

**PENGEMBANGAN APLIKASI EDUKASI BERBASIS ANDROID
SEBAGAI SARANA BELAJAR PENGUCAPAN KOSAKATA
BAHASA INGGRIS**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muh. Asdar
NIM: 155150400111130



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

PENGEMBANGAN APLIKASI EDUKASI BERBASIS ANDROID SEBAGAI SARANA
BELAJAR PENGUCAPAN KOSAKATA BAHASA INGGRIS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muh. Asdar
NIM: 155150400111130

Skripsi ini telah diuji dan dinyatakan lulus pada
17 Juni 2019
Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 197408232000121001

Faizatul Amalia, S.Pd., M.Pd.
NIK: 2013098608212001

Mengetahui
Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 197408232000121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 13 Mei 2019

[materai 6000]

Muh. Asdar

NIM: 155150400111130



PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga penelitian skripsi yang berjudul “Pengembangan Aplikasi Edukasi Berbasis Android Sebagai Sarana Belajar Pengucapan Kosakata Bahasa Inggris” ini dapat terselesaikan.

Penulis menyadari bahwa penelitian ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Herman Tolle, Dr. Eng., S.T, M.T. selaku dosen pembimbing satu skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Ibu Faizatul Amalia, S.Pd., M.Pd. selaku dosen pembimbing dua skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibunda dan ayahanda beserta seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Bapak Herman Tolle, Dr. Eng., S.T, M.T. selaku ketua Jurusan Sistem Informasi.
5. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku ketua Program Studi Sistem Informasi.
6. Seluruh civitas akademika Sistem Informasi Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian skripsi ini.
7. Seluruh responden yang telah membantu memberikan saran dan meluangkan waktu untuk kelancaran penelitian.
8. Seluruh teman-teman penulis khususnya teman-teman di UKM Nol Deraat Film yang selalu menyemangati ketika bertemu dengan bertanya “skripsi sudah sampai mana?” atau “kapan wisuda?”.
9. Arif Ardhan, Faris Genis dan Felix Yosa yang telah membantu kelancaran penulisan laporan penelitian ini dengan cara membagikan informasi mengenai laporan skripsi.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap laporan penelitian ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 13 Mei 2019

Penulis

muh.asdar17@gmail.com



ABSTRAK

Muh. Asdar, Pengembangan Aplikasi Edukasi Berbasis Android Sebagai Sarana Belajar Pengucapan Kosakata Bahasa Inggris

Pembimbing: Herman Tolle, Dr. Eng., S.T, M.T. dan Faizatul Amalia, S.Pd., M.Pd.

Penggunaan *smartphone* android sudah menjadi bagian dalam kehidupan manusia untuk mempermudah memenuhi kebutuhan. Bahasa Inggris merupakan bahasa internasional yang banyak digunakan di dunia saat ini, oleh karena itu sudah banyak aplikasi yang dapat membantu belajar bahasa Inggris, namun masih banyak orang yang salah mengucapkan kosakata bahasa Inggris yang benar, karena itu dirancang sebuah aplikasi belajar pengucapan kosakata bahasa Inggris berbasis *smartphone* android, maka pengguna dapat belajar bahasa Inggris secara digital dengan mudah kapanpun dan dimanapun. Karena tujuan utama aplikasi untuk melatih pengucapan kosakata bahasa Inggris, maka pada penelitian ini diusulkan aplikasi android dengan menerapkan *voice recognition* yakni menggunakan *Application programming interface (API) text to speech* dan *speech to text*. Aplikasi dibuat dengan bahasa pemrograman java, database SQLite dan API google *text to speech* dan *speech to text*. Pengembangan aplikasi menggunakan metode *prototyping* karena aplikasi diharapkan sesuai dengan kebutuhan calon pengguna dan cara belajar yang diterapkan pada aplikasi telah sesuai untuk melatih pengucapan kosakata bahasa Inggris. Pengujian aplikasi menggunakan pengujian *requirement validation testing* dengan hasil 100% valid, *usability testing* dengan hasil system usability score 85 yang berarti aplikasi layak digunakan dan pengujian implementasi akurasi google *speech* API dengan tingkat akurasi 100% dengan syarat pengucapan dari pengguna haruslah jelas dan tidak terlalu cepat.

Kata kunci: *voice recognition*, metode *prototyping*, pengujian *requirement validation*, pengujian *usability*, pengujian akurasi google *speech* API.

ABSTRACT

Muh. Asdar, Development of Android-Based Educational Application as a Learning Tool for English Vocabulary Pronunciation

Supervisors: Herman Tolle, Dr. Eng., S.T, M.T. and Faizatul Amalia, S.Pd., M.Pd.

The use of android smartphones already part in people's lives to make ends meet. The English language is the international language that is widely used in the world today, because it's been a lot of Applications that can help learning the language of the English, but still a lot of people wrong or incorrectly pronounces the English language vocabulary, therefore designed an Application learn pronunciation of vocabulary English based android smartphone, users can learn the language of United Kingdom digitally with ease whenever and wherever. Because the purpose of Application to train pronunciation and vocabulary of the language of United Kingdom, then this proposed research on android Applications by applying voice recognition using Application programing interface (API) text to speech and speech to text. Applications created with the java Programming language, SQLite database API and google Text to speech and Speech to text. Application development using the method of prototyping because the Application is expected to match the needs of prospective users and how to learn that Application has been applied according to train pronunciation of the English language vocabulary. Testing the Application using requirement validation testing with 100% valid results, usability testing with the results of the sytem usability score is 85 which means Application feasibility and testing the accuracy of the google speech APIS with 100% accuracy with the terms of the pronunciation of the user must be clearly and not too fast.

