

**Perancangan Latihan TOEFL Menggunakan Framework Code Igniter  
untuk English Course Centre Banjarnegara**

**Ginangar Setyo Widodo**

**A11.2008.04325**

**Program Studi Teknik Informatika**

**Universitas Dian Nuswantoro**

**2013**

---

**Abstrak**

TOEFL (*Test Of English as Foreign Language*) yang kini berperan penting, baik untuk studi lanjut maupun seleksi kepegawaian, dituntut memiliki kredibilitas dalam penilaiannya. Sehingga ETS (Educational Testing Service), organisasi yang mengelola TOEFL sejak tahun 1963, beranjak ke metode TIBT (TOEFL Internet Based Test). Metode ini menggantikan PBT (Paper Based Test) yang tak lagi efisien. Dalam laporan ini penulis membuat rancangan TIBT untuk memenuhi kebutuhan LPK ECC Banjarnegara, sebagai rujukan untuk meningkatkan kemampuan berbahasa Inggris masyarakat Banjarnegara.

Penulis menggunakan framework Code Igniter (CI) untuk membuat sistem TIBT. Sebab CI memiliki keunggulan, seperti; gratis, ringan dan dapat diperluas. Sedangkan dalam perancangan sistem, pendekatan UML (*Unified Modeling Language*) digunakan.

Sebagai hasilnya, rancangan TIBT untuk ECC Banjarnegara memudahkan peserta untuk melakukan tes dari jarak jauh dan mengurangi penggunaan kertas untuk kegiatan tes ini.

Kegiatan TIBT ini, juga meningkatkan efisiensi dan daya saing ECC Banjarnegara sebagai suatu lembaga pendidikan. Namun sistem penulis masih membutuhkan *speech recognizer* untuk mengoptimalkan fitur *Speaking Test* dalam rangkaian TIBT.

Kata Kunci : TOEFL, TIBT, Code Igniter

## **1. Pendahuluan**

### **1.1 Latar Belakang**

Kemampuan berbahasa Inggris seseorang menjadi penting dalam kehidupan sehari-hari, karena kecakapan ini mulai diperhitungkan dalam dunia pendidikan maupun dunia kerja. Maka dari itu, mengikuti kursus menjadi perlu supaya mahir berbahasa Inggris, yang sesuai dengan penjelasan pasal 26 ayat 5 Undang-undang Nomor 20 Tahun 2003. [5] Kemudian guna melaksanakan fungsi kursus di daerah asal penulis, terdapat lembaga pendidikan English Course Center (ECC) Banjarnegara.

*Test Of English as Foreign Language* (TOEFL) digunakan untuk menilai kemampuan seseorang dalam berbahasa Inggris, memiliki beberapa metode/pendekatan. Adalah TOEFL *Paper Based Test* (PBT), metode konvensional populer yang selama ini digunakan. Namun metode ini sudah menjadi tidak efisien. Dikatakan demikian karena TOEFL PBT sangat terbatas dari segi waktu dan tempat. Misalnya pelaksanaan PBT di ECC, peserta harus ada di ruang tes saat TOEFL berlangsung. Sementara peserta PBT tidak seluruhnya berdomisili di kota Banjarnegara. Oleh karena itu, dibutuhkan

penerapan metode TOEFL *Internet Based Test* (TIBT). TIBT yang dikenal khalayak awam sebagai tes TOEFL *online*, yaitu tes yang bisa diakses melalui *browser* komputer dengan internet sebagai media perantara, dipilih karena memungkinkan peserta dari luar daerah untuk bisa mengikuti tes. Selain itu TIBT merupakan metode terkini dan terpercaya yang telah diakui oleh Educational Testing Service (ETS), sebuah lembaga sertifikasi TOEFL internasional. [12]

TIBT sendiri merupakan salah satu bentuk penerapan e-Learning berbasis web. Dimana TIBT ini dirancang menggunakan *framework* Code Igniter (CI). CI adalah sebuah *framework* dengan berbagai keunggulan, misalnya; gratis, ringan dan dapat diperluas. [13]

Beberapa ulasan di atas menjadi dasar untuk membuat aplikasi latihan TOEFL berbasis web untuk ECC Banjarnegara.

## 1.2 Rumusan Masalah

Mengetahui latar belakang yang telah dikemukakan sebelumnya, masalah dalam penelitian ini meruncing kepada pembuatan rancangan sistem tes TOEFL *online* yang baik untuk ECC Banjarnegara.

