul Mahasiswa

Modul mahasiswa hanya dapat diakses oleh mahasiswa yang terdaftar.

Modul mahasiswa bertanggung jawab untuk menangani kegiatan mahasiswa saat

mengambil kuis. Daftar berikut ini merupakan *use case* utama dari modul mahasiswa untuk sistem kuis *online adaptif*, penjelasan lebih lanjutnya diuraikan dalam Lampiran A.

a. Use Case Login Mahasiswa

Use case ini mengizinkan mahasiswa untuk login ke sistem

b. Use Case Mahasiswa Mengambil Kuis

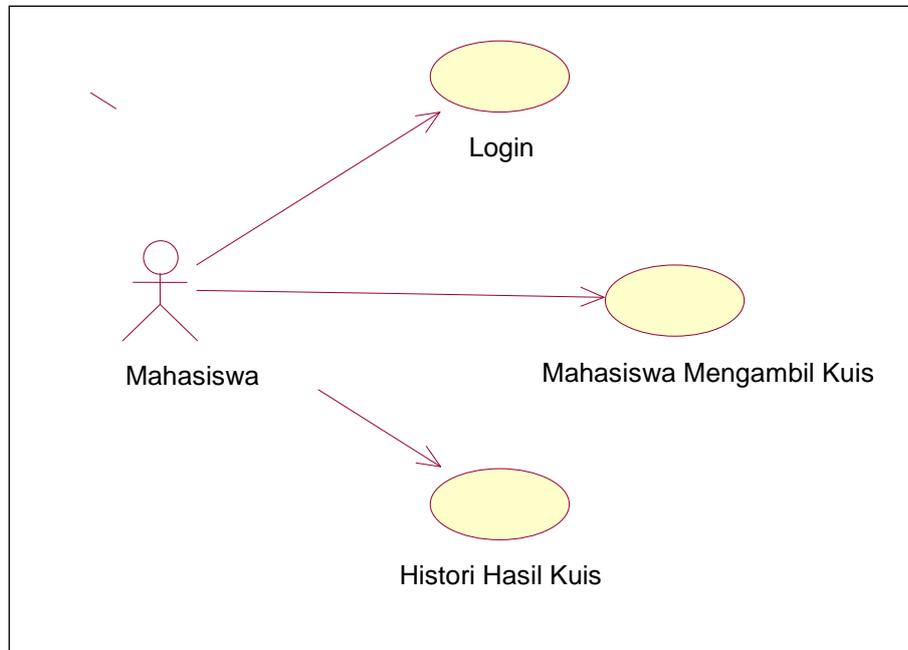
Use case ini mengizinkan mahasiswa untuk mengambil kuis berdasarkan mata kuliah yang dipilih. Hal ini juga menunjukkan tingkat kesulitan mahasiswa saat ini dan juga tingkat kesulitan yang direkomendasikan untuk mahasiswa. Pada saat mahasiswa mengambil kuis, mahasiswa akan disajikan sekumpulan pertanyaan berdasarkan tingkat kesulitan yang direkomendasikan. Setelah sistem mengirimkan pertanyaan, sistem melakukan pengecekan dan menampilkan umpan balik untuk para mahasiswa yang meliputi hasil mahasiswa, dan tingkat kesulitan pengetahuan domain yang direkomendasikan berikutnya.

c. Use Case Histori Hasil Kuis Mahasiswa

Use case ini mengizinkan mahasiswa untuk menampilkan tingkat performa mereka, mengetahui kekuatan dan kelemahan yang dihadapi mahasiswa dalam mata kuliah berdasarkan nilai (skor) mereka.

Gambar 3.6 menunjukkan diagram *use case* utama untuk modul Mahasiswa. *Use case* dan deskripsi lengkap untuk modul Mahasiswa disajikan pada Lampiran A. Semua *use case* utama tersebut kemudian diterjemahkan sebagai fungsionalitas sistem kuis *online adaptif*. Tabel 3.6 menampilkan daftar

semua fungsi yang diidentifikasi dan didukung oleh *use case* untuk modul Mahasiswa.



Gambar 3.6 Diagram Use Case Utama untuk Modul Mahasiswa.

Tabel 3.14 Fungsi yang diidentifikasi dan didukung oleh Use Case untuk Modul Mahasiswa

Fungsi	Use Case
History Hasil Kuis Mahasiswa	Use Case Menampilkan History Hasil Kuis Mahasiswa <u>Use Case Include</u> Use Case Include Menampilkan Mata Kuliah berdasarkan Semester (Lampiran A untuk diagram dan deskripsi use case Mengelola History Hasil Kuis Mahasiswa)
Login Mahasiswa	Use Case Login Mahasiswa (Lampiran A untuk diagram dan deskripsi use case Login Mahasiswa)

Lanjutan Tabel 3.15 Fungsi yang diidentifikasi dan didukung oleh Use Case untuk Modul Mahasiswa

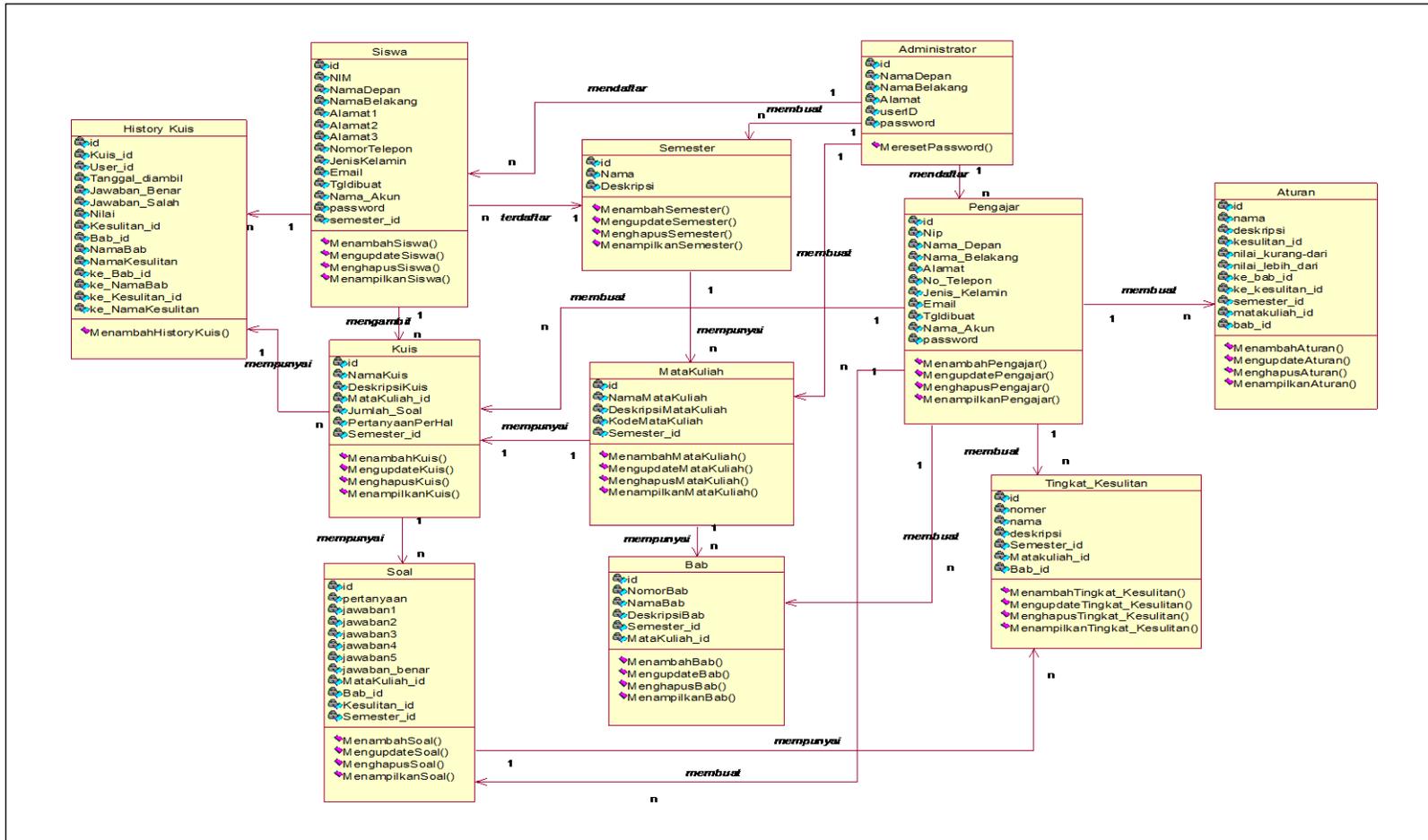
Fungsi	Use Case
Mahasiswa Mengambil Kuis	Use Case Menampilkan Tingkat Kesulitan Mahasiswa Saat ini dan Berikutnya Use Case Mengerjakan Kuis Use Case Memilih Pertanyaan Use Case Konfigurasi Kuis <u>Use Case Include</u> Use Case Include Menampilkan Bab berdasarkan Mata Kuliah (Lampiran A untuk diagram dan deskripsi use case Mahasiswa Mengambil Kuis)

3.3.3 Desain Sistem

Pada tahap desain sistem, desain teknis mulai didefinisikan yang berisi dari desain terinci untuk masing-masing *use case*. Semua *use case* yang disajikan dalam tahap analisis akan diperinci dan disajikan dalam tiga diagram yaitu *sequence diagram*, *class diagram*, dan *diagram aktifitas*. *Sequence diagram* digunakan untuk menggambarkan interaksi yang diatur dalam urutan waktu. *Class diagram* digunakan untuk memodelkan *class-class* yang akan digunakan, yang nantinya akan juga ditransformasikan dalam bentuk tabel-tabel suatu database sebagai tempat penyimpanan data, sedangkan *diagram aktifitas* untuk memodelkan proses yang berlangsung dalam sistem kuis online adaptif ini. Desain *user interface* juga akan digambarkan dalam tahapan ini sebagai bentuk antar muka sistem dengan pengguna.

3.3.3.1 Class Diagram Sistem Kuis Online Adaptif

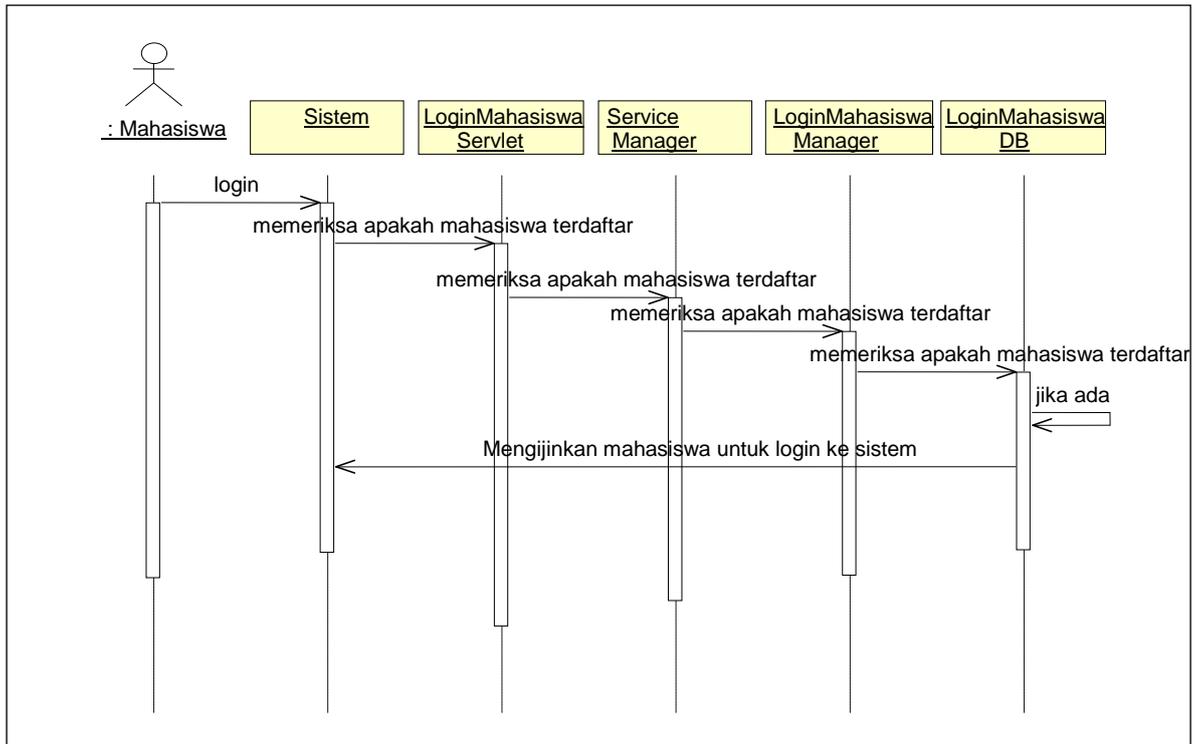
Sistem kuis online adaptif ini didesain dengan menggunakan pendekatan berorientasi objek, class yang merupakan *blue print* dari sebuah objek digambarkan dengan *class diagram* seperti yang ditunjukkan dalam Gambar 3.6



