



UNIVERSITAS INDONESIA

**IMPLEMENTASI SIMPLE-O
PADA VIRTUAL LEARNING ENVIRONMENT MOODLE**

SKRIPSI

**NARA BRAHMA PRAMODANA
0606074193**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**



UNIVERSITAS INDONESIA

**IMPLEMENTASI SIMPLE-O
PADA VIRTUAL LEARNING ENVIRONMENT MOODLE**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana

**NARA BRAHMA PRAMODANA
0606074193**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Nara Brahma Pramodana

NPM : 0606074193

Tanda Tangan :

Tanggal : 15 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Nara Brahma Pramodana
NPM : 0606074193
Program Studi : Teknik Elektro
Judul Skripsi : Implementasi SIMPLE-O pada Virtual Learning Environment Moodle

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir. Anak Agung Putri Ratna, M.Eng (.....)

Penguji : Muhammad Salman ST., MIT (.....)

Penguji : Ir. Endang Sriningsih MT, Si (.....)

Ditetapkan di : Depok

Tanggal : 9 Juli 2010

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah SWT karena atas segala rahmat dan karunia-Nya lah saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dari berbagai pihak. Oleh karena itu, saya mengucapkan terima kasih kepada :

1. Ibu Dr. Ir. Anak Agung Putri Ratna, M.Eng, selaku pembimbing yang membantu memberikan arahan dan nasihat sehingga saya dapat menyelesaikan skripsi ini;
2. Para peneliti sebelum ini yang juga memberikan sumber bacaan yang banyak bagi saya;
3. Teman – teman satu bimbingan dan satu angkatan dengan saya, Boma Anantasatya Adhi, Gregorius Handoyo, Meirisal Dwi Walidi, Vivi Zulvi Fauziah, yang telah banyak bertukar pikiran dan bekerjasama dalam menyelesaikan skripsi ini;
4. Orang tua dan keluarga saya yang selalu memberikan dukungan kepada saya; dan
5. Seluruh Sivitas akademika Departemen Teknik Elektro yang tidak dapat saya sebutkan satu persatu.

Penulis mohon maaf apabila terdapat kekurangan atau kesalahan dalam penulisan skripsi ini, oleh karena itu penulis mengharapkan kritik dan saran yang dapat melengkapi kekurangan yang ada pada skripsi ini.

Akhir kata, semoga Tuhan YME berkenan membalas kebaikan semua pihak yang telah membantu. Semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, Juni 2010

Nara Brahma Pramodana

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Indonesia, saya bertanda tangan di bawah ini :

Nama : Nara Brahma Pramodana
NPM : 0606074193
Program studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul :

**IMPLEMENTASI SIMPLE-O
PADA VIRTUAL LEARNING ENVIRONMENT MOODLE**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta sebagai pemegang Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 15 Juni 2010

Yang menyatakan

Nara Brahma Pramodana

ABSTRAK

Nama : Nara Brahma Pramodana
Praogram Studi : Teknik Elektro
Judul : Implementasi SIMPLE-O pada Virtual Learning Environment Moodle

Pada Skripsi ini akan dibahas mengenai penggunaan sebuah sistem penilaian esai otomatis dengan menggunakan metode *Latent Semantic Analysis* (LSA) pada sebuah media pembelajaran elektronik. Sistem penilaian esai yang digunakan adalah SIMPLE-O yang merupakan sistem penilaian esai otomatis yang menggunakan LSA. Pada skripsi ini suatu bagian perhitungan esai yang ada pada SIMPLE-O diimplementasikan ke dalam sebuah *Virtual Learning Environment* (VLE) yang bernama Moodle. Dengan dilakukannya pengintegrasian SIMPLE-O dengan Moodle, suatu jawaban esai pelajar dapat dihitung nilainya dengan melalui VLE ini. Pada sistem ini, jawaban pelajar dihitung secara bersama-sama sehingga waktu prosesnya memiliki hubungan linier dengan banyaknya jawaban pelajar yang dihitung. Kesimpulan dari skripsi ini yaitu aplikasi SIMPLE-O dapat berhasil diimplementasikan pada Moodle, jumlah *attempt* yang dihitung berbanding lurus dengan lama waktu pengekseskusion halaman perhitungan. Rata-rata waktu yang dibutuhkan untuk memproses 5 *attempt* sebesar 32,983 detik, untuk 10 *attempt* sebesar 78,678 detik, dan untuk 15 *attempt* sebesar 113,752 detik.

Kata kunci: SIMPLE-O, Moodle, *attempt*,

ABSTRACT

Name : Nara Brahma Pramodana
Study Program : Electrical Engineering
Title: : The Implementation of SIMPLE-O on Virtual Learning Environment Moodle

This paper will present about the use of automated essay grading system with Latent Semantic Analysis (LSA) in an electronic leaning media. Essay grading system used the SIMPLE-O which is the automated essay grading using LSA. In this paper, an essay calculating part of the SIMPLE-O is implemented into a Virtual Learning Environment (VLE) called Moodle. By doing SIMPLE-O and Moodle integration, student essay answer can be calculated through this VLE. In this system, the student answer calculated together so that the processing time has a linear relationship with the number of student answers are counted. The conclusion of this paper is the SIMPLE-O can be successfully implemented on Moodle, the amount of calculated attempt is proportional to the execution time of the calculation pages. Average time required to process five attempts is 32.983 seconds, for ten attempts is 78.678 seconds, and for the fifteen attempts is 113.752 seconds.

Keywords: SIMPLE-O, Moodle, attempt.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN OROSINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	v
ABSTRAK	vi
<i>ASBTRACT</i>	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Penulisan.....	2
1.3 Pembatasan Masalah.....	2
1.4 Metodologi Penulisan	3
1.5 Sistematika Penulisan	3
BAB 2 MOODLE, LSA, DAN PROGRAM PENDUKUNG	4
2.1 Moodle	4
2.2 <i>E-Learning</i>	4
2.3 <i>Latent Semantic Analysis</i>	5
2.4 Program Pendukung.....	6
2.4.1 PHP (<i>Hypertext Preprocessor</i>)	6
2.4.2 MySQL	6
2.4.3 Apache	8
2.4.4 JAMA (Java Matrix).....	8
BAB 3 IMPLEMENTASI SISTEM.....	9
3.1 Sistem Penilaian Esai Otomatis (SIMPLE-O)	9
3.2 Konsep Dasar Sistem Implementasi	10
3.3 Skenario Implementasi	11
BAB 4 IMPLEMENTASI DAN ANALISA	14
4.1 Sistem Implementasi.....	14
4.2 Hasil Implementasi	15
4.2.1 Modul Kuis	15
4.3 Analisa Implementasi	17
BAB 5 KESIMPULAN.....	25
DAFTAR REFERENSI	26

DAFTAR GAMBAR

Gambar 3.1 Algoritma perhitungan nilai mahasiswa pada SIMPLE-O.....	9
Gambar 3.2 Diagram Penggunaan SIMPLE-O pada Moodle	11
Gambar 3.3 Algoritma <i>regrade</i> pada Moodle	12
Gambar 4.1 Halaman awal Moodle dimana SIMPLE-O diimplementasikan	14
Gambar 4.2 Tampilan mahasiswa mengikuti ujian	15
Gambar 4.3 Tampilan dosen melihat hasil ujian	16
Gambar 4.4 Tampilan setelah pemilihan <i>regrade</i>	16
Gambar 4.5 Hasil jawaban pertama yang dimasukkan ke SIMPLE-O	19
Gambar 4.6 Hasil jawaban pertama yang dimasukkan ke Moodle	19
Gambar 4.7 Hasil jawaban kedua yang dimasukkan ke SIMPLE-O	20
Gambar 4.8 Hasil jawaban kedua yang dimasukkan ke Moodle	20
Gambar 4.9 Hasil percobaan dengan 5 <i>attempt</i>	22
Gambar 4.10 Hasil percobaan dengan 10 <i>attempt</i>	23
Gambar 4.11 Hasil percobaan dengan 15 <i>attempt</i>	24

DAFTAR TABEL

Tabel 4.1 Hasil percobaan 5 attempt.....	21
Tabel 4.2 Hasil percobaan 10 attempt.....	22
Tabel 4.3 Hasil percobaan 15 attempt.....	23



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi pada zaman sekarang ini pesat sekali. Mulai dari teknologi informasi sampai pada teknologi transportasi yang tidak kalah pesatnya. Perkembangan teknologi tersebut dipicu oleh berkembangnya pemikiran manusia sehingga dapat menghasilkan berbagai macam penemuan-penemuan baru yang dapat meningkatkan taraf hidup manusia. Sistem pendidikan yang lebih maju tentunya membuat seseorang menjadi lebih kreatif.

