

# APLIKASI PEMBELAJARAN BAHASA INGGRIS PADA PUSAT PENDIDIKAN MASTER BERBASIS ANDROID

Ivan Justin<sup>1</sup>, Genrawan Hoendarto<sup>2</sup>, Lina<sup>3</sup>

<sup>123</sup>Teknik Informatika Universitas Widya Dharma Pontianak

e-mail: <sup>1</sup>paradoxicalmatters1962@gmail.com, <sup>2</sup>genrawan.hoendarto@widyadharm.ac.id, <sup>3</sup>linalo\_77@yahoo.com

## *Abstract*

*Since the end of World War II, the English language has seen a resurgence in use, following the American-assisted booming economies of the postwar world. Since then, it has become the language used in international communications, as countries from East to West begin the process of globalization, many have adopted and championed English as the language foremost in use to bridge the linguistic gap between nations. This importance cannot be overstated: English has made its mark in Indonesian solely through loanwords adapted from English neologisms. Terms such as “computer,” or “cellphone” were indeed borrowed from the English language. Education in this area is crucial for the nation to remain competitive on the world stage. Concurrent with this development, however, is the rise of smartphone devices piloted by the Android OS developed by the Google corporation. As these devices have become ubiquitous in daily life, the author has chosen the Android platform as a baseline for the application. To accomplish this objective, the author uses a descriptive method as a model of research design, with the data being supplied by data collection methods such as interview with relevant subjects, observation of usage, and the study of literature. For the analytics and design of this project, the methods used are object-oriented description through the use of the Unified Modeling Language (UML) and database design is done through the MySQL and SQLite softwares, both of which have their own uses within their application. This research has resulted in an interactive English learning app, where users can read materials and exercise their English. The conclusion is that an application makes it easier for users to access, retain, and exercise their knowledge through the use of a ubiquitous electronic device. Suggestions for further development include the inclusion of more content, and more varieties of exercises.*

**Keywords:** *Android, Development, Learning, English language*

## **Abstrak**

Sejak berakhirnya Perang Dunia II, bahasa Inggris telah mengalami kenaikan dalam hal penggunaan, terutama setelah *booming* pada sektor ekonomi berbagai negara di dunia paska-perang. Bahasa Inggris telah menjadi bahasa yang digunakan dalam komunikasi internasional, setelah berbagai negara dari Timur atau Barat memulai proses globalisasi, dengan mayoritas mengangkat Inggris sebagai bahasa jembatan perbedaan linguistik berbagai bangsa. Pentingnya Inggris tidak dapat dikesalkan: berbagai kata pinjam dalam bahasa Indonesia berasal dari neologisme yang diadaptasi dari bahasa Inggris. Istilah seperti “komputer,” atau “ponsel” dipinjam dari bahasa Inggris. Pembelajaran dalam bidang ini penting bagi bangsa agar dapat berkompetisi pada panggung dunia. Seiring dengan perkembangan bahasa Inggris adalah naiknya alat-alat *smartphone* yang memakai OS Android yang dikembangkan oleh perusahaan Google. Karena alat-alat ini telah menjadi umum pada kehidupan sehari-hari, penulis telah memutuskan untuk memakai platform Android sebagai *baseline* aplikasi. Untuk mencapai tujuan ini, penulis menggunakan metode deskriptif sebagai model perancangan penelitian, dengan data yang diambil menggunakan metode wawancara dengan subyek yang relevan, observasi penggunaan bahasa, dan studi literatur. Untuk analisa dan perancangan, digunakan metode deskripsi berorientasi objek menggunakan *Unified Modeling Language* (UML) dan perancangan database yang dilakukan menggunakan software *MySQL* dan *SQLite*. Hasil penelitian ini adalah sebuah aplikasi pembelajaran bahasa Inggris yang interactive, di mana *user* dapat membaca dan melatih kephahaman materi mereka. Kesimpulan adalah aplikasi membuat mudah pengguna dalam mengakses, memelihara, dan melatih kemampuan mereka dengan menggunakan objek elektronik yang umum. Saran untuk pembangunan selanjutnya berupa pembuatan konten yang lebih besar, dan varietas latihan yang lebih banya.