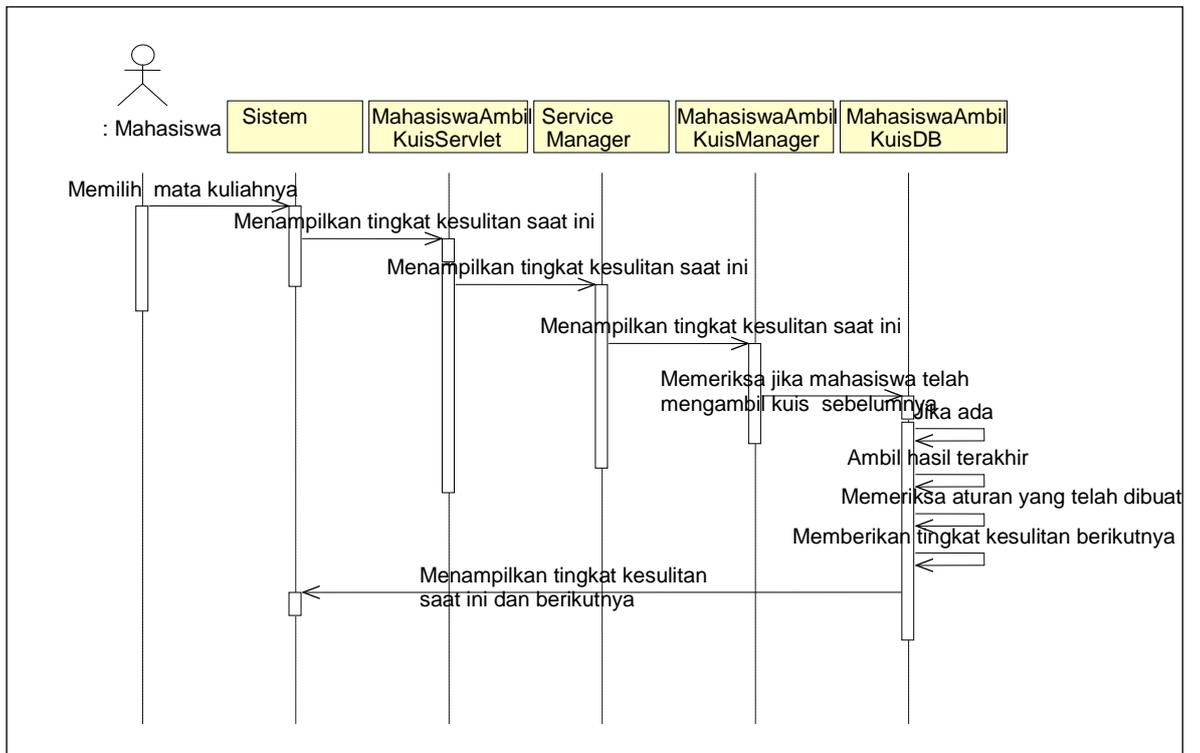
Gambar 3.6 Class Diagram Sistem Kuis Online Adaptif

3.3.3.2 Sequence Diagram untuk masing-masing use case

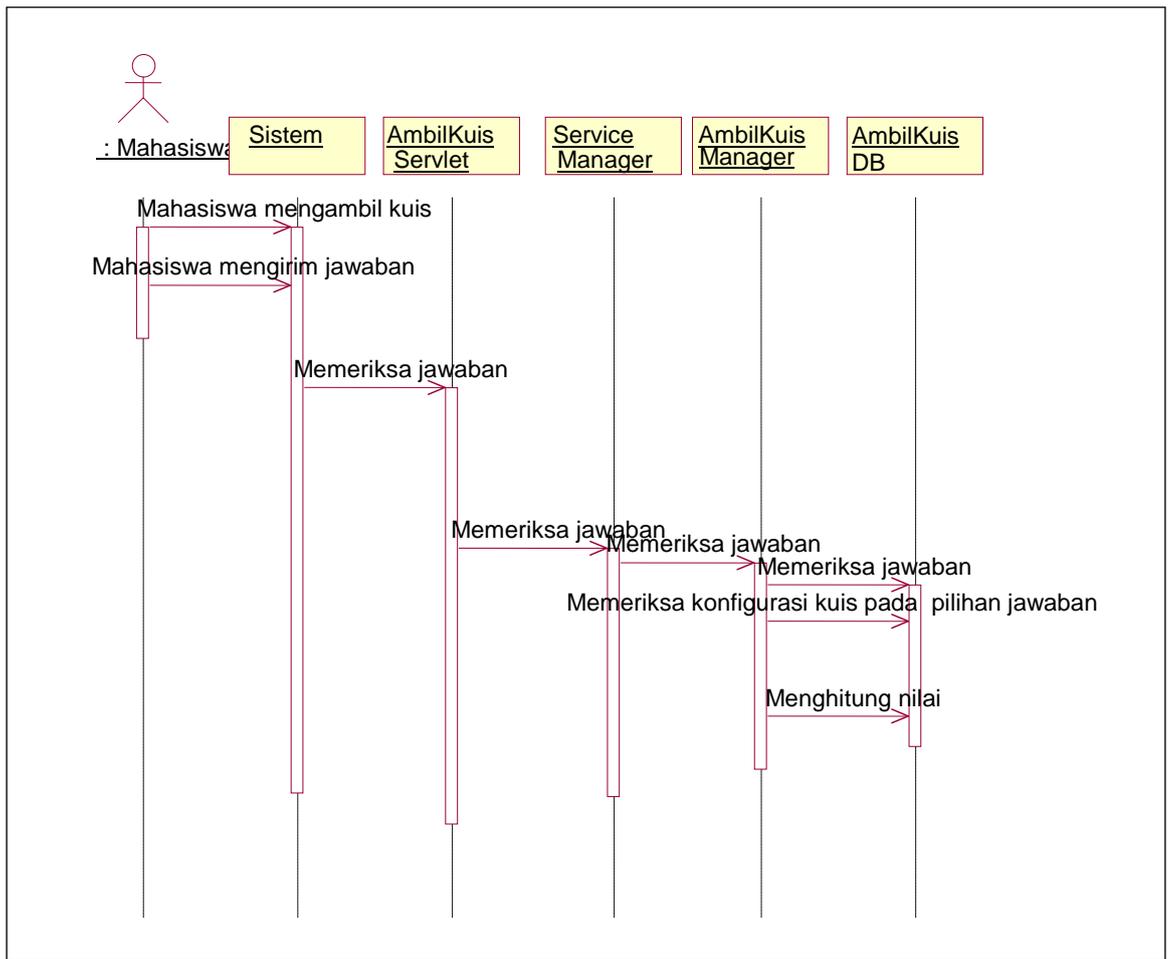
Seperti dijelaskan sebelumnya, setiap *use case* merupakan salah satu skenario dari sistem. Untuk masing-masing skenario, hanya satu *sequence diagram* yang disajikan. Namun demikian, untuk menyederhanakan, hanya skenario modul mahasiswa disajikan. Sebelum mahasiswa diperbolehkan untuk mengambil kuis, mahasiswa diharuskan untuk login ke sistem untuk memvalidasi apakah mahasiswa terdaftar dalam sistem. *Sequence diagram* untuk login mahasiswa ditunjukkan pada Gambar 3.6.a. Fungsi *Mahasiswa Mengambil Kuis* terdiri dari lima *use case*, oleh karena itu ada lima *sequence diagram* disajikan. *Sequence diagram* untuk 'Menampilkan tingkat kesulitan pengetahuan domain sekarang dan berikutnya' disajikan pada Gambar 3.6.b. Gambar 3.6.c menunjukkan *sequence diagram* 'Mengerjakan Kuis' dan Gambar 3.6.d menunjukkan *sequence diagram* 'Berikan Feedback'. *Use case* 'Memilih Pertanyaan' disajikan pada Gambar 3.6.e. *Use case* "Konfigurasi Kuis" disajikan pada Gambar 3.6.f. Diagram terakhir untuk modul mahasiswa yaitu 'Tampilkan Histori Hasil Mahasiswa' ditunjukkan pada Gambar 3.6.g. *Sequence diagram* lainnya disajikan dalam Lampiran .



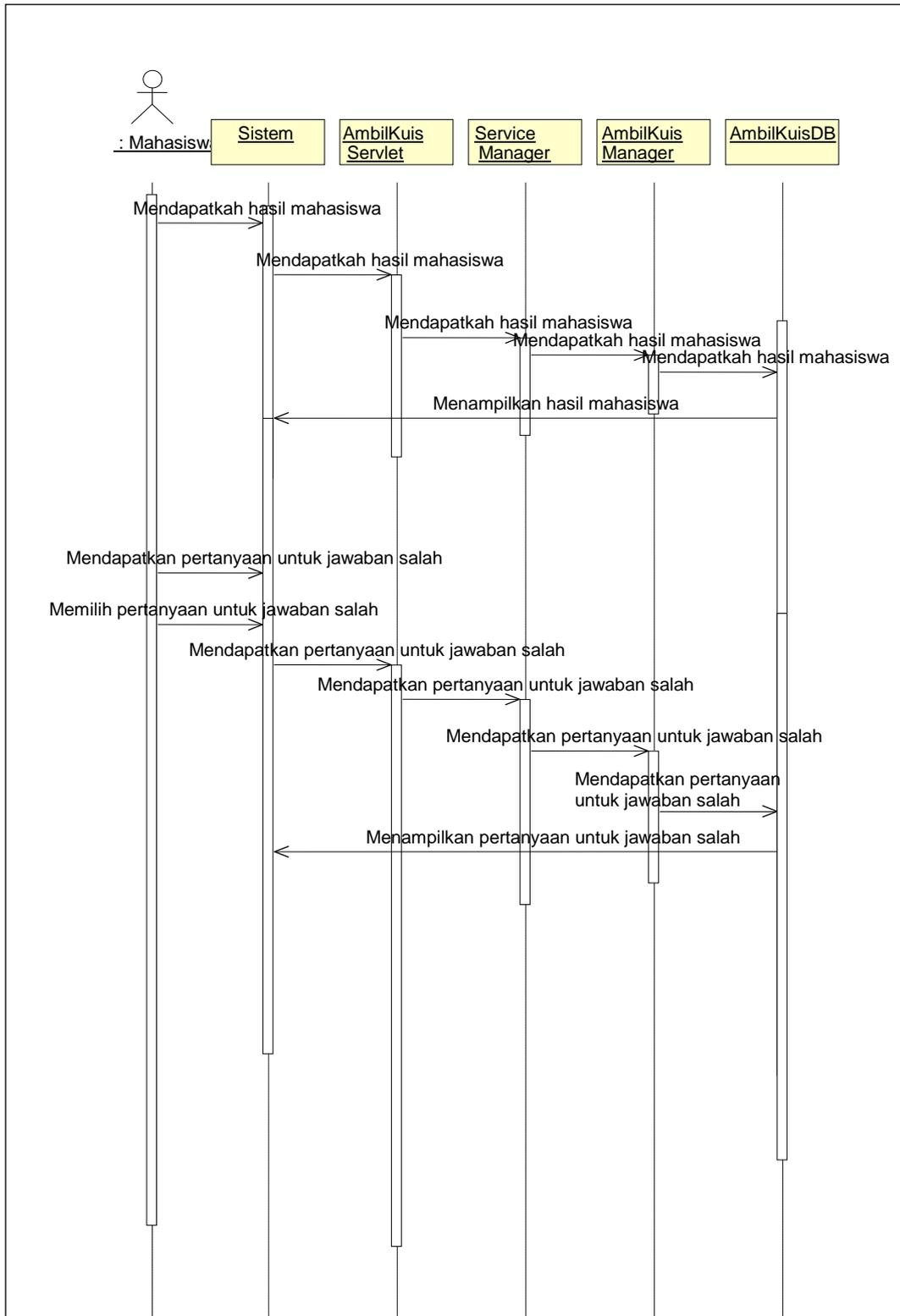
Gambar 3.6.a Sequence Diagram Login Mahasiswa



