

RANCANG BANGUN APLIKASI TES *TOEFL* MENGUNAKAN ALGORITMA *QUICK SORT* BERBASIS KOMPUTER

Nuri David Maria Veronika¹, Yulia Darnita²

^{1,2}Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu
Jl. Bali PO BOX 118. Telp (0736) 227665, Fax (0736) 26161, Bengkulu 38119

¹ misz_nuricuby@gmail.com

²yuliadarnita@gmail.com

Abstrak: Bahasa Inggris merupakan salah satu bahasa yang paling penting, karena memiliki peran sebagai bahasa internasional. Untuk mengetahui tingkat pemahaman seseorang terhadap bahasa Inggris diperlukan sebuah tes uji pemahaman bahasa Inggris, salah satunya dengan mengikuti tes *TOEFL*. Hasil tes ini dapat digunakan ketika kita ingin melanjutkan studi, melamar pekerjaan ataupun berpergian keluar negeri. Dalam penelitian ini penulis menerapkan sebuah algoritma *Quick Sort* yang fungsinya untuk mengetahui perankingan terhadap seluruh peserta tes berdasarkan nilai yang mereka peroleh. Dalam proses pembuatan aplikasi ini peneliti menggunakan Bahasa Pemrograman *VB.Net* dan database *Microsoft Access*. Penelitian ini dilakukan dengan mencari data soal tes *TOEFL* pada beberapa *website* yang menyediakan data soal tersebut serta data dari perpustakaan Universitas Muhammadiyah Bengkulu. Sedangkan metode pengembangan sistem yang digunakan adalah Prototipe yang terdiri dari 4 fase yaitu analisa kebutuhan sistem, desain sistem, pengujian sistem, implementasi.

Kata Kunci : Tes *Toefl*, *Quick Sort*, *Prototipe*, Skor.

Abstract: *English is one of the most important language, because it has a role as an international language. to determine a person's level of understanding of the English language test required an understanding of the English language test, one of them by following the Toefl Test. Results of this test can be used when we want to continue their studies, applying for a job or is traveling out of the country. In this study the authors apply a Quick Sort algorithm whose function is to determine perankingan to all participants test based on the value they receive. Based on these problems, the author intends to lift the title "Design Build Toefl Test Applications Using Computer-Based Quick Sort algorithm. In the process of making this application researchers using VB.Net Programming Language and Microsoft Access databases. This research was conducted by searching the data Problem Toefl test on several websites that provide data about these as well as data from Bengkulu while Muhammadiyah University library system development methods used by the author is a prototype that consists of four phases, namely the analysis of system requirements, system design, system testing, implementation.*

Keywords: *Toefl Test, Quick Sort, Prototype, Score.*

I. PENDAHULUAN

Bahasa Inggris merupakan salah satu bahasa yang paling penting, karena memiliki peran sebagai bahasa internasional. Salah satu cara untuk mengembangkan pengetahuan tentang bahasa Inggris haruslah dengan belajar baik itu di instansi pendidikan, tempat kursus maupun secara otodidak. Namun untuk mengetahui tingkat pemahaman seseorang terhadap bahasa Inggris diperlukan sebuah tes uji pemahaman bahasa Inggris, salah satunya dengan mengikuti *Test Of English as a Foreign Language (TOEFL)*. Hasil tes ini dapat digunakan ketika kita ingin melanjutkan studi, melamar pekerjaan ataupun berpergian keluar negeri. Dalam penelitian ini peneliti memilih algoritma *Quick Sort*, karena

algoritma ini merupakan salah satu algoritma pengurutan modern dan dapat dengan cepat melakukan pengurutan data. Cara kerja Algoritma *Quick Sort* adalah dengan membandingkan suatu elemen (*pivot*) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil daripada *pivot* tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada *pivot* tersebut terletak di sebelah kanan. Sehingga dengan demikian telah terbentuk dua *sublist* kiri dan *sublist* kanan dari *pivot* [1]. Algoritma *Quick Sort* akan digunakan untuk proses perangkingan terhadap nilai peserta tes *TOEFL*.

II. LANDASAN TEORI

Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada [2].

Perangkat lunak aplikasi (bahasa Inggris: *software application*) adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan.

Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu [3]. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi dua yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

TOEFL (Test of English as a Foreign Language) merupakan test yang digunakan untuk mengetahui tingkat kemampuan seseorang dalam berbahasa Inggris[4]. *Test* ini ditujukan bagi orang yang tidak menggunakan bahasa Inggris sebagai bahasa ibu. Umumnya, *test* ini digunakan sebagai salah satu prasyarat bagi seseorang yang ingin melanjutkan studi atau bekerja di suatu negara yang menggunakan Bahasa Inggris dalam komunikasi sehari-hari. Selain *TOEFL*, masih ada beberapa jenis test lain yang hampir sama, yaitu *IELTS*, *TOEIC* dan *ESOL*. *TOEFL* dikembangkan dan dikeluarkan oleh *ETS (Educational Testing Service)* yang berpusat di New Jersey, USA. *Test* ini pertama kali diperkenalkan pada tahun 1963.

Saat ini ada 3 macam *test TOEFL* yang dikeluarkan oleh ETS, yaitu PBT (*Paper Based Test*) *TOEFL*, CBT (*Computer Based Test*) *TOEFL* dan iBT (*Internet Based Test*) *TOEFL*. Dari tiga tipe ini, yang sering digunakan adalah PBT dan iBT.

a. PBT *TOEFL*

PBT merupakan bentuk *TOEFL Test* yang menggunakan lembar jawaban berupa kertas. Ini merupakan sistem test pertama yang dikeluarkan oleh ETS. Dalam PBT, *test* yang dilakukan meliputi *listening*, *structure* dan *reading*. Rentangan skor yang diberikan adalah 310-677. Selain itu, waktu yang diberikan untuk mengerjakan test ini sekitar 2-2,5 jam. Tapi, sejak bulan April 2006, test ini sudah tidak digunakan lagi di Indonesia. Saat ini, *test* ini masih digunakan di negara yang belum memiliki sistem test iBT.

b. CBT *TOEFL*

CBT merupakan test yang tak lagi menggunakan kertas. Semua test dilakukan dalam suatu komputer menggunakan *software* tertentu. Sistem *test* ini muncul pada tahun 1998. Kemampuan yang diujikan meliputi *listening*, *structure*, *reading* dan *writing*. Pada sistem ini, rentangan skornya adalah 0-300 dengan waktu pengerjaan antara 2-2,5 jam.

c. IBT *TOEFL*

Test dengan sistem iBT merupakan *test* yang paling populer dan banyak digunakan. Sistem test ini juga berbasis komputer, hanya saja *test* ini terhubung dengan jaringan internet, dengan demikian, *test* dilakukan secara *online*. Sistem *test* ini mulai dipublikasikan dan digunakan pada tahun 2005, tapi baru digunakan di Indonesia pada tahun 2006. Kemampuan yang diujikan meliputi *reading*, *listening*, *writing* dan *speaking*. Walau sistem ini banyak digunakan, tapi masih ada beberapa negara yang menggunakan sistem PBT.

Pengurutan data (*sorting*) didefinisikan sebagai suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Algoritma pengurutan adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen [1]. Ada dua macam urutan yang biasa digunakan dalam proses pengurutan yaitu

- a. Urut naik (*ascending*) yaitu dari data yang mempunyai nilai paling kecil sampai paling besar
- b. Urut turun (*descending*) yaitu data yang mempunyai nilai paling besar sampai paling kecil.

Terdapat berbagai macam algoritma pengurutan data diantaranya adalah *Selection sort*, *Heap Sort*, *Quick sort* dan lain sebagainya. Algoritma *Selection sort* digunakan untuk membandingkan elemen yang sekarang dengan elemen yang berikutnya sampai dengan elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan kemudian ditukar. Kalau algoritma *Heap Sort* adalah algoritma pengurutan yang memanfaatkan struktur *heap*. Nilai *key* setiap *node* selalu lebih besar dari nilai *key* semua keturunannya. Sedangkan algoritma *Quick Sort* merupakan algoritma yang membandingkan suatu elemen (*pivot*) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil daripada *pivot* tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada *pivot* tersebut terletak di sebelah kanan. Sehingga dengan demikian telah terbentuk dua *sublist* kiri dan *sublist* kanan dari *pivot* [1].