**Kata Kunci:** Android, Pembangunan, Pembelajaran, Bahasa Inggris

## 1. PENDAHULUAN

Pada dekade-dekade akhir abad ke-20 dan mulainya abad ke-21, bahasa Inggris telah melesat menjadi bahasa utama bidang telekomunikasi dan teknologi. Dominasi Amerika pada bidang ekonomi membuat bahasa Inggris menjadi pilihan yang alami untuk sebuah bahasa universal dalam sains, teknologi, dan informasi. Pada dunia modern, bahasa Inggris telah menjadi sebuah bahasa yang digunakan untuk telekomunikasi internasional dan universal. Bahasa Indonesia, yang tidak memiliki hubungan kekerabatan yang nampak dengan bahasa Inggris, telah meminjam sebagian besar istilah teknologinya dari neologisme-neologisme yang pertama-tama dibuat dalam bahasa Inggris (contohnya, komputer dari “computer”, dan ponsel dari “cellphone.”)

Oleh karena itu, sang penulis bermaksud mengembangkan aplikasi untuk Pusat Pendidikan Master sebagai pembantu pembelajaran bahasa Inggris dengan menggunakan buku *Cambridge Grammar in Use* sebagai acuan pembuatan kurikulum. Mengenai *platform* pembangunan, Android merupakan *platform* yang tepat karena ketersediannya yang luas dan terjangkau. Sebagai tambahan kepada pertimbangan penulis, Android memiliki karakteristik sebuah *multimedia-compatible device* dan memenuhi keperluan sekolah untuk lingkungan pembelajaran yang lebih interaktif dan bervariasi.

## 2. METODE PENELITIAN

2.1 Rancangan Penelitian, Teknik Pengumpulan Data, Metode Analisa dan Perancangan Sistem, Alat-alat Pembangunan Aplikasi.

2.1.1 Rancangan Penelitian

Penulis membaca dan mempelajari materi yang diterbitkan sehubungan dengan masalah yang diteliti atau dengan cara mengumpulkan data melalui referensi dan literatur yang terkait dalam menyelesaikan masalah yang diteliti.

2.1.2 Teknik Pengumpulan Data

Adapun Metode pengumpulan data yang digunakan oleh penulis antara lain:

2.1.2.1 Studi Literatur

Penulis membaca dan mempelajari materi yang diterbitkan sehubungan dengan masalah yang diteliti atau dengan cara mengumpulkan data melalui referensi dan literatur yang terkait dalam menyelesaikan masalah yang diteliti.

2.1.2.2 Wawancara

Penulis mencoba menanyakan kepada pemilik properti dan staf pengajar agar dapat menyesuaikan program agar memenuhi kebutuhan objek penelitian. Dalam proses ini, penulis mencoba untuk mencari *feedback* untuk terlebih mengembangkan aplikasi program pembantu pembelajaran.

2.1.2.3 Metode Analisa dan Perancangan Sistem

Penulis menggunakan *Unified Modeling Language* (disingkat menjadi UML) dengan tujuan memvisualisasikan, membangun konsep dan mendokumentasikan sistem yang dikembangkan secara sistematis dan detail.

2.1.2.4 Alat-alat Pembangunan Aplikasi

Perancangan dilakukan dengan menggunakan Android Studio sebagai IDE, Java dan PHP sebagai bahasa pemrograman utama dan SQLite Server sebagai DBMS utama. Perancangan dapat melibatkan bahasa pemrograman lain sebagaimana memenuhi kebutuhan.

2.2 Landasan Teori

2.2.1 Data

Data adalah arus fakta mentah yang mewakili kejadian yang terjadi pada organisasi sebelum dapat diorganisir dan diatur menjadi bentuk yang dapat dipahami dan digunakan. [1] Data terdiri dari fakta mentah, seperti nomor pegawai, total jam pekerjaan pada satu minggu, angka bagian inventori, atau jumlah unit yang diproduksi, dan data mempunyai nilai yang kecil diluar keberadaannya. [2]

2.2.2 Informasi

Informasi adalah data yang telah dibentuk menjadi sebuah tatanan yang berarti dan berguna bagi manusia. [1] Informasi adalah kumpulan data yang diorganisasi dan diproses agar ia memiliki nilai lain selain nilai fakta-fakta tersebut secara individu. [2]

### 2.2.3 Unified Modeling Language (UML)

UML merupakan sebuah alat komunikasi antar *developer* yang mengedepankan ekspresi grafik menggunkan notasi dibanding dengan komunikasi berbentuk *text*. Tujuan UML adalah memberikan teknik permodelan untuk semua *domain* aplikasi dalam pengembangan *software*. [3] Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.” [4]

### 2.2.4 Android

Android adalah *software stack* untuk piranti *mobile*. *Stack* ini terdiri dari aplikasi, sebuah mesin virtual yang menjalankan aplikasi, *middleware* (*software* yang berada di atas system operasi dan menyediakan berbagai layanan untuk *virtual machine* dan aplikasi-aplikasinya) dan sistem operasi Linux. [5]

