

PEMBUATAN APLIKASI TOEFL SEBAGAI MEDIA PELATIHAN BAHASA INGGRIS BERBASIS WEB

Yongki Yonatan Marbun¹⁾, R. Rizal Isnanto²⁾, Kurniawan Teguh Martono²⁾
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstrak - Penggunaan internet telah memberikan kemudahan bagi pengguna dalam mengakses informasi. Salah satu layanan yang dari teknologi internet adalah web. Web selain berfungsi sebagai media informasi, juga dapat dijadikan sebagai media pelatihan dan pembelajaran. Salah satu contohnya yaitu penggunaan web dalam proses pelatihan dan pembelajaran TOEFL (Test of English as Foreign Language). TOEFL merupakan sebuah tes yang digunakan untuk mengukur kemampuan bahasa Inggris seseorang. Sementara itu untuk mendapatkan skor TOEFL yang tinggi bukanlah hal yang mudah. Biasanya peserta tes harus belajar dan mengerjakan soal TOEFL yang terdapat pada buku untuk mengukur kemampuannya. Metode belajar dengan menggunakan buku ini, membutuhkan waktu yang relatif banyak. Seseorang harus menjawab soal, kemudian memeriksa jawaban, menghitung jumlah soal yang benar, dan kemudian mengkonversi jumlah jawaban yang benar serta menghitung jumlah skor yang didapat. Dengan metode belajar yang seperti itu akan membutuhkan tambahan waktu.

Perancangan aplikasi TOEFL pada penelitian ini bertujuan untuk membuat aplikasi TOEFL yang dapat digunakan untuk menguji kemampuan diri dimana dalam pengaksesan aplikasi ini membutuhkan internet. Aplikasi ini berbasis web dengan menggunakan bahasa pemrograman PHP dan basisdata MySQL dengan menggunakan metode Waterfall pada proses perancangannya. Perencanaan sistem menggunakan metode terstruktur dengan mendefinisikan DFD, ERD, serta diagram alir sistem.

Pengujian aplikasi TOEFL ini menggunakan metode pengujian Black Box. Dimana berdasarkan pengujian black box yang sering disebut pengujian fungsionalitas semua tombol dapat digunakan dan menampilkan tanggapan sesuai keinginan dan pengujian oleh pengguna (Beta testing) dengan hasil lebih besar dari 82%, dan pengujian reliability menggunakan aplikasi WAPT 8.5 versi trial dengan $R = 0.987$ dan $error\ rate = 0.013$ yang berarti berdasarkan standar Telcordia perangkat ini memenuhi kriteria Reliability. Untuk selanjutnya, aplikasi ujian TOEFL berbasis web ini dapat dikembangkan dengan menggunakan bahasa pemrograman dan basisdata yang lain, dibuat dalam platform yang berbeda, serta dikembangkan dengan

materi yang sesuai dengan kebutuhan dalam pengujian kemampuan bahasa Inggris seperti TOEF, TOEIC, dan IELTS.

Kata Kunci : Internet, TOEFL, Web, Black Box, Reliability

I. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi web yang sangat cepat, aplikasi-aplikasi web yang bermunculan juga semakin banyak. Meningkatnya teknologi web ini dipengaruhi juga karena kehadiran Internet. Kehadiran Internet dapat memudahkan kehidupan, salah satunya pada bidang pendidikan. Pada saat ini, sudah banyak situs web yang menyediakan media pembelajaran dan ujian *online* yang semakin interaktif serta mudah untuk dipelajari.

Pada zaman sekarang ini, bahasa Inggris merupakan salah satu bahasa internasional yang sering digunakan. Salah satu pelatihan bahasa Inggris yang sudah tidak asing lagi adalah TOEFL (*Test of English as Foreign Language*). Biasanya TOEFL digunakan untuk mempersiapkan diri bagi mereka yang ingin belajar ke luar negeri. Tetapi saat ini TOEFL juga digunakan oleh instansi pendidikan, seperti pada perkuliahan sebagai syarat masuk dan syarat kelulusan.

Untuk melatih kemampuan dalam menjawab soal-soal TOEFL tersebut, calon peserta membeli buku TOEFL dari toko buku untuk mendapatkan contoh soal-soal TOEFL yang biasanya digunakan pada ujian TOEFL. Metode belajar dengan menggunakan buku ini, membutuhkan waktu yang relatif banyak. Seseorang harus menjawab soal, kemudian memeriksa jawaban, menghitung jumlah soal yang benar, dan kemudian menghitung jumlah skor yang didapat. Dengan metode belajar yang seperti itu akan membutuhkan tambahan waktu.

Oleh karena itu, untuk memberikan alternatif dalam menguji kemampuan sebelum mengikuti ujian TOEFL PBT dibutuhkan suatu sarana bantuan yang harus bersifat hemat waktu (*time saving*), dan mempermudah dalam melakukan pengujian kemampuan diri. Permasalahan ini dapat diselesaikan dengan menggunakan fasilitas Internet yang dapat digunakan sebagai sarana untuk menguji kemampuan terhadap

materi yang telah dipelajari yaitu berupa tes TOEFL untuk menguji kemampuan diri.

1.2 Tujuan

Tujuan dari Tugas Akhir ini adalah: Membuat aplikasi ujian TOEFL sebagai media pelatihan berbasis web yang dapat digunakan untuk mengukur kemampuan diri.

1.3 Batasan Masalah

Dalam penulisan tugas akhir ini, penulis membatasi pembahasan dalam hal berikut:

1. Aplikasi Test TOEFL dirancang dan dibuat dengan menggunakan bahasa pemrograman PHP dan MySQL dimana aplikasinya berbasis web dan diakses berbasis desktop.
2. Tugas akhir ini terbatas hanya membahas cara perancangan dan pembuatan aplikasi ujian TOEFL.
3. Aplikasi ini hanya berupa aplikasi ujian TOEFL tanpa pembahasan dimana soal-soal yang disediakan diambil dari buku dan ujian dalam aplikasi ujian TOEFL ini meliputi: *Reading*, *Grammar*, dan *Listening*. Pada bagian *Listening*, aplikasi ini dilengkapi dengan dukungan pengaksesan *audio* (.wav atau .mp3).
4. Pengguna aplikasi ini adalah orang-orang yang ingin menguji kemampuan diri dalam bahasa Inggris.
5. Aplikasi yang dibuat merupakan aplikasi yang mirip dengan tes TOEFL PBT (*Paper Based Test*). Aplikasi ujian TOEFL ini hanya sebuah simulasi, yang digunakan untuk mengukur kemampuan diri pengguna bukan sebagai penilaian akhir hasil tes TOEFL dan tidak untuk menggantikan tes yang sebenarnya.

II. LANDASAN TEORI

2.1 TOEFL

TOEFL merupakan kependekan dari *Test of English as a Foreign Language*, yang merupakan ujian yang dilakukan untuk mengukur sejauh mana kemampuan bahasa Inggris seseorang. Tes ini digunakan untuk keperluan mendaftar ke lembaga pendidikan di Amerika Serikat maupun di negara lain. Tes ini pertama kali diadakan pada tahun 1963 dan telah diikuti oleh jutaan orang di seluruh dunia^[3]. TOEFL PBT merupakan jenis tes TOEFL yang pertama kali digunakan. Dilakukan secara manual menggunakan kertas jawaban dimana peserta ujian harus menghitamkan pilihan jawaban dengan menggunakan pensil^[4].