Pada penelitian ini penulis menggunakan algoritma *Quick sort*, Algoritma *Quick sort* ini juga dikenal sebagai *Partition-Exchange Sort* atau disebut sebagai *Sorting* Pergantian Pembagi. Pada kasus terburuknya, algoritma ini membuat perbandingan $O(n^2)$, malapung kejadian seperti ini sangat langka. *Quick sort* sering lebih cepat dalam praktiknya dari pada algoritma $O(n \log n)$ yang lainnya. Dan juga, urutan dan referensi lokalisasi memori *quicksort* bekerja lebih baik dengan menggunakan *cache* CPU, jadi keseluruhan *sorting* dapat dilakukan hanya dengan ruang tambahan $O(\log n)$.

VB .NET adalah bahasa pemrograman untuk membuat aplikasi berbasis *Windows*, aplikasi *form* *Web ASP .NET*, layanan *Web XML* dan aplikasi *mobile* seperti komputer *Palm* dan *Pocket PC*. *VB .NET* dibangun di atas fondasi *.NET Framework* (lingkungan kerja *.NET*).

Microsoft .NET telah didesain dari dasar dengan internet sebagai fokus utamanya (walaupun tidak secara eksklusif hanya untuk pengembang internet saja). Banyak inovasi baru yang berada dalam *platform* ini akan mengatasi keterbatasan dari *tool-tool* dan teknologi lama.

Dengan menggunakan alat ini, para programmer dapat membangun aplikasi *Windows Form*, Aplikasi web berbasis *ASP*, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*.

Bahasa *VisualBasic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas *.NET Framework*. Peluncurannya mengundang kontroversi,

mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

Microsoft Visual Basic.NET memiliki banyak fasilitas baru dan ditingkatkan seperti *inheritance*, *interface*, dan *overloading* yang menjadikannya sebagai bahasa pemrograman berorientasi objek yang tangguh. *Object Oriented Programming* (OOP) merupakan kumpulan objek yang saling berinteraksi satu dengan lainnya. OOP akan mendekomposisikan masalah dunia nyata dan dinamakan *class* ataupun *tipe*.

Microsoft Access merupakan salah satu *software* pengolah database yang berjalan di bawah sistem operasi *Microsoft Windows*. *Microsoft Access* adalah salah satu program yang dapat melakukan manajemen *database* merupakan cara pengolahan data terutama dalam teknologi komputer dan digunakan sebagai pengaturan data [5]. *Database* adalah sekumpulan informasi yang diorganisasikan dan diinformasikan dalam bentuk yang spesifik.

Model perangkat lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung. UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Pada saat ini telah banyak versi UML yang digunakan dalam tahap analisis dan perancangan sistem. Versi yang terbaru saat ini adalah UML versi 2.5, namun versi yang telah diterima oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object oriented* dan *software component* adalah UML versi 2.0. [6]. UML versi 2.0 tersebut terdiri dari empat belas teknik diagram, seperti yang dinyatakan oleh Dennis dkk. [6]:

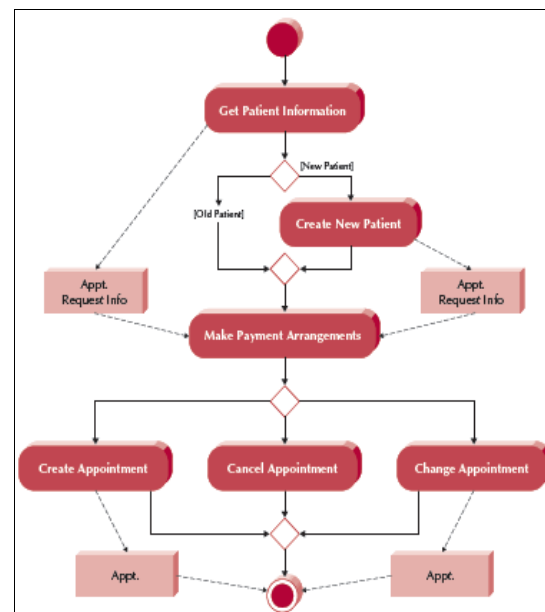
The Version 2.0 of the UML defines a set of fourteen diagramming techniques used to model a system. The diagrams are broken into two major groupings: one for modeling structure of a system and one for modeling behavior. The structure modeling diagrams include class, object, package, deployment, component, and composite structure diagrams. The behavior modeling diagrams include activity, sequence, communication, interaction overview, behavior state machine, protocol state machine, and use case diagrams.