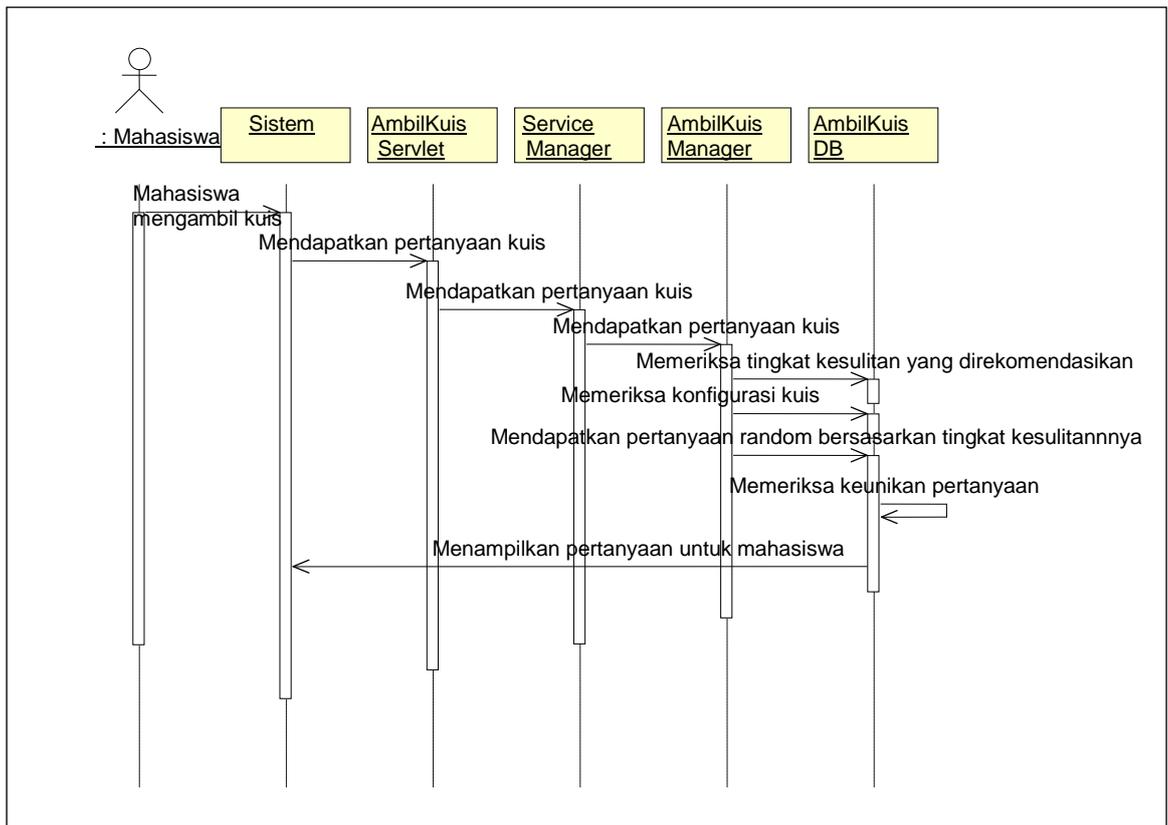
Gambar 3.6.b Sequence Diagram Menampilkan tingkat kesulitan pengetahuan domain mahasiswa saat ini dan berikutnya



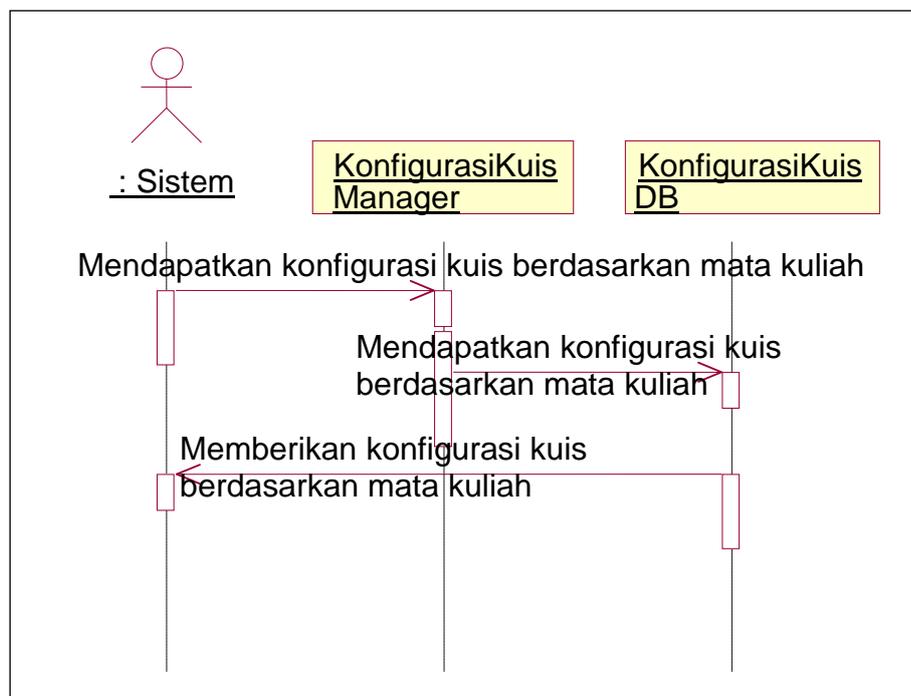
Gambar 3.6.c Sequence Diagram Mengerjakan Kuis



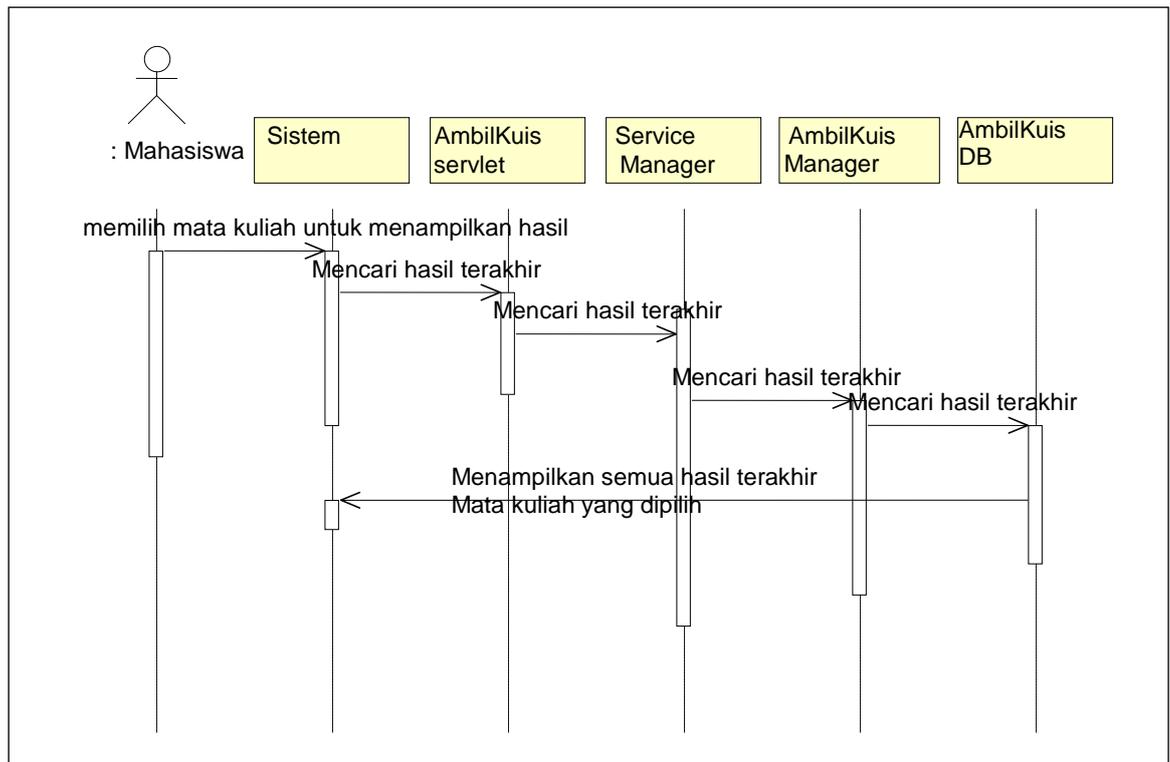
Gambar 3.6.d Sequence Diagram Memberikan Umpan Balik



Gambar 3.6.e Sequence Diagram Memilih Pertanyaan



Gambar 3.6.f Sequence Diagram Mendapatkan Konfigurasi Kuis



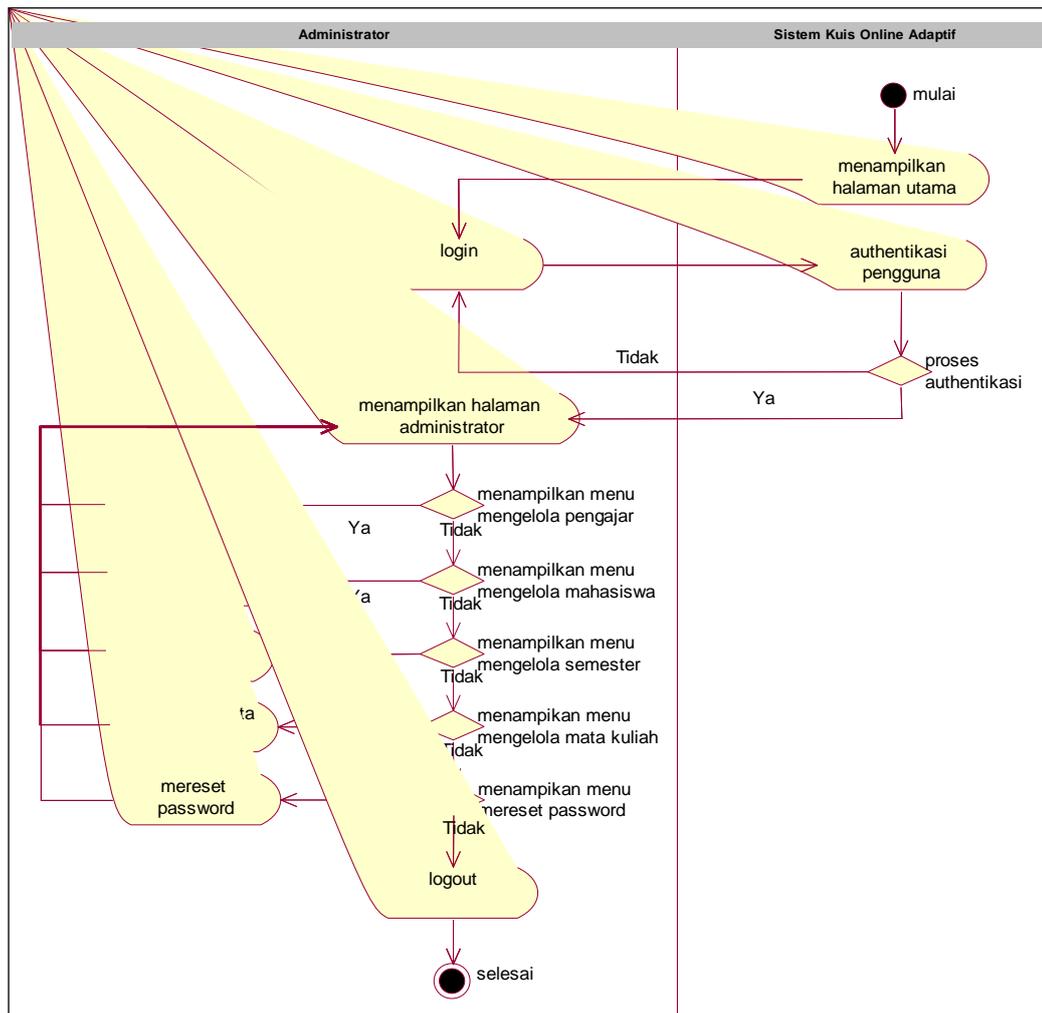
Gambar 3.6.g Sequence Diagram Menampilkan Histori Hasil Kuis Mahasiswa

3.3.3.3 Pemodelan Proses

Berdasarkan identifikasi *use case* yang ada terutama *use case* menurut modul yang ada, akan digambarkan proses secara rinci dari masing-masing *use case* tersebut dengan menggunakan *Diagram Aktifitas*. Setiap *use case* merupakan satu skenario dari sistem, sehingga untuk setiap skenario, hanya satu *Diagram Aktifitas* disajikan. *Diagram Aktifitas* yang akan digambarkan hanya pada proses yang terjadi pada modul Administrator, modul Pengajar, dan modul Mahasiswa.

3.3.3.1.1 Diagram Aktifitas Administrator

Diagram Aktifitas Administrator menggambarkan aktifitas atau proses yang dilakukan oleh seorang administrator terhadap sistem kuis *online adaptif*, berikut ini adalah gambar Diagram Aktifitas Administrator.