### 2.2.5 MySQL

MySQL adalah sebuah bentuk dari *Structured Query Language* yang dirancang menggunakan bahasa Inggris dan didesain untuk mengizinkan permintaan sederhana dari database. [6] MySQL adalah sistem database yang dapat menyimpan informasi dan kemudian diintegrasikan dengan PHP untuk menciptakan aplikasi web yang berfungsi penuh. [7]

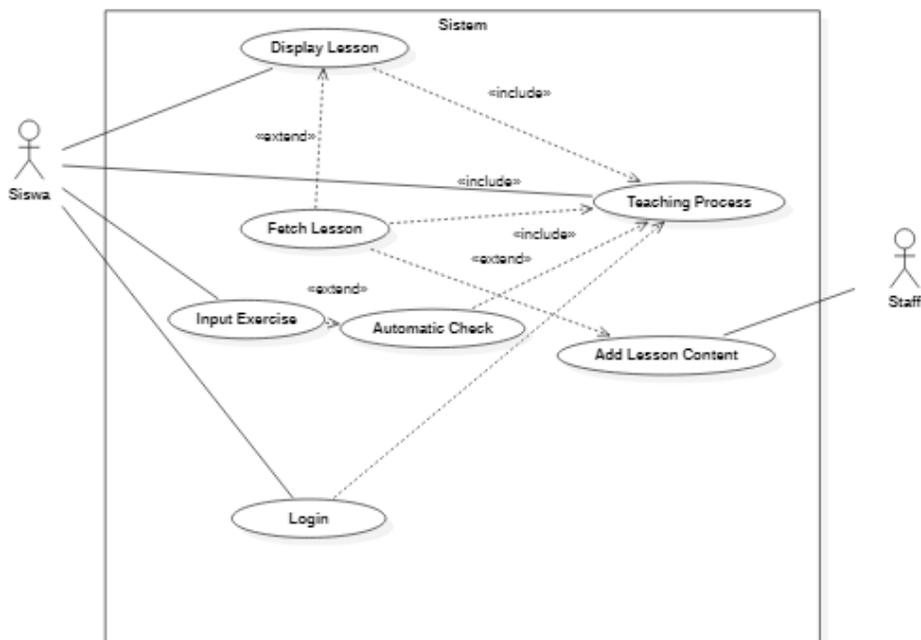
## 3. HASIL DAN PEMBAHASAN

### 3.1 UML Sistem Usulan

Agar perancangan aplikasi pembantu pembelajaran bahasa Inggris berbasis Android ini dapat berjalan sesuai dengan harapan penulis, maka penulis menggunakan diagram *Unified Modeling Language* (UML) untuk menjabarkan proses-proses yang ada. Berikut adalah diagram UML dari sistem yang diusulkan:

#### 3.1.1 Use Case Diagram

Perancangan diagram *use case* bertujuan untuk memberikan gambaran atas interaksi-interaksi yang dapat digunakan oleh *user* aplikasi yang dirancang. Pada saat aplikasi pertama dijalankan, *user* atau pengguna harus menghadapi *form login*, dimana *user* harus memasukkan *username* dan *password* yang diberikan untuk memverifikasi keberadaan *user* tersebut di dalam sistem.



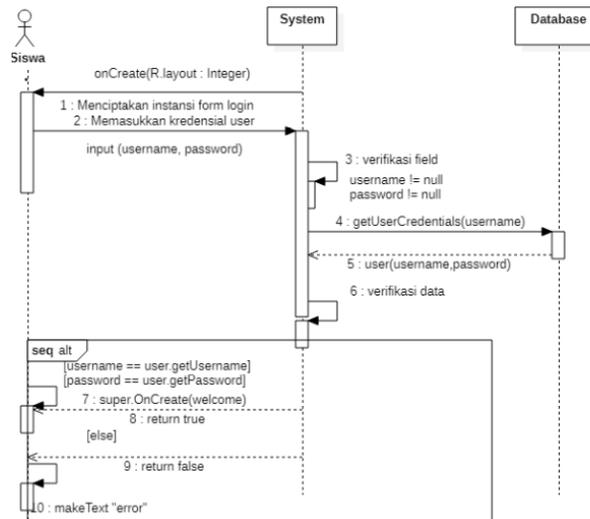
Gambar 1. Use Case Diagram Sistem Usulan

Setelah sistem yang dirancang mengkonfirmasi keberadaan *user* di dalam sistem, *user* (yang merupakan siswa yang memilih untuk mengikuti kursus di Pusat Pendidikan Master) dapat menjalankan *lesson-lesson* yang telah diberikan oleh penulis. Dalam *lesson-lesson* tersebut terdapat latihan, atau *exercises* yang dapat dikerjakan

oleh siswa. Setelah siswa mengerjakan latihan-latihan tersebut, maka sistem akan melakukan proses *auto-checking* pada hasil latihan yang dikerjakan olehnya. Tergantung dari hasil latihan tersebut, pekerjaan siswa dapat dikonversikan menjadi *token* yang digunakan oleh *progress tracker* untuk membuka pelajaran-pelajaran yang lebih tinggi.