UML versi 2.0 mendefinisikan empat belas diagram UML yang dapat digunakan untuk memodelkan suatu sistem. Diagram UML yang digunakan dibagi menjadi dua kelompok utama: satu untuk pemodelan struktur suatu sistem dan satu lagi untuk pemodelan sifat suatu sistem. *structure diagrams* terdiri dari *class diagram*, *object diagram*, *package diagram*, *deployment diagram*, *component diagram*, dan *composite structure diagram*. Sedangkan *behavior diagrams* terdiri dari *activity diagram*, *sequence diagram*, *communication diagram*, *interaction overview diagram*, *behavior state machine diagram*, *protocol state machine diagram*, dan *use case diagram*. Namun pada Aplikasi ini hanya menggunakan dua diagram yaitu diagram activity dan diagram use case. Menurut Dennis, dkk [6] yang menyatakan bahwa:

Activity diagrams provide the analyst with the ability to model processes in an information system. Activity diagrams can be used to model workflows, individual use cases, or the decision logic contained within an individual method. They also provide an approach to model parallel processes.

Berdasarkan pernyataan-pernyataan diatas dapat ditarik kesimpulan bahwa *activity diagram*

menunjukkan aktivitas sistem dalam bentuk aksi-aksi dan seorang analis dapat menggunakan *activity diagram* untuk memodelkan proses dalam suatu sistem informasi. Diagram ini juga dapat digunakan untuk alur kerja model, *use case individual*, atau logika keputusan yang terkandung dalam metode individual. Berikut ini gambar *diagram activity*.



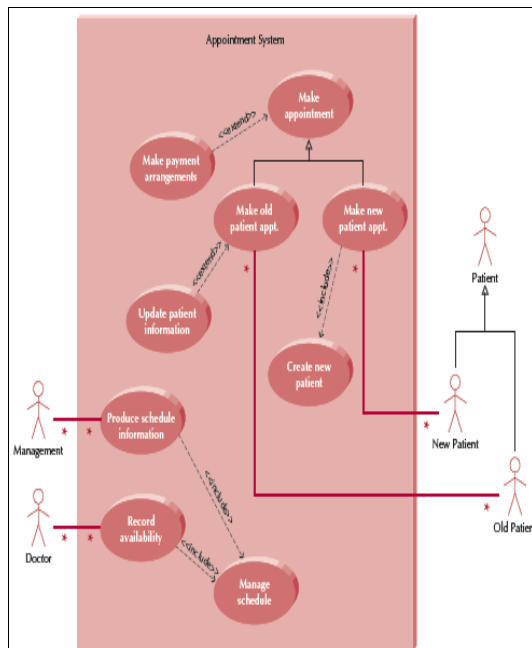
Gambar 1. Activity Diagram

Use Case diagram menggambarkan fungsionalitas dari sebuah sistem yang mempresentasikan sebuah interaksi antara aktor dan sistem. Hal tersebut hampir sama seperti Dennis dkk.[6] yang menyatakan :

Use case diagrams allow the analyst to model the interaction of an information system and its environment. The environment of an information system includes both the end user and any external system that interacts with the information system. The primary use of the use case diagram is to provide a means to document and understand the requirements of the evolving information system. Use cases and use case diagrams are some of the most important tools

that are used in object-oriented systems analysis and design.

Berdasarkan pernyataan-pernyataan tersebut, *use case diagram* memungkinkan analisis untuk memodelkan interaksi antara sistem informasi dan lingkungannya. Lingkungan sistem informasi mencakup pengguna akhir dan setiap sistem eksternal yang berinteraksi dengan sistem informasi. Penggunaan utama dari *use case diagram* adalah untuk menyediakan sarana dalam mendokumentasikan dan memahami persyaratan sistem informasi yang sedang berkembang. *Use case* dan *diagram use case* adalah beberapa alat yang paling penting untuk digunakan dalam analisis dan desain sistem berorientasi objek.