## 1.3 Batasan Masalah

Supaya tidak melenceng dari judul dan perumusan masalah yang ditentukan, perlu ada batasan masalah. Maka dari itu, pembahasan hanya akan berorientasi pada sistem tes TOEFL yang mencakup; tampilan, eksekusi ujian/tes, dan kalkulasi skor hasil tes.

## 1.4 Tujuan Penelitian

Sebuah situs web sebagai prototipe sistem berbasis internet yang mampu mengakomodir kegiatan tes TOEFL guna rujukan memperoleh

sertifikat. Sebuah *website* yang bisa diakses oleh peserta tes TOEFL kapan saja, dan dimana saja.

## 1.5 Manfaat Penelitian

### 1.5.1 Bagi Pemilik Perusahaan

Menambah jumlah dan kualitas layanan yang ada di lembaga pendidikan ECC Banjarnegara. Kemudian membangun *image* yang baik di kancah nasional, karena produk ini nantinya akan menjadi komoditi yang mampu menaikkan daya saing ECC.

### 1.5.2 Bagi Universitas Dian Nuswantoro

Menjadi tolok ukur keberhasilan institusi dalam mendidik mahasiswanya. Dan laporan penelitian dapat dijadikan sebagai koleksi pustaka yang menyediakan informasi serta ilmu lebih bagi rekan-rekan mahasiswa juga pembaca lainnya.

## 2. Tinjauan Pustaka

### 2.1 TOEFL

#### 2.1.1 Konsep Dasar TOEFL

TOEFL adalah suatu tes yang digunakan untuk mengukur kemampuan berbahasa Inggris bagi orang yang menggunakan bahasa Inggris sebagai bahasa keduanya (*non-native*). Sedangkan Menurut Patricia N. Sulliuian and Grace Y. Q. Zhong, "*TOEFL is the porpose of this is to measure your ability to read and understand North American English.*" TOEFL dimaksudkan sebagai alat ukur atau evaluasi atas kemampuan berbahasa Inggris seseorang. [10]

### 2.1.2 Perkembangan TOEFL

Untuk metode pelaksanaan tes TOEFL sesuai urutan perkembangannya terdiri dari:

Tabel 2.1 Perkembangan TOEFL

Stages	Construct	Content
1. The First TOEFL Test 1964–1979	Discrete components of language skills and knowledge	Multiple-choice items assessing vocabulary, reading comprehension, listening comprehension, knowledge of correct English structure and grammar
2. A Suite of TOEFL Tests 1979–2005	Original constructs (listening, reading, structure and grammar) retained but two additional ones added—writing ability and speaking ability	In addition to multiple-choice items assessing the original constructs, separate constructed-response tests of writing, the <i>TWE</i> <sup>®</sup> test and speaking, the <i>TSE</i> <sup>®</sup> test, were developed
3. The TOEFL iBT <sup>®</sup> Test 2005–present	Communicative competence—the ability to put language knowledge to use in relevant contexts	Academic tasks were developed that require the integration of receptive and productive skills such as listening, reading and writing or speaking, as well multiple-choice items for listening and reading

Sumber: TOEFL iBT<sup>®</sup> Research (2011). *Insight — TOEFL<sup>®</sup> Program History*.

Seperti halnya yang diungkapkan oleh TOEFL Indonesia, tes TOEFL menurut urutan perkembangannya dinyatakan sebagai berikut:

- a. TOEFL Paper Based Test  
Terdiri dari 3 *section*: Listening, Structure, and Reading.  
*Maximum Score*: 677.
- b. TOEFL Computer Based Test  
Terdiri dari 4 *section*: Listening, Structure, Reading, Writing.  
*Maximum Score*: 300.
- c. TOEFL Internet Based Test  
Terdiri dari 4 *section*: Listening, Reading, Writing, Speaking.  
*Maximum score*: 120. [15]

### 2.1.3 TOEFL Internet Based Test (TIBT)

TIBT menekankan pada keterampilan terpadu dan pengukuran keempat keterampilan berbahasa, termasuk *speaking*. Isi dari tes ini merupakan benar adanya, dan bahasanya bersifat konsisten

dengan bahasa yang digunakan di kehidupan sehari-hari, lingkungan akademik sesungguhnya.