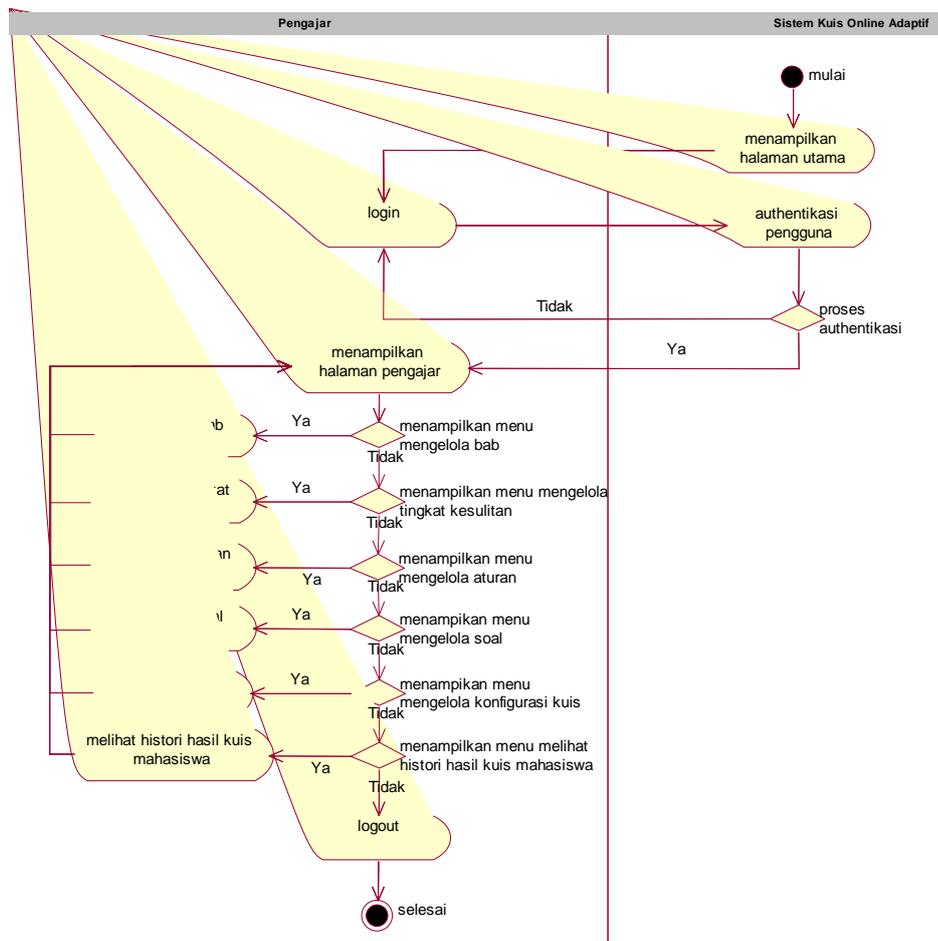


Gambar 3.7 Diagram Aktifitas Administrator.

Pada Gambar 3.7, seorang administrator diharuskan melakukan proses autentikasi untuk dapat masuk ke pengaturan halaman administrator yang berisi menu mengelola pengajar, mengelola mahasiswa, mengelola semester, mengelola mata kuliah, dan mereset password.

3.3.3.1.2 Diagram Aktifitas Pengajar

Diagram Aktifitas Pengajar menggambarkan aktifitas atau proses yang dilakukan oleh seorang pengajar terhadap sistem kuis *online adaptif*, berikut ini adalah gambar Diagram Aktifitas Pengajar.



Gambar 3.8 Diagram Aktifitas Pengajar.

Pada Gambar 3.8, seorang pengajar diharuskan melakukan proses autentikasi untuk dapat masuk ke pengaturan halaman pengajar yang berisi menu mengelola bab, mengelola tingkat kesulitan, mengelola aturan, mengelola soal, mengelola konfigurasi kuis dan melihat histori hasil kuis mahasiswa.

3.3.3.1.3 Diagram Aktifitas Mahasiswa

Diagram Aktifitas Mahasiswa menggambarkan aktifitas atau proses yang dilakukan oleh seorang mahasiswa terhadap sistem kuis *online adaptif* terutama pada saat melakukan kuis, berikut ini adalah gambar *Diagram Aktifitas* Mahasiswa.

mengambil kuis, mahasiswa akan diminta mengerjakan pre test yang sudah disediakan untuk menguji kemampuan mahasiswa sekaligus menentukan awal mulai bab dan tingkat (level) pengetahuan dalam kuis tersebut. Aturan adaptif yang dibuat untuk pre test ini menggunakan IF THEN rule, sebagai berikut :

```
IF mata_kuliah = mata_kuliah_saas_inis AND  
nilai >= nilai_batas_bawah AND nilai <= nilai_batas_atas AND  
waktu <= waktu_yang_ditentukan  
THEN  
no_bab = no_bab_yang_sesuai  
tingkat_kesulitan = tingkat_kesulitan_yang_sesuai
```

Hasil dari pre test disimpan ke dalam histori hasil kuis mahasiswa dan mahasiswa ditunjukkan link untuk mengerjakan awal kuis pada bab dan level sesuai dengan hasil yang didapat, atau kemungkinan sudah dianggap berkompeten dalam kuis tersebut. Proses penilaian berdasarkan aturan adaptif yang diterapkan dalam penilaian kuis ini adalah sebagai berikut :

```
IF mata_kuliah = mata_kuliah_saas_inis AND  
no_bab = no_bab_saas_inis AND  
tingkat_kesulitan = tingkat_kesulitan_saas_inis AND  
nilai >= nilai_batas_bawah AND nilai <= nilai_batas_atas AND  
waktu <= waktu_yang_ditentukan  
THEN  
no_bab = no_bab_yang_sesuai  
tingkat_kesulitan = tingkat_kesulitan_yang_sesuai
```

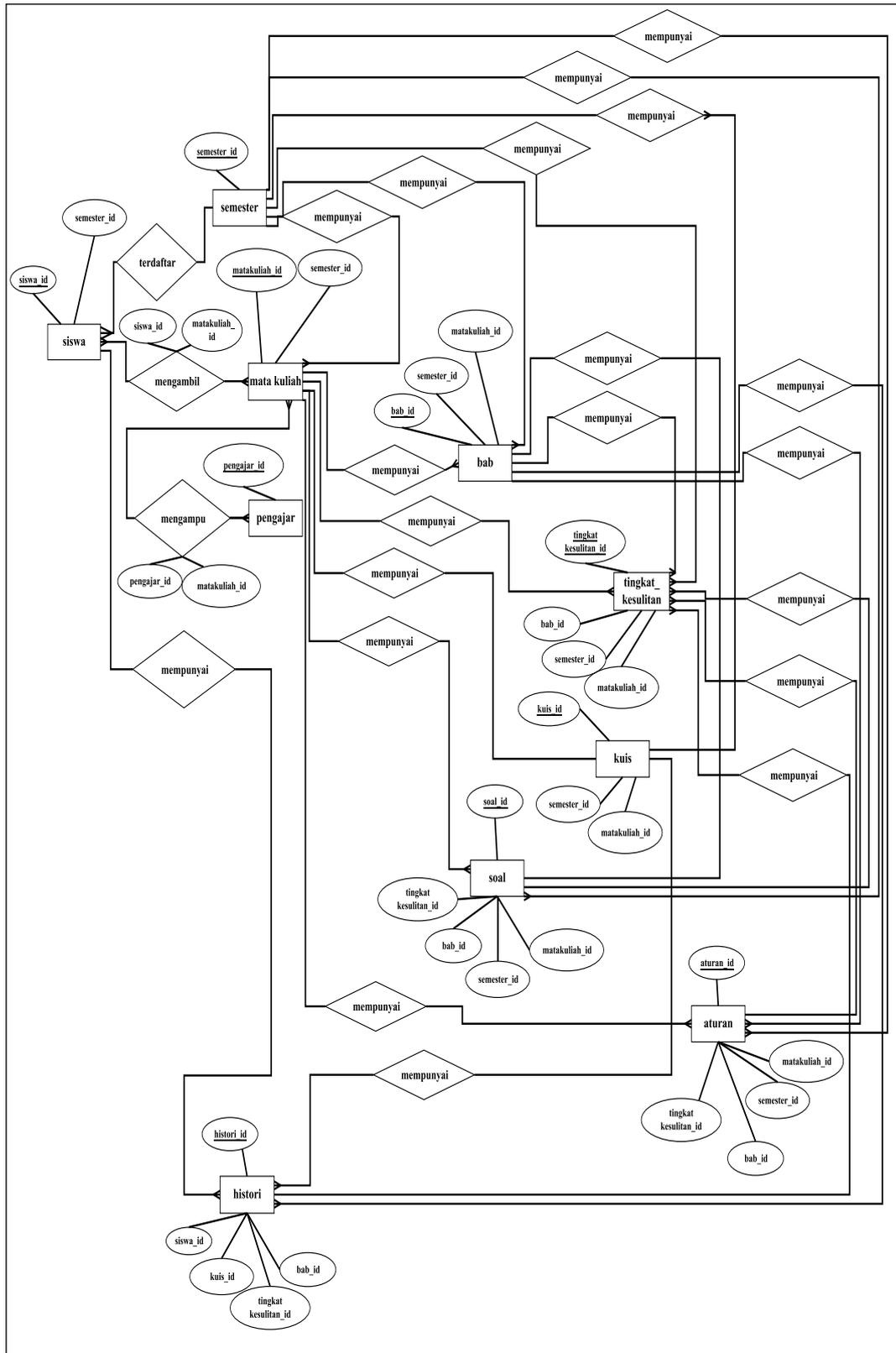
Hasil dari kuis tersebut disimpan ke dalam histori hasil kuis mahasiswa dan mahasiswa ditunjukkan link untuk mengerjakan kuis berikutnya sesuai dengan hasil yang didapat, sampai bab terakhir dengan tingkat kesulitan 3 atau mahasiswa menghentikan kuisnya untuk sementara waktu. Mahasiswa yang akan melanjutkan kuisnya berdasarkan histori terakhir hasil kuis mahasiswa tersebut, jadi tiap mahasiswa akan mengerjakan kuis sesuai dengan subjek dan tingkat kesulitannya

yang mereka peroleh melalui penyesuaian (adaptif). Semua aturan adaptif ditentukan oleh pengajar masing-masing.

3.3.3.4 Pemodelan Data

3.3.3.4.1 Diagram Hubungan Entitas Sistem Kuis Online Adaptif

Pemodelan data diawali dengan merancang *entity relationships diagram (ERD)* / diagram hubungan entitas yang terdiri dari 10 entitas (*entity*) yaitu pengajar, mahasiswa, semester, mata kuliah, bab, tingkat kesulitan, kuis, soal, history kuis, dan aturan, ditambah dua hubungan (*relationships*) yang menjadi entitas (*entity*) yaitu hubungan “mengambil” antara entitas mahasiswa dengan mata kuliah dan hubungan “mengampu” antara entitas pengajar dengan mata kuliah seperti terlihat pada Gambar 3.10.



Gambar 3.10 ERD Sistem Kuis Online Adaptif

Pada Gambar 3.10, masing-masing *entity* mempunyai relasi dan mapping kardinalitas terhadap *entity* lainnya sebagai berikut ;

1. *Entity* Pengajar mempunyai relasi *mengampu* terhadap *entity* Mata Kuliah dengan mapping kardinalitas *many to many*.
2. *Entity* Mahasiswa mempunyai relasi *terdaftar* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
3. *Entity* Mahasiswa mempunyai relasi *mengambil* terhadap *entity* Mata Kuliah dengan mapping kardinalitas *many to many*.
4. *Entity* Mahasiswa mempunyai relasi *mempunyai* terhadap *entity* Histori dengan dengan mapping kardinalitas *one to many*.
5. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
6. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Bab dengan mapping kardinalitas *one to many*.
7. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Tingkat Kesulitan dengan mapping kardinalitas *one to many*.
8. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Kuis dengan mapping kardinalitas *one to one*.
9. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Soal dengan mapping kardinalitas *one to many*.
10. *Entity* Mata Kuliah mempunyai relasi *mempunyai* terhadap *entity* Aturan dengan mapping kardinalitas *one to many*.
11. *Entity* Bab mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.