Gambar 2. Use Case Diagram

III. METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

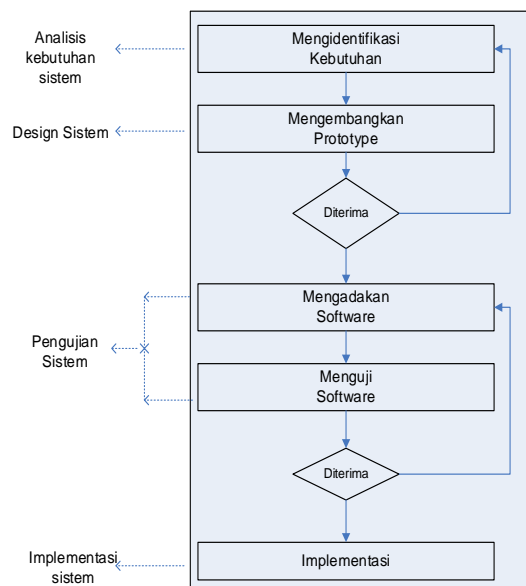
1. Studi Kepustakaan (*Library Research*). Penelitian ini dilakukan untuk pengambilan

data yang bersifat teori yang kemudian digunakan sebagai literatur penunjang guna mendukung penelitian yang dilakukan. Data ini diperoleh dari buku-buku dan sumber lainnya yang dapat dijadikan acuan yang ada kaitannya dengan masalah yang diteliti.

2. Studi Lapangan (*Field Research*). Data diperoleh secara langsung pada objek penelitian. Adapun teknik pengumpulan data yang digunakan adalah :

- a. Observasi yang dilakukan dengan cara mengadakan pengamatan secara langsung terhadap masalah yang sedang diteliti, dengan maksud untuk membandingkan keterangan-keterangan yang diperoleh dengan kenyataan.
- b. Dokumentasi yang dilakukan dengan cara mengumpulkan data berupa soal-soal *toefl* beserta jawabannya yang akan diujicobakan menggunakan aplikasi yang dibangun penulis.

3. Metode Pengembangan Sistem

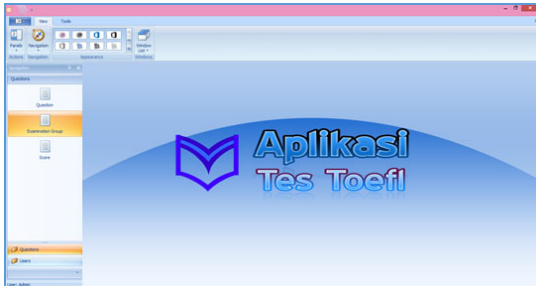


Gambar 3. Tahapan Model *Prototipe*

IV. HASIL DAN PEMBAHASAN

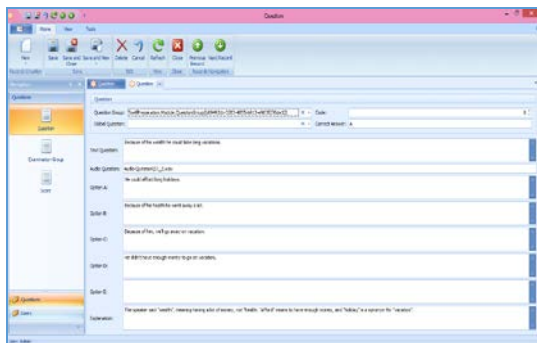
Berikut tampilan halaman utama aplikasi tes *toefl* untuk *admin*.

