

PERANCANGAN APLIKASI TES BAHASA INGGRIS ONLINE (ATBIO) DENGAN SOAL ACAK

Hady Pranoto

Computer Science Department, School of Computer Science, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
hadi_pranoto2001@yahoo.co.id

ABSTRACT

English proficiency can be measured through several tests that are recognized internationally like TOEFL. To measure the English proficiency of 'Binusians' with TOEFL standards and provide rapid and credible test results, an program design of online English test application (ATBIO) is created by displaying a random question, in which every examinee gets different questions. Prior to the design, first performed profile analysis of the questions in the TOEFL test held by Binus University Language Center. Following is the design of the program using standard UML 2.0 notation. Furthermore, the author submitted the draft to be built by the program developers using java script or ASP. With credible, fast and accurate results of ATBIO, it is expected that people, departments or organizations concerned with these test results can immediately make decisions and take necessary actions.

Keywords: *Design of online English test application, TOEFL standard, random questions, Binus University Language Center*

ABSTRAK

Kemampuan berbahasa Inggris dapat diukur melalui beberapa tes yang diakui di dunia internasional, salah satunya adalah TOEFL. Untuk mengukur kemampuan Bahasa Inggris para Binusian dengan standar TOEFL dan memberikan hasil tes yang cepat serta dapat dipertanggung jawabkan, sebuah perancangan program Aplikasi Tes Bahasa Inggris Online (ATBIO) dibuat dengan menampilkan pertanyaan acak, di mana setiap peserta uji mendapatkan pertanyaan berbeda. Sebelum dilakukan perancangan, pertama-tama dilakukan analisis profile dari soal dalam tes TOEFL yang dimiliki oleh Language Center Binus University. Kemudian dilakukan desain program berdasarkan daur hidup pengembangan sistem (SDLC) dengan pendekatan Object Oriented menggunakan standar notasi UML 2.0. Selanjutnya penulis menyerahkan rancangan untuk dibangun oleh pengembang program menggunakan java script atau ASP. Dengan hasil ATBIO yang dapat dipertanggung jawabkan, cepat dan akurat, diharapkan orang, departemen atau organisasi yang berkepentingan dengan hasil tes ini dapat segera mengambil keputusan dan mengambil tindakan yang dibutuhkan.

Kata kunci: *perancangan Aplikasi Tes Bahasa Inggris Online (ATBIO), standar TOEFL, soal acak, Binus University Language Center*

PENDAHULUAN

Salah satu kompetensi yang harus dimiliki oleh Binusian adalah *language proficiency* 'kemampuan berbahasa'. Melalui *language proficiency* yang baik proses komunikasi akan lebih baik lagi, proses penyampaian idea atau gagasan akan berjalan lebih lancar. Seseorang yang memiliki *language proficiency* yang baik akan lebih lancar dalam berkomunikasi, dapat mengerti tentang gagasan yang sedang dibicarakan, setidaknya mereka memiliki perbendaharaan kosa kata yang lebih banyak.

Dalam upaya meningkatkan *language proficiency* bagi para Binusian, Binus University Language Center menyediakan beberapa program, di antaranya: mata kuliah *English Entrant*, *English Infocus*, dan *English Savvy* untuk mahasiswa; *English Course Development Program* (ECDP) untuk karyawan dan dosen; *English for Teaching Purposes* (ETP) dan *English for Academic Purposes* (EAP) yang dikhususkan untuk dosen.

Language Center menggunakan aplikasi tes Bahasa Inggris yang berbasis web untuk mengukur hasil pembelajaran dari program *language proficiency* yang dibuat. Pada jadwal tertentu aplikasi tersebut dijalankan untuk mendapatkan hasil evaluasi. Aplikasi yang dijalankan menampilkan sebuah set soal yang sudah ditentukan sebelumnya, dan peserta tinggal menjawab pertanyaan yang ditampilkan dan kemudian aplikasi akan melakukan kalkulasi nilai yang dihasilkan berdasarkan jawaban yang benar. Jumlah set soal yang sudah selesai dibuat saat ini adalah sejumlah 8 set soal. Jumlah set tersebut adalah jumlah set yang terlalu sedikit. Probabilitas seseorang untuk mendapat soal yang sama pada saat mengikuti tes yang kedua adalah sebanyak 1/8 atau 12.5 %. Angka probabilitas tersebut adalah sebuah angka yang cukup besar, jika seseorang mendapat soal tes yang sama mereka ada kemungkinan masih mengingat jawaban dari tes yang sebelumnya, hal ini menurunkan persentase kredibilitas tes tersebut merupakan cerminan kemampuan dari peserta tes tersebut.