Perkembangan teknologi tidak hanya pada metode pembelajaran yang ada saja. Media yang digunakan untuk proses pembelajaran juga sangat berkembang. Mulai digunakannya berbagai perangkat digital dalam proses pembelajaran membuat proses belajar menjadi semakin menarik dan memudahkan pengajar dan siswa. Semakin berkembangnya dunia teknologi informasi, proses pembelajaran dapat menggunakan berbagai macam media. Berbagai buku yang tersedia secara *online* dan berbagai macam sumber lain seperti video, java applet, dll. Berkembangnya dunia teknologi informasi juga membuat timbulnya metode belajar secara *online* sehingga dapat diakses oleh siapa saja, dimana pun orang tersebut berada selama masih dapat tersambung dengan koneksi internet. Mulai dari bahan pembelajaran sampai dengan proses evaluasi hasil pembelajaran dapat dilakukan secara *online*.

Terlepas dari ada atau tidaknya sarana pendukung untuk melaksanakan kegiatan belajar mengajar dimana saja ini, apabila proses belajar mengajar dapat dilakukan dari mana saja, semakin banyak pula orang yang dapat belajar sehingga pemerataan tingkat pendidikan dapat tercapai. Adanya sistem pembelajaran secara *online* ini tidak luput dari perlunya sistem evaluasi yang *online* juga agar pembelajaran ini dapat berjalan dengan efisien. Metode evaluasi yang digunakan bisa berupa jenis soal objektif atau berupa soal esai.

Soal pilihan ganda memang terlihat lebih mudah dan kurang dapat mengevaluasi pelajar. Tetapi, dengan memberikan teknik menjawab yang lebih rumit, terdapat lebih dari satu jawaban benar misalnya, soal pilihan ganda cukup menggambarkan kemampuan dari pelajar. Soal esai tidak menyediakan jawaban



pada siswa, sehingga siswa harus menjawab dengan berdasarkan kemampuan yang dimilikinya dan tentunya lebih dapat mengevaluasi pelajar. Pelajar dapat menjawab dengan lebih leluasa dibandingkan dengan soal pilihan berganda.

Tetapi, timbul masalah dalam proses penilaian soal esai. Soal pilihan ganda dapat dengan mudah dinilai dengan mesin karena jawaban pelajar dapat dicocokkan dengan kunci jawaban secara langsung. Pada soal esai, cukup sulit untuk mencocokkan jawaban pelajar dengan jawaban pengajar apabila menggunakan mesin. Dengan berkembangnya ilmu pengetahuan timbul berbagai macam metode untuk membuat suatu sistem penilaian esai otomatis.

Beberapa contoh metode penilaian esai otomatis yang ada yaitu *Project Essay Grader (PEG)*, *Electronic Essay Rater (E-RATER)*, dan *Latent Semantic Analysis (LSA)* [12][4]. Selain itu, di Universitas Indonesia telah dikembangkan pula sistem penilaian esai otomatis dengan menggunakan metode *Latent Semantic Analysis* dimana penggunaannya pada Bahasa Indonesia. Sistem ini dinamakan SIMPLE-O dan salah satu perkembangannya saat ini telah digunakan kata bobot sebagai tambahan referensi penilaian esai tersebut [1][6][7].

Sistem penilaian esai otomatis dengan menggunakan teknik *Latent Semantic Analysis* yang disebut SIMPLE-O diharapkan dapat memudahkan pengguna Moodle sebagai sarana untuk menilai suatu ujian esai dengan menggunakan komputer.

1.2 Tujuan Penulisan

Tujuan dari penulisan skripsi ini adalah untuk melakukan implementasi sebuah sistem penilaian esai otomatis dengan metode *Latent Semantic Analysis* yang bernama SIMPLE-O ke dalam sebuah *Virtual Learning Environment (VLE)* yang bernama Moodle

1.3 Pembatasan Masalah

Pada skripsi ini, akan dilakukan implementasi SIMPLE-O pada Moodle, dimana yang diimplementasikan dari SIMPLE-O hanya bagian perhitungan jawaban mahasiswa saja dan dilakukan modul *quiz* yang terdapat pada Moodle.

1.4 Metodologi Penulisan

Metodologi penulisan yang digunakan pada skripsi ini adalah:

1. Studi literatur, yaitu dengan membaca dari beberapa materi yang ada di internet.
2. Teknik observasi, yaitu dengan meninjau kembali script pemrograman yang telah dibuat.

1.5 Sistematika Penulisan

Skripsi ini dibagi menjadi lima bab, dimana pada masing-masing bab akan menjelaskan sebagai berikut:

1. Bab 1: Pendahuluan
Pada bab ini, akan dijelaskan mengenai latar belakang, tujuan, pembatasan masalah, metodologi penulisan, dan sistematika penulisan.
2. Bab 2: Moodle, LSA, dan Program Pendukung
Pada bab ini, akan dijelaskan mengenai Moodle, *E-Learning*, *Latent Semantic Analysis*, dan Program pendukung.
3. Bab 3: Implementasi Sistem
Pada bab ini, akan dijelaskan mengenai SIMPLE-O, mengenai konsep dasar sistem implementasi, dan skenario implementasi.
4. Bab 4: Implementasi dan Analisa
Pada bab ini, akan dijelaskan mengenai sistem implementasi, hasil implementasi, dan analisa implementasi
5. Bab 5: Kesimpulan
Pada bab ini, akan dijelaskan mengenai kesimpulan yang dapat diambil dari pembahasan skripsi ini.

BAB 2 MOODLE, LSA, DAN PROGRAM PENDUKUNG

2.1 Moodle

Moodle merupakan sebuah akronim dari “Modular Object-Oriented Dynamic Learning Environment”. Moodle merupakan sebuah perangkat piranti lunak yang berbasis *open-source* dan tersedia gratis. Moodle juga biasa disebut sebagai *Course Management System (CMS)*, *Learning Management System*, atau *Virtual Learning Environment (VLE)*. Dengan menggunakan moodle, seorang pengajar dapat membuat suatu lingkungan belajar dimana pengajar tersebut dapat menyediakan materi, menyediakan tempat untuk mengumpulkan tugas, melaksanakan *quiz*, dan berbagai kegiatan lain yang dilakukan secara *online*. Moodle merupakan perangkat lunak *open source*, sehingga pengguna dapat mendapatkan, menggunakan, sekaligus mengubah Moodle ini sehingga sesuai dengan apa yang diinginkan, tentunya tetap dibawah lisensi GNU. [3]

Moodle merupakan salah satu dari VLE yang paling *user-friendly* jika dibandingkan dengan perangkat lunak lain yang juga berlisensi *open source* dan di desain secara khusus untuk pembuatan sistem belajar *online* yang berkualitas. Moodle memiliki dokumentasi yang sangat baik, dukungan yang kuat untuk keamanan dan administrasi. Moodle, sebagai suatu sistem *e-learning*, telah dikembangkan oleh berbagai spesialis yang berkontribusi untuk pengembangan Moodle itu sendiri. Moodle, sebagai VLE, juga memiliki suatu fitur yaitu situs moodle.org yang merupakan tempat berkumpulnya para pengguna Moodle termasuk para administrator sistem, perancang instruksional, dan pengembang. Situs moodle.org ini menjadi pusat pertukaran informasi dan kolaborasi di kalangan pengguna Moodle. Moodle pada saat ini tidak hanya digunakan di universitas saja tetapi juga sudah digunakan di sekolah-sekolah, organisasi *non-profit*, dan perusahaan pribadi [9].