Tes ini memiliki empat sesi:

- a. **Reading**, mengukur kemampuan untuk memahami uraian membaca.
- b. **Listening**, mengukur kemampuan untuk memahami bahasa Inggris sebagai bahasa pengantar yang digunakan di kampus dan universitas.
- c. **Speaking**, mengukur kemampuan untuk berbicara bahasa Inggris.
- d. **Writing**, mengukur kemampuan untuk menulis sebagaimana yang telah menjadi kelayakan di universitas.

Muatan dari TIBT ini berdasarkan “*corpus*” atau basis data dari bahasa verbal dan nonverbal yang saat ini menampung lebih dari 2,7 juta kata, dihimpun dari berbagai institusi pendidikan di Amerika Serikat. [16]

### 2.2 Framework Code Igniter

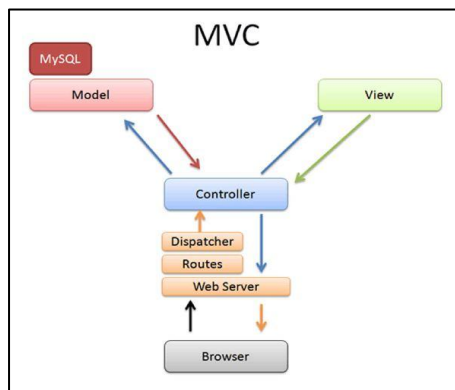
CodeIgniter (CI) adalah sebuah kerangka pembangunan aplikasi atau mudahnya disebut *toolkit*, untuk *developer* yang akan membuat aplikasi web dengan PHP. Tujuan CI adalah supaya pembangunan aplikasi lebih cepat dibanding menulis *source code* dari awal, karena CI telah menyediakan banyak library untuk proses-proses yang sering digunakan pada suatu aplikasi, dan juga dengan kemudahan dalam menggunakan library tersebut serta kesederhaan penggunaannya. [13]

CodeIgniter ditulis (dibuat) oleh Rick Ellis, seorang musisi rock yang menjadi *programmer*. Ellis membangun perusahaan

bernama Ellis Lab, yang mengembangkan beberapa produk unggulan yang salah satunya adalah CodeIgniter. [11] CI diperkenalkan kepada publik pada tanggal 28 Februari 2006. CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller *development pattern*. [3]

### 2.2.1 Konsep MVC CI

Konsep MVC adalah konsep pemisahan antara *logic* dengan tampilan dan database. Manfaat konsep ini adalah, membuat *coding logic* lebih *simple*, karena sudah dipisah dengan *code* untuk tampilan dan membuat *programmer* dapat bekerja secara terpisah dengan *designer*. *Programmer* mengerjakan *logic*, sedangkan *designer* berkecukupan dengan design dan tampilan. [13]



Gambar 2.1 Perbedaan CodeIgniter dengan Framework Lain  
 Sumber: Id, Ibnu Daqiqil, M.Ti. (2011). *Framework CodeIgniter - Sebuah Panduan dan Best Practice*.

Adapun komponen-komponen MVC menurut Ibnu Daqiqil, antara lain:

#### a. Model

Model berhubungan dengan data dan interaksi ke database atau

*webservice*. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun *webservice*. Biasanya di dalam model akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data *website*. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

#### b. View

View berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman web, RSS, javascript dan lain-lain. *Programmer* harus menghindari adanya logika atau pemrosesan data di View. Di dalam View hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam View jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari Model dan Controller.

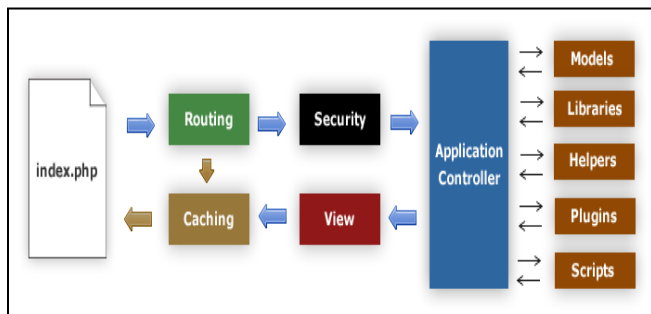
#### c. Controller

Controller bertindak sebagai penghubung data dan View. Di

dalam Controller inilah terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada Model. Tugas Controller adalah menyediakan berbagai variabel yang akan ditampilkan di View, memanggil Model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input. [3]

### 2.2.2 Karakteristik CI

Karakter CodeIgniter yang begitu khas dijelaskan oleh Candra Utama (2011) melalui alur proses:



Gambar 2.2 Alur Proses CodeIgniter

Sumber: Utama, Candra (2011).  
CodeIgniter Framework — IT507 -  
Rekayasa Web.