Untuk meningkatkan kualitas hasil tes tersebut perlu dirancang sebuah aplikasi tes Bahasa Inggris menggunakan metode acak soal. Dengan mengacak soal yang ditampilkan, jumlah variasi set soal akan bertambah dari jumlah set soal sebelumnya yang berjumlah delapan set. Dengan bertambahnya variasi probabilitas seorang peserta tes mendapatkan soal sama akan lebih kecil lagi. Pada akhirnya kemungkinan pencontekan jawaban akan sangat kecil sehingga hasil tes seorang peserta uji dapat dipercaya kredibilitasnya.

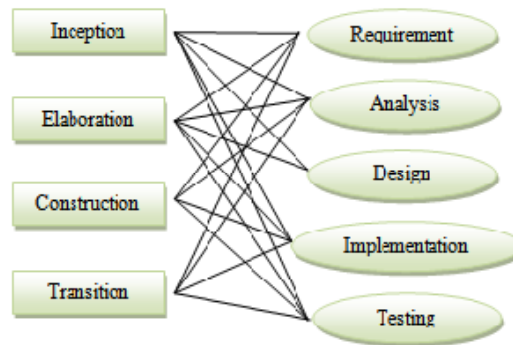
METODE

Dalam penelitian ini penulis mendasari penelitiannya pada pengamatan dokumen, aktifitas pelaksanaan tes dan mengamati alur kerja dari aplikasi yang sudah ada. Setelah mengamati atau melakukan observasi terhadap dokumen dan user, penulis melanjutkannya dengan mendesain sistem berdasarkan daur hidup pengembangan sistem (SDLC) dengan pendekatan Object Oriented (Britton & Doake, 2005). Pada dasarnya proses yang terjadi pada pendekatan obyek oriented dengan tradisional daur hidup sistem hampir sama, hanya kadang dibingungkan oleh istilah istilah yang digunakan.

SDLC dengan Pendekatan Object Oriented

SDLC dengan pendekatan Object Oriented yang penulis jalankan adalah pendekatan meliputi empat proses (Gambar 1). Pertama, *inception* – pada proses ini pekerjaan inisialisasi dan persetujuan tentang batasan batasan proyek dibuat. Ini termasuk proses membuat business case untuk proyek tersebut, analisis resiko yang mungkin muncul dan seberapa besar sistem ini akan dibangun. Proses ini

hampir mirip dengan *feasibility study*. Kedua, *elaboration* – proses ini berusaha membentuk arsitektur dasar sistem dan rencana pengimplementasian. Pada tahap ini proses desain sudah dibuat termasuk batasan dan berapa banyak biaya yang dibutuhkan. Ketiga, *construction* – pada proses ini pengkodean program sudah dilaksanakan. Proses pembangunan sistem ini seringkali mengalami beberapa pengulangan dalam prosesnya sampai kepada *beta release* dari sistem. Selanjutnya dilakukan pengujian dan perbaikan untuk menyempurnakan sistem. Keempat, *transition* – proses ini adalah proses transisi penyerahan sistem dari pengembang kepada klien. Pada proses ini pembetulan akhir atas kesalahan-kesalahan sistem tetap dilaksanakan, terlebih kesalahan-kesalahan yang muncul pada tahap konstruksi.



Gambar 1. Tahapan pembangunan sistem dalam sudut pandang *object oriented*.