12. *Entity* Bab mempunyai relasi *mempunyai* terhadap *entity* Soal dengan mapping kardinalitas *one to many*.
13. *Entity* Bab mempunyai relasi *mempunyai* terhadap *entity* Tingkat Kesulitan dengan mapping kardinalitas *one to many*.
14. *Entity* Bab mempunyai relasi *mempunyai* terhadap *entity* Aturan dengan mapping kardinalitas *one to many*.
15. *Entity* Tingkat Kesulitan mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
16. *Entity* Kuis mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
17. *Entity* Kuis mempunyai relasi *mempunyai* terhadap *entity* Histori dengan mapping kardinalitas *one to many*.
18. *Entity* Soal mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
19. *Entity* Soal mempunyai relasi *mempunyai* terhadap *entity* Tingkat Kesulitan dengan mapping kardinalitas *one to many*.
20. *Entity* Aturan mempunyai relasi *mempunyai* terhadap *entity* Semester dengan mapping kardinalitas *many to one*.
21. *Entity* Aturan mempunyai relasi *mempunyai* terhadap *entity* Tingkat Kesulitan dengan mapping kardinalitas *one to many*.
22. *Entity* Histori mempunyai relasi *mempunyai* terhadap *entity* Bab dengan mapping kardinalitas *many to one*.
23. *Entity* Histori mempunyai relasi *mempunyai* terhadap *entity* Tingkat Kesulitan dengan mapping kardinalitas *one to many*.

3.3.3.4.2 Rancangan Database Sistem Kuis Online Adaptif

Berdasarkan diagram *entity relationship* akan ditransformasikan ke dalam bentuk tabel-tabel pendukung utama database sistem kuis online adaptif ini. Adapun tabel-tabel pendukung utama database sistem ini adalah sebagai berikut

1. Tabel pengajar

Tabel pengajar digunakan untuk menyimpan data pengajar yang mengelola sistem kuis online adaptif ini dengan field id sebagai primary key.

Tabel 3.16 Struktur Tabel Pengajar

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nip	int(11)	Nomor Induk Pegawai
nama_depan	varchar(50)	Nama depan pengajar
nama_belakang	varchar(100)	Nama belakang pengajar
alamat1	varchar(255)	Alamat pengajar
alamat2	varchar(255)	Alamat pengajar
alamat3	varchar(255)	Alamat pengajar
no_telpon	varchar(25)	Nomor telepon pengajar
jenis_kelamin	enum('Wanita', 'Pria')	Jenis kelamin pengajar
email	varchar(50)	Email pengajar
tgl_dibuat	datetime	Tanggal akun dibuat
nama_akun	varchar(10)	User ID pengajar
kata_sandi	text	Password pengajar

2. Tabel Pengajar_Matakuliah

Tabel Pengajar_Matakuliah digunakan untuk menyimpan ID pengajar dan ID mata kuliah yang merupakan hasil relasi antar tabel pengajar dan tabel mata kuliah dengan field pengajar_id sebagai foreign key yang merujuk ke field id dalam tabel pengajar dan field matakuliah_id sebagai foreign key yang merujuk ke field id dalam tabel mata kuliah.

Tabel 3.17 Struktur Tabel Pengajar_Matakuliah

Nama Field	Tipe Data	Keterangan
pengajar_id	int(4)	ID pengajar (foreign key)

Nama Field	Tipe Data	Keterangan
matakuliah_id	int(4)	ID mata kuliah (foreign key)

3. Tabel mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa yang mengikuti sistem kuis online adaptif ini dengan field id sebagai primary key, dan field semester_id sebagai foreign key yang merujuk ke field id dalam tabel semester.

Tabel 3.18 Struktur Tabel Mahasiswa

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nim	int(11)	Nomor Induk Mahasiswa
nama_depan	varchar(50)	Nama depan mahasiswa
nama_belakang	varchar(100)	Nama belakang mahasiswa
alamat1	varchar(255)	Alamat mahasiswa
alamat2	varchar(255)	Alamat mahasiswa
alamat3	varchar(255)	Alamat mahasiswa
no_telpon	varchar(25)	Nomor telepon mahasiswa
jenis_kelamin	enum('Wanita', 'Pria')	Jenis kelamin mahasiswa
email	varchar(50)	Email mahasiswa
tgl_dibuat	datetime	Tanggal akun dibuat
nama_akun	varchar(10)	User id mahasiswa
kata_sandi	text	Password mahasiswa
semester_id	smallint(2)	ID semester (foreign key)
kelas	varchar(20)	Kelompok mahasiswa

4. Tabel Mahasiswa_Matakuliah

Tabel Mahasiswa_Matakuliah digunakan untuk menyimpan ID mahasiswa dan ID mata kuliah yang merupakan hasil relasi antar tabel mahasiswa dan tabel mata kuliah dengan field mahasiswa_id sebagai foreign key yang merujuk ke field id dalam tabel mahasiswa dan field matakuliah_id sebagai foreign key yang merujuk ke field id dalam tabel mata kuliah.

Tabel 3.20 Struktur Tabel Mahasiswa_Matakuliah

Nama Field	Tipe Data	Keterangan
mahasiswa_id	int(4)	ID mahasiswa (foreign key)
matakuliah_id	int(4)	ID mata kuliah (foreign key)

5. Tabel Semester

Tabel semester digunakan untuk menyimpan data semester yang ditentukan dalam sistem kuis online adaptif ini dengan field id sebagai primary key.

Tabel 3.21 Struktur Tabel Semester

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nama	varchar(50)	Nama Semester
deskripsi	text	Deskripsi Semester

6. Tabel Matakuliah

Tabel matakuliah digunakan untuk menyimpan data mata kuliah tiap semester yang ada dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan field semester_id sebagai foreign key yang merujuk ke field id dalam tabel semester.

Tabel 3.22 Struktur Tabel Matakuliah

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nama	varchar(100)	Nama mata kuliah
deskripsi	text	Deskripsi mata kuliah
kode	varchar(15)	Kode mata kuliah
semester_id	smallint(2)	ID semester (foreign key)

7. Tabel Bab

Tabel bab digunakan untuk menyimpan data bab (pokok bahasan) tiap mata kuliah yang ada dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan field

semester_id sebagai foreign key yang merujuk ke field id dalam tabel semester, serta field matakuliah_id sebagai foreign key juga yang merujuk ke field id dalam tabel matakuliah.

Tabel 3.23 Struktur Tabel Bab

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nomer	smallint(2)	Nomor Bab
nama	varchar(50)	Nama Bab
deskripsi	text	Deskripsi Bab
semester_id	smallint(2)	ID semester (foreign key)
matakuliah_id	int(4)	ID matakuliah (foreign key)
pretes	tinyint(4)	Status Pretest apa tidak

8. Tabel Kuis

Tabel kuis digunakan untuk menyimpan data kuis yang ada dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan field semester_id sebagai foreign key yang merujuk ke field id dalam tabel semester, serta field matakuliah_id sebagai foreign key juga yang merujuk ke field id dalam tabel matakuliah.

Tabel 3.24 Struktur Tabel Kuis

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nama	varchar(50)	Nama Kuis
deskripsi	text	Deskripsi Kuis
matakuliah_id	int(4)	ID mata kuliah (foreign key)
jumlah_soal	int(3)	Jumlah soal dalam kuis
soal_per_halaman	int(3)	Jumlah soal per halaman
semester_id	int(4)	ID semester (foreign key)

9. Tabel Soal

Tabel soal digunakan untuk menyimpan data soal yang ada dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan empat field sebagai foreign key, yaitu

- a. field semester_id yang merujuk ke field id dalam tabel semester,

- b. field matakuliah_id yang merujuk ke field id dalam tabel matakuliah,
- c. field bab_id yang merujuk ke field id dalam tabel bab,
- d. field kesulitan_id yang merujuk ke field id dalam tabel tingkat kesulitan.

Tabel 3.25 Struktur Tabel Soal

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
pertanyaan	text	Pertanyaan
jawaban1	text	Pilihan jawaban 1
jawaban2	text	Pilihan jawaban 2
jawaban3	text	Pilihan jawaban 3
jawaban4	text	Pilihan jawaban 4
jawaban5	text	Pilihan jawaban 5
jawaban_benar	char(255)	Kunci jawaban multiple choice
matakuliah_id	int(4)	ID mata kuliah (foreign key)
bab_id	int(4)	ID bab (foreign key)
kesulitan_id	int(4)	ID kesulitan (foreign key)
semester_id	int(4)	ID semester (foreign key)
true_false	tinyint(4)	Kunci jawaban true/false
Fill_in_the_blank	varchar(200)	Kunci jawaban fill in the blank
Jenis_soal	tinyint(4)	Status Jenis Soal

10. Tabel TingkatKesulitan

Tabel tingkat kesulitan digunakan untuk menyimpan data tingkat kesulitan yang ditentukan dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan tiga field sebagai foreign key, yaitu

- field semester_id yang merujuk ke field id dalam tabel semester,
- field matakuliah_id yang merujuk ke field id dalam tabel matakuliah,
- field bab_id yang merujuk ke field id dalam tabel bab,

Tabel 3.26 Struktur Tabel TingkatKesulitan

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nomer	smallint(2)	Nomor tingkat kesulitan
nama	varchar(50)	Nama tingkat kesulitan
deskripsi	text	Deskripsi tingkat kesulitan
semester_id	smallint(2)	ID semester (foreign key)
matakuliah_id	int(4)	ID mata kuliah (foreign key)
bab_id	int(4)	ID bab (foreign key)

11. Tabel Aturan

Tabel aturan digunakan untuk menyimpan data aturan (rule) yang ditentukan dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan empat field sebagai foreign key, yaitu

- a. field semester_id yang merujuk ke field id dalam tabel semester,
- b. field matakuliah_id yang merujuk ke field id dalam tabel matakuliah,
- c. field bab_id yang merujuk ke field id dalam tabel bab,
- d. field kesulitan_id yang merujuk ke field id dalam tabel tingkat kesulitan.

Tabel 3.27 Struktur Tabel Aturan

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
nama	varchar(50)	Nama aturan
deskripsi	text	Deskripsi aturan
kesulitan_id	int(4)	ID kesulitan (foreign key)
nilai_kurang_dari	int(3)	Batas bawah nilai
nilai_lebih_dari	int(3)	Batas atas nilai
ke_bab_id	int(4)	ID bab yang direkomendasikan
ke_kesulitan_id	int(4)	ID kesulitan yang direkomendasikan
semester_id	smallint(2)	ID semester (foreign key)
matakuliah_id	int(4)	ID mata kuliah (foreign key)
bab_id	int(4)	ID bab (foreign key)

12. Tabel Histori

Tabel histori digunakan untuk menyimpan data histori hasil kuis mahasiswa yang dalam sistem kuis online adaptif ini dengan field id sebagai primary key dan empat field sebagai foreign key, yaitu

- field kuis_id yang merujuk ke field id dalam tabel kuis,
- field user_id yang merujuk ke field id dalam tabel mahasiswa,
- field bab_id yang merujuk ke field id dalam tabel bab,
- field kesulitan_id yang merujuk ke field id dalam tabel tingkat kesulitan.

Tabel 3.29 Struktur Tabel Histori

Nama Field	Tipe Data	Keterangan
id	int(4)	Primary key
kuis_id	int(4)	ID kuis (foreign key)
user_id	int(4)	User ID mahasiswa (foreign key)
Tgl_diambil	datetime	Tanggal kuis diambil
jawaban_benar	int(4)	Jumlah jawaban benar
jawaban_salah	int(4)	Jumlah jawaban salah
nilai	int(3)	Nilai kuis
kesulitan_id	int(4)	ID tingkat kesulitan (foreign key)
nama_kesulitan	varchar(20)	Nama tingkat kesulitan
bab_id	int(4)	ID bab (foreign key)
nama_bab	varchar(20)	Nama bab
ke_bab_id	int(4)	ID bab yang direkomendasikan
ke_nama_bab	varchar(255)	Nama bab yang direkomendasikan
ke_kesulitan_id	int(4)	ID kesulitan yang direkomendasikan
ke_nama_kesulitan	varchar(20)	Nama kesulitan yang direkomendasikan

3.3.3.4.3 Relationship antar tabel

Tabel-tabel dalam database sistem kuis online adaptif ini saling berelasi satu sama lain, seperti yang digambarkan dalam Gambar 3.11, dengan penjelasan sebagai berikut :