Gambar 2.2 selanjutnya dijelaskan sebagai berikut:

- a. **Index.php** berfungsi sebagai pengendali awal, menginisialisasi

sumber daya utama yang dibutuhkan CodeIgniter.

- b. **Router** memeriksa paket HTTP *request* untuk menentukan aksi apa yang harus dilakukan oleh sistem.
- c. Jika **cache** tersedia, maka halaman langsung dikirim ke browser, eksekusi sistem yang normal akan dilewati.
- d. **Security**. Sebelum *Application Controller* dieksekusi, paket HTTP *request* dan semua data yang dikirimkan pengguna akan disaring terlebih dahulu oleh *Security Class*.
- e. **Application Controller** menginisialisasi model, library utama, *helpers* dan semua sumberdaya yang dibutuhkan untuk setiap *request*.
- f. Antarmuka aplikasi (**View**) yang sudah disiapkan dikirimkan ke *browser*. Jika *caching* diaktifkan, maka View akan disimpan sementara untuk *request* yang sama berikutnya. [13]

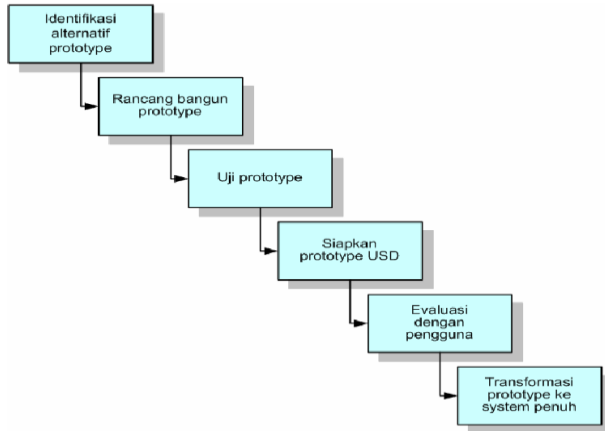
## 2.3 Prototype

### 2.3.1 Konsep Prototype

*Prototyping* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan (Howard, 1997). [8]

### 2.3.2 Tahapan Prototype

Tahap-tahap dalam *prototyping* boleh dikatakan merupakan tahap-tahap yang dipercepat. Strategi utama dalam *prototyping* adalah kerjakan yang mudah terlebih dahulu dan sampaikan hasil kepada pengguna sesegera mungkin.



Gambar 2.3 Tahapan Prototype

Sumber: Mulyanto, Aunur R. (2008).

*Rekayasa Perangkat Lunak Jilid 1 untuk SMK.*

## 3. Metode Penelitian

### 3.1 Analisis Penelitian

Analisis yang dijabarkan dalam beberapa langkah seperti berikut :

#### a. Mengidentifikasi Masalah

TOEFL yang selama ini diadakan oleh ECC Banjarnegara adalah *Paper Based Test* (PBT), yaitu tes yang mengandalkan kertas dan alat tulis dalam pelaksanaannya. Cara konvensional ini belakangan dipandang sudah tidak efisien lagi. Dikatakan demikian karena PBT yang ada sangat terbatas dari segi waktu dan tempat.

#### b. Memahami Kerja dari Sistem

Menjelaskan mekanisme TOEFL PBT di ECC, tes dilaksanakan dalam suatu ruangan yang dilengkapi piranti audio. Setiap peserta memiliki lembar soal dan jawaban di mejanya. Waktu pengerjaan untuk setiap bagiannya sudah ditentukan, secara sekuensial sehingga peserta tidak diperkenankan mengerjakan soal dari bagian tes yang sebelumnya.

#### c. Menganalisis Kelemahan Sistem

Kelemahan sistem lama yang selama ini dijalankan adalah keterbatasan waktu dan tempat pelaksanaan tes yang terjadwal hanya diadakan kurang dari empat kali dalam satu tahun. PBT juga dianggap boros sumber daya karena setiap menjelang pelaksanaannya harus terlebih dahulu dicetak lembar soal dan jawabannya.