Unified Modeling Language (UML)

UML adalah sebuah bahasa pemodelan dengan kegunaan umum yang terstandarisasi dalam bidang software engineering, standar ini di keluarkan oleh Object Management Group (OMG). Di dalam UML terdapat notasi grafik dan text untuk menciptakan model-model visual dari sistem yang dibuat. “UML digunakan untuk menspesifikasikan, memvisualisasi, membangun dan menstruktur segala sesuatu yang ditemukan dalam proses pembangunan sistem” (UML® Resource Page) seperti: orang yang terlibat dalam sistem, menggambarkan sistem bisnis, menggambarkan logika dari sistem, aktifitas dalam sistem, memperlihatkan bahasa pemrograman dalam sistem, menggambarkan skema *database*, dan memperlihatkan mekanisme penggunaan ulang komponen yang sudah ada. UML digunakan dalam proses pembuatan sistem berbasis *object oriented*. Diagram UML dapat mengambalikan dua gambaran yang berbeda tentang model sistem yaitu: (1) *static (structural)* – menggambarkan bentuk statis dari sistem menggunakan attribut obyek, operasi dan hubungan antar obyeknya. Diagram yang termasuk dalam bentuk ini adalah class diagram dan composite structure diagram; (2) *dynamic (behavioral)* – menggambarkan bentuk dinamis dari tingkah laku sistem dengan menampilkan kolaborasi antar obyeknya dan perubahan internal status keadaan dari obyeknya. Diagram yang masuk kedalam kategori ini adalah *sequence diagram*, *activity diagram* dan *state machine diagram*.

Problem Domain

Dalam tahap perancangan penulis melakukan pengamatan terhadap proses ujian yang berlangsung di dalam ruangan laboratorium Language Center. Dalam proses ini penulis mengamati bahwa dalam proses ujian ini peserta uji masih dapat mencontek jawaban dari peserta uji lainnya. Hal itu dapat terjadi karena soal yang dimunculkan ke layar komputer peserta uji yang satu sama dengan soal yang dimunculkan ke peserta uji lainnya, sehingga tentu saja jawaban benarnya adalah sama. Untuk itu penulis mempunyai ide untuk merancang sebuah aplikasi tes yang memunculkan soal yang berbeda antara orang yang satu dengan yang lain, tentu saja soal yang berbeda ini harus dirancang

memiliki bobot yang sama antara soal yang sejenis, apakah ada kemungkinan dua komputer yang bersebelahan menampilkan soal yang sama? Tentu saja mungkin, tapi probabilitas ini akan kecil sekali jika kita menyiapkan variasi soal yang berbeda dalam jumlah yang banyak. Dengan hilangnya aktifitas mencontek tersebut kredibilitas dari hasil tes yang akan dikeluarkan lebih baik lagi. Untuk itu penelitian kali ini menambah jumlah variasi soal tanpa menambah jumlah set soal.

HASIL DAN PEMBAHASAN

Pengamatan Proses Tes

Pada tulisan kali ini penulis akan mengambil contoh pada tes menggunakan tes TOEFL saja, penulis mengamati dokumen Paper Base TOEFL Test. Keputusan tersebut diambil atas dasar bahwa konsep yang sama juga dapat diaplikasikan pada tes Bahasa Inggris yang menggunakan standar TOEIC. Pengamatan pada dokumen dititik beratkan pada mengamati anatomi soal, apakah melalui anatomi soal yang ada, penulis dapat menambahkan jumlah variasi soal dengan mengkombinasikan anatomi soal tersebut, anatomi soal TOEFL adalah sebagai berikut:

Listening

Dalam standard TOEFL, Listening section dibagi menjadi tiga bagian yaitu: (1) *part A – Short Conversation*; (2) *part B – Longer Conversation*; (3) *part C – Long Talks*.

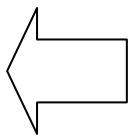
Structure

Structure section dibagi menjadi dua bagian yaitu: (1) *Structure*; (2) *Written Expression*.