TOEFL terdiri dari tiga bagian tes yang menjadi dasar, yaitu *Listening Comprehension*, *Structure and Writtem Expression*, dan *Reading Comprehension*.

Berikut ini adalah cara untuk menghitung skor TOEFL^[6]:

1. Jumlahkan skor hasil konversi
Jumlah = skor *listening* + skor *structure and written expression* + skor *reading*
2. Bagi hasil penjumlahan diatas dengan 3

Jumlah = Jumlah : 3

3. Kalikan hasil pembagian dengan 10
Skor TOEFL = hasil pembagian * 10

2.2 Aplikasi Berbasis Web

Salah satu keuntungan dari aplikasi berbasis web adalah kemudahan dalam mengakses data yang tidak terbatas tempat dan waktu, sedangkan kerugiannya adalah waktu yang diperlukan untuk menunggu saat sistem sedang memproses permintaan pengguna, sehingga waktu tunggu terhadap proses pada aplikasi berbasis web diperlukan waktu tunggu yang lebih lama. Hal ini berbeda dengan aplikasi berbasis desktop dimana aplikasi tersebut harus melakukan proses instalasi aplikasi terlebih dahulu di komputer yang akan digunakan untuk mengakses data, namun memiliki keunggulan dalam proses pelaksanaan instruksi, tanpa memerlukan waktu tunggu ke *server*^[9].

2.3 PHP

PHP adalah singkatan dari "PHP: *Hypertext Preprocessor*", yaitu bahasa pemrograman universal yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan dapat digunakan bersamaan dengan HTML.

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca *resume online*-nya. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi pengunjung, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan *script* buatan Rasmus Lerdorf, hingga akhirnya Rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP)^[13].

2.4 MySQL

MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executetable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan dapat diperoleh dengan cara *men-download* (mengunduh) di Internet secara gratis^[15].

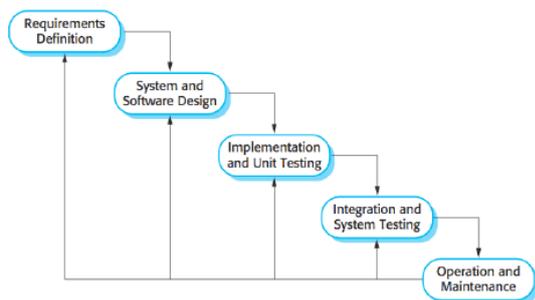
Basisdata adalah suatu *tools* yang digunakan untuk menyimpan informasi, mengambil informasi kapanpun dibutuhkan, dan mengatur informasi yang tersimpan. Jika digambarkan Lemari *File* merupakan suatu basisdata^[17]. Basisdata merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi. Artinya, basisdata merupakan tempat penyimpanan data yang besar, dimana dapat digunakan oleh banyak pengguna.

2.5 CSS

CSS atau *Cascading Style Sheet* adalah bahasa pemrograman yang digunakan untuk mendeskripsikan bagaimana suatu konten akan ditampilkan. Kita memberikan karakteristik tampilan dari elemen yang ada pada HTML menggunakan CSS^[18].

2.6 Metode Waterfall

Model pengembangan ini dimulai dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak dapat kembali atau mengulang ke tahap sebelumnya^[20]. Langkah-langkah pemodelan *Waterfall* dapat dilihat pada Gambar 1:



Gambar 1 metode *Waterfall*^[21]

Tahapan pada pemodelan perangkat lunak menggunakan model *waterfall* terdiri atas definisi kebutuhan, perancangan sistem dan perangkat lunak, Pengujian unit dan implementasi, pengujian sistem dan integrasi, dan penerapan program dan pemeliharaan^[21].

2.7 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial^[22]. Data yang telah terkumpul melalui angket, kemudian diolah ke dalam bentuk kuantitatif, yaitu dengan cara menetapkan skor jawaban dari pertanyaan yang telah dijawab responden.

Berdasarkan data hasil kuesioner, persentase masing-masing jawaban dapat dihitung dengan rumus^[22]:

$$Y = \frac{X}{i} \times 100\%$$

Keterangan :

Y = Nilai presentase

i = Skor ideal (skor tertinggi Likert dikali dengan jumlah responden)

X = Total skor

Berikut ini kriteria interpretasi skor untuk skala Likert:

- $0\% \leq Y \leq 20\%$ = Sangat Lemah
- $21\% \leq Y \leq 40\%$ = Lemah
- $46\% \leq Y \leq 60\%$ = Cukup
- $61\% \leq Y \leq 80\%$ = Kuat
- $81\% \leq Y \leq 100\%$ = Sangat Kuat

2.8 Pengujian Reliability

Pengujian *reliability* dapat dilakukan dengan mengukur frekuensi dan tingkat kegagalan, keakuratan hasil output, *mean-time-to-failed* (MTTF), kemampuan untuk pulih dari kegagalan dan prediktabilitas program^[23]. Analisis kualitas aspek *reliability* dilakukan dengan menggunakan rumus dari Nelson yaitu^[24]:

$$R = 1 - \frac{f}{n} = 1 - r$$

Dimana:

R = *reliability*

f = *total failure*

n = *total test case*

r = *error rate*

III. PERANCANGAN SISTEM

3.1 Analisis Kebutuhan Sistem

Untuk memahami sifat dan fungsi perangkat lunak yang akan dibangun, analisa harus memahami informasi dan juga tingkah laku yang harus dimiliki sistem. Pada analisa kebutuhan ini dilakukan pencarian terhadap kebutuhan-kebutuhan yang dibutuhkan dan harus dimiliki oleh sistem sehingga nanti pengerjaan aplikasi dapat memperoleh hasil yang diinginkan.

Identifikasi masukan merupakan tahapan untuk mengumpulkan informasi yang dibutuhkan oleh sistem aplikasi yang nantinya akan di proses oleh sistem. Data masukan ini diperoleh dari admin dan juga dari sisi pengguna. Dari sisi admin, data masukan adalah berupa pertanyaan dan jawaban yang nantinya akan diajukan sistem kepada pengguna, audio dan teks *reading*. Dari sisi pengguna, data masukan berupa identitas diri yang akan di inputkan pada saat pengguna ingin membuat akun agar pengguna bisa masuk ke dalam sistem. Selain itu, masukan dari sisi pengguna juga berupa jawaban dari pertanyaan-pertanyaan yang diajukan oleh sistem dan juga berupa pesan yang ingin disampaikan pengguna kepada admin.