## 3.2 Pengembangan Sistem

Metode yang sesuai dalam pengembangan sistem TIBT adalah *prototype*, yang terdiri dari:

### 3.2.1 Identifikasi

Adapun *prototyping model* yang penulis gunakan adalah *prototype* yang *reusable*. Dimana dari segi *prototype level*-nya mencakup *input/output* (antar muka pengguna), *processing* (perawatan file dasar dan proses-proses transaksi), dan *system* (model lengkap dari perangkat lunak).

### 3.2.2 Rancang Bangun

Dalam perancangan digunakan CASE *software* (*Computer Aided System Engineering*) seperti Enterprise Architect,

lalu pengolahan database SQL oleh phpMyAdmin.

Dalam pengembangan sistem diterapkan prinsip OOD (*Object Oriented Design*) metode UML (*Unified Modeling Language*) yang digambarkan oleh *Use Case Diagram*, *Class Diagram* dan *Sequence Diagram*.

### 3.2.3 Pengujian

Pengujian dari sistem ini dilakukan dengan simulasi langsung di hadapan peserta pada komputer yang dilengkapi *offline server* seperti WAMP Server. Simulasi ini dilakukan menggunakan komputer dan *headset* yang disediakan penulis. Secara acak, sampel yang didapat adalah lima orang siswa ECC. Pelaksanaan tes secara estafet di satu komputer dari peserta pertama hingga akhir. Prosesi simulasi ini juga dibimbing dan diawasi oleh Pak Zain, pendiri ECC.

### 3.2.4 Prototype USD (*User's System Diagram*)

USD pada *prototype* ini lebih lanjut dijelaskan melalui *Use Case Diagram* dan *Scenario Diagram*.

### 3.2.5 Evaluasi

Tahap ini melibatkan Pak Zain sebagai pengguna produk, yang menentukan kelayakan TIBT rancangan penulis.

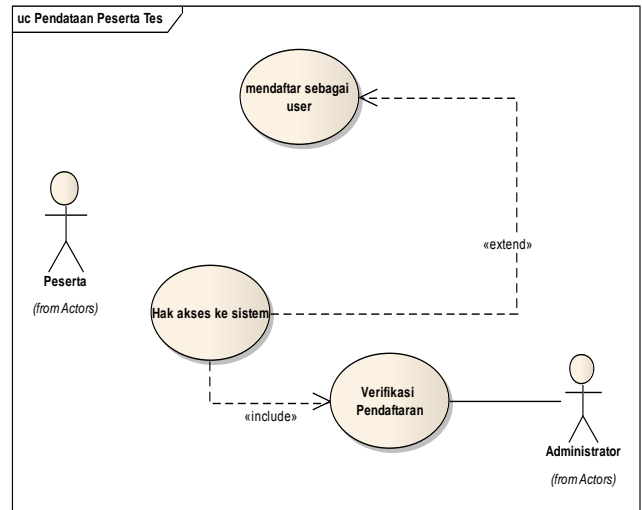
### 3.2.6 Transformasi

Setelah evaluasi dan revisi dari tahap pengujian selesai, produk *final* penulis lalu diterapkan pada beberapa unit komputer ECC yang dilengkapi server lokal terintegrasi, yang sepenuhnya diterapkan sebagai TIBT di ECC Banjarnegara.