Keywords: voice recognition, prototyping methods, testing requirements testing, usability testing, testing the accuracy of Google speech API.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
DAFTAR LAMIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Aplikasi Belajar Pengucapan Kosakata Bahasa <i>Inggris</i>	6
2.3 Android	7
2.4 <i>Voice Recognition</i>	8
2.5 <i>Google Speech API</i>	9
2.6 Android Studio	9
2.7 Metode Pengembangan <i>Prototyping</i>	10
2.8 Pengujian <i>Black Box</i>	11
2.9 <i>Requirement validation Testing</i>	12
2.10 Pengujian <i>Usability</i>	12
2.11 <i>System Usability Scale</i>	13
BAB 3 METODOLOGI	14
3.1 Studi Literatur	15

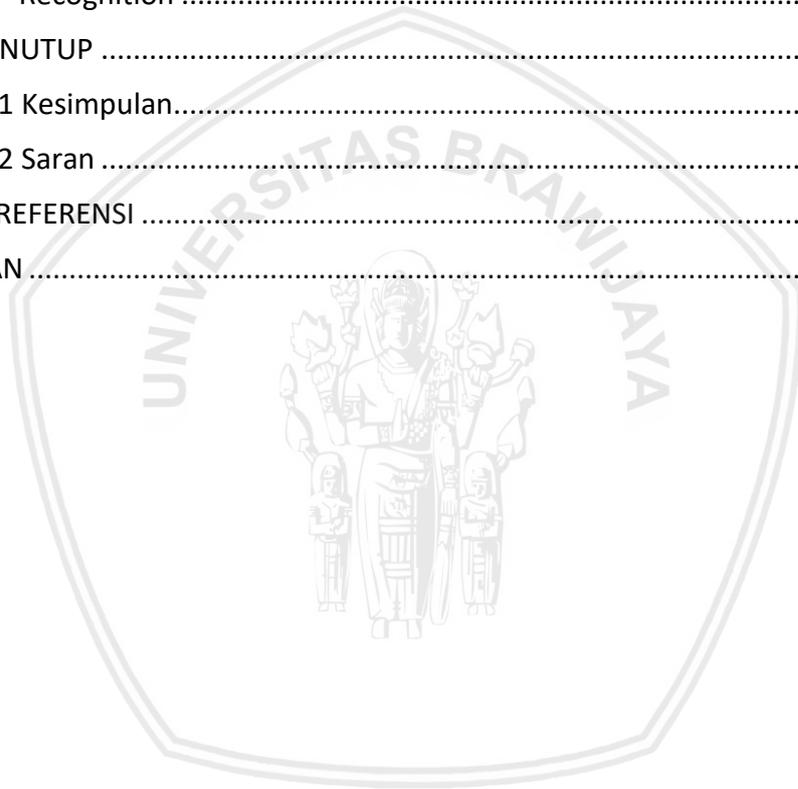


3.2 Analisis Kebutuhan dan Perancangan	15
3.3 Implementasi	17
3.4 Pengujian dan Analisis	17
3.5 Kesimpulan dan Saran	18
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN	19
4.1 Gambaran Umum Sistem.....	19
4.2 Skenario Aplikasi Belajar Pengucapan Kosakata Bahasa Inggris	20
4.2.1 Konten Aplikasi Belajar pengucapan Kosakata Bahasa Inggris ...	20
4.2.2 Alur Aplikasi Belajar pengucapan Kosakata Bahasa Inggris	21
4.3 Identifikasi Aktor.....	22
4.4 Kebutuhan Fungsional Dan <i>Non</i> Fungsional.....	22
4.5 <i>Use Case</i> diagram.....	24
4.6 <i>Use Case Scenario</i>	26
4.6.1 <i>Use case Scenario</i> Menampilkan Soal.....	26
4.6.2 <i>Use case Scenario</i> Mengeluarkan Suara Soal.....	27
4.6.3 <i>Use case Scenario</i> Menjawab Soal	27
4.6.4 <i>Use case Scenario</i> Menilai Jawaban.....	28
4.6.5 <i>Use case Scenario</i> Menampilkan Jumlah Kata Benar	28
4.6.6 <i>Use case Scenario</i> Manampilkan Sisa Waktu.....	29
4.6.7 <i>Use case Scenario</i> Hitung Total Poin.....	30
4.6.8 <i>Use case Scenario</i> Menyimpan <i>Top score</i> Baru	30
4.6.9 <i>Use case Scenario</i> Menampilkan <i>Top score</i>	31
4.6.10 <i>Use case Scenario</i> Menampilkan Tugas Harian.....	32
4.6.11 <i>Use case Scenario</i> Memperbarui tugas harian	32
4.6.12 <i>Use case Scenario</i> Menampilkan Kategori Belajar.....	33
4.6.13 <i>Use case Scenario</i> Menampilkan Kelas Tantangan	33
4.7 Perancangan	34
4.7.1 <i>Activity</i> diagram	34
4.7.2 Diagram Kelas.....	38
4.7.3 Desain <i>Database</i>	39
4.7.4 <i>Screenflow</i>	40
4.7.5 Perancangan Antarmuka.....	42

4.8 Implementasi <i>Prototype</i>	53
4.8.1 Halaman <i>Splashscreen</i>	53
4.8.2 Halaman Utama	54
4.8.3 Halaman Pilih Kategori	54
4.8.4 Halaman Soal Belajar	55
4.8.5 Halaman <i>Popup</i> Hasil Jawab Soal dan Penilaian Belajar	55
4.8.6 Halaman Pilih Kelas Tantangan	56