1. Tabel mahasiswa memiliki field id sebagai primary key yang berelasi dengan
 - a. field mahasiswa_id sebagai foreign key dalam tabel mahasiswa_matakuliah dengan kardinalitas *one to many*.
 - b. field user_id sebagai foreign key dalam tabel histori dengan kardinalitas *one to many*.
2. Tabel pengajar memiliki field id sebagai primary key yang berelasi dengan field pengajar_id sebagai foreign key dalam tabel pengajar_matakuliah dengan kardinalitas *one to many*.
3. Tabel semester memiliki field id sebagai primary key yang berelasi dengan
 - a. field semester_id sebagai foreign key dalam tabel mahasiswa dengan kardinalitas *one to many*.
 - b. field semester_id sebagai foreign key dalam tabel mata kuliah dengan kardinalitas *one to many*.
 - c. field semester_id sebagai foreign key dalam tabel bab dengan kardinalitas *one to many*.
 - d. field semester_id sebagai foreign key dalam tabel kuis dengan kardinalitas *one to many*.
 - e. field semester_id sebagai foreign key dalam tabel soal dengan kardinalitas *one to many*.
 - f. field semester_id sebagai foreign key dalam tabel tingkat kesulitan dengan kardinalitas *one to many*.
 - g. field semester_id sebagai foreign key dalam tabel aturan dengan kardinalitas *one to many*.
4. Tabel mata kuliah memiliki field id sebagai primary key yang berelasi dengan

- a. field `matakuliah_id` sebagai foreign key dalam tabel `mahasiswa_matakuliah` dengan kardinalitas *one to many*.
 - b. field `matakuliah_id` sebagai foreign key dalam tabel `bab` dengan kardinalitas *one to many*.
 - c. field `matakuliah_id` sebagai foreign key dalam tabel `bab` dengan kardinalitas *one to many*.
 - d. field `matakuliah_id` sebagai foreign key dalam tabel `kuis` dengan kardinalitas *one to many*.
 - e. field `matakuliah_id` sebagai foreign key dalam tabel `soal` dengan kardinalitas *one to many*.
 - f. field `matakuliah_id` sebagai foreign key dalam tabel `tingkat kesulitan` dengan kardinalitas *one to many*.
 - g. field `matakuliah_id` sebagai foreign key dalam tabel `aturan` dengan kardinalitas *one to many*.
5. Tabel `bab` memiliki field `id` sebagai primary key yang berelasi dengan
 - a. field `bab_id` sebagai foreign key dalam tabel `soal` dengan kardinalitas *one to many*.
 - b. field `bab_id` sebagai foreign key dalam tabel `tingkat kesulitan` dengan kardinalitas *one to many*.
 - c. field `bab_id` sebagai foreign key dalam tabel `aturan` dengan kardinalitas *one to many*.
 - d. field `bab_id` sebagai foreign key dalam tabel `histori` dengan kardinalitas *one to many*.
 6. Tabel `kuis` memiliki field `id` sebagai primary key yang berelasi dengan field `kuis_id` sebagai foreign key dalam tabel `histori` dengan kardinalitas *one to many*.
 7. Tabel `tingkat kesulitan` memiliki field `id` sebagai primary key yang berelasi dengan
 - a. field `kesulitan_id` sebagai foreign key dalam tabel `soal` dengan kardinalitas *one to many*.

- b. field `kesulitan_id` sebagai foreign key dalam tabel aturan dengan kardinalitas *one to many*.
- c. field `kesulitan_id` sebagai foreign key dalam tabel histori dengan kardinalitas *one to many*.

3.3.3.5 Desain User Interface

Desain *user interface* dari sistem kuis online adaptif ini terdiri dari dua desain, yaitu desain *user interface* dari sisi *back-end*, dan desain *user interface* dari sisi *front-end*. *User interface* dari sisi *back-end* lebih banyak nantinya digunakan oleh administrator dan pengajar dalam mengelola sistem ini, sedangkan *user interface* dari sisi *front-end* ditujukan untuk mahasiswa yang mengikuti sistem kuis online adaptif ini.

3.3.3.5.1 Desain User Interface Sisi Back-End

Desain *user interface* dari sisi *back-end* mengadopsi dari desain *user interface* yang sering digunakan oleh komponen dalam *content management system*. Berikut ini desain *user interface* yang akan digunakan dari sisi *back-end* :

1. Desain Halaman Pengelolaan Pengajar

Halaman ini nantinya akan digunakan oleh administrator untuk mengelola data pengajar dengan hanya menampilkan NIP, Nama, E-mail, Jenis Kelamin, dan Tanggal dibuat yang berkaitan dengan seorang pengajar yang dilengkapi kotak cek (*check box*) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (*delete*), edit, dan tambah (*new*) seorang pengajar. Desain halaman pengelolaan pengajar ditunjukkan dalam Gambar 3.12



Gambar 3.12 Desain Halaman Pengelolaan Pengajar

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record pengajar dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record pengajar nantinya dapat dilakukan dengan meng-klik icon *New*. Desain

halaman tambah pengajar ditunjukkan dalam Gambar 3.13, *user interface* ini nantinya berhubungan dengan tabel pengajar.

Gambar 3.13 Desain Halaman Tambah Pengajar

2. Desain Halaman Pengelolaan Mahasiswa

Halaman ini nantinya akan digunakan oleh administrator untuk mengelola data mahasiswa dengan hanya menampilkan NIM, Nama, E-mail, Jenis Kelamin, dan Tanggal dibuat yang berkaitan dengan seorang mahasiswa yang dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) seorang mahasiswa. Desain halaman pengelolaan mahasiswa ditunjukkan dalam Gambar 3.14

Gambar 3.14 Desain Halaman Pengelolaan Mahasiswa

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record mahasiswa dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record mahasiswa nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah mahasiswa ditunjukkan dalam Gambar 3.15, *user interface* ini nantinya berhubungan dengan tabel mahasiswa dan tabel semester.

TAMBAH SISWA

Details

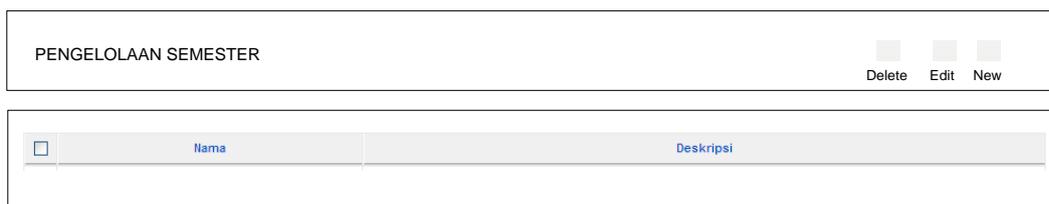
NIK:	<input type="text"/>
Nama Lengkap:	<input type="text"/>
Nama Keluarga:	<input type="text"/>
Semester:	Pilih Semester <input type="button" value="v"/>
Alamat:	<input type="text"/>
Handphone:	<input type="text"/>
Jenis Kelamin:	Pria <input type="button" value="v"/>
Alamat-1:	<input type="text"/>
Alamat-2:	<input type="text"/>
Alamat-3:	<input type="text"/>
No. Telepon:	<input type="text"/>
Tanggal Masuk:	2011-04-24 <input type="button" value="v"/>
Mata Kuliah:	Pengenalan Teknologi Informasi <input type="button" value="v"/>
Uraian Detail:	<input type="text"/>
Mata Studi Baru:	<input type="text"/>
Mata Studi Verifikasi:	<input type="text"/>

Gambar 3.15 Desain Halaman Tambah Mahasiswa

3. Desain Halaman Pengelolaan Semester

Halaman ini nantinya akan digunakan oleh administrator untuk mengelola data semester dengan menampilkan Nama dan Deskripsi berkaitan dengan semester, yang dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) semester.

Desain halaman pengelolaan semester ditunjukkan dalam Gambar 3.16



The image shows the header and table structure of the Semester Management page. The header is a rectangular box containing the text "PENGLOLAAN SEMESTER" on the left and three icons labeled "Delete", "Edit", and "New" on the right. Below the header is a table with two columns: "Nama" and "Deskripsi". A checkbox is located at the beginning of the table row.

Gambar 3.16 Desain Halaman Pengelolaan Semester

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record semester dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record semester nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah semester ditunjukkan dalam Gambar 3.17, *user interface* ini nantinya berhubungan dengan tabel semester.



The image shows the header and form of the Add Semester page. The header is a rectangular box containing the text "TAMBAH SEMESTER" on the left and two icons labeled "Save" and "Cancel" on the right. Below the header is a form titled "Details" with two input fields: "Nama:" and "Deskripsi:".

Gambar 3.17 Desain Halaman Tambah Semester

4. Desain Halaman Pengelolaan Mata Kuliah

Halaman ini nantinya akan digunakan oleh administrator untuk mengelola data mata kuliah dengan menampilkan ID, Kode, Nama, Deskripsi, dan Semester yang berkaitan dengan mata kuliah, dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) mata kuliah. Desain halaman pengelolaan mata kuliah ditunjukkan dalam Gambar 3.18

<input type="checkbox"/>	ID	Kode	Nama	Deskripsi	Semester
--------------------------	----	------	------	-----------	----------

Gambar 3.18 Desain Halaman Pengelolaan Mata Kuliah

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record mata kuliah dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record mata kuliah nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah mata kuliah ditunjukkan dalam Gambar 3.19, *user interface* ini nantinya berhubungan dengan tabel mata kuliah dan tabel semester.

Details

Kode:	<input type="text"/>
Nama:	<input type="text"/>
Deskripsi:	<input type="text"/>
Semester:	Pilih Semester <input type="button" value="v"/>

Gambar 3.19 Desain Halaman Tambah Mata Kuliah

5. Desain Halaman Pengelolaan Bab

Halaman ini nantinya akan digunakan oleh pengajar untuk mengelola data bab dengan menampilkan Nama, Deskripsi, Mata Kuliah dan Semester yang berkaitan dengan bab, dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan

disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) bab. Desain halaman pengelolaan bab ditunjukkan dalam Gambar 3.20



Gambar 3.20 Desain Halaman Pengelolaan Bab

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record bab dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record bab nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah bab ditunjukkan dalam Gambar 3.21, *user interface* ini nantinya berhubungan dengan tabel bab, tabel semester dan tabel mata kuliah.



Gambar 3.21 Desain Halaman Tambah Bab

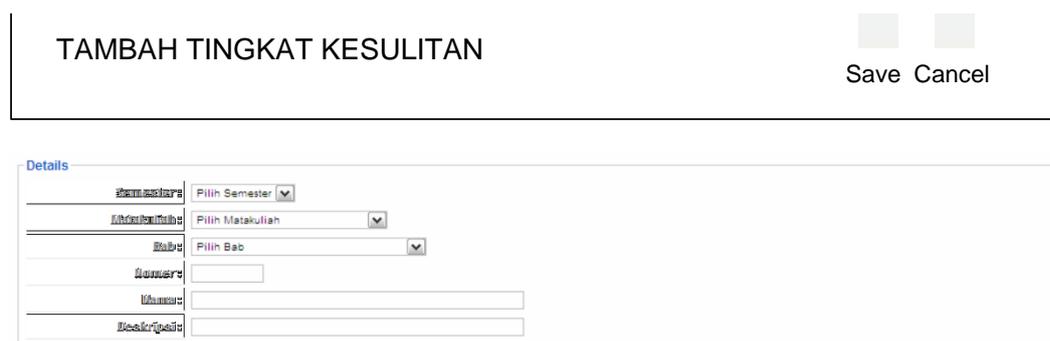
6. Desain Halaman Pengelolaan Tingkat Kesulitan

Halaman ini nantinya akan digunakan oleh pengajar untuk mengelola data tingkat kesulitan dengan menampilkan Nama, Deskripsi, Bab, dan Mata Kuliah yang berkaitan dengan tingkat kesulitan, dilengkapi kotak cek (*check box*) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (*delete*), edit, dan tambah (*new*) tingkat kesulitan. Desain halaman pengelolaan tingkat kesulitan ditunjukkan dalam Gambar 3.22



Gambar 3.22 Desain Halaman Pengelolaan Tingkat Kesulitan

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record tingkat kesulitan dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record tingkat kesulitan nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah tingkat kesulitan ditunjukkan dalam Gambar 3.23, *user interface* ini nantinya berhubungan dengan tabel tingkat kesulitan, tabel semester, tabel mata kuliah, dan tabel bab.



Gambar 3.23 Desain Halaman Tambah Tingkat Kesulitan

7. Desain Halaman Pengelolaan Kuis

Halaman ini nantinya akan digunakan oleh pengajar untuk mengelola data kuis dengan menampilkan ID, Nama, Deskripsi, dan Mata Kuliah yang berkaitan dengan kuis, dilengkapi kotak cek (*check box*) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (*delete*), edit, dan tambah (*new*) kuis. Desain halaman pengelolaan kuis ditunjukkan dalam Gambar 3.24

PENGELOLAAN KUIS				<input type="button" value="Delete"/> <input type="button" value="Edit"/> <input type="button" value="New"/>
<input type="checkbox"/>	ID	Nama	Deskripsi	Matakuliah

Gambar 3.24 Desain Halaman Pengelolaan Kuis

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record kuis dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record kuis nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah kuis ditunjukkan dalam Gambar 3.25, *user interface* ini nantinya berhubungan dengan tabel kuis, tabel semester, dan tabel mata kuliah.