2.2 E-Learning

E-Learning merupakan suatu metode dalam pembelajaran dimana proses dari pembelajaran tersebut dilakukan melalui media elektronik. Biasanya *e-learning* dilakukan melalui komputer yang terhubung dengan internet atau dapat



juga melalui intranet [8]. Media penyampaian *e-learning* sangat bermacam-macam. Pembelajaran dapat disampaikan melalui teks misalnya berupa e-book. Pembelajaran juga dapat disampaikan berupa video atau audio. Dengan adanya konsep pembelajaran ini, pengajar dan pelajar tidak perlu bertatap muka secara langsung.

Pada konsep *e-learning* ini, proses evaluasi dari pembelajaran dapat dilakukan secara *online*, mulai dari pemberian soal ujian, menjawab soal ujian, sampai proses penilaian-pun dapat dilakukan secara *online*. Hal ini dapat menguntungkan bagi pengajar dan pelajar karena metode ini dapat menjadi efektif dan efisien. Dengan berkembangnya sistem penilaian dengan menggunakan komputer maka ujian dengan metode pembelajaran ini dapat menjadi lebih cepat dan akurat.

2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) merupakan sebuah teknik matematis yang digunakan untuk mengekstrak dan menyimpulkan kumpulan dari kata-kata dalam jumlah yang besar. Perhitungan matematisnya dilakukan dengan memetakan ada atau tidaknya kata dari kelompok kata yang ada pada matriks. Metode ini menggunakan teknik analisa vektor pada suatu matriks, yaitu teknik aljabar linier *Singular Value Decomposition* (SVD). Pada SVD, matriks akan didekomposisi menjadi tiga komponen matriks. Komponen matriks pertama mendeskripsikan entitas baris sebagai vektor ortogonal matriks. Komponen matriks kedua mendeskripsikan matriks diagonal yang berisi skalar dan yang ketiga adalah matriks entitas kolom sebagai vektor ortogonal matriks [2].

Setelah didapatkan tiga buah matriks dari proses SVD, proses berikutnya yang perlu dilakukan adalah mereduksi dimensi dari matriks dengan cara mengurangi dimensi dari matriks kedua yang berupa matriks diagonal yang bernilai skalar. Pengurangan dimensi matriks diagonal ini dilakukan dengan cara mengeset semua nilai diagonal matriks ke dua menjadi nol kecuali diagonal dimensi yang dipilih. Dan jika ketiga komponen matriks tersebut dikalikan maka menghasilkan matriks rekonstruksi yang lain dengan tujuan untuk nilai korelasi yang diinginkan [2].

Secara matematis sebuah matriks dapat didekomposisi dengan baik jika memiliki nilai faktor yang kecil dibandingkan dengan dimensi terkecil dari matriks awalnya. Sehingga rekonstruksi matriks terbaik dihasilkan pada saat nilai faktor lebih kecil dari jumlah faktor yang digunakan [2].

2.4 Program Pendukung

Dalam pembuatan program essay grading berbasis web ini, menggunakan sistem yang dibangun dengan bahasa scripting PHP yang menggandeng database MySQL dan scripting HTML. Pada proses perhitungan matriks di dalam program essay grading ini, menggunakan program JAMA (Java Matrix) yang di program ulang menjadi program PHP. Web server yang digunakan adalah Apache.

2.4.1 PHP (Hypertext Preprocessor)

PHP banyak digunakan oleh programmer dengan alasan memiliki kemiripan syntax dengan bahasa C/C++, Java, dan Perl. PHP merupakan bahasa scripting *open source* dengan menggunakan lisensi GPL (GNU Public Licence) yang dapat digunakan dengan gratis dan bebas. PHP biasa digunakan untuk membuat aplikasi web yang bersifat dinamis.

PHP menyerupai dengan HTML dimana kode-kode yang dibuat tidak perlu di-*compile* sebelum digunakan. Kode yang dibuat akan diproses saat diperlukan [5].

2.4.2 MySQL

MySQL merupakan software yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source*. MySQL awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. Selanjutnya pengembangan MySQL berada dibawah naungan perusahaan MySQL AB.

Pada tanggal 16 Januari 2008 Sun Microsystems, Inc mengumumkan aksi korporasi - akuisisi terhadap MySQL AB sehingga menjadikan Sun sebagai salah satu perusahaan dengan produk platform *open source* terbesar seperti Java, OpenSolaris dan akhirnya MySQL [4].

Berselang setahun kemudian, tepatnya pada tanggal 20 April 2009 giliran Oracle melakukan akuisisi terhadap Sun Microsystems.

Sebagai software DBMS, MySQL memiliki beberapa fitur, diantaranya :

1. Multiplatform

MySQL dapat digunakan dalam banyak platform, seperti Windows, Linux, Unix dan platform lainnya.

2. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

3. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam sebuah perintah (*query*). Sebagai contoh dapat dilihat perintah MySQL berikut ini:

```
mysql> SELECT CONCAT(first_name, ' ', last_name)
      -> FROM citizen
      -> WHERE income/dependents > 10000 AND age >
      30;
```

4. Antar Muka

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

5. Andal, cepat dan mudah digunakan

MySQL tergolong sebagai *database server* yang andal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.

6. Jaminan keamanan akses

MySQL mendukung pengamanan database dengan berbagai kriteria pengaksesan. Contohnya, pada MySQL dapat dimungkinkan untuk mengatur user tertentu agar bisa mengakses data yang bersifat pribadi dimana user lainnya tidak dapat mengakses data tersebut.

7. Dukungan SQL

MySQL mendukung perintah SQL (*Structured Query Language*) yang merupakan bahasa yang digunakan untuk mengakses database relasional [4].

2.4.3 Apache

Server HTTP Apache atau Server Web/WWW Apache adalah web server yang dapat dijalankan di banyak sistem operasi seperti Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya, yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web ini menggunakan HTTP. Web Server Apache berbasiskan *open source* dan mulai populer di internet semenjak tahun 1996.

Supaya dokumen-dokumen web berupa HTML ataupun PHP bisa diakses oleh browser maka dokumen-dokumen tersebut perlu diletakkan dalam direktori khusus yang diatur oleh Apache.

2.4.4 JAMA (Java Matrix)

Pada awal pengembangan SIMPLE-O, untuk melakukan perhitungan SVD menggunakan program matlab, sehingga hasil perhitungan dapat dihitung dalam program matlab tersebut. Namun untuk mendapatkan hasil dari perhitungan tersebut membutuhkan waktu yang cukup lama untuk satu orang user saja, sehingga dibutuhkan suatu program yang memungkinkan mempercepat proses perhitungan dengan meninjau banyaknya user saat melakukan ujian bersama-sama.

Dengan alasan tersebut, maka dipilih Java Matrix yang dapat mengerjakan proses perhitungan SVD lebih cepat dibandingkan dengan Matlab. JAMA yang digunakan merupakan skrip PHP untuk perhitungan matriks kompleks. Class dari package JAMA akan sering digunakan dalam operasi matriks seperti perkalian matriks, transpose, dan inverse.