Reading

Dalam section Reading, soal tes TOEFL terbagi menjadi lima buah paragraph yang masing-masing dikelompokkan pada: (1) paragraf 1: 9 soal; (2) paragraf 2: 9 soal; (3) paragraf 3: 11 soal; (4) paragraf 4: 10 soal; (5) paragraf 5: 11 soal. Pada tiap set soal yang sudah ada, jumlah soal tiap paragraf bisa berbeda-beda, tetapi penulis melihat bahwa ada pola yang sama tiap-tiap variasi soal tes, bahwa tiap paragraph memiliki pertanyaan sebanyak 10 soal. Untuk saat ini hanya ada satu macam variasi tes yang sudah ditetapkan sebelumnya oleh administrator, sehingga dimasing-masing layar komputer peserta uji menampilkan isi ujian yang sama. Untuk lebih jelasnya dapat dilihat pada Gambar 2.

t ID	Tes	Name	Variation	Va
r 001	Use	Name 1	r 1	Va
r 002	Use	Name 2	r 1	Va
r 003	Use	Name 3	r 1	Va
r 004	Use	Name 4	r 1	Va
r 005	Use	Name 5	r 1	Va



Variasi 1
Reading
Structure
Listening

Gambar 2. Variasi Soal (semua peserta menggunakan variasi soal yang sama)

Untuk konsep yang pertama ini kesempatan peserta uji untuk mencontek besar sekali, karena probabilitas seseorang peserta tes mendapatkan soal yang sama adalah 100%. Untuk mencegah hal ini terjadi penulis mempunyai ide untuk mengacak variasi set soal yang akan ditampilkan dilayar komputer peserta tes. Untuk itu kita memanfaatkan anatomi dari soal-soal yang menggunakan standard TOEFL tersebut. Sebagai data contoh kita asumsikan kita akan menggunakan 3 variasi set soal untuk dipakai dalam perhitungan variasi soal yang mungkin dibentuk melalui 3 variasi soal tersebut.

Peluang Variasi Soal

Sebelumnya ada pertanyaan bagaimanakah memperbanyak variasi soal tanpa menambah jumlah set yang dibuat. Caranya adalah dengan memanfaatkan anatomi dari soal TOEFL tersebut, dengan mengkombinasikan bagian-bagian dalam soal TOEFL tersebut maka jumlah kombinasi yang didapat akan lebih banyak lagi.

Mari kita hitung peluang kejadian yang kita bisa hitung dari variasi set soal tersebut. Untuk kita harus sepakat dulu terhadap penamaan yang akan kita gunakan. Untuk soal listening Part-A akan kita gunakan simbol A, untuk Part-B simbol B dan Part-C simbol C. Pada bagian Structure yang memiliki dua bagian kita simbolkan dengan S untuk bagian Structure, dan W untuk bagian Written Expression. Pada bagian reading kita gunakan simbol Ra sebagai Reading paragraf 1, Rb untuk Reading paragraf 2, Rc untuk Reading paragraf 3, Rd untuk Reading paragraf 4, dan Re untuk Reading paragraf 5, sedangkan untuk variasi set soal awal kita gunakan angka sebagai penomoran untuk variasi soal 1. Jadi kalau sebuah soal dengan kombinasi A1B2C3S2W3Ra2Rb1Rc2Rd1Re3, berarti soal tersebut merupakan kombinasi soal dari:

- Listening-Part A dari variasi set soal ke 1(A1) -> n1
- Listening-Part B dari variasi set soal ke 2(B2) -> n2
- Listening-Part C dari variasi set soal ke 3(C3) -> n3
- Structure-Part Structure dari variasi set soal ke 2(S2) -> n4
- Structure-Part Written Expression dari variasi set soal ke 3(W3) -> n5
- Reading-Part Paragraf 1 dari variasi set soal ke 2(Ra2) -> n6
- Reading-Part Paragraf 2 dari variasi set soal ke 1(Rb1) -> n7
- Reading-Part Paragraf 3 dari variasi set soal ke 2(Rc2) -> n8
- Reading-Part Paragraf 4 dari variasi set soal ke 1(Rd1) -> n9
- Reading-Part Paragraf 5 dari variasi set soal ke 3(Re3) -> n10

Sebuah peluang adalah pernyataan numerik tentang kemungkinan suatu kejadian yang dapat terjadi. Dalam hal ini peluang dapat diartikan sebagai suatu ukuran terhadap kepastian dan ketidakpastian. Nilai peluang adalah lebih besar atau sama dengan 0 dan lebih kecil atau sama dengan 1. Artinya jika peluang suatu kejadian sama dengan 0 berarti peluang kejadian tersebut adalah mustahil akan terjadi, jika nilai peluangnya adalah 1 maka peluang kejadian tersebut adalah pasti terjadi. (Lungan, 2006).