### 3.1.2 Diagram Sekuensial Login

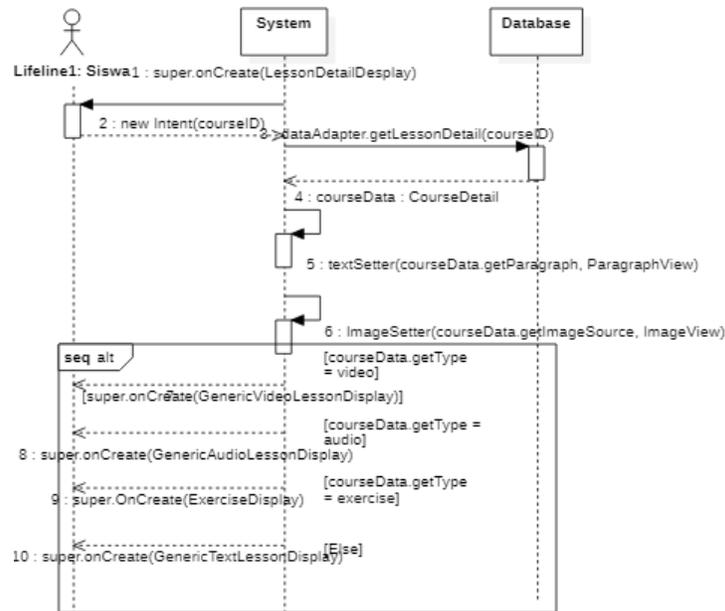
Ketika pengguna (dalam kasus ini siswa) mengakses aplikasi, maka sistem akan menampilkan sebuah form *login*. Setelah form *login* tersebut diisi, maka sistem akan membandingkan *username* dan *password* yang dikirim dengan *record* yang berada di database. Setelah perbandingan selesai maka sistem mengirim pesan konfirmasi yang berisikan hasil perbandingan tersebut.



Gambar 2. Diagram Sekuensial Login Sistem Usulan

### 3.1.3 Diagram Sekuensial Lesson

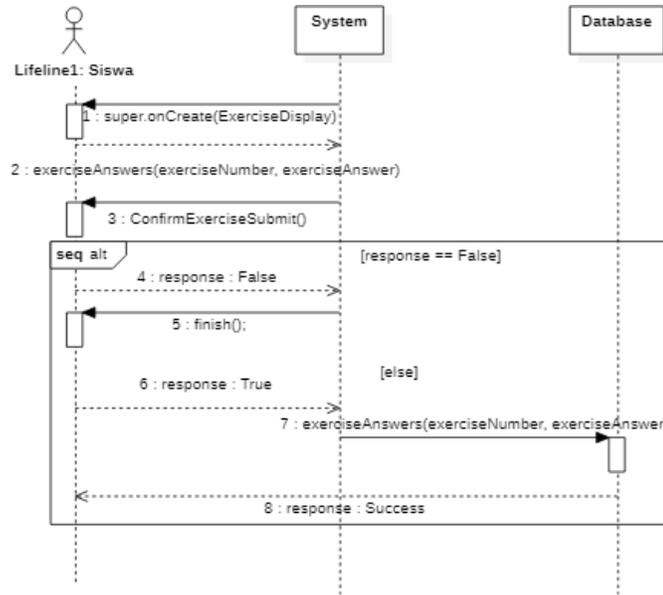
Setelah siswa memilih *lesson* yang ditampilkan pada sidebar, maka sistem akan mengambil *metadata lesson* pada database (yang berisikan *file path*, nama *lesson*, dan spesifikasi *lesson* lainnya). Setelah *metadata* diambil maka sistem akan mengambil *datafile* dari *path* yang dispesifikasi. Setelah semua spesifikasi telah dipenuhi, maka program akan membangun (*build* atau *load*) *lesson*, yang kemudian akan ditampilkan kepada siswa.