Identikasi keluaran merupakan tahapan untuk menentukan tahapan hasil dalam bentuk skor TOEFL yang diperoleh dari jawaban-jawaban yang dimasukan oleh pengguna ke dalam sistem. Selain itu, keluaran dari sistem ini juga berupa pertanyaan-pertanyaan, keterangan soal, dan pilihan jawaban yang sudah diinputkan admin ke dalam sistem. Kemudian pengguna akan memberi respon dengan cara memasukkan jawaban yang dianggap paling benar terhadap soal-soal yang ditampilkan oleh sistem. Jawaban-jawaban kemudian akan diperiksa oleh sistem apakah benar atau salah berdasarkan jawaban yang benar yang telah diinputkan oleh admin. Jumlah benar dari jawaban benar yang dimasukkan oleh pengguna akan mempengaruhi jumlah skor TOEFL yang akan didapatkan pengguna. Semakin banyak jumlah jawaban yang benar, maka semakin tinggi pula skor TOEFL yang akan didapatkan oleh pengguna tersebut.

Berikut ini merupakan aktor yang berperan di dalam aplikasi TOEFL dan berikut deskripsi aktivitas yang dapat dilakukan oleh masing-masing aktor seperti ditunjukkan pada Tabel 1.

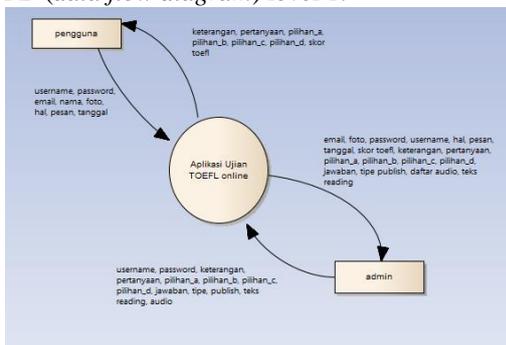
Tabel 1 Aktor yang terlibat

| No | Aktor | Deskripsi |
|----|------------------------|--|
| 1. | Pengguna (user) | Pengguna dapat mendaftarkan diri untuk membuat akun, dapat melakukan <i>log in</i> untuk masuk ke dalam sistem, dapat mengubah <i>password</i> , menjawab soal-soal yang disediakan sistem, melihat skor TOEFL, dan memberikan pesan kepada admin. |
| 2. | Admin | Admin memiliki hak akses tertinggi untuk mengelola soal-soal yang akan dibutuhkan sistem, mengelola akun pengguna, melihat daftar <i>user</i> dan skor TOEFL, dan mengelola kotak pesan pengguna. |

Langkah selanjutnya setelah menentukan kebutuhan sistem adalah membuat desain atau gambaran umum dari sistem dengan menggunakan metode terstruktur yaitu *context diagram*, *data flow diagram* serta penggambaran proses melalui diagram alir.

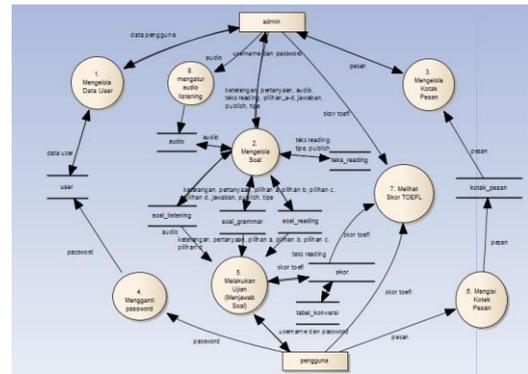
3.2 Proses Desain Sistem

Tahapan selanjutnya adalah proses desain sistem sebagai gambaran dasar bentuk sistem. Dalam melakukan desain sistem terdapat beberapa hal yang harus dibuat yaitu desain proses kerja, desain basis data dan desain tampilan sistem. Pemodelan desain proses kerja menggunakan metode terstruktur, yaitu *context diagram* dan DFD (*data flow diagram*) level 1.



Gambar 2. Diagram konteks aplikasi ujian TOEFL

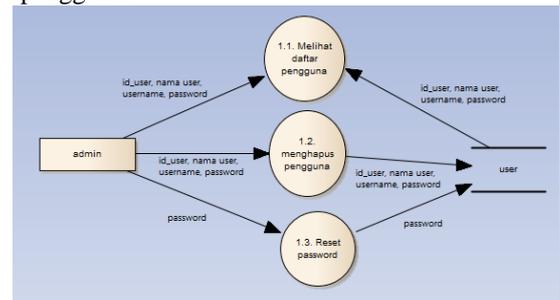
Gambar 2 merupakan gambaran umum tentang sistem meliputi entitas luar (pengguna, dan admin), proses (aplikasi ujian TOEFL), serta data apa saja yang mengalir pada sistem. Selain itu, pada sistem ini nantinya juga akan terdapat beberapa penyimpanan data (*user*, *audio*, *kotak_pesan*, *soal_listening*, *soal_grammar*, *soal_reading*, *teks_reading*, *tabel_nilai*, dan *tabel_konversi*). Proses yang lebih rinci dari diagram konteks tersebut dapat dijabarkan lagi dalam bentuk *Data Flow Diagram* (DFD) level 0.



Gambar 3 Data Flow Diagram Level 0

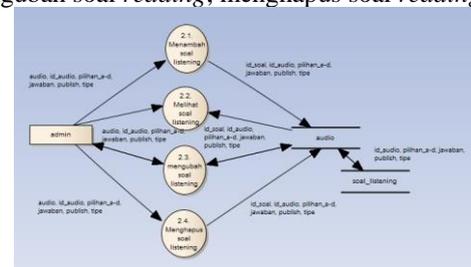
Gambar 3 menunjukkan DFD Level 0 yang menggambarkan proses apa saja yang terdapat dalam sistem pakar. Dalam gambar tersebut, terdapat delapan proses utama yang menghubungkan entitas maupun proses ke penyimpanan data. Delapan proses tersebut diantaranya adalah:

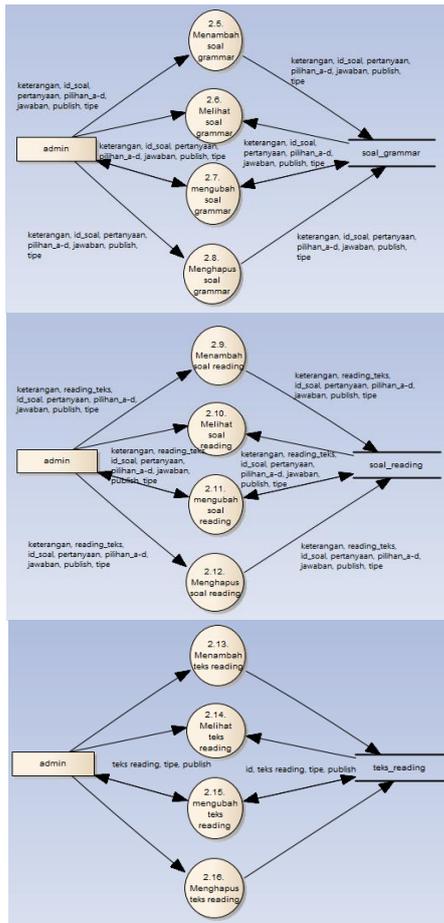
1. Proses 1 yaitu proses mengelola data pengguna. Pada proses DFD level 1.1 ini, proses mengelola data pengguna ini dapat dijelaskan lagi menjadi proses melihat data pengguna, dan menghapus data pengguna.