a. Halaman Utama

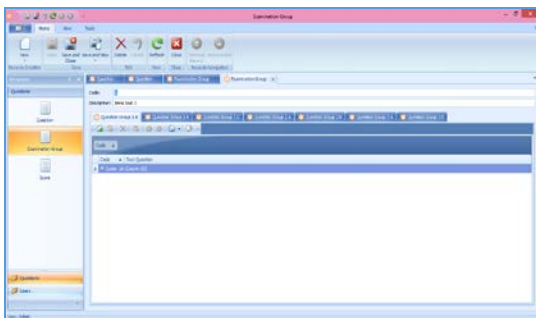


Gambar 4. Halaman Utama Aplikasi Tes *Toefl* untuk *Admin*

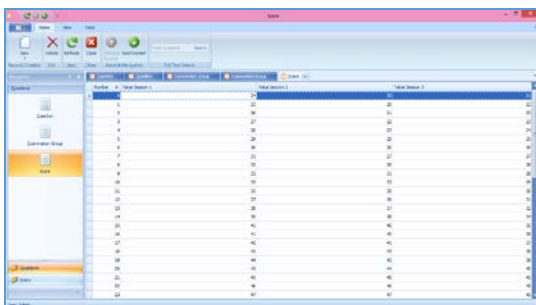
b. Halaman Menu *Question*



Gambar 5. Halaman Sub Menu *Question* Aplikasi Tes *Toefl*

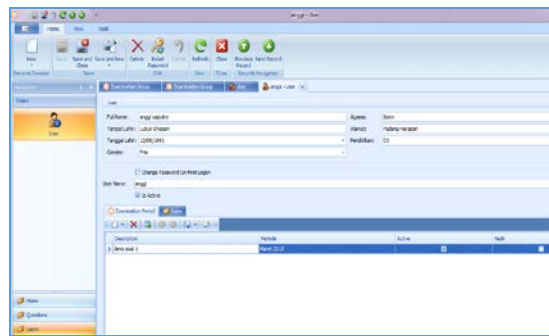


Gambar 6. Halaman Sub Menu *Examination Group* Aplikasi Tes *Toefl*



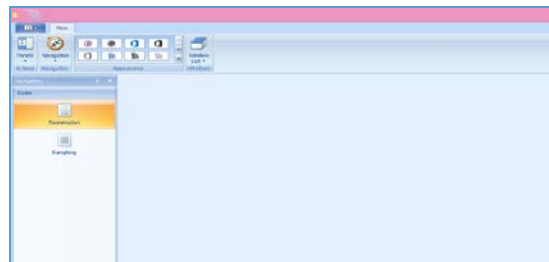
Gambar 7. Halaman Sub Menu *Score* Aplikasi Tes *Toefl*

c. Halaman Menu *User*



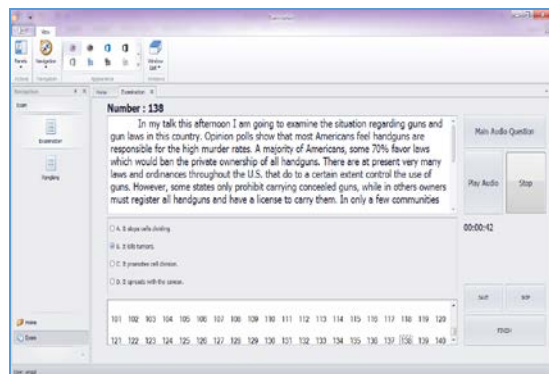
Gambar 8. Halaman Menu Pengguna Aplikasi Tes *Toefl*

d. Halaman untuk Pengguna



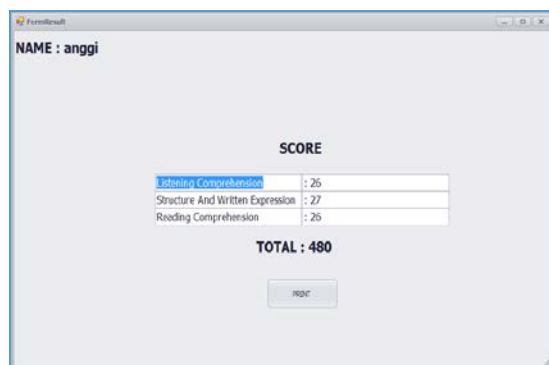
Gambar 9. Halaman Utama Aplikasi Tes *Toefl* untuk Pengguna

e. Halaman *Examination*



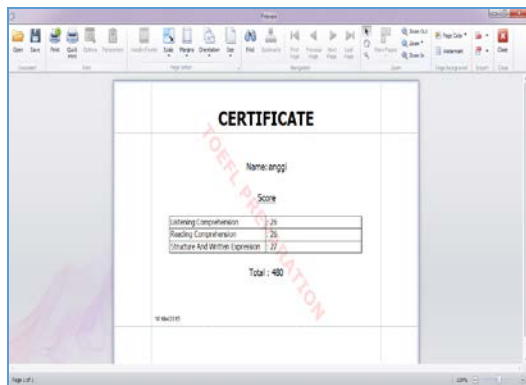
Gambar 10. Halaman *Examination* Aplikasi Tes *Toefl*

f. Halaman *Result*



Gambar 11. Halaman *Result* Aplikasi Tes *Toefl*

g. Tampilan *Print Out*



Gambar 12. Tampilan *Print Out* Tes *Toefl* Aplikasi Tes *Toefl*

h. Halaman Rangkang

Number	Name	Score
1	aroppi saputra	480
2	adwan affandi	447
3	kamilia perhini	443
4	doni iskanadar	410
5	himan setiadi	413
6	erda liliana sari	400
7	galih perhini	390
8	hendoko	387
9	ade irawan	380
10	fizi vandi	377
11	eka maryana	370
12	oudi julianto	353
13	nia safini	350
14	indah Widya Sari	350
15	rhendra aldhyan	350
16	lovi fatmahanifah	347