Gambar 3. Diagram Sekuensial Lesson Sistem Usulan

### 3.1.4 Diagram Sekuensial Exercise

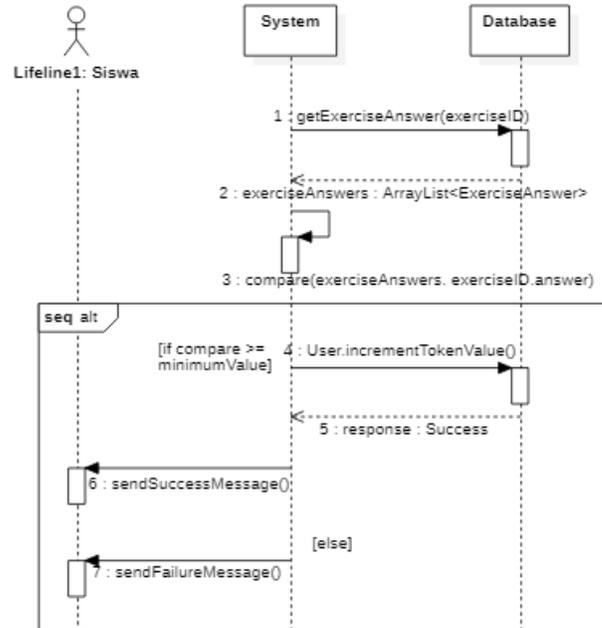
Dalam proses pengerjaan *lesson*, aplikasi akan menampilkan form latihan atau *exercise* kepada siswa. Setelah pengerjaan selesai, maka sistem merekam respon siswa dan mensortir data jawaban siswa berdasarkan ID siswa, menunggu hasil pengecekan *autocheck* untuk mendapatkan *token* yang di-*feed* ke dalam proses *progress tracker*.



Gambar 4. Diagram Sekuensial Exercise Sistem Usulan

### 3.1.5 Diagram Sekuensial Autocheck

Pertama sistem akan meminta data jawaban dari database. Setelah data jawaban telah diperoleh, maka sistem akan meminta data kunci jawaban dari database, yang akan dibandingkan dengan data jawaban untuk menghasilkan *token* yang akan diberikan pada *progress tracker*, serta memberikan *feedback* kepada siswa.

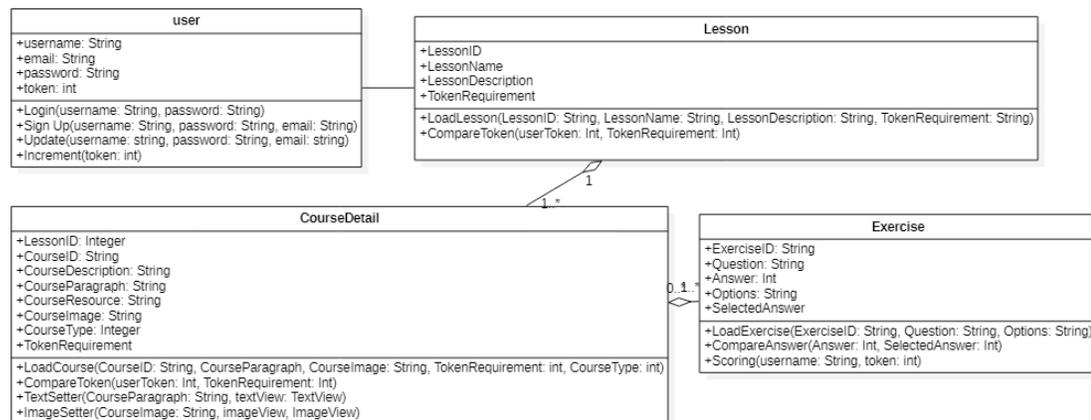


Gambar 5. Diagram Sekuensial Autocheck Sistem Usulan

### 3.1.6 Diagram Kelas

Diagram kelas merupakan model statis yang digambarkan untuk mengilustrasikan struktur dan deskripsi kelas serta hubungan antar kelas. Berikut merupakan kelas yang berada pada aplikasi pembantu pembelajaran Bahasa Inggris pada Pusat Pendidikan Master:

- User**  
Terdiri dari *property-property*: *username* dan *password*
- Lesson**  
Terdiri dari *property-property*: *id lesson*, *nama lesson*, *token requirement*, dan *path-path* yang digunakan untuk menyusun lesson (seperti file XML).
- Progress Tracker**  
Terdiri dari *property-property*: *username*, *id lesson* dan *token value*. *Token value* akan dipakai untuk mengkalkulasikan apakah *user* dapat mengakses tingkatan-tingkatan pelajaran yang lebih tinggi.