Berhubungan dengan peluang ada dalil yang mengatakan jika suatu operasi dapat diselesaikan dengan $n1$ cara, dan masing masing dapat dilanjutkan dengan $n2$ cara, dan seterusnya sampai dengan deretan ke k dengan nk cara, maka deretan k dari operasi tersebut dapat diselesaikan dengan:

$$N = n1 \times n2 \times \dots \times nk \text{ cara}$$

Rumus diatas dapat diterapkan pada anatomi soal TOEFL untuk mendapatkan jumlah variasi yang bisa didapat dari tiga set soal. Jadi banyaknya kemungkinan kombinasi soal(cara) yang berbeda adalah

$$\begin{aligned}
N &= n_1 * n_2 * n_3 * n_4 * n_5 * n_6 * n_7 * n_8 * n_9 * n_{10} \\
&= 3 * 3 * 3 * 3 * 3 * 3 * 3 * 3 * 3 * 3 \\
&= 3^n = 3^{10} = 59049 \text{ kemungkinan}
\end{aligned}$$

Dengan kombinasi soal seperti itu jenis soal yang dihasilkan akan menghasilkan jumlah kombinasi yang sangat banyak. Maka peluang untuk mendapatkan kombinasi set soal dengan kode A1B2C3S2W3Ra2Rb1Rc2Rd1Re3 adalah:

$$P(A1 B2C3S2W3Ra2Rb1Rc2Rd1Re3) = 1 / 59049 = 0.00169\%$$

Jika dilihat peluang hasil tersebut di atas sangatlah kecil, kemungkinan bahwa dua orang yang berdekatan akan mendapatkan soal yang sama.

Jika proses testing dilakukan di laboratorium *software*, proses pengacakan hanya terjadi pada *structure section* dan *reading section* saja. Hal ini karena pada *Listening section* suara dikeluarkan secara streaming ke *speaker* di dalam ruangan, jadi tidak memungkinkan menggunakan suara yang berbeda. Melihat kondisi tersebut maka kombinasi berbeda yang mungkin terjadi berkurang jumlahnya menjadi:

$$\begin{aligned}
N &= n_1 * n_2 * n_3 * n_4 * n_5 * n_6 * n_7 * n_8 * n_9 * n_{10} \\
&= 3 * 3 * 3 * 3 * 3 * 3 * 3 * 3 \\
&= 2187
\end{aligned}$$

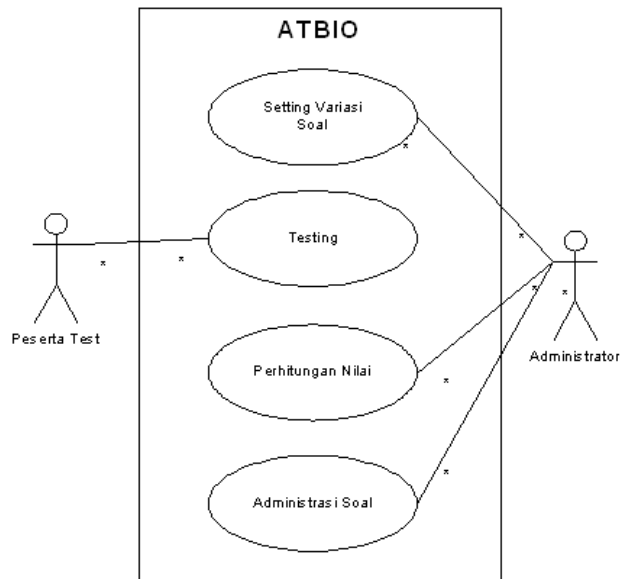
Maka peluang untuk mendapatkan kombinasi set soal dengan kode S2W3Ra2Rb1Rc2Rd1Re3 adalah:

$$P(S2W3Ra2Rb1Rc2Rd1Re3) = 1 / n_4 * n_5 * n_6 * n_7 * n_8 = 1/2187 = 0.0004572$$