TAMBAH KUIS		<input type="button" value="Save"/> <input type="button" value="Cancel"/>
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Details</p> <p>Semester : <input type="text" value="Pilih Semester"/></p> <p>Matakuliah : <input type="text" value="Pilih Matakuliah"/></p> <p>Nama : <input type="text"/></p> <p>Deskripsi : <input type="text"/></p> <p>Jumlah Soal : <input type="text" value="0"/></p> <p>Soal Per Halaman : <input type="text" value="0"/></p> </div>		

Gambar 3.25 Desain Halaman Tambah Kuis

8. Desain Halaman Pengelolaan Soal

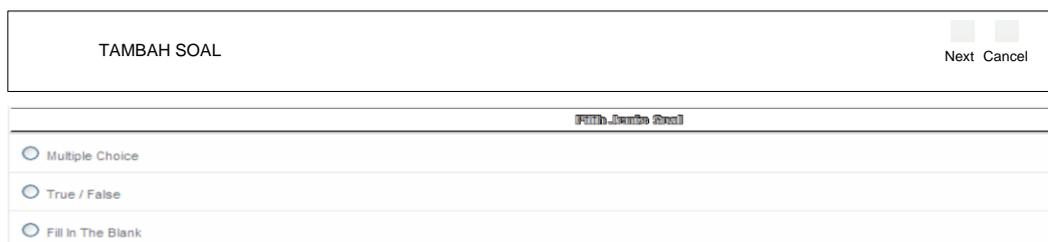
Halaman ini nantinya akan digunakan oleh pengajar untuk mengelola data soal dengan menampilkan ID, Pertanyaan, Mata Kuliah, Bab, dan Tingkat Kesulitan yang berkaitan dengan soal, dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) soal. Desain halaman pengelolaan kuis ditunjukkan dalam Gambar 3.26



<input type="checkbox"/>	ID	Pertanyaan	Matakuliah	Bab	Tingkat Kesulitan
--------------------------	----	------------	------------	-----	-------------------

Gambar 3.26 Desain Halaman Pengelolaan Soal

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record kuis dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record kuis nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman pemilihan jenis soal ditunjukkan dalam Gambar 3.27, yang memuat tiga pilihan jenis soal, dalam desain halaman tersebut juga disediakan icon *Next* untuk menuju halaman tambah soal sesuai dengan jenis soal yang dipilih.



Pilih Jenis Soal

- Multiple Choice
- True / False
- Fill in The Blank

Gambar 3.27 Desain Halaman Pemilihan Jenis Soal

Desain halaman tambah soal terdiri dari tiga desain sesuai dengan tipe soal yang didukung oleh kuis ini, yaitu *multiple choice* (pilihan ganda) ditunjukkan dalam Gambar 3.28, *true/false* (benar/salah) ditunjukkan dalam Gambar 3.29, dan *fill in the blank* (isian

singkat) ditunjukkan dalam Gambar 3.30. Ketiga *user interface* ini nantinya berhubungan dengan tabel soal, tabel semester, tabel mata kuliah, tabel bab, dan tabel tingkat kesulitan.

Gambar 3.28 Desain Halaman Tambah Soal Multiple Choice

Gambar 3.29 Desain Halaman Tambah Soal True/False

Gambar 3.30 Desain Halaman Tambah Soal Fill in the Blank

9. Desain Halaman Pengelolaan Aturan

Halaman ini nantinya akan digunakan oleh pengajar untuk mengelola data aturan dengan menampilkan Nama Rule, Tingkat Kesulitan, Bab, Mata Kuliah, Semester, Nilai Lebih Dari, Nilai Kurang Dari, Bab Tujuan, dan Level Tujuan yang berkaitan dengan aturan, dilengkapi kotak cek (check box) untuk memilih, dalam halaman tersebut juga akan disediakan icon untuk melakukan operasi hapus (delete), edit, dan tambah (new) aturan. Desain halaman pengelolaan aturan ditunjukkan dalam Gambar 3.31.

PENGELOLAAN ATURAN										
<input type="checkbox"/>	Nama Rule	Tingkat Kesulitan	Bab	Matakuliah	Semester	Nilai Lebih Dari	Nilai Kurang Dari	Bab Tujuan	Level Tujuan	

Gambar 3.31 Desain Halaman Pengelolaan Aturan

Kotak cek (*check box*) yang disediakan digunakan untuk memilih satu atau lebih record aturan dengan tujuan untuk dihapus atau diperbaiki datanya, sedangkan untuk menambah record aturan nantinya dapat dilakukan dengan meng-klik icon *New*. Desain halaman tambah aturan ditunjukkan dalam Gambar 3.32, *user interface* ini nantinya berhubungan dengan tabel aturan, tabel semester, tabel mata kuliah, tabel bab, tabel tingkat kesulitan, .

Pengelolaan Aturan

Aturan Detail

Matakuliah:

Bab:

Tingkat Kesulitan:

Nama Rule:

Deskripsi:

Jika Nilai Lebih Dari:

Dan Nilai Kurang Dari:

Waktu: Menit

Matakuliah:

Ke Bab:

Ke Tingkat Kesulitan:

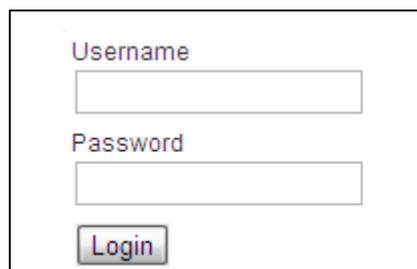
Gambar 3.32 Desain Halaman Tambah Aturan

3.3.3.5.2 Desain User Interface Sisi Front-End

Desain *user interface* dari sisi *front-end* inilah yang nantinya sering digunakan oleh mahasiswa sebagai peserta kuis online adaptif ini. Berikut ini desain *user interface* yang akan digunakan dari sisi *front-end* :

1. Desain Halaman Login Mahasiswa

Halaman ini nantinya akan digunakan oleh mahasiswa untuk melakukan *login* ke sistem kuis online adaptif , dalam halaman *login* tersebut, disediakan kotak isian user name dan password. Desain halaman *login* mahasiswa ditunjukkan dalam Gambar 3.33. *User interface* ini nantinya berhubungan dengan tabel mahasiswa.

The image shows a simple login form with a white background and a thin black border. It contains three elements: a text input field labeled 'Username' in blue text, a second text input field labeled 'Password' in blue text, and a rectangular button labeled 'Login' in blue text with a light gray gradient and a thin black border.

Gambar 3.33 Desain Halaman Login Mahasiswa

2. Desain Halaman Kuis Mahasiswa

Halaman ini nantinya akan digunakan oleh mahasiswa untuk mengerjakan soal-soal kuis yang diambil. Desain halaman kuis mahasiswa ditunjukkan dalam Gambar 3.34. *User interface* ini nantinya berhubungan dengan tabel mahasiswa, tabel kuis, tabel soal, dan tabel histori kuis.

KUIS

Daftar pertanyaan beserta pilihan jawaban

Pertanyaan Multiple Choice

- Pilihan jawaban1
- Pilihan jawaban2
- Pilihan jawaban3
- Pilihan jawaban4
- Pilihan jawaban5

Pertanyaan True/False

- Benar
- Salah

Pertanyaan Fill in the Blank

Submit

Gambar 3.34 Desain Halaman Kuis Mahasiswa

3. Desain Halaman Histori Hasil Kuis Mahasiswa

Halaman ini nantinya akan digunakan oleh mahasiswa untuk melihat histori hasil kuis yang pernah dikerjakan. Desain halaman kuis mahasiswa ditunjukkan dalam Gambar 3.35. *User interface* ini nantinya berhubungan dengan tabel mahasiswa, tabel kuis, dan tabel histori kuis.



Tanggal	Bab	Level	Nilai
---------	-----	-------	-------

Gambar 3.35 Desain Halaman Output Histori Hasil Kuis Mahasiswa

3.3.3.6 Pengembangan Komponen Sistem Kuis Online Adaptif

Komponen sistem ini nantinya memiliki dua bagian utama yaitu bagian administrator dan bagian site. Bagian administrator atau *back-end* menyediakan antarmuka untuk administrator dan pengajar dalam melakukan konfigurasi dan manajemen terhadap berbagai aspek dari komponen, antarmuka ini dapat diakses melalui aplikasi administrator *content management system*. Bagian site atau *front-end* menyediakan antarmuka untuk mahasiswa menggunakan sistem kuis online adaptif ini.

Dalam metode pembuatan komponen terdiri atas beberapa hal yang perlu untuk diperhatikan yaitu :

1. Struktur dari komponen
2. Proses eksekusi komponen
3. Registrasi basis data
4. Pembuatan toolbar.

3.3.3.6.1 Struktur Komponen

Content Management System Joomla menggunakan pola penamaan khusus yang digunakan oleh semua komponen joomla. Tiap-tiap komponen yang ada di dalam sistem

memiliki nama yang unik dan pada nama-nama tersebut tidak mengandung spasi. Kode-kode yang menyusun suatu komponen dibagi ke dalam 2 direktori, tiap-tiap direktori mengandung nama komponen yang menggunakan prefik *com_*. Misalnya nama sebuah komponen adalah mahasiswa, maka untuk pembuatan komponen tersebut dua buah direktori yang harus dibuat adalah :

1. Sebuah direktori yang dinamakan *com_mahasiswa* yang diletakkan di dalam direktori yang bernama *components* untuk keperluan *front-end*.
2. Sebuah direktori yang bernama *com_mahasiswa* yang dibuat di dalam direktori *components*, dimana direktori *components* ini terdapat di dalam direktori administrator untuk keperluan *back-end*.

Ketika komponen di-load dari *front-end*, Joomla akan mencari sebuah file yaitu file komponen dimana file komponen tersebut memiliki nama yang unik yang berekstensi *.php*. Oleh karena itu dibuat file *mahasiswa.php* (dalam contoh diatas nama komponen adalah mahasiswa) di dalam direktori *components/com_mahasiswa*. Demikian juga halnya dengan proses menjalankan komponen di *back-end* memerlukan sebuah file dengan nama yang diawali dengan *admin.* dan diikuti dengan nama komponen dan akhiran yang berekstensi *.php* (*admin.mahasiswa.php*). File *admin.mahasiswa.php* dibuat di dalam direktori *administrator/components/com_mahasiswa*. Kedua file tersebut untuk sementara dibiarkan kosong.

3.3.3.6.2 Proses Eksekusi Komponen

Semua *front-end request* dalam Joomla ditangani oleh *index.php* di dalam direktori *root*. File *index.php* ini nanti yang akan diakses ketika pengguna akan mengakses situs. Komponen-komponen yang berbeda dapat di-load dengan mengatur setting pada variabel *option* dalam *URL GET string*. Jika misalnya instalasi Joomla dilakukan pada web server lokal di dalam direktori yang dinamakan Joomla, maka

URL yang bisa digunakan untuk mengakses situs adalah : <http://localhost/joomla/index.php> sedangkan komponen front end dapat diakses melalui alamat : http://localhost/joomla/index.php?option=com_mahasiswa melalui internet browser. Jika belum ada penambahan kode pada file *reviews.php*, tidak akan ada komponen (nama komponen) yang akan ditampilkan pada layar . Untuk dapat menampilkan komponen tersebut pada *front-end*, harus dilakukan pengkodean pada file *reviews.php* yang ada di dalam direktori *components/com_mahasiswa/*, sedangkan untuk dapat menampilkan komponen dalam *back-end*, harus dilakukan pengkodean pada file *admin.mahasiswa.php* yang ada dalam direktori *administrator/components/com_mahasiswa/*.

3.3.3.6.3 Registrasi Komponen ke dalam Basis Data

Meskipun bagian *front end* dari komponen bisa diakses dengan menggunakan suatu URL misalnya http://localhost/joomla/index.php?option=com_mahasiswa, cara ini tidak akan bisa diterima oleh user. User memerlukan cara yang lebih mudah dipahami, cara yang *user friendly*. Untuk mengakomodasi hal tersebut diperlukan navigasi. Navigasi ini bisa dilakukan jika komponen telah melakukan registrasi ke dalam basis data joomla.