Gambar 6. Diagram Kelas Aplikasi Sistem Usulan

## 3.2 Tampilan Aplikasi Pembantu Pembelajaran Bahasa Inggris

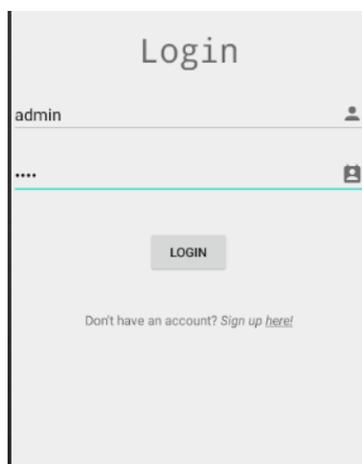
### 3.2.1 Tampilan Halaman Login

Bagian antarmuka yang akan ditunjukkan oleh aplikasi kepada pengguna adalah *activity login* beserta *layout*-nya. Pada susunan UI ini, terdapat beberapa komponen penting yang membuat *activity* dan program secara umum dapat beroperasi. Aplikasi akan mengambil *input username* dan *password* lewat *EditText* yang berhubungan. Tujuan pengambilan *input* ini adalah untuk memverifikasi *user* dengan akun yang hendak diakses dan melakukan autentifikasi apabila *user* mempunyai hak. Hal ini dilakukan agar *user* dapat mengakses fitur-fitur yang didapatkan melalui *token* yang berada pada akun *user*. Apabila *user* tidak memiliki akun di dalam sistem, maka mereka dapat mendaftarkan keberadaan mereka di dalam sistem dengan menekan *TextView* yang disediakan di bawah tombol *login*. Setelah *TextView* tersebut ditekan, maka *activity login* akan menjalankan proses *finish()*; dan menginisialisasikan *intent* untuk menampilkan tampilan *activity signup*.

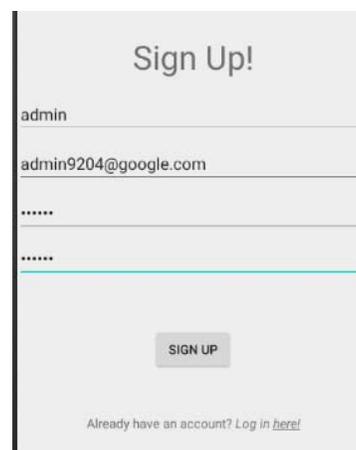
### 3.2.2 Tampilan Halaman Sign Up

Setelah *activity login* selesai dan *activity signup* dimulai, maka tampilan *layout* yang diberikan oleh *signup* tidak jauh berbeda dari *login*. *EditText* yang disediakan menjadi empat: *username*, *email*, *password*, beserta konfirmasi. Karena tujuan *user* adalah untuk mendaftarkan akun baru ke dalam sistem yang disediakan, maka aplikasi harus meminta informasi yang lebih banyak dari *user*. *User* harus mengisi semua *field* yang disediakan oleh aplikasi, dan mengisi *field* konfirmasi *password* dengan benar. Apabila ada *field* yang kosong, maka aplikasi akan memberikan *error*, dan meminta *user* untuk mengisi semua informasi yang telah diminta oleh program.

Setelah *user* berhasil meng-*input* semua informasi yang diminta oleh aplikasi dengan benar, menekan tombol *signup* akan membuat aplikasi melakukan fungsi *POST* pada *server MySQL* yang disediakan oleh penulis untuk mengirim *query* informasi ke dalam database. Jika ada data duplikat, maka *server* akan memberikan *echo error* dan meminta *user* untuk memilih *username* yang berbeda. Jika berhasil, *server* akan memberikan *echo* berbentuk string yang bertujuan untuk mengirimkan pesan sukses kepada *user*. Sebagai tambahan, setelah proses *signup* berhasil dilakukan, maka *user* akan di *login* secara otomatis.



Gambar 7. Tampilan Login



Gambar 8. Tampilan Signup

### 3.2.3 Tampilan Halaman Beranda

Setelah *user* menyelesaikan proses *login* dan *signup* secara sukses, maka aplikasi akan melakukan transisi ke *activity* beranda. *Activity* ini merupakan pusat pengoperasian aplikasi. Pada bagian atas *activity* terdapat string yang mengucapkan nama *user* dalam *TextView*, yang akan menggunakan informasi yang terdapat pada *global variable currentUser*, yang disusun berdasarkan *class userCredentials*. Tujuan penyimpanan ini agar aplikasi dapat mensortir dan mengunci *lesson* dengan benar sesuai dengan *token* yang dimiliki oleh *user* yang *login* ke dalam sistem.