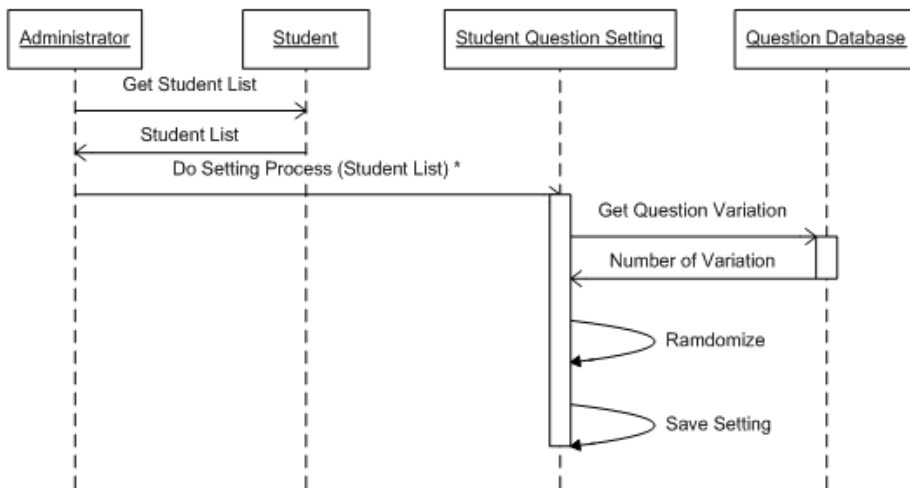
Rancangan Sistem

Penulis memberikan sebuah nama untuk aplikasi ini, yaitu Aplikasi Tes Bahasa Inggris Online (ATBIO). Sistem akan dibangun akan dibagi menjadi dua bagian subsistem yaitu bagian *front end* dan bagian *back end*. Bagian *front end* akan digunakan sebagai antar muka sistem dengan pengguna atau peserta tes, dan bagian *back end* akan digunakan sebagai antar muka antara sistem dan administrator dalam mengelola soal-soal tes dan melakukan perhitungan *score*. Penjelasan dapat dilihat dari *use case* di bawah ini (Gambar 3). Karena menggunakan variasi soal yang berbeda-beda untuk tiap peserta uji, perlu dilakukan *setting* jenis variasi soal terlebih dahulu dan disimpan ke dalam *database*. Hal ini dilakukan sebelum ujian dilaksanakan agar tidak mengganggu kinerja *server* yang akan digunakan sebagai *server* ujian. Proses ini dilakukan oleh subsistem *back-end*, proses ini dilakukan sekali saja sebelum sesi tes dilaksanakan. Untuk lebih jelasnya dapat dilihat pada Gambar 4 di bawah ini.

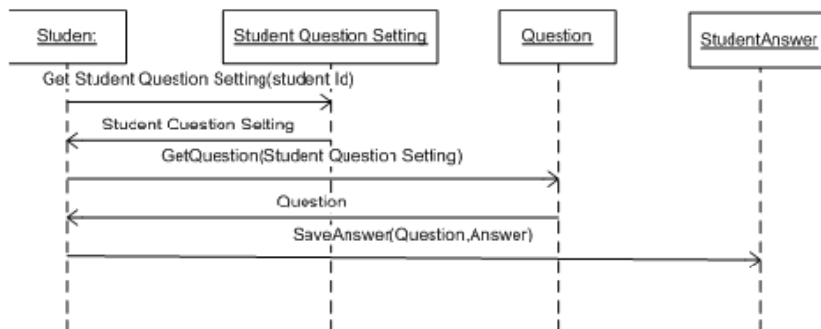
Para peserta memulai tes dan memasukan id-binusiannya. Kemudian sistem akan mengecek *setting* yang sudah diset sebelumnya dan menampilkan soal ujian berdasarkan set tersebut. Peserta tinggal menjawab pertanyaan yang diajukan dan saat mengklik jawaban, maka sistem akan langsung menyimpan jawaban peserta tersebut kedalam *question answer database*. Setelah proses ujian selesai, akan dilakukan proses perhitungan nilai pada subsistem *back-end*. Proses perhitungan ini akan membandingkan jawaban peserta tes dengan jawaban dari pertanyaan tersebut, sesuai dengan variasi set soal ujian yang sudah disetting sebelumnya. Jika jawaban benar, *score* akan bertambah satu dan disimpan kembali kedalam *database*. Setelah proses membandingkan selesai, dilanjutkan dengan proses penghitungan nilai akhir berdasarkan standar TOEFL yang sudah ditetapkan sebelumnya (Gambar 6). Setelah proses tersebut selesai, data yang berisi hasil proses perhitungan nilai tersebut oleh SLC diserahkan ke Language Center untuk digunakan atau didistribusikan ke unit-unit bersangkutan.