Registrasi ini dilakukan dengan cara menambahkan sebuah baris pada tabel komponen *jos_components* (dengan asumsi *database prefix* yang digunakan adalah *jos_*). Registrasi ini dilakukan dengan menggunakan query jika bekerja dengan menggunakan command-line interface dengan contoh sebagai berikut :

```
INSERT INTO jos_components (name, link, admin_menu_link,
admin_menu_alt,'option',admin_menu_img,params)VALUES ('Komponen
Mahasiswa','option=com_mahasiswa','Mengelola Mahasiswa',
'com_mahasiswa','js/ThemeOffice/component.png','')
```

Jika bekerja dengan menggunakan *GUI* atau *web based database manager* seperti *phpMyAdmin*, caranya dengan mengisi *Komponen Mahasiswa* untuk *name*, *option=com_reviews* untuk *link* dan *admin_menu_link*, *Mengelola Mahasiswa* untuk

admin_menu_alt, *com_reviews* untuk *option* dan *js/ThemeOffice/component.png* untuk *admin_menu_img.png*. *Field-field* yang lain yang terdapat pada tabel *jos_components* dibiarkan kosong, dimana *default* untuk *field menuid*, *parent*, *ordering*, *iscore* adalah 0, sedangkan *default* untuk *enabled* adalah 1. Dengan menambahkan *record* tersebut di atas, sistem akan memperoleh beberapa informasi dasar tentang komponen yang akan ditambahkan.

Setelah selesai melakukan registrasi komponen pada basis data joomla, komponen yang akan ditambahkan (dalam contoh di atas adalah Komponen Mahasiswa) akan tampil pada *pulldown menu* pada menu *Components*. Hal ini membuktikan bahwa komponen Komponen Mahasiswa telah terdaftar, sehingga link untuk *front end* bisa dibuat. Caranya dengan melalui menu pada bagian administrator joomla, yaitu : pertama klik *Menus – Main Menu* dan klik *New*, lalu pilih *Komponen Mahasiswa* pada tampilan layar, lalu mengisikan *Mahasiswa* untuk *name*. Setelah disimpan dengan klik *save*, link untuk komponen baru sudah bisa terlihat pada bagian *Main Menu* pada tampilan layar *front end* joomla.

Setelah melakukan semua proses di atas, coding untuk pembuatan komponen bisa dilakukan. Hal yang perlu menjadi catatan adalah bahwa semua *front end request* harus melalui http://localhost/joomla/index.php?option=com_mahasiswa sedangkan semua *back end request* yang dilakukan pengguna harus melalui http://localhost/joomla/administrator/index.php?option=com_mahasiswa. Pengkode-an komponen dapat memanfaatkan toolbar, kelas-kelas basis data dan file-file pustaka yang sudah tersedia di dalam joomla

3.3.3.6.4 Pembuatan Toolbar Komponen

Meskipun bagian *front end* dari komponen bisa diakses dengan menggunakan suatu URL misalnya http://localhost/joomla/index.php?option=com_mahasiswa, cara ini tidak akan bisa diterima oleh user. User memerlukan cara yang lebih mudah dipahami, cara

yang *user friendly*. Untuk mengakomodasi hal tersebut diperlukan navigasi. Navigasi ini bisa dilakukan jika komponen telah melakukan registrasi ke dalam basis data Joomla.

Semua komponen yang terdapat pada core Joomla menggunakan toolbar dengan buttons yang sama, misalnya button untuk saving, deleting, editing dan publishing. Button yang sudah disediakan oleh core Joomla ini juga dapat dimanfaatkan untuk toolbar pada komponen baru yang akan dibuat. Pembuatan toolbar ini dimulai dengan membuat sebuah file PHP dengan nama (dalam contoh kasus komponen Mahasiswa) yang mengikuti suatu aturan penamaan atau pola tertentu misalnya *toolbar.mahasiswa.html.php* di dalam direktori *administrator/components/com_mahasiswa*, lalu gunakan kode-kode dibawah ini :

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
class TOOLBAR_reviews
{
function_NEW()
{
JToolBarHelper::save();
JToolBarHelper::apply();
JToolBarHelper::cancel();
}

function_DEFAULT()
{
JToolBarHelper::title( JText::_('Komponen Mahasiswa'),
'generic.png' );
JToolBarHelper::publishList();
JToolBarHelper::unpublishList();
JToolBarHelper::editList();
JToolBarHelper::deleteList();
JToolBarHelper::addNew();
}
}
?>
```

File – file yang berisi kode-kode output biasanya diatur di dalam suatu kelas. Misalkan dalam contoh kode di atas terdapat kelas yang bernama *TOOLBAR_mahasiswa*. Tiap-tiap anggota fungsi yang ada di dalam kelas tersebut mewakili toolbars yang berbeda-beda. Kelas *JToolBarHelper* berisi fungsi-fungsi yang dapat menghasilkan semua tag *HTML* yang diperlukan untuk membangun toolbars. Untuk dapat menampilkan toolbar yang

sudah ditentukan sebelumnya, perlu menambahkan beberapa kode ke dalam file *toolbar.mahasiswa.php* di dalam direktori *administrator/components/com_mahasiswa*.

Kode tersebut adalah sebagai berikut :

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
require_once(JApplicationHelper::getPath('toolbar_html' ));

switch($task)
{
case 'edit':
case 'add':
    TOOLBAR_reviews::_NEW();
    break;
default:
    TOOLBAR_reviews::_DEFAULT();
    break;
}
?>
```

Variabel *request \$task* secara otomatis akan teregistrasi ke dalam *global scope*, dan akan digunakan untuk mengarahkan aliran logik di dalam komponen. Jadi, toolbar dasar yang ada pada core joomla dapat dimanfaatkan untuk komponen yang akan ditambahkan dan dapat ditampilkan pada beberapa layar yang berbeda.

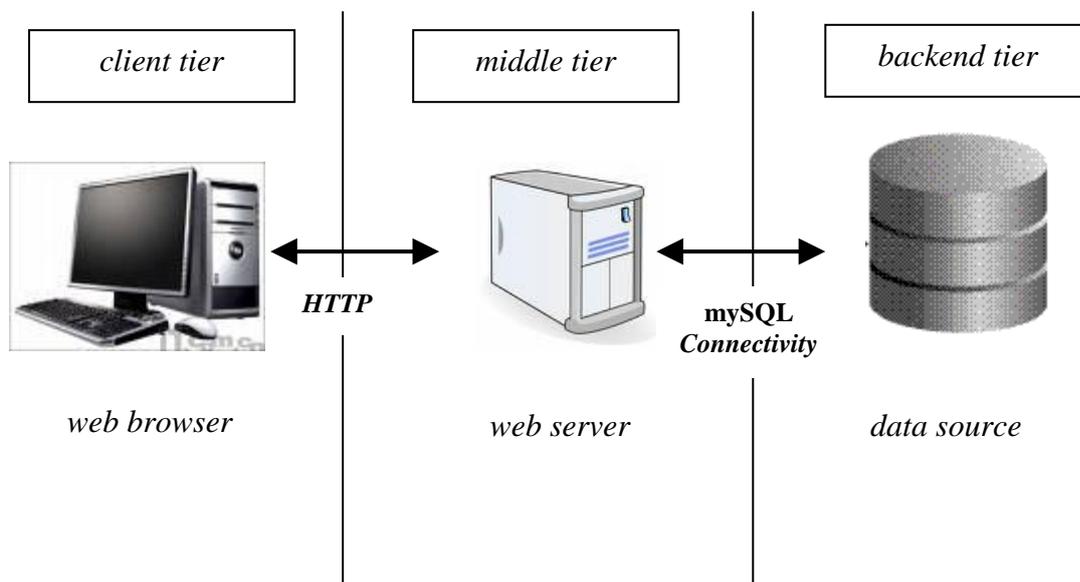
3.3.3.7 Arsitektur Sistem

Sistem kuis online adaptif ini dikembangkan berdasarkan arsitektur *three-tier*, sesuai dengan pemisahan fisik antara tiga bagian yaitu:

1. Browser atau aplikasi GUI - untuk *client tier*
2. *Web Server* atau *Application Server* - untuk *middle tier*
3. *Database Server* (RDBMS atau Relational Database) - untuk backend tier

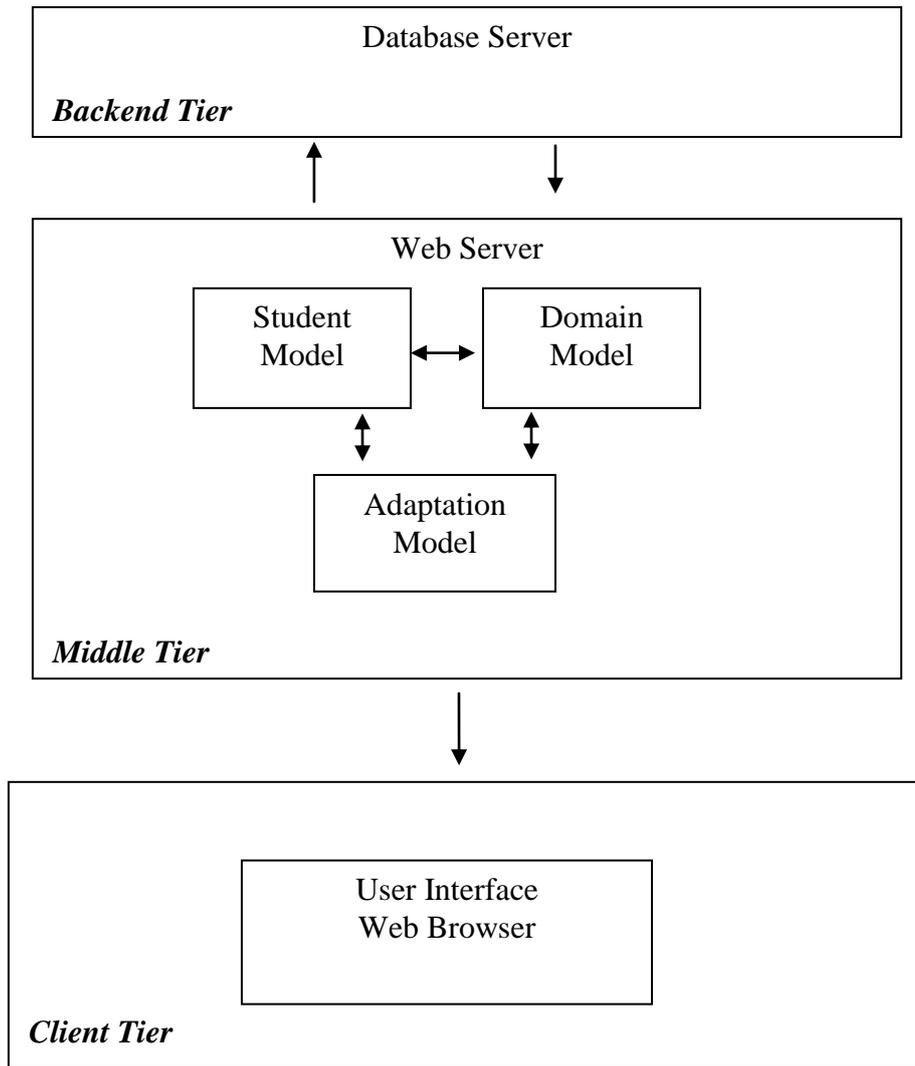
Gambar 3.36 menunjukkan tampilan fisik arsitektur *three-tier* sistem kuis online adaptif. *Client* yang berinteraksi dengan user melalui *web browser*, mengirim permintaan ke *web server (middle tier)* dengan menggunakan koneksi internet melalui protokol HTTP (*hypertext transfer protocol*). *Web server* (tempat aplikasi sistem kuis online adaptif berada) memproses permintaan berdasarkan alur yang ditetapkan. Output dari proses akan dikirim ke dalam

database dengan menggunakan koneksi database *mySQL* untuk memperbaharui data (*backend tier*).



Gambar 3.36 Pandangan Fisik Arsitektur Three-Tier Sistem Kuis online Adaptif

Pandangan secara logical dari arsitektur tingkat tinggi untuk sistem kuis online adaptif ditunjukkan pada Gambar 3.37. Lapisan *client tier* berisi *web browser* bagi pengguna untuk berinteraksi dengan *middle tier*. Lapisan *middle tier* berisi *web server* yang menyimpan semua logika bisnis ditentukan oleh pengajar untuk *student model*, *domain model* dan *adaptation model*, sedangkan *backend-tier*, berisi *database server* yang menyimpan semua data untuk sistem kuis online adaptif.



Gambar 3.37 Pandangan Logik Arsitektur Three-Tier Sistem Kuis Online Adaptif

3.4 Kesulitan-kesulitan

Kesulitan-kesulitan dalam penelitian ini yaitu pada saat tahapan *coding* harus menyesuaikan dengan aturan kerangka kerja (*framework*) dari *content management system* Joomla, hal ini merupakan suatu keharusan dalam membuat ekstensi untuk sebuah *content management system*. Penyesuaian ini baik dari sisi *back-end* maupun *front-end*, dikarenakan konsep *codingnya* berbeda.