Gambar 13. Halaman Rangkang Aplikasi Tes *Toefl*

Perangkingan dilakukan terhadap 50 orang pengguna yang telah selesai menjawab pertanyaan-pertanyaan pada aplikasi tes *Toefl* ini, dan sebagian besar dari mereka mendapatkan nilai antara 300-350 pada tes *toefl* ini. Untuk lebih lengkapnya dapat dilihat pada tabel dibawah ini.

Tabel 1. Persentase Perolehan Nilai Pengguna

Range Skor	Total Pengguna	Persentase
300-350	36	72%
351-400	9	18%
401-450	4	8%
451-500	1	2%

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil pembahasan, perancangan serta pembuatan Aplikasi Tes *Toefl*, maka dapat ditarik kesimpulan sebagai berikut:

1. Dari 50 orang pengguna aplikasi ini didapatkan data bahwa yang mendapat skor antara 300-350 ada 36 orang (72 %), yang mendapat nilai antara 351 – 400 ada 9 orang (18 %), yang mendapat nilai 401 – 450 ada 4 orang (8 %) dan yang mendapat nilai antara 451 – 500 ada 1 orang (2 %).
2. Kelemahan dari sistem ini adalah hanya dapat berjalan pada sistem operasi *Windows* dan tidak dapat diakses dari luar jaringan, sedangkan kelebihan adalah aplikasi ini dapat digunakan untuk tes *toefl* ataupun latihan tes *toefl* berbasis komputer baik itu digunakan untuk pribadi ataupun pihak penyelenggara Tes *Toefl* berbasis komputer.
3. Aplikasi ini juga dapat mengetahui langsung nilai hasil tes yang di dapatkan berdasarkan pengisian jawaban pada Aplikasi Tes *Toefl* sebelumnya. Serta dapat mengetahui perangkingan yang didapatkan oleh setiap pengguna setelah mereka selesai menjawab semua soal pada Aplikasi Tes *Toefl* ini.

B. Saran

Saran yang dapat diberikan dari hasil penelitian skripsi ini adalah:

1. Aplikasi ini masih banyak kelemahannya dan diharapkan dapat dikembangkan lebih luas tidak hanya untuk komputer desktop atau laptop tetapi juga diharapkan dapat dikembangkan untuk dapat dibuat berbasis *web* ataupun *mobile*.
2. Diharapkan aplikasi ini dapat dikembangkan agar dapat diakses dari luar jaringan.

REFERENSI

- [1] Pamuji, Eko.2013.*Pembuatan Aplikasi Pembelajaran Andromath Berbasis Android*. Jurusan Sistem Informasi. Sekolah Tinggi Manajemen Informatika dan Komputer. Amikom Yogyakarta.Yogyakarta.
- [2] Qudsi, Dini Hidayatul, Dini Nurmalasari dan Zainal Arifin. 2012. *Toefl For Android*. Jurusan Komputer. Politeknik Caltex Riau, Pekanbaru.
- [3] Prasetyo, Didik Dwi.2006.*Pemrograman Aplikasi Database dengan Visual Basic .NET 2005 dan MS Access*. PT Elex Media Komputindo. Jakarta
- [4] Riyanti, Yuni Dwi, Purwono Hendradi dan Mukhtar Hanafi. 2014. *Pemanfaatan Functional Dependencies (FD) Untuk Peningkatan Efektifitas Validasi Data Pengguna Telepon Seluler*. Jurusan Teknik Informatika. Fakultas Teknik. Universitas Muhammadiyah Magelang.
- [5] Dennis, Alan, Barbara Haley Wixom dan David Tegarden. 2005. *System Analysis and Design with uml version 2.0*. United States of America: A Wiley-Interscience Publication
- [6] Ningsih, Endang Pujiati, Bambang Siswoyo, Riffa Haviani.2014. *Analisis Perbandingan Algoritma Metode Pengurutan Metode Quick Sort, Metode Pengurutan Selection Sort dan Metode Pengurutan Heap Sort*. Sekolah Tinggi Manajemen Informatika dan Komputer. Amikom Yogyakarta.Yogyakarta.