Gambar 4. DFD Level 1 Proses 1 (mengelola data *user*)

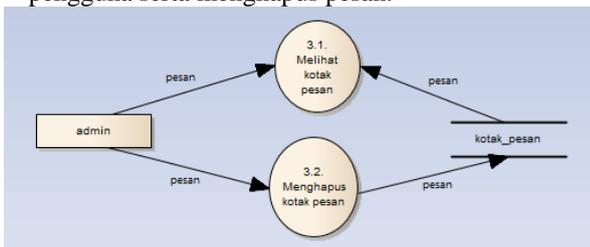
2. Proses 2 yaitu mengelola soal yang dibutuhkan untuk aplikasi ujian TOEFL, beberapa proses yang dapat dilakukan dalam mengelola soal diantaranya adalah menambah soal *listening*, melihat soal *listening*, mengubah soal *listening*, menghapus soal *listening*, menambah soal *grammar*, melihat soal *grammar*, mengubah soal *grammar*, menghapus soal *grammar*, menambah soal *reading*, melihat soal *reading*, mengubah soal *reading*, menghapus soal *reading*.





Gambar 5. DFD Level 1 Proses 2 (Mengelola Soal)

3. Proses 3 yaitu mengelola kotak pesan, beberapa proses yang dapat dilakukan dalam mengelola kotak pesan adalah melihat kotak pesan yang diisi oleh pengguna serta menghapus pesan.



Gambar 6. DFD Level 1 Proses 3 (Mengelola kotak pesan)

Setelah melakukan perancangan alur kerja sistem, tahapan selanjutnya adalah melakukan perancangan basisdata. Dalam melakukan perancangan basisdata, beberapa tahapan yang dilakukan adalah menentukan entitas apa saja yang akan terlibat. Entitas yang terlibat dan dibuat sendiri dalam penelitian ini sebanyak sembilan buah, diantaranya: *user*, *soal_listening*, *soal_grammar*, *soal_reading*, *tabel_nilai*, *kotak_pesan*, *audio*, *teks_reading*, *tabel_konversi*. Setelah menentukan entitas apa saja yang terlibat, tahapan selanjutnya adalah menentukan atribut dari tiap entitas tersebut. Tabel 1 hingga 9 adalah perincian dari atribut pada tiap-tiap entitas. Basisdata yang digunakan pada sistem ini seperti ditunjukkan pada Tabel 2 – Tabel 10.

Tabel 2 Tabel *user*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|--------------------|-------------|-------------|--------------------------------|
| 1. | <i>id_user</i> | INT(4) | Unique key | Kolom untuk ID <i>user</i> |
| 2. | <i>username</i> | VARCHAR(50) | Primary Key | Kolom untuk <i>username</i> |
| 3. | <i>password</i> | VARCHAR(32) | - | Kolom untuk <i>password</i> |
| 4. | <i>email</i> | VARCHAR(50) | - | Kolom untuk email <i>user</i> |
| 5. | <i>gambar_user</i> | VARCHAR(50) | - | Kolom untuk Gambar <i>user</i> |
| 6. | <i>level</i> | VARCHAR(5) | - | Kolom untuk level |

Tabel 3 Tabel *soal_listening*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|-----|-------------------|-------------------|-------------|-----------------------------|
| 1. | <i>id soal</i> | INT(4) | Primary Key | Kolom untuk ID soal |
| 2. | <i>id_audio</i> | VARCHAR(100) | - | Kolom untuk ID <i>Audio</i> |
| 3. | <i>keterangan</i> | VARCHAR(10000) | - | Kolom untuk keterangan |
| 4. | <i>pilihan a</i> | VARCHAR(500) | - | Kolom untuk pilihan A |
| 5. | <i>pilihan b</i> | VARCHAR(500) | - | Kolom untuk pilihan B |
| 6. | <i>pilihan c</i> | VARCHAR(500) | - | Kolom untuk pilihan C |
| 7. | <i>pilihan d</i> | VARCHAR(500) | - | Kolom untuk pilihan D |
| 8. | <i>Jawaban</i> | VARCHAR(100) | - | Kolom untuk jawaban |
| 9. | <i>publish</i> | ENUM('yes', 'no') | - | Kolom untuk pilihan publish |
| 10. | <i>tipe</i> | INT(2) | - | Kolom untuk tipe soal |

Tabel 4 Tabel *soal_grammar*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|-------------------|-------------------|-------------|-----------------------------|
| 1. | <i>id soal</i> | INT(4) | Primary Key | Kolom untuk ID soal |
| 2. | <i>pertanyaan</i> | VARCHAR(1000) | - | Kolom untuk pertanyaan |
| 3. | <i>pilihan a</i> | VARCHAR(500) | - | Kolom untuk pilihan A |
| 4. | <i>pilihan b</i> | VARCHAR(500) | - | Kolom untuk pilihan B |
| 5. | <i>pilihan c</i> | VARCHAR(500) | - | Kolom untuk pilihan C |
| 6. | <i>pilihan d</i> | VARCHAR(500) | - | Kolom untuk pilihan D |
| 7. | <i>Jawaban</i> | VARCHAR(100) | - | Kolom untuk jawaban |
| 8. | <i>publish</i> | ENUM('yes', 'no') | - | Kolom untuk pilihan publish |
| 9. | <i>tipe</i> | INT(2) | - | Kolom untuk tipe soal |

Tabel 5 Tabel *soal_reading*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|-----|-------------------|-------------------|-------------|-------------------------------|
| 1. | <i>id soal</i> | INT(4) | Primary Key | Kolom untuk ID soal |
| 2. | <i>id_reading</i> | VARCHAR(6) | - | Kolom untuk ID <i>reading</i> |
| 3. | <i>pertanyaan</i> | VARCHAR(1000) | - | Kolom untuk pertanyaan |
| 4. | <i>pilihan a</i> | VARCHAR(500) | - | Kolom untuk pilihan A |
| 5. | <i>pilihan b</i> | VARCHAR(500) | - | Kolom untuk pilihan B |
| 6. | <i>pilihan c</i> | VARCHAR(500) | - | Kolom untuk pilihan C |
| 7. | <i>pilihan d</i> | VARCHAR(500) | - | Kolom untuk pilihan D |
| 8. | <i>Jawaban</i> | VARCHAR(100) | - | Kolom untuk jawaban |
| 9. | <i>publish</i> | ENUM('yes', 'no') | - | Kolom untuk pilihan publish |
| 10. | <i>tipe</i> | INT(2) | - | Kolom untuk tipe soal |

Tabel 6 Tabel *tabel_nilai*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|------------------------|-----------|-------------|------------------------------------|
| 1. | <i>id nilai</i> | INT(4) | Primary Key | Kolom untuk ID nilai |
| 2. | <i>id_user</i> | INT(4) | - | Kolom untuk ID <i>user</i> |
| 3. | <i>nilai_listening</i> | INT(4) | - | Kolom untuk nilai <i>listening</i> |
| 4. | <i>nilai_grammar</i> | INT(4) | - | Kolom untuk nilai <i>grammar</i> |
| 5. | <i>nilai_reading</i> | INT(4) | - | Kolom untuk nilai <i>reading</i> |
| 6. | <i>nilai_akhir</i> | INT(4) | - | Kolom untuk nilai akhir |
| 7. | <i>tanggal</i> | DATE | - | Kolom untuk tanggal |