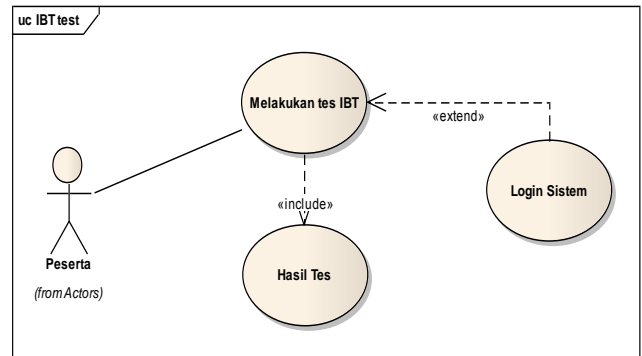
## 4. Hasil dan Pembahasan

## 4.1 Perancangan

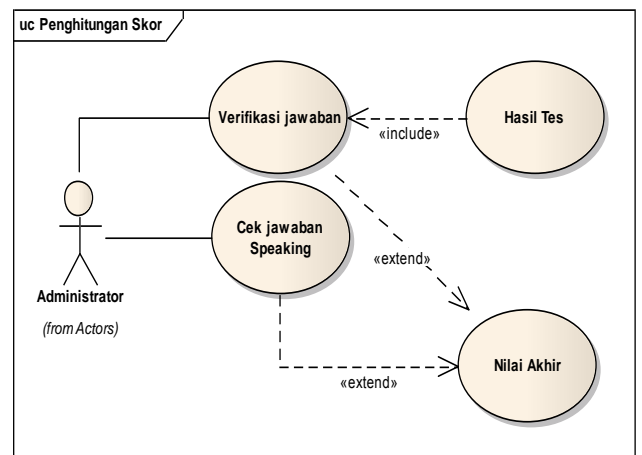
### 4.1.1 Pemodelan *Use Case Diagram*



Gambar 4.1 Diagram *Use Case* Pendataan Peserta Tes

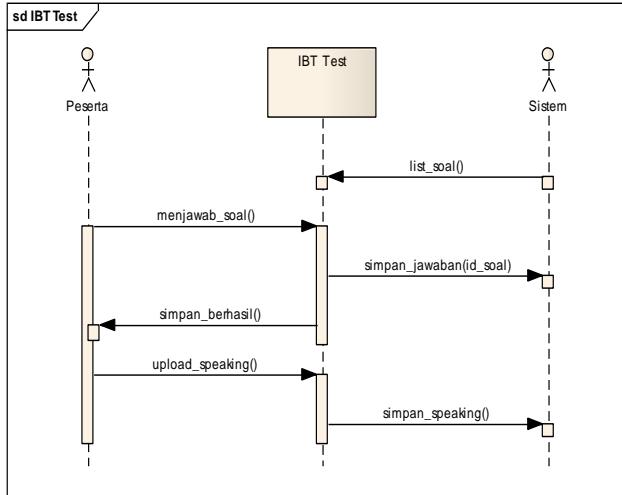


Gambar 4.2 Diagram *Use Case* IBT Test



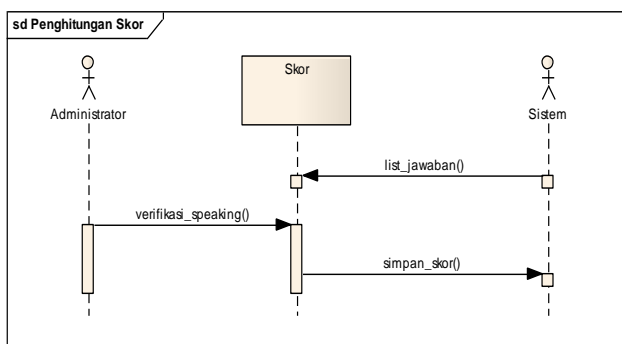
Gambar 4.3 Diagram *Use Case*  
Penghitungan Skor

#### 4.1.2 Pemodelan *Sequence Diagram*



Gambar 4.4 *Sequence Diagram* IBT Test

Pada gambar 4.4 ditunjukkan proses tes *online* melalui antarmuka. Sistem menampilkan semua soal yang tersedia. Peserta kemudian menjawab soal yang telah disediakan pada antarmuka dan selanjutnya disimpan. Untuk soal *speaking*, Peserta upload file *speaking* melalui antarmuka.



Gambar 4.5 *Sequence Diagram*  
Penghitungan Skor

Proses yang ditampilkan pada gambar 4.5 adalah proses penghitungan skor yang dilakukan oleh sistem. Sistem menampilkan peserta yang telah melakukan tes, untuk kemudian diverifikasi oleh administrator. Khusus untuk jawaban *speaking*, Administrator memberikan nilai secara manual untuk kemudian disimpan ke dalam sistem. Selanjutnya sistem akan menghitung skor secara otomatis.

#### 4.1.3 Antarmuka Sistem

Gambar 4.6 Desain Antarmuka untuk  
Halaman Utama

Halaman utama yang ditampilkan pada sistem ini adalah halaman yang dapat diakses oleh semua pengunjung, baik yang sudah terdaftar maupun yang belum terdaftar. Pada halaman utama terdapat empat elemen, yaitu *header*, *menu*, *content*, dan *login*.

Gambar 4.7 Desain Antarmuka Halaman  
Peserta

Setelah melakukan login, user akan dibawa menuju ke halaman akun peserta. Pada halaman ini, terdapat elemen *Header*, *Menu*, dan *Content* seperti pada halaman utama namun *Custom* untuk tiap peserta berbeda. Pada bagian menu, peserta dapat memilih untuk tes, melihat hasil, atau cetak hasil tes. *Content* dapat berisi berita atau berupa soal yang ditampilkan oleh sistem.