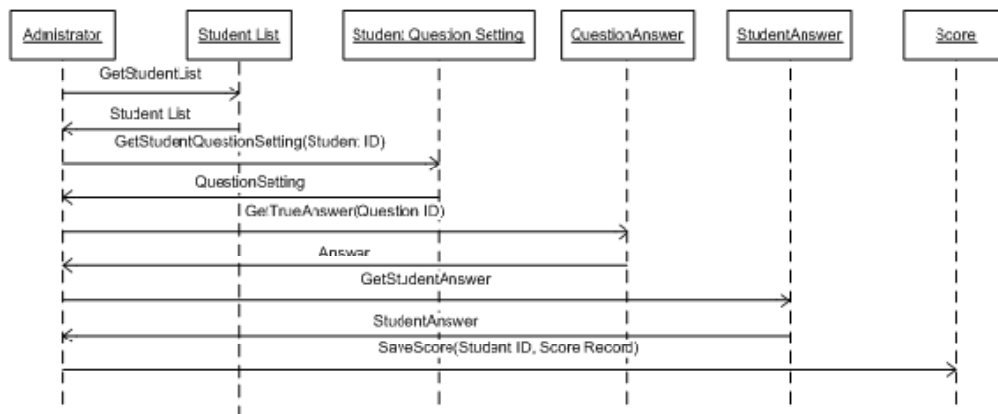
Gambar 3. Use case ATBIO system.



Gambar 4. Sequence diagram pengacakan soal dan penyimpanan setting.



Gambar 5. Sequence diagram proses menampilkan soal dan menyimpan jawaban.



Gambar 6. Sequence diagram proses koreksi dan menyimpan hasil penilaian.

PENUTUP

Berdasarkan hasil analisis dan perhitungan peluang, penulis menyimpulkan bahwa dengan cara mengacak soal yang ditampilkan akan memperbesar jumlah variasi set soal yang dibuat dengan tiga buah set soal akan didapat sebanyak 2187 kombinasi soal. Dengan membuat tampilan berbeda pada setiap komputer akan mengurangi kesempatan seorang peserta uji melakukan kecurangan, sehingga hasil tes dapat dipertanggungjawabkan kredibilitasnya dan kualitas dari tes yang diselenggarakan akan dapat meningkat dengan sendirinya. Konsep acak yang diterapkan pada Aplikasi Tes Bahasa Inggris Online (ATBIO) yang dibangun dapat mengurangi kecurangan peserta uji, sehingga dengan sendirinya akan meningkatkan validitas kemampuan peserta uji menjadi lebih valid lagi. Bagi peorangan dapat menjadi motivasi untuk meningkatkan kualitas kemampuan berbahasa dirinya. Sedangkan bagi institusi dapat mengunakannya sebagai acuan dalam mengambil kebijakan tertentu.

DAFTAR PUSTAKA

- Britton, C., & Doake, J. (2005). *A Student Guide to Object Oriented Development*. Burlington MA: Elsevier Butterworth Heineman.
- Lungan, R. (2006). *Aplikasi Statistika & Hitung Peluang*. Jakarta: Graha Ilmu.
- UML® Resource Page. (n.d.). *UML* Retrieved from UML® Resource Page, <http://www.uml.org/>