Tabel 7 Tabel *kotak_pesan*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|-----------------|----------------|-------------|--------------------------------|
| 1. | <i>id pesan</i> | INT(4) | Primary Key | Kolom untuk ID pesan |
| 2. | <i>hal</i> | VARCHAR(1000) | - | Kolom untuk hal |
| 3. | <i>nama</i> | VARCHAR(50) | - | Kolom untuk nama pemberi pesan |
| 4. | <i>email</i> | VARCHAR(50) | - | Kolom untuk email |
| 5. | <i>pesan</i> | VARCHAR(10000) | - | Kolom untuk pesan |
| 6. | <i>tanggal</i> | DATE | - | Kolom untuk tanggal |

Tabel 8 Tabel *audio*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|-------------------|--------------|-------------|---------------------------------|
| 1. | <i>id_audio</i> | INT(4) | Primary Key | Kolom untuk ID <i>audio</i> |
| 2. | <i>nama_audio</i> | VARCHAR(200) | - | Kolom untuk nama <i>audio</i> |
| 3. | <i>status</i> | VARCHAR(10) | - | Kolom untuk status <i>audio</i> |

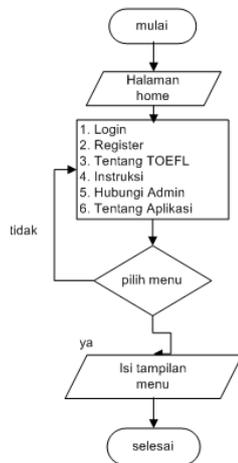
Tabel 9 Tabel *teks_reading*

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|---------------------|-------------------|-------------|---------------------------------|
| 1. | id <i>reading</i> | INT(4) | Primary Key | Kolom untuk ID <i>reading</i> |
| 2. | <i>reading</i> teks | VARCHAR(10000) | - | Kolom untuk teks <i>reading</i> |
| 3. | publish | ENUM('yes', 'no') | - | Kolom untuk publish |
| 4. | tipe | INT(2) | - | Kolom untuk tanggal |

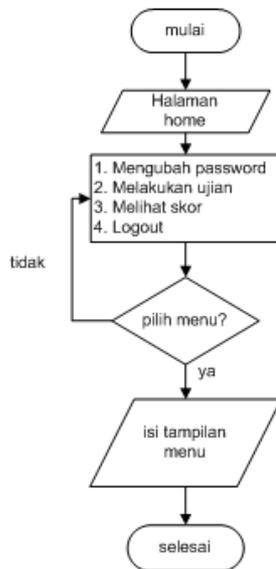
Tabel 10 Tabel tabel_konversi

| No | Kolom | Tipe Data | Keterangan | Deskripsi |
|----|---------------------------|-----------|-------------|---------------------------------------|
| 1. | id | INT(4) | Primary Key | Kolom untuk ID |
| 2. | jumlah benar | INT(4) | - | Kolom untuk jumlah benar |
| 3. | konversi <i>listening</i> | INT(4) | - | Kolom untuk konversi <i>listening</i> |
| 4. | konversi <i>grammar</i> | INT(4) | - | Kolom untuk konversi <i>grammar</i> |
| 5. | konversi <i>reading</i> | INT(4) | - | Kolom untuk konversi <i>reading</i> |

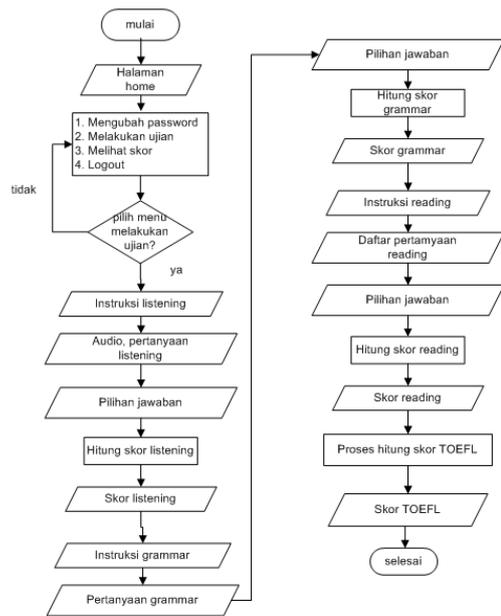
Tahapan selanjutnya adalah melakukan perancangan diagram alir yang menggambarkan aliran proses yang terdapat didalam sistem. Gambaran tersebut digambarkan dalam diagram alir dari aplikasi TOEFL yang ditunjukkan pada Gambar 7 – Gambar 10 berikut.