Setelah satu *lesson* dipilih, maka *activity* beranda akan memulai sebuah *activity CoursesList*, yang berfungsi untuk me-load dan menampilkan semua *courses* yang terdapat di bawah *lesson* tersebut. Untuk membantu *activity CoursesList* dalam melakukan tugas ini, maka *activity* beranda akan mendorong (*push*) informasi tambahan dalam bentuk ID dari *class lesson* tersebut, yang akan menampilkan *courses* yang berada di bawah *lesson* tersebut.

### 3.2.4 Tampilan Halaman CoursesList

Proses program selanjutnya adalah menampilkan daftar *courses* yang dimiliki oleh sebuah *lesson*. Setelah *activity CoursesList* diluncurkan oleh program, maka hal pertama yang akan dilakukan oleh aplikasi adalah mengambil data *class Lesson* yang di-pass oleh *activity* beranda. Menggunakan data ini, aplikasi akan melakukan proses *query* terhadap database SQLite yang berada di dalam program. *Query* ini berfungsi untuk mengambil *preview courses* yang berada di bawah naungan sebuah *lesson*, menggunakan ID yang diasosiasikan kepada sebuah *lesson* tertentu untuk mendapatkan daftar *courses* yang terasosiasi dengan *lesson* tersebut. Dari daftar ini, pengguna dapat melihat *courses* yang dapat diakses menggunakan *token* yang dimiliki. *User* kemudian dapat memilih salah satu *lesson* yang tersedia, yang akan membuat *activity* untuk mendorong *class* yang terasosiasi dengan urutan *display* ke *activity* selanjutnya, yakni *activity lesson* atau *exercise*. *Activity CoursesList* kemudian akan dipindah ke *back stack* agar dapat diakses waktu *user* ingin melihat *courses* yang lain.

### 3.2.5 Tampilan Halaman Lesson

Ketika *user* menekan sebuah *lesson* dari daftar *courses* yang tersedia, maka *activity lesson* akan menerima input dari *activity coursesList*. *Activity lesson* kemudian akan memproses input tersebut menurut sebuah *generic layout* yang disediakan penulis. Dalam input yang diterima oleh *activity* sebelumnya adalah sebuah *class coursesDetail* yang berisi rujukan-rujukan terhadap *resources* yang berada di dalam APK aplikasi. Rujukan-rujukan ini memiliki tipe-tipe seperti paragraf (di dalam penulisan kode: "*A1\_first\_paragraph*") atau gambar, yang dapat berupa alamat URL atau nama *drawable* yang berada di dalam APK aplikasi.

### 3.2.6 Tampilan Halaman Exercise

Untuk mengakses tingkatan *lesson* yang lebih tinggi, *user* harus mengerjakan latihan atau *exercise* yang diberikan program. Pada *coursesList*, maka *user* harus memilih *lesson* yang ditandai dengan *exercise*. Melakukan klik terhadap *CardView exercise* akan mengirim *user* ke *activity exercise*. Aplikasi akan mengirim *query* rujukan-rujukan kepada database SQLite yang berada di dalam program. *Query* ini akan mengambil soal-soal yang berada di dalam database dan menyusunnya ke dalam bentuk yang dapat dicerna dan dikerjakan oleh *user*. *Query* juga akan

mengambil kunci jawaban yang berada di dalam database, sehingga soal dapat diperiksa sesegera mungkin setelah *user* menekan tombol *submit*.



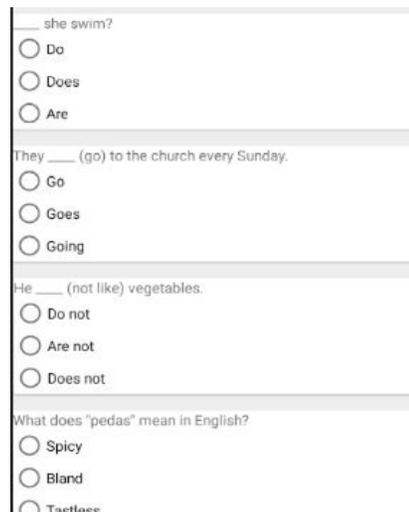
Gambar 9. Tampilan Beranda



Gambar 10. Tampilan Lesson



Gambar 11. Tampilan CoursesList



Gambar 12. Tampilan Exercise

### 3.3 Lingkungan Pengembangan Program

Berikut ini adalah spesifikasi yang dipakai baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) dalam pengembangan aplikasi diantaranya :

#### 3.3.1 Hardware

##### a. Laptop

- 1) *Processor* Intel Core i7-4510U 2.0GHZ
- 2) *Memory* 4 GB RAM
- 3) NVIDIA GeForce 840M