## 4.2 Implementasi Sistem

Hasil implementasi terhadap rancangan model sistem adalah sebuah situs atau aplikasi berbasis web yang dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL yang dibantu dengan penggunaan WAMP Server sebagai server web lokal dengan URL lokal *localhost*. Kebutuhan untuk menjalankan sistem ini adalah sebuah komputer yang dilengkapi dengan server web lokal dan *web browser*.

## 5. Penutup

### 5.1 Kesimpulan

Dengan adanya Sistem Latihan TOEFL ECC Banjarnegara ini, akan memudahkan peserta tes untuk melakukan tes dari jarak jauh. Selain itu, dengan adanya tes *online* ini, akan mengurangi penggunaan kertas untuk kegiatan tes ini.

Kegiatan tes *online* ini, juga meningkatkan efisiensi dan daya saing karena keberhasilan suatu perusahaan tergantung dari penggunaan teknologi dalam unit usahanya.

### 5.2 Saran

Setelah menyelesaikan tugas akhir ini, ada beberapa kekurangan pada karena keterbatasan waktu dan sumber daya. Beberapa kekurangan tersebut dirangkum pada saran-saran di bawah ini.

1. Sistem ini belum dapat menangani secara langsung tes yang berupa *speaking*, karena tes tersebut membutuhkan sebuah metode yang

disebut *speech recognizing*. Sehingga peserta harus meng-*upload* jawaban ke dalam sistem.

2. Sistem ini belum dapat menangani pendaftaran dan pembayaran secara *online*, sehingga perlu adanya proses yang menangani pendaftaran dan pembayaran secara langsung.

## DAFTAR PUSTAKA

- [1] Adri, Muhammad, S.Pd., M.T. (2008). *Pemrograman Web — Materi 1 Konsep Dasar Web Engineering*. Modul Kuliah. Universitas Negeri Padang.
- [2] Boulton, Helen (2008). *Managing e-Learning: What are the Real Implications for Schools?* United Kingdom: Nottingham Trent University.
- [3] Id, Ibnu Daqiqil, M.Ti. (2011). *Framework CodeIgniter — Sebuah Panduan dan Best Practice*. Pekanbaru: Koder.
- [4] Jogyanto, HM., MBA., AKT., Ph.D. (2008). *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
- [5] Kemendiknas (2010). *Apa dan Bagaimana Pembinaan Kursus dan Kelembagaan*. Jakarta: Ditjen Pendidikan Formal dan Informal.
- [6] Kusnendar, Jajang (2008). *Rekayasa Perangkat Lunak*. Modul Kuliah. Universitas Kuningan.
- [7] Muhadji, Kinta (2007). *PHP & MySQL Web Development*. Jurnal Elektronik. IlmuKomputer.com.

- [8] Mulyanto , Aunur R. (2008). *Rekayasa Perangkat Lunak Jilid 1 untuk SMK*. Jakarta: Departemen Pendidikan Nasional.
- [9] Pressman Ph.D, Roger S. (2009). *Software Engineering: A Practitioner's Approach (Seventh Edition)*. New York: McGraw-Hill.
- [10] Primastuti, Syifa, S.Kom. (2011). *Sistem Pembelajaran untuk Pelatihan dan Test TOEFL Menggunakan VoIP*. Jurnal Ilmiah. Institut Teknologi Sepuluh Nopember Surabaya.
- [11] Sofwan, Akhmad (2010). *Belajar PHP dengan Framework Code Igniter*. Jakarta: IKC.
- [12] TOEFL iBT® Research (2011). *Insight — TOEFL® Program History*. New Jersey: ETS.
- [13] Utama, Candra (2011). *CodeIgniter Framework — IT507 - Rekayasa Web*. Bandung: UNPAS.
- [14] Yulianto, Ardhian Agung, et. al. (2010). *Analisis dan Desain Sistem Informasi*. Modul Kuliah. Polteknik Telkom Bandung.
- [15] <http://www.toeflindonesia.com/>, diakses tanggal 24 April 2012.
- [16] [http://www.itc-indonesia.com/index.php?option=com\\_content&view=article&id=91](http://www.itc-indonesia.com/index.php?option=com_content&view=article&id=91), diakses tanggal 3 Mei 2012.