BAB 3 IMPLEMENTASI SISTEM

3.1 Sistem Penilaian Esai Otomatis (SIMPLE-O)

Aplikasi penilaian esai otomatis yang dikembangkan [1,6,7] merupakan sebuah aplikasi yang berbasis web. Aplikasi ini dibuat dengan menggunakan pemrograman PHP dan untuk proses penghitungannya menggunakan JAMA (Java Matrix). Selain itu, pada pemrograman ini juga digunakan sistem database MySQL untuk penggunaan databasenya dan menggunakan HTML untuk membuat *user interface*-nya.

Dalam penggunaannya pada skripsi ini, salah satu bagian dari SIMPLE-O ini dimasukkan ke dalam suatu Aplikasi *Virtual Learning Environment* (VLE) yang bernama Moodle. Bagian yang ditambahkan ke Moodle ini berupa suatu algoritma perhitungan jawaban yang telah dimasukkan oleh pelajar. Jawaban pelajar tersebut akan dibandingkan dengan jawaban pengajar yang telah diubah menjadi matriks dan akan menjadi matriks referensi. Untuk lebih jelasnya, algoritma dari modul perhitungan dapat dilihat pada Gambar 3.1.

```
Proc ujian ( )
[load matkul untuk user dari database]
while matkul != EOF
    [tampilkan mata kuliah yang dipilih]
ewhile
read (pil); if pil = Back then goto Halaman_Muka;
else
if pil = Lihat Soal then
    write ("Soal", soal[ ]); write ("Jawaban");
    read jawab_mhs[ ];
    if submit = true then
        [cek persamaan kata-kata jawaban mahasiswa]
        [bentuk matriks dari jawaban mahasiswa];
        [bandingkan matriks jawaban dengan referensi];
        [kirim nilai ke tabel database];
    else
        ( );
else
    ( );
Eproc
```

Gambar 3.1 Algoritma perhitungan nilai mahasiswa pada SIMPLE-O

3.2 Konsep Dasar Implementasi

Implementasi sistem penilaian esai otomatis ini bertujuan sebagai aplikasi yang dapat membantu dalam proses evaluasi dari sebuah atau beberapa jawaban esai yang telah dikerjakan secara online. Aplikasi SIMPLE-O akan diimplementasikan pada suatu *Virtual Learning Environment* berbasis web yang dinamakan Moodle. Pemilihan Moodle sebagai sarana ini didasarkan pada alasan berikut ini:

1. Moodle merupakan perangkat lunak yang berbasis *open source* sehingga pengguna dapat dengan gratis mengunduh perangkat lunak tersebut, menggunakan, mengubah, dan bahkan mendistribusikan perangkat lunak tersebut dibawah lisensi GNU. [3]
2. Moodle dapat digunakan oleh semua server yang mensupport PHP.
3. Moodle telah digunakan di seluruh dunia oleh berbagai macam sekolah, universitas, dan perusahaan. Moodle memiliki kredibilitas yang sangat tinggi.
4. Moodle merupakan LMS dan VLE yang dapat menunjang pengajar untuk membuat tempat untuk mengumpulkan tugas, menilai tugas yang telah dikerjakan oleh pelajar, membagikan link atau file, dan lain-lain.

Karena aplikasi ini berbasis web, maka akses untuk penggunaan sistem ini dapat dilakukan dari mana saja selama akses internet tersedia. Beberapa keuntungan dari sistem ini yaitu:

1. Pelajar dan pengajar dapat melakukan proses belajar mengajar secara online tanpa ada batasan jarak dan waktu, selama terdapat komputer dan akses internet.
2. Pemeriksaan evaluasi pelajar dilakukan oleh mesin, sehingga meringankan tugas pengajar untuk memeriksa hasil belajar dari para pelajarnya.

Secara umum, sistem ini pertama-tama akan membagi user ke dalam dua jenis, user yang telah terdaftar dan user yang belum terdaftar. Apabila user belum terdaftar, maka user tidak dapat menggunakan sistem ini. User yang sudah terdaftar dibagi lagi ke dalam 3 jenis *role*, yaitu administrator, pengajar, dan pelajar. Masing-masing user memiliki wewenang masing-masing yaitu:

1. Administrator, memiliki wewenang tertinggi dalam sistem ini. Dapat melakukan semua pekerjaan yang dapat dilakukan di sistem ini, termasuk

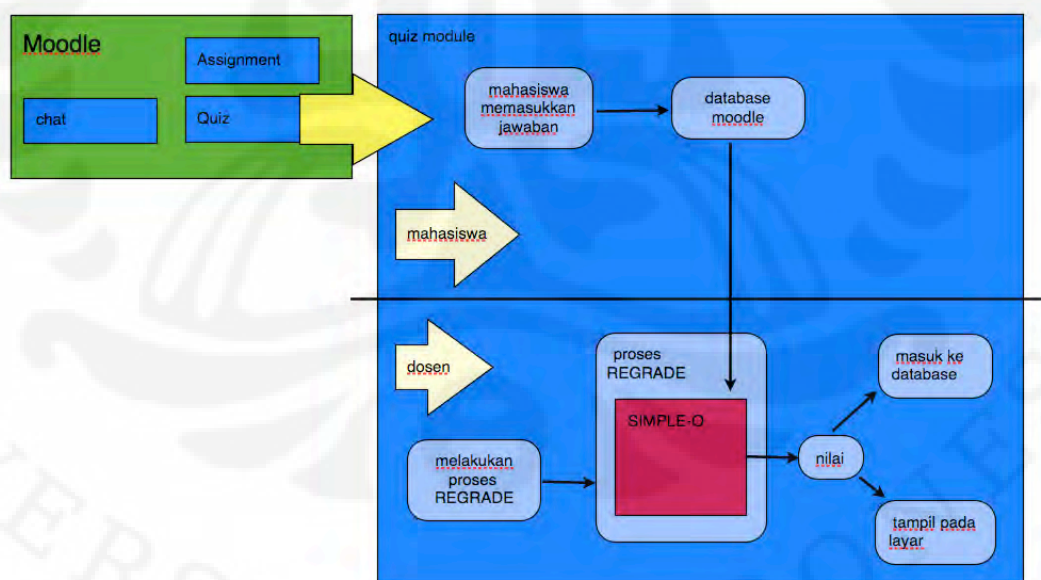
mengatur permasalahan teknis dari Moodle, menambahkan user, dan mengatur *role* dari user.

2. Pengajar, memiliki wewenang untuk menambahkan topik diskusi, memberikan jawaban referensi, menentukan bobot pada jawaban, menentukan kelompok, ikut berdiskusi pada semua kelompok, dan melihat nilai mata kuliah yang diajarkan.
3. Pelajar, memiliki wewenang untuk ikut berdiskusi pada kelompoknya saja, mensubmit tugas yang diberikan, mengunduh materi yang disediakan oleh pengajar, mengikuti kuis.

3.3 Skenario Implementasi

Pada bagian ini akan dilakukan dijelaskan bagaimana proses implemetasi SIMPLE-O pada Moodle dan proses bagaimana pengevaluasian hasil belajar mahasiswa yang dilakukan dengan Moodle yang telah diintegrasikan dengan SIMPLE-O.

Secara teknis penggunaan dari SIMPLE-O dapat dilihat pada Gambar 3.2. Moodle memiliki berbagai macam modul *activity* yang sudah *built-in* diantaranya



Gambar 3.2 Diagram Penggunaan SIMPLE-O pada Moodle

yaitu *assignment*, *chat*, *quiz*, dan *lesson*. Penggunaan aplikasi SIMPLE-O pada skripsi ini, digunakan untuk menilai hasil pekerjaan mahasiswa yang menggunakan modul kuis pada Moodle. Proses penggabungan ini belum

sepenuhnya terintegrasi dengan seluruh sistem Moodle. Proses penghitungan tidak langsung dilakukan sesudah mahasiswa menyelesaikan sebuah kuis, tetapi harus dieksekusi secara manual oleh *teacher* yang bersangkutan atau oleh seorang *administrator* melalui fitur *regrade* yang ada pada halaman dosen.