##### b. Smartphone

- 1) *Oppo* A37
- 2) *Resolution* 1280 x 720 (HD+)
- 3) *Android* 5.1.1 (Lollipop)
- 4) *Qualcomm MSM8916 Snapdragon 410 Quad-core Cortex-A53*
- 5) *Ram* 2 GB

#### 3.3.2 Software

- c. Sistem operasi Windows 7 (64 Bit)

- d. Android Studio
- e. MySQL Server
- f. Internet

#### 4. KESIMPULAN

Setelah melakukan pengujian terhadap aplikasi pembantu pembelajaran bahasa Inggris selesai dilakukan, kesimpulan yang dapat ditarik dari hasil pengujian tersebut adalah sebagai berikut:

- a. Hasil pengujian pada program menghasilkan bukti bahwa program dapat memenuhi kebutuhan multimedia Pusat Pendidikan Master yang mempunyai performa konsisten.
- b. Aplikasi dapat dijalankan pada Android versi 5.1 dan di atasnya.
- c. Aplikasi memiliki kemampuan multimedia berupa memutar video *YouTube*, menampilkan *image* yang relevan dalam kurikulum agar dapat menampilkan *lesson* yang menarik.
- d. Sebuah server dibutuhkan untuk meng-*host* data *user* dan agar data dapat konsisten pada berbagai situasi. Fungsi *login* dan *signup* membutuhkan sebuah *server* inang yang dapat memverifikasi informasi *user* dan melakukan *update* pada data *user* jika dibutuhkan perubahan, contohnya pada penambahan variabel *token* pada database *MySQL*.
- e. Aplikasi dapat memeriksa hasil latihan yang diberikan kepada *user*, sebagaimana diuraikan pada Bab IV.
- f. Program membagi materi menurut kelas dan kompetensi melalui sistem *token*.

#### 5. SARAN

Adapun saran-saran yang dapat digunakan untuk pengembangan lebih lanjut adalah sebagai berikut:

- a. Infrastruktur yang memadai agar *update* data program dapat dilangsungkan secara online, tanpa harus melalui intervensi dari *programmer*.
- b. Penambahan varietas *exercise* yang lebih menarik dan kreatif.
- c. Penambahan bahan-bahan multimedia lain agar proses pembelajaran dapat dibantu menggunakan fitur-fitur yang lebih menarik
- d. Penambahan fitur *backup* dan *recovery* pada bagian server *MySQL* agar data pelajar tidak hilang.
- e. Penambahan *lesson* yang dapat mencapai tingkatan *Intermediate*, agar program dapat digunakan oleh pengguna yang memiliki kemampuan yang Inggris yang lebih tinggi.
- f. Penambahan lokalisasi *American English* menggunakan *resources* yang berkaitan dengan bahasa tersebut.
- g. Penambahan fitur pemeriksaan manual agar terdapat latihan-latihan berbentuk *essay*.
- h. Memuktahirkan versi *Android* yang digunakan agar dapat menggunakan fitur-fitur dari tingkatan API yang lebih tinggi dari versi 5.1.

#### UCAPAN TERIMA KASIH

Dalam penelitian ini, penulis telah banyak mendapat bantuan bimbingan, data, saran, dan dukungan moril dari berbagai pihak, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada civitas akademika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak, kepada pembimbing, kepada staf pengajar dan manajemen di Pusat Pendidikan Master dan kepada pihak-pihak lain yang sudah sangat membantu penulis secara teknis dan moril dalam menyelesaikan penelitian ini.

#### DAFTAR PUSTAKA

- [1] Laudon, Ken, dan Laudon, Jane P. (2012). *Management Information Systems, 12th Edition*. New York: Pearson.
- [2] Stair, Robert M., dan Reynolds, George W. (2014). *Fundamentals of Information Systems, Eighth Edition*. Boston: Cengage Learning.
- [3] Rumpe, Bernhard. (2016). *Modeling with UML*. New York: Springer.
- [4] Nugroho, Bunafit (2010). *Aplikasi Pemrograman Web Dinamis Dengan PHP dan Mysql*. Yogyakarta: Gava Media.
- [5] Friesen, Jeff (2014). *Learn Java for Android Development, Third Edition*. New York: Apress.

- [6] Nixon, Robin (2014). *Learning PHP, MySQL, JavaScript, CSS dan HTML5*. Sebastopol: O'Reilly Media.
- [7] Suehring, S., dan Valade, J. (2013). *PHP, MySQL, Javascript, dan HTML5, All-in-one For Dummies*. New Jersey: John Wiley dan Sons, Inc.