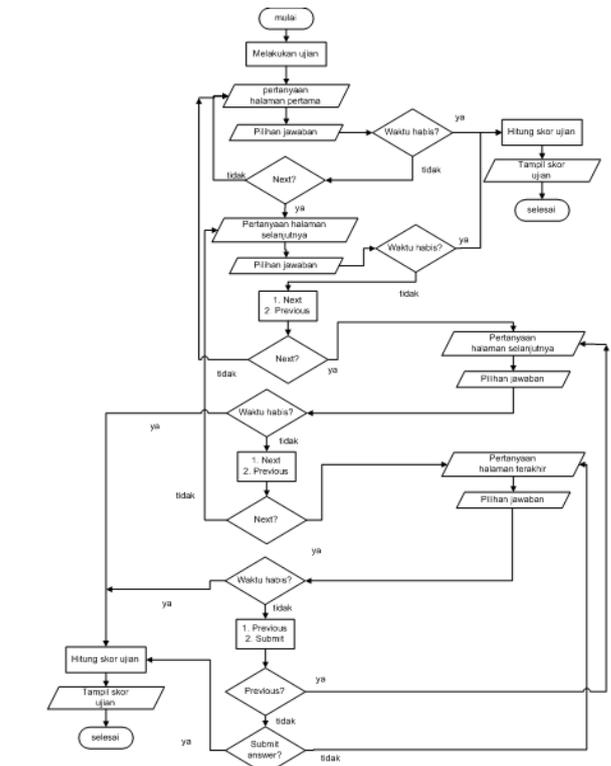
Gambar 7. Diagram Alir beranda sistem



Gambar 8 Diagram Alir halaman user

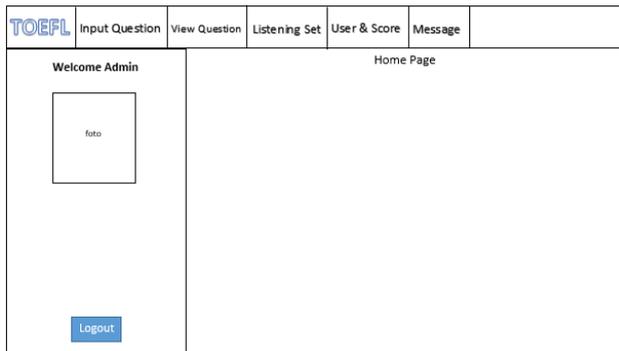


Gambar 9 Diagram Alir melakukan ujian

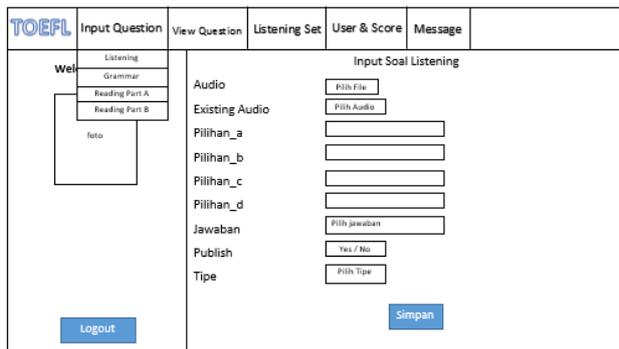


Gambar 10 Diagram Alir melakukan ujian grammar

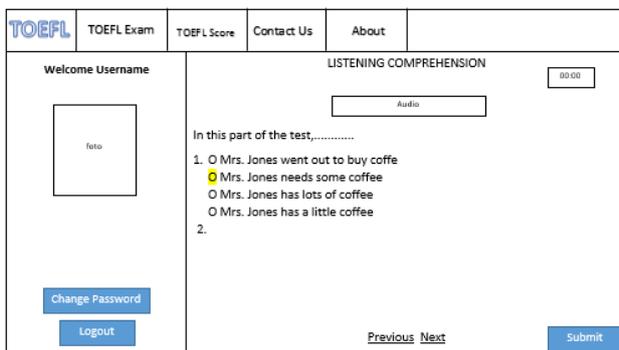
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga dapat memberikan gambaran untuk calon pengguna aplikasi. Selain itu, dengan melakukan perancangan antarmuka diharapkan dapat mempermudah dalam melakukan implementasi serta memudahkan pengembang sistem dalam membangun aplikasi. Beberapa contoh tampilan antarmuka perancangan aplikasi TOEFL dapat ditunjukkan pada Gambar 11 – Gambar 14.



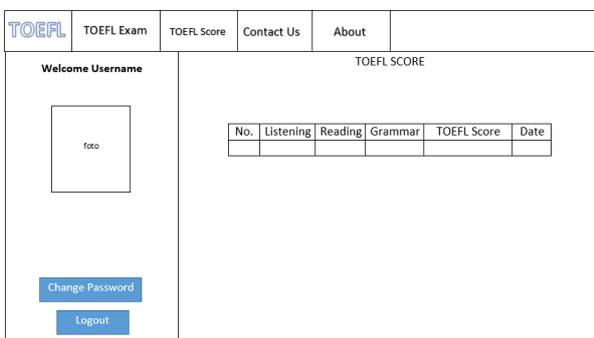
Gambar 11 Perancangan halaman admin



Gambar 12 Perancangan menambah pertanyaan



Gambar 13 Perancangan antarmuka user melakukan ujian



Gambar 14 Perancangan antarmuka skor user

aplikasi. Pengguna yang dimaksud adalah mahasiswa yang berada disekitar peneliti.

1. Perangkat Keras

Perangkat keras yang digunakan dalam membangun sistem ini adalah PC Desktop dengan spesifikasi adalah sebagai berikut:

- Prosesor : Intel® Core(TM) i3-370M CPU@2.10GHZ
- Sistem Operasi : Windows 8 64-bit
- RAM : 2.00 GB
- Mouse Standar
- Keyboard Standar

2. Perangkat Lunak

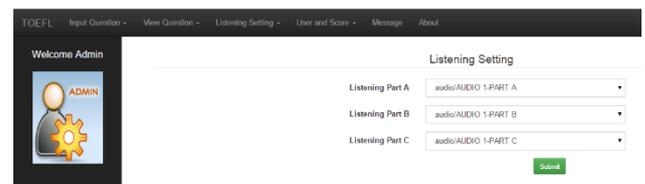
Perangkat lunak yang digunakan dalam mengimplementasikan sistem adalah web browser berbasis desktop, baik chrome, mozilla, opera, dan lainnya.

3. Implementasi Antarmuka Sistem

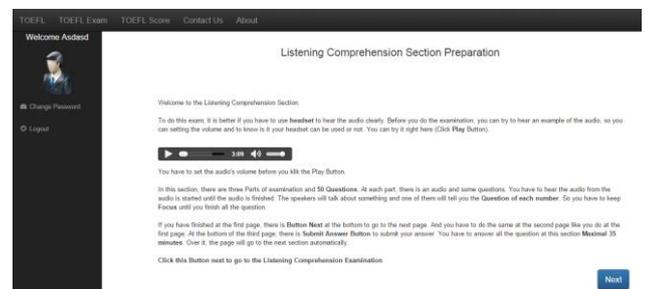
Tampilan aplikasi ketika dijalankan akan ditunjukkan pada Gambar 15 – Gambar 20.



Gambar 15 Tampilan melihat pertanyaan



Gambar 16 tampilan mengatur audio



Gambar 17 tampilan sebelum ujian listening

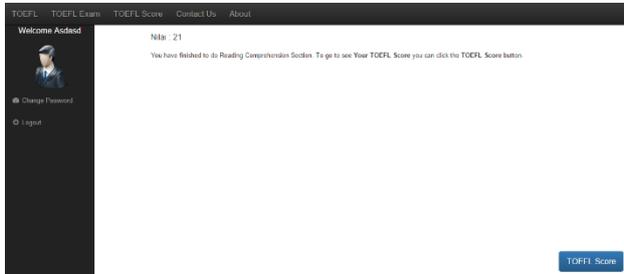
IV. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

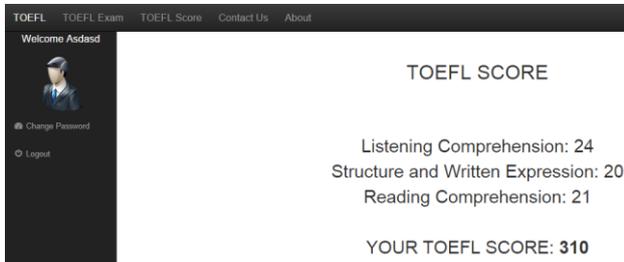
Langkah pertama dalam proses implementasi, adalah mencari pengguna yang bersedia untuk menggunakan



Gambar 18 tampilan melakukan ujian *listening*



Gambar 19 tampilan setelah mealkukan ujian setiap sesi



Gambar 20 tampilan skor toefl

4.2 Pengujian Aplikasi

Pengujian pada tahap ini bermaksud untuk menguji aplikasi serta memastikan aplikasi dapat berjalan seperti tujuan dan seluruh fungsi dapat digunakan. Pengujian yang dilakukan menggunakan metode *black-box*. Pengujian *black-box* merupakan pengujian yang lebih berfokus pada sisi fungsionalitas aplikasi dan tingkat keberhasilan pengujian diukur dari terpenuhinya tujuan dari fungsi yang sudah dibuat.

1. Pengujian Fungsionalitas

Pengujian aplikasi ini dibuat berupa tabel pengujian *black-box* dari fungsi yang ada dalam aplikasi. Tabel 11 menunjukkan hasil pengujian fungsi-fungsi yang terdapat pada aplikasi.