Pada saat user memilih *regrade* pertama-tama halaman tersebut akan memasukkan jawaban referensi yang dibuat dosen. Jawaban referensi dari dosen terdiri dari kata bobot dan kata kunci serta matriks referensi. Basis data yang berisi jawaban referensi ini harus dimasukkan secara manual ke salah satu tabel baru yang ada di basis data Moodle. Tabel jawaban referensi diambil dari tabel yang ada pada basis data SIMPLE-O. Selain itu, perlu juga dilakukan perubahan pada kode soal agar nomor soal dari SIMPLE-O sesuai dengan basis data Moodle. Jawaban referensi tersebut dimasukkan ke dalam suatu array. Selanjutnya diambil basis data persamaan kata-kata, diambil dari tabel kata-kata SIMPLE-O, lalu dimasukkan ke dalam suatu array.

```

proc regrade ()
[ambil data referensi jawaban berdasarkan kode quiz]
[ambil data referensi kata-kata]
[ambil nama-nama user yang mengikuti ujian]
[ambil data soal dan attempt yang di-regrade]
for [semua soal]
    for [semua attempt]
        [dilakukan perhitungan jawaban mahasiswa];
        [nilai dimasukkan ke array]
    efor
efor
for [semua attempt]
    [gabung semua nilai];
efor
for [semua attempt]
    [masukkan nilai ke database];
    [tampilkan nilai yang telah dihitung ke layar];
efor
eproc

```

Gambar 3.3 Algoritma *regrade* pada Moodle

Proses selanjutnya, dilakukan pengambilan nama-nama user yang mengikuti ujian tersebut lalu jawaban dari mereka dimasukkan ke dalam array. Setelah itu, Moodle melakukan proses pengambilan soal yang akan dihitung dan

selanjutnya akan dilakukan *looping* untuk melakukan pengecekan jawaban mahasiswa. Proses *looping* dilakukan secara dua tahap. Berdasarkan urutan soal terlebih dahulu, lalu berdasarkan *attempt* atau user yang diperiksa. Tiap user yang mengikuti ujian memiliki nomor urutan *attempt* berdasarkan urutan waktu. Dengan kata lain, prosesnya akan mengecek salah satu soal untuk semua user dan selanjutnya akan mengecek soal berikutnya untuk semua user pula, begitu seterusnya sampai semua soal selesai diperiksa. Selanjutnya, dilakukan perhitungan nilai dan pemasukan ke dalam sistem basis data. Sistem ini belum dapat diintegrasikan ke dalam sistem *grading* pada Moodle, sehingga nilai yang sudah didapatkan disimpan pada tabel yang ditambahkan pada basis data Moodle dan ditampilkan ke layar. Untuk lebih jelasnya dapat dilihat pada Gambar 3.3.

BAB 4 IMPLEMENTASI DAN ANALISA

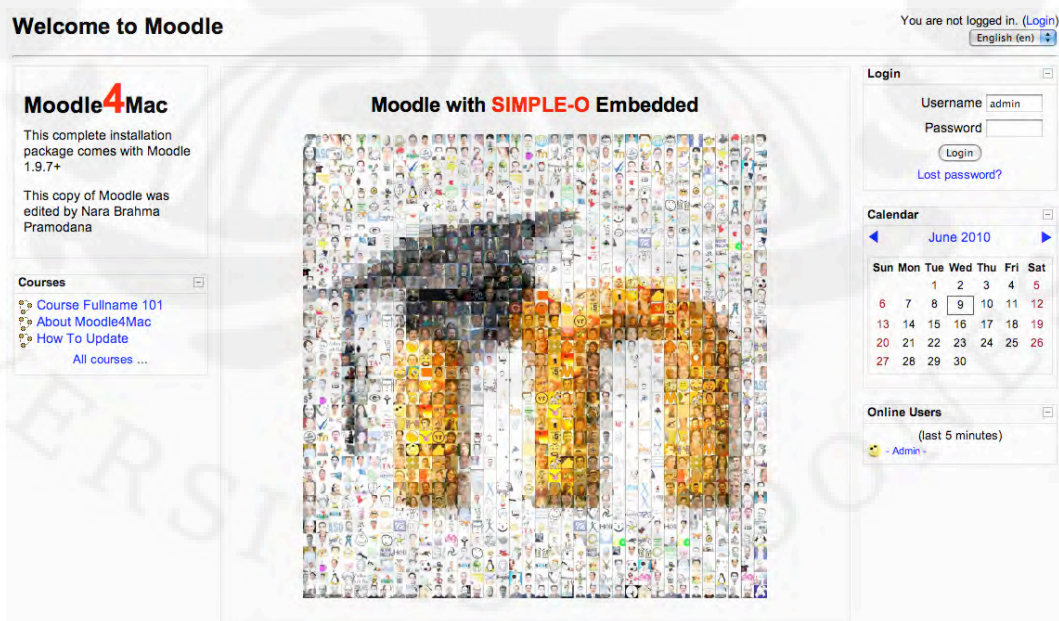
4.1 Sistem Implementasi

Aplikasi SIMPLE-O dan Moodle merupakan suatu aplikasi yang berbasis web. Kedua aplikasi ini menggunakan bahasa pemrograman PHP dan sistem penyimpanan datanya menggunakan basis data MySQL. Sistem ini diimplementasikan pada sebuah laptop dengan spesifikasi hardware sebagai berikut:

- Merk dan Tipe: Apple MacBook Air
- Prosesor: Intel Core 2 Duo (P7500) 1.6 GHz
- RAM: 2 GB, 667 MHz, DDR2 SDRAM

Sedangkan perangkat lunak yang digunakan dalam proses pengimplementasian SIMPLE-O pada Moodle sebagai berikut:

- Sistem Operasi: Mac OS X Leopard 10.5.8
- Moodle 1.9.7+
- MAMP 1.7.2 termasuk PHP 5.2.6 dan MySQL 5.0.41
- NetBeans IDE 6.8
- Web Browser: Google Chrome ver. 5.0.375.70 dan Safari ver.5.0, 5.10.1



Gambar 4. 1 Halaman awal Moodle dimana SIMPLE-O diimplementasikan

4.2 Hasil Implementasi

Dalam pengimplementasian SIMPLE-O pada Moodle, bagian perhitungan jawaban mahasiswa pada SIMPLE-O diintegrasikan ke dalam modul *quiz* yang ada pada Moodle. Berikut ini, akan ditampilkan beberapa fitur dari modul *quiz* yang telah dikembangkan. Setiap bagian akan dijelaskan bagaimana suatu bagian tersebut dibuka oleh user yang memiliki role sebagai dosen dan sebagai mahasiswa. Untuk user administrator tidak dibahas karena user dengan role tersebut memiliki akses untuk melakukan semua fungsi yang ada.

4.2.1 Modul Kuis

Modul ini merupakan sebuah modul yang memang dipersiapkan oleh Moodle untuk membuat suatu ujian. Pengaturan yang dapat dilakukan diantaranya yaitu membuat berbagai macam soal termasuk soal esai dan soal pilihan berganda, mengatur waktu untuk membuka atau menutup kuis tersebut, mengatur berapa

The screenshot shows a Moodle quiz page titled "Tes Kuis Pertama" for "Course Fullname 101". The user is logged in as "mhs4 mhs4". The quiz is in "Attempt 1".

Question 1: "Sebutkan dan jelaskan mode transmisi yang disebut simplex." (Marks: 1). The answer provided is: "sistem transmisi yang hanya memungkinkan satu data untuk dikirim (dialirkan) dalam satu waktu jenis mode transmisi ini tidak mungkin terjadinya pengiriman data secara dua arah dalam waktu yang bersamaan".

Question 2: "Jelaskan apa yang dimaksud dengan internet protocols?" (Marks: 1). The answer provided is: "satu set peraturan tentang koneksi internet dengan komputer yang mengatur bagaimana data dikirim kemana data dikirim dan sebagainya".

Gambar 4. 2 Tampilan mahasiswa mengikuti ujian

banyak seorang user mahasiswa dapat mengikuti kuis tersebut, dan mengetahui siapa saja yang telah mengikuti kuis tersebut termasuk sudah berapa kali user tersebut mengikuti kuis, tentunya apabila memang diatur untuk dapat mengikuti

kuis tersebut lebih dari satu kali. Apabila menggunakan user mahasiswa, maka tampilan mengikuti kuis dapat dilihat pada Gambar 4.2 dengan menggunakan user

The screenshot shows the Moodle interface for a course named 'Course Fullname 101'. The quiz is titled 'Tes Kuis Pertama'. The page displays a table of quiz results for two users: mhs2 and mhs3. The table includes columns for 'First name / Surname', 'Started on', 'Completed', 'Time taken', 'Grade/100', and eight question items (#1 to #8). Both users have a grade of 0. Below the table, there are options to 'Select all / Deselect all', 'Delete selected attempts', and download the report in ODS, Excel, or text format. There are also preference settings for the page and report.

First name / Surname	Started on	Completed	Time taken	Grade/100	#1	#2	#3	#4	#5	#6	#7	#8
mhs2	8 June 2010, 05:27 pm	8 June 2010, 05:28 pm	1 min 13 secs	0	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5
mhs3	8 June 2010, 07:01 pm	8 June 2010, 07:02 pm	1 min 31 secs	0	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5
Overall average				0	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5	/12.5

Gambar 4. 3 Tampilan dosen melihat hasil ujian

dosen, user tersebut dapat membuat kuis baru, mengatur pilihan untuk kuis tersebut, dan melakukan *regrade* untuk melakukan penilaian dengan SIMPLE-O. Gambar 4.3. Untuk hasil apabila *regrade* dipilih, dapat dilihat pada Gambar 4.4.

The screenshot shows the Moodle interface for the same course, but now displaying the 'Regrading Quiz "Tes Kuis Pertama"' page. It lists attempts that change during regrading as hyperlinks to the question review window. Below this, there is a 'REVIEW ALL' section with a summary of user scores for different attempts. The page also includes a footer with a Moodle Docs link.

Regrading Quiz "Tes Kuis Pertama"

Attempts that change during regrading are displayed as hyperlinks to the question review window

- Regrading "Soal 1.1", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.2", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.3", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.4", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.5", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.6", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.7", Attempts: #20 #19 #22 #23 #24
- Regrading "Soal 1.8", Attempts: #20 #19 #22 #23 #24

REVIEW ALL
Pada Quiz 4

- USERID 7 dengan NAMA user mhs3 pada ATTEMPT 20 mendapatkan nilai sebesar 61.97375
- USERID 5 dengan NAMA user mhs2 pada ATTEMPT 19 mendapatkan nilai sebesar 63.435
- USERID 8 dengan NAMA user mhs4 pada ATTEMPT 22 mendapatkan nilai sebesar 74.28875
- USERID 9 dengan NAMA user mhs5 pada ATTEMPT 23 mendapatkan nilai sebesar 79.255
- USERID 10 dengan NAMA user mhs6 pada ATTEMPT 24 mendapatkan nilai sebesar 76.5125

This page took 30.272148132324 seconds to be processed
While counting for 5 attempts

Moodle Docs for this page

Gambar 4. 4 Tampilan setelah pemilihan *regrade*

4.3 Analisa Implementasi

Pada skripsi ini aplikasi SIMPLE-O dapat diimplementasikan ke dalam Moodle walaupun masih belum sepenuhnya terintegrasi dengan sistem Moodle, tetapi proses untuk melakukan penilaian sebuah jawaban esai sudah dapat bekerja. Untuk pemasukan soal referensi masih harus menggunakan SIMPLE-O dan selanjutnya dilakukan modifikasi database sehingga sesuai dengan sistem yang digunakan pada Moodle. Untuk membuktikan bahwa program berhasil diimplementasikan, dilakukan sebuah percobaan yang membandingkan hasil perhitungan sebuah jawaban mahasiswa yang dihitung di SIMPLE-O dan di Moodle yang telah diintegrasikan dengan SIMPLE-O. Percobaan dilakukan sebanyak dua kali, soal yang dipakai yaitu:

1. Sebutkan dan jelaskan mode transmisi yang disebut simplex!
2. Jelaskan apa yang dimaksud dengan internet protocols!
3. Jelaskan apa yang dimaksud dengan HUB!
4. Jelaskan apa yang anda ketahui mengenai UTP!
5. Jelaskan apa yang dimaksud dengan LAN!
6. Jelaskan perbedaan antara topologi fisik dengan topologi logic!
7. Jelaskan apa yang dimaksud dengan jaringan peer to peer!
8. Jelaskan cara kerja CSMA/CD secara singkat!

Sedangkan jawaban dari soal tersebut yang digunakan sebagai jawaban referensi adalah sebagai berikut:

1. Transmisi simplex atau disebut dengan transmisi satu arah adalah transmisi tunggal, transmisi yang dilakukan pada hanya satu arah saja
2. Internet protocol adalah salah satu set aturan atau rule untuk komunikasi dalam dan antara komputer di jaringan. Banyak protokol terdiri atas suite (atau group) protokol stack pada layer. Layer ini tergantung pada operasi dari layer yang lainnya dalam suit, supaya berfungsi dengan benar.
3. Hub dapat mengembangkan cakupan sinyal dengan menerima kemudian meregenerasinya dan mengirim keluar ke port yang lain. Trafik akan dikirim keluar oleh seluruh port dari hub. Memungkinkan banyak collisions pada segmen jaringan dan sering bukan merupakan solusi yang

baik. Juga disebut concentrators karena mereka melayani sebagai pusat titik koneksi untuk LAN.

4. Pada kabel UTP terdapat dua atau empat pasang kabel. Kabel jenis ini mempunyai efek pembatalan (cancellation effect) untuk pengurangan interferensi yang disebabkan oleh electromagnetic interface (EMI) dan radio frequency interference (RFI). Sangat umum untuk pengkabelan jaringan. Mempunyai jangkauan dari 328 ft (100 meter).
5. Suatu group / kumpulan komputer yang terinerkoneksi dibawah kontrol administrasi yang sama. Dapat berupa jaringan lokal kecil yang di-instal di perkantoran. Dapat terdiri dari jaringan lokal yang terhubung yang terdiri dari beberapa ratus host, di-instal pada beberapa gedung dan lokasi. LAN ini mempunyai daerah cakupan yang sempit, paling jauh sekitar 1 km. Kampus termasuk LAN.
6. Topologi Fisik adalah layout fisik dari komponen yang ada di jaringan. Sedangkan Topologi logic menentukan bagaimana host mengakses ke medium untuk berkomunikasi lintas jaringan.
7. Jaringan peer to peer adalah jaringan komputer yang dapat berbagi files, mengirim pesan, mencetak pada shared printer. Pada jaringan tersebut, masing-masing komputer mempunyai kemampuan dan tanggung jawab yang sama. Setiap pengguna memutuskan data mana dan devais mana yang akan di-share. Pada jaringan ini tidak ada titik pusat kontrol pada jaringan. Jumlah anggota jaringan, paling baik bila terdiri dari 10 atau kurang dari 10 komputer.
8. Seluruh host atau komputer yang terhubung, mendengarkan jaringan saat ingin mengirimkan data. Bila komputer tidak mendeteksi tidak ada host yang mengirimkan data, maka host tersebut mencoba untuk mengirimkan data. Bila tidak ada host lain yang mengirimkan data, maka transmisi tersebut berhasil dan data akan sampai pada tujuan dengan sukses. Jika ada host atau komputer lain mengirimkan atau mentransmisikan data pada saat yang bersamaan, tabrakan atau collision akan terjadi pada media jaringan. Host atau komputer pertama yang mendeteksi tabrakan atau collision tersebut, maka akan mengirim sinyal yang menyatakan bahwa

terjadi tabrakan atau collision dan memberitahukan ke seluruh host untuk menghentikan pengiriman dan untuk menjalankan algoritma backoff. Seluruh komputer yang tergabung pada jaringan tersebut akan menghentikan program dan mencoba kembali proses pengirimannya setelah periode waktu tertentu (random).

Gambar 4. 5 Hasil jawaban pertama yang dimasukkan ke SIMPLE-O

Gambar 4. 6 Hasil jawaban pertama yang dimasukkan ke Moodle

Berdasarkan soal dan jawaban tersebut diatas, dicoba untuk menghitung dua buah jawaban mahasiswa. Berikut ini merupakan gambar perbandingan antara perhitungan jawaban mahasiswa pada aplikasi SIMPLE-O dan pada Moodle. Pada Gambar 4.5 pada bagian bawah, dapat dilihat hasil dari perhitungan adalah “score 63,435”.

The screenshot shows the 'Sistem Penilaian Essay Otomatis' interface. The header is blue with the title 'Sistem Penilaian Essay Otomatis' and a welcome message. On the left, there is a navigation menu with options like 'PENGUNJUNG', 'Mahasiswa', 'Libre Nilai', 'Pilih Ujian', and 'Logout'. The main content area displays a user's score of 74.34875, which is circled in red. Below the score, there is a message: 'This page took 2.8100030422211 seconds to be processed'. The background features a large, faint watermark of the Universitas Indonesia logo.

Gambar 4. 7 Hasil jawaban kedua yang dimasukkan ke SIMPLE-O

Setelah itu, jawaban yang sama dimasukkan ke dalam Moodle. Hasilnya sama dengan perhitungan yang dilakukan dengan SIMPLE-O, untuk lebih jelasnya dapat dilihat pada Gambar 4.6 pada bagian bawah tertulis “USERID 5

The screenshot shows the Moodle interface. The main content area displays a user's score of 63.435, which is circled in red. Below the score, there is a message: 'Moodle Docs for this page'. At the bottom, there is a login status: 'You are logged in as - Admin - (Logout)'. The background features a large, faint watermark of the Universitas Indonesia logo.

Gambar 4. 8 Hasil jawaban kedua yang dimasukkan ke Moodle

dengan NAMA user mhs2 pada ATTEMPT 19 mendapatkan nilai sebesar 63,435”. Selanjutnya, dilakukan lagi pemasukan jawaban yang lain pada SIMPLE-O dan hasilnya dapat dilihat pada Gambar 4.7 pada bagian bawah terdapat tulisan “Score 74,34875”. Jawaban tersebut dimasukkan juga ke dalam Moodle. Setelah dilakukan proses *regrade* akan terlihat “USERID 7 dengan NAMA user mhs3 pada ATTEMPT 20 mendapatkan nilai sebesar 74,34875”.

Selain ditampilkan hasil jawaban yang telah dimasukkan halaman Moodle juga akan memunculkan seluruh mahasiswa yang pernah mengikuti kuis tersebut juga dilakukan penghitungan ulang pada kuis tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 4.8.

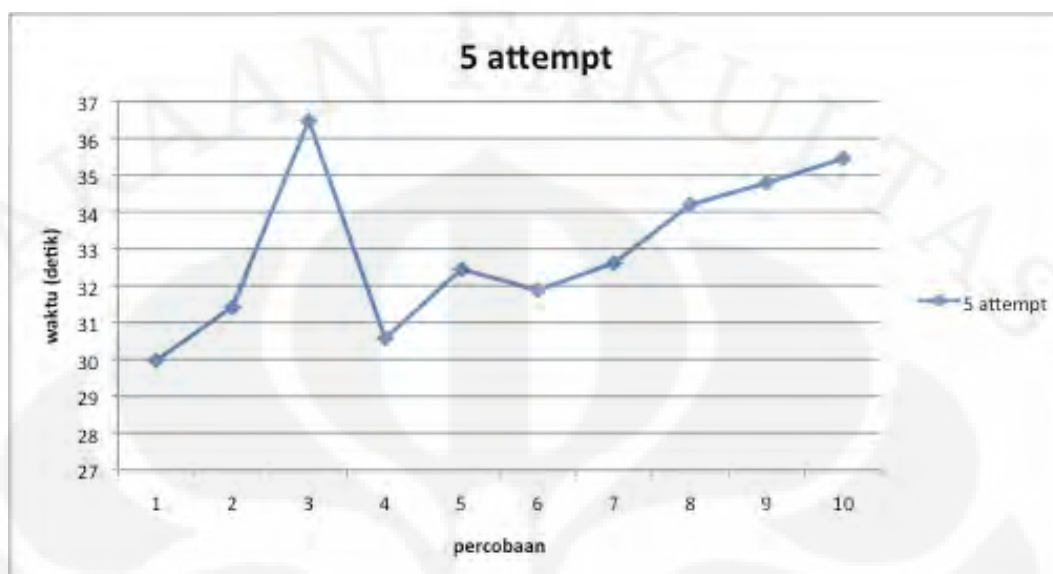
Pada skripsi ini, juga dilakukan pengujian performansi dari SIMPLE-O yang telah diintegrasikan dengan Moodle ini. Proses pengujian dilakukan dengan menghitung waktu yang dibutuhkan untuk memproses suatu halaman dari Moodle dimana pada halaman tersebut SIMPLE-O diintegrasikan. Pada pengujian ini, dilakukan tiga macam skenario yaitu menghitung jawaban sebanyak 5 *attempt*, 10 *attempt*, dan 15 *attempt*. Setiap skenario tersebut dilakukan pengulangan percobaan sebanyak sepuluh kali.

Tabel 4. 1 Hasil Percobaan dengan 5 *Attempt*

Percobaan	Waktu (s)
1	29.969
2	31.420
3	36.480
4	30.570
5	32.447
6	31.881
7	32.614
8	34.202
9	34.791
10	35.455
Rata-Rata	32.983
@attempt	6.597

Setelah dilakukan skenario untuk menghitung sebanyak 5 *attempt*, didapatkan bahwa rata-rata waktu yang dibutuhkan untuk melakukan proses perhitungan 5 *attempt* adalah sebesar 32,982 detik. Dari hasil tersebut dapat dihitung bahwa perhitungan tiap *attempt* pada skenario ini yaitu sebesar 6,59

detik. Untuk data lebih detail pada tiap percobaan lebih jelasnya dapat dilihat pada Tabel 4.1 dan untuk grafik hasil percobaan dapat di lihat pada Gambar 4.9.



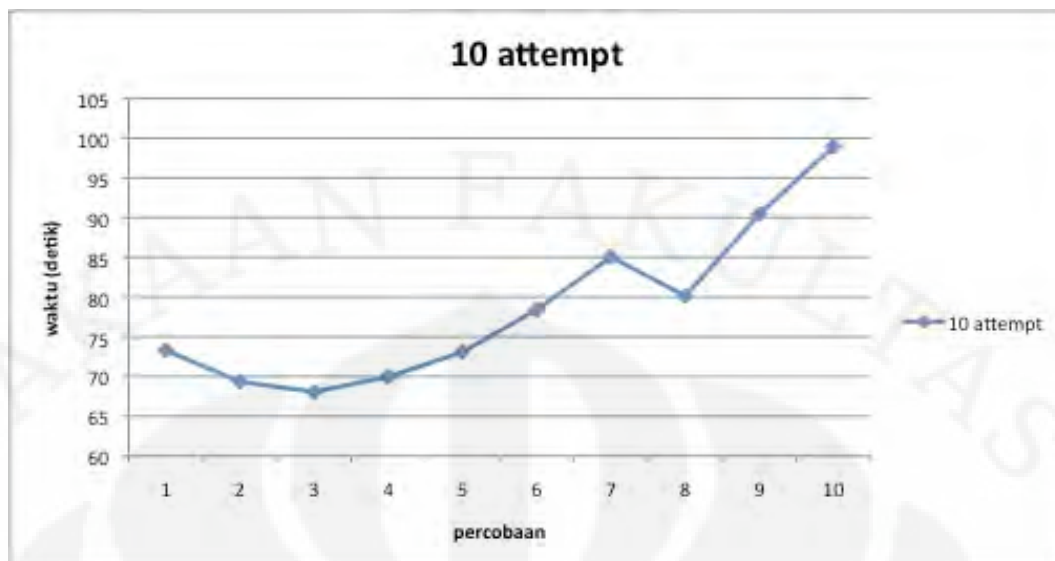
Gambar 4. 9 Hasil percobaan dengan 5 attempt

Pada skenario perhitungan 10 attempt didapatkan bahwa rata-rata waktu yang dibutuhkan untuk melakukan proses perhitungan untuk 10 attempt adalah sebesar 78,678 detik. Dari hasil tersebut dapat dihitung bahwa perhitungan tiap

Tabel 4. 2 Hasil Percobaan dengan 10 Attempt

Percobaan	Waktu (s)
1	73.320
2	69.351
3	68.064
4	69.967
5	73.086
6	78.408
7	85.063
8	80.132
9	90.440
10	98.948
Rata-Rata	78.678
@attempt	7.868

attempt pada skenario ini yaitu sebesar 7,868 detik. Untuk data lebih detail pada tiap percobaan dapat dilihat pada Tabel 4.2 dan untuk grafik hasil percobaan dapat dilihat pada Gambar 4.10.



Gambar 4. 10 Hasil percobaan dengan 10 *attempt*

Pada skenario perhitungan yang ketiga, dilakukan perhitungan 15 *attempt*. Setelah dilakukan percobaan didapatkan bahwa rata-rata waktu yang dibutuhkan untuk melakukan proses perhitungan untuk 15 *attempt* adalah sebesar 113,752

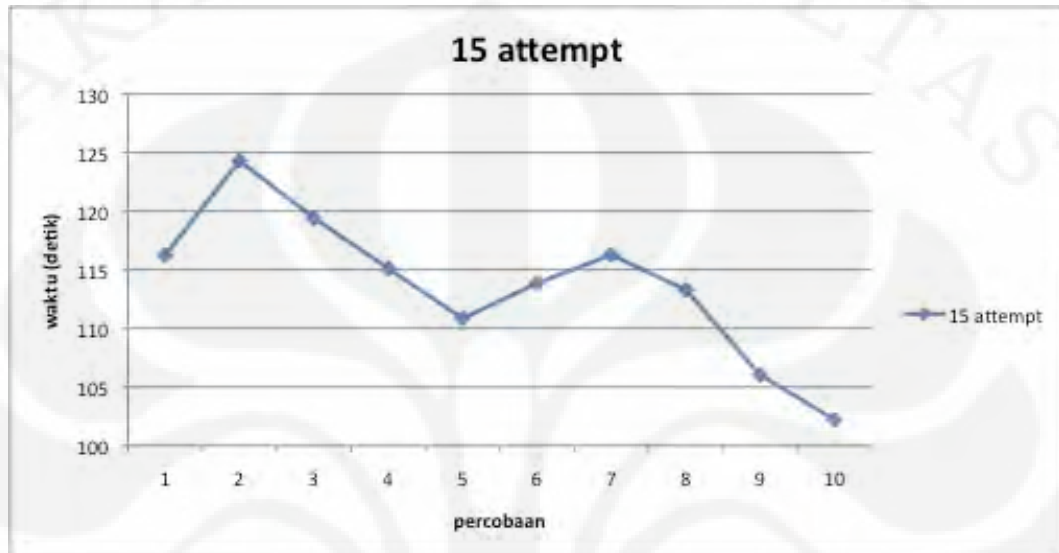
Tabel 4. 3 Hasil Percobaan dengan 15 *Attempt*

Percobaan	Waktu (s)
1	116.234
2	124.306
3	119.414
4	115.102
5	110.844
6	113.850
7	116.277
8	113.274
9	106.028
10	102.189
Rata-Rata	113.752
@attempt	7.583

detik. Dari hasil tersebut dapat dihitung bahwa perhitungan tiap *attempt* pada skenario ini yaitu sebesar 7,583 detik. Untuk data lebih detail pada tiap percobaan, dapat dilihat Gambar 4.11

Setelah dilakukan perhitungan, dapat diambil kesimpulan bahwa jumlah *attempt* yang dihitung berpengaruh langsung terhadap waktu pemrosesan halaman dimana SIMPLE-O diintegrasikan ke dalamnya. Semakin banyak *attempt* yang dihitung, semakin lama pula waktu yang dibutuhkan untuk melakukan

perhitungan. Selain itu, dapat disimpulkan pula bahwa antara program SIMPLE-O dimana perhitungan dilakukan langsung setelah seorang pelajar men-submit jawabannya dengan SIMPLE-O yang telah diintegrasikan ke Moodle dimana pengekseskuan perhitungan jawaban dilakukan oleh pengajar memiliki hasil yang sama.



Gambar 4. 11 Hasil percobaan dengan 15 *attempt*

BAB 5 KESIMPULAN

Pada skripsi ini dapat diambil beberapa kesimpulan, yaitu:

1. Aplikasi SIMPLE-O dapat berhasil diimplementasikan pada sebuah *Virtual Learning Environment* yang bernama Moodle.
2. Jumlah *attempt* yang dihitung berbanding lurus dengan lama waktu pengesekusian halaman perhitungan.
3. Rata-rata waktu yang dibutuhkan untuk memproses 5 attempt sebesar 32,983 detik, untuk 10 attempt sebesar 78,678 detik, dan untuk 15 attempt sebesar 113,752 detik.



DAFTAR REFERENSI

- [1] Budiarto, B., Hartanto, D., & Ratna, A.A.P., (2007, April). SIMPLE: Sistem penilaian esei otomatis untuk menilai ujian dalam bahasa indonesia, *Jurnal Makara Seri Teknologi*, 11.
- [2] Landauer, T.K, Foltz, P.W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259 – 284.
- [3] Moodle Documentation (n.d.) June 14, 2010. <http://docs.moodle.org>
- [4] MySQL :: The World's Most Popular Open Source Database (n.d.) June 13, 2010. <http://www.mysql.com>
- [5] PHP Manual (n.d.). June 13, 2010. <http://id.php.net/manual/en/index.php>
- [6] Ratna, A.A.P., et al., ed. (2007, December). *SIMPLE-O: Web based automated essay grading system using latent semantic analysis method for indonesian language considering weight word and word synonym*. Paper presented at The 10th International Conference on Quality in Research, Faculty of Engineering, University of Indonesia, Depok, Indonesia.
- [7] Ratna, A.A.P., et al., ed. (2006, September). SIMPLE: Sistem penilaian esei otomatis berbasis WEB dengan metode latent semantic analysis yang digunakan pada bahasa indonesia dengan penambahan kata bobot. *Journal of Technology*. 3.
- [8] WorldWideLearn (n.d.). *What is e-learning*. December 17, 2009. <http://www.worldwidelearn.com/elearning-essentials/index.html>
- [9] Zedan, Husein. Al-Ajlan, Ajlan. (2008) *Why Moodle*. May 26, 2010. 12th IEEE Workshop on Future Trends of Distributed Computing Systems. <http://ieeexplore.ieee.org>