Tabel 11. Pengujian fungsi login dan register

| Nama Pengujian | Bentuk Pengujian | Hasil Yang Diharapkan | Hasil Pengujian |
|---------------------------------|--|--|-----------------|
| Pengujian pilihan menu | Mengklik tombol Menu | Menampilkan pilihan melakukan <i>login</i> sebagai pengguna atau admin | Berhasil |
| Pengujian <i>login</i> pengguna | Mengklik tombol Login | Menampilkan tampilan pilihan form <i>login</i> | Berhasil |
| Pengujian pendaftaran pengguna | Mengklik pilihan <i>register</i> | Menampilkan <i>field</i> melakukan pendaftaran pengguna | Berhasil |
| Pengujian <i>field</i> | Mengisi semua <i>field</i> pendaftaran | Menampilkan tampilan untuk | Berhasil |

| | | | |
|--|---|--|----------|
| pendaftaran berhasil | pengguna dengan benar | melakukan <i>login</i> | |
| Pengujian <i>field</i> pendaftaran gagal | Mengisi <i>field</i> pendaftaran namun masih ada <i>field</i> yang kosong | Menampilkan pemberitahuan pendaftaran gagal | Berhasil |
| Pengujian <i>login</i> sebagai user | Mengklik pilihan <i>login</i> | Menampilkan <i>field login</i> pengguna | Berhasil |
| Pengujian <i>field login</i> | Tidak mengisi <i>field login</i> | Menampilkan pemberitahuan <i>field</i> harus diisi | Berhasil |
| Pengujian <i>login</i> sukses | Mengisi <i>field username</i> dan <i>password</i> yang sudah terdaftar | Masuk ke halaman pengguna dengan level user | Berhasil |
| Pengujian <i>login</i> gagal | Mengisi <i>field username</i> dan <i>password</i> yang tidak terdaftar | Menampilkan pemberitahuan tidak terdaftar | Berhasil |
| Pengujian <i>Logout</i> | Mengklik tombol Logout | Menampilkan halaman publik aplikasi | Berhasil |
| Pengujian <i>login</i> sebagai admin | Mengklik pilihan <i>login</i> | Menampilkan <i>field login</i> pengguna | Berhasil |
| Pengujian <i>field login</i> | Tidak mengisi <i>field login</i> | Menampilkan pemberitahuan <i>field</i> harus diisi | Berhasil |
| Pengujian <i>login</i> sukses | Mengisi <i>field username</i> dan <i>password</i> yang sudah terdaftar | Masuk ke halaman pengguna dengan level sebagai admin | Berhasil |
| Pengujian <i>login</i> sebagai admin gagal | Mengisi <i>field username</i> dan <i>password</i> yang tidak terdaftar | Menampilkan pemberitahuan tidak terdaftar | Berhasil |
| Pengujian <i>Logout</i> | Mengklik tombol Logout | Menampilkan halaman publik aplikasi | Berhasil |

Pada pengujian menu halaman *home* yang terdiri dari 7 pertanyaan, menu input pertanyaan yang terdiri dari 13 pertanyaan, melihat pertanyaan yang terdiri dari 16 pertanyaan, mengatur *audio listening* yang terdiri dari 2 pertanyaan, menu mengelola akun dan menampilkan skor yang terdiri dari 5 pertanyaan, menu mengelola pesan yang terdiri dari 2 pertanyaan, menu mengubah kata sandi yang terdiri dari 3 pertanyaan, menu melakukan ujian yang terdiri dari 17 pertanyaan, menu melihat skor TOEFL pengguna yang terdiri dari 1 pertanyaan dengan hasil pengujian menunjukkan bahwa semua item pengujian adalah berhasil.

Berdasarkan hasil pengujian fungsionalitas yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan pada suatu saat aplikasi digunakan. Sehingga membutuhkan proses *maintenance* untuk lebih mengetahui kekurangan dari aplikasi.

2. Pengujian Pengguna

Pengujian Beta merupakan pengujian yang dilakukan oleh pengguna akhir perangkat lunak sebagai

responden dari kuesioner yang berisi pertanyaan mengenai *usability* dari aplikasi. Jawaban hasil kuesioner kemudian dihitung dengan rumus Skala Likert^[22] berikut ini untuk mendapatkan kesimpulan.

$$Y = \frac{X}{i} \times 100\%$$

Keterangan :

Y = Nilai presentase

i = Skor ideal (skor tertinggi Likert dikali dengan jumlah responden)

X = Total skor

Tabel 12 menunjukkan hasil perolehan data dari pertanyaan-pertanyaan kuesioner yang diajukan ke 20 responden.

Tabel 12 Hasil persentase beta

| No. | Pertanyaan | Persentase |
|-----|--|------------|
| 1. | Apakah aplikasi mudah dipelajari? | 83% |
| 2. | Apakah aplikas mudah digunakan? | 87% |
| 3. | Apakah aplikasi sesuai dengan TOEFL PBT? | 84% |
| 4. | Apakah aplikas cukup membantu? | 82% |
| 5. | Cukup menarik? | 85% |
| 6. | Apakah mudah mengakses ke menu aplikasi? | 87% |
| 7. | Apakah kebutuhan pengguna telah terpenuhi? | 86% |
| 8. | Apakah menu dan tampilan mudah diingat? | 82% |
| 9. | Apakah setiap menu berjalan baik? | 89% |

Tabel 14 menunjukkan nilai-nilai hasil pengujian *usability* terhadap masing-masing pertanyaan memiliki persentase terendah 82%, dimana bila hasil persentase berada diatas 80% berarti berada pada daerah kuat – sangat kuat. Hal ini dapat diartikan bahwa aplikasi TOEFL yang diuji sudah memberikan nilai yang baik terhadap penerimaan dari sisi pengguna dalam 5 aspek *usability*.

3. Pengujian *Reliability*

Berdasarkan uji skenario *stress testing website* dengan *tool* WAPT 8.5 *trial version*, diketahui hasil pengujian seperti ditunjukkan pada Tabel 13.

Tabel 13 Hasil pengujian *reliability*

| No | Kategori | Sukses | Gagal |
|-------|-----------------|--------|-------|
| 1 | <i>Sessions</i> | 155 | 6 |
| 2 | <i>Pages</i> | 126 | 0 |
| 3 | <i>Hits</i> | 411 | 3 |
| Total | | 692 | 9 |

Berdasarkan hasil pengujian *reliabilitas* menggunakan WAPT 8.5 *trial version* seperti ditunjukkan pada Tabel 15 maka dapat dihitung nilai *error rate* dan *Reliability* sebagai berikut.

$$r = \frac{f}{n} = \frac{9}{692} = 0.013$$

Keterangan:

f = total failure yaitu 9

n = total test case yaitu 692

r = error rate yaitu 0.013

$$R = 1 - r = 1 - 0.013 = 0.987$$

Keterangan:

r = error rate yaitu 0.013

R = reliability yaitu 0.987

Dari perhitungan diatas dapat diketahui bahwa nilai *error rate* (r) yaitu 0.013 dan nilai *reliability* (R) yaitu 0.987. Maka setelah nilai *reliability* dicocokkan dengan standar *Telcordia*, dapat disimpulkan bahwa perangkat ini dapat diterima dan memenuhi kriteria standar *reability* yang ditetapkan oleh *Telcordia*..

V. PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisis dan pengujian pada aplikasi TOEFL, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi ujian TOEFL berbasis web yang dibuat sudah dapat digunakan untuk menguji kemampuan bahasa Inggris dengan hasil akhir berupa skor TOEFL.
2. Berdasarkan hasil pengujian *black box*, aplikasi sudah berjalan dengan baik dengan semua tombol dapat digunakan dan menampilkan tanggapan sesuai dengan yang seharusnya.
3. Berdasarkan hasil pengujian *beta*, hasil kuesioner dari pengujian aplikasi dapat disimpulkan bahwa perangkat lunak yang dibangun mudah dipelajari, mudah digunakan, fungsi yang terdapat didalam aplikasi sesuai dengan ujian TOEFL PBT biasanya, dapat digunakan untuk menilai kemampuan diri, antarmuka yang digunakan menarik, mudah untuk mengakses ke menu aplikasi, kebutuhan pengguna sudah sesuai dengan menu dan fasilitas yang telah disediakan, menu dan tampilan mudah untuk diingat, dan setiap menu dalam aplikasi ini dapat berjalan dengan baik dimana berdasarkan hasil rata-rata pengujian aplikasi memiliki persentasi \geq dari 80%, yang berarti hasil pengujian dari setiap pertanyaan berada dalam persentase hasil kuat sampai dengan sangat kuat. Hal ini menunjukkan bahwa aplikasi memberikan nilai baik terhadap penerimaan dari sisi pengguna dalam 5 aspek *usability*.
4. Berdasarkan hasil pengujian *reliability*, didapat bahwa nilai *error rate* (r) yaitu 0.013 dan nilai *reliability* (R) yaitu 0.987, dimana berdasarkan standar *Telcordia* dapat disimpulkan bahwa perangkat ini memenuhi kriteria standar *reability*.

5.2. Saran

Saran yang diberikan dalam upaya pengembangan aplikasi yang lebih baik dikemudian hari.

1. Dapat dikembangkan aplikasi Ujian TOEFL sebagai Media Pelatihan Bahasa Inggris Berbasis Web dengan menggunakan bahasa pemrograman selain PHP dan mesin basis data selain MySQL agar dapat dibandingkan kinerjanya.
2. Perlu dilakukan penelitian lanjutan untuk mengembangkan tes kemajuan Bahasa Inggris yang lebih bervariasi, sesuai dengan kebutuhan yang ada di pasar, seperti TOEP, IELTS, TOEIC, dan sebagainya sehingga pengguna dapat menilai kemampuan Bahasa Inggris sesuai dengan kebutuhannya.
3. Aplikasi Ujian TOEFL sebagai Media Pelatihan Bahasa Inggris Berbasis Web ini dapat dikembangkan lebih lanjut dengan membangun aplikasi yang berbeda *platform*.

DAFTAR PUSTAKA

- [1] Parwanto, T., "Perancangan Aplikasi Simulasi TOEFL (Test of English as Foreign Language)," no. 5-10-2012, p. 3, 2011.
- [2] Triyadi, E., "Perancangan Aplikasi Tes TOEFL Mobile Menggunakan Teknologi J2me Pada Ponsel Berbasis Java Sebagai Media Pembelajaran," 2011.
- [3] Pratiwi, A., Mayasari, A. R., dan Rahmasari, G., *100% Best Guide to TOEFL Up to 600*, Jakarta: Laskar Aksara, 2011.
- [4] Hinkel, E., *TOEFL Test Strategies*, Jakarta: Binarupa Aksara, 2005.
- [5] Purmaning, E. R., Ayuningtyas, A. K., Hudha, Nurul, *Tip & Trik Melejitkan Skor TOEFL*, Jakarta: Cmedia, 2012.
- [6] Fauzy, E. K., Yudiashari F., dan S, Lidya Cristina, *Break Down! Soal-Soal TOEFL*, Bandung: RuangKata, 2013.
- [7] Supriyanto, A., *Pengantar Teknologi Informasi*, Jakarta: Salemba Infotek, 2005.
- [8] Bakri, Z., dan Hidayaningsih, P. S., "Internet : Pengiriman *file*," vol. VII, p. 24, 1997.
- [9] Noviyanto, F., dan Ashari, A., "Penerapan Teknologi Ria untuk Membangun Aplikasi Web dengan Pengaksesan Real Time (Studi Kasus: Pemesanan Handphone Online)," vol. V, no. 1, p. 486, 2013.
- [10] Noviyanto, F., dan Wahyu, D. D., "Rich Internet Application (RIA) untuk Aplikasi Sistem Informasi Klinik Kesehatan," vol. III, no. 2, pp. 341-349, 2015.
- [11] Dharma, F., "Pengaruh Structural Assurance dan Perceived Reputation Terhadap Trust Pengguna Internet di Sistem E-commerce," p. 6, 2006.
- [12] F, M. M., *PHP Tutorial Book for Beginner*, Yogyakarta: Notebook, 2014.
- [13] L. Dwiartara, "Menyelam dan Menaklukkan Samudra PHP," *Ilmuwebsite. com*, p. 3, 2012. [15] Dananjaya, J., *Folklor Indonesia Ilmu, Gosip, Dongeng, dan lain lain*. Jakarta: Grafitipers, 1984.
- [14] Raharjo, B., Heryanto, I., dan K., E. R., *Modul Pemrograman WEB (HTML, PHP, & MySQL)*, Bandung: Modula, 2012.
- [15] Kadir, A., *Tuntutan Praktis: Belajar Basisdata Menggunakan MySQL*, Yogyakarta: Andi, 2008.
- [16] Kadir, A., *Dasar Pemrograman Web Dinamis Menggunakan PHP*, Yogyakarta: Andi, 2002.
- [17] Solichin, A., "MySQL 5: Dari Pemula Hingga Mahir," p. 6, 2010.
- [18] Elisabeth, R., dan Eric, F., "Head First HTML and CSS 2nd Edition," p. 28, 2012.
- [19] Ariona, R., "Belajar HTML dan CSS - Tutorial Fundamental dalam Mempelajari HTML dan CSS," p. 58, 2013.
- [20] academia, "academia," [Online]. Available: https://www.academia.edu/7585995/BAB_I_Model_Waterfall. [Accessed 21 April 2015].
- [21] Sommerville, I., *Software Engineering, Eighth Edition*. Addison-Wesley, 2007.
- [22] Sugiyono, *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung, Indonesia: Alfabeta, 2009.
- [23] Pressman, Roger S., *Software Engineering: A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill Companies, Inc, 2010.
- [24] Tian, J., "Evaluating Web Software Reliability Based on Workload and Failure Data Extracted From Server Logs," p.754, 2004.
- [25] Asthana, A., dan Olivieri, J., "Quantifying Software Reliability and Readiness," 2009.
- [26] Sudarmono, E., *THE KING TOEFL Tips & Trik plus Bank Soal*. Jakarta Selatan: Wahyumedia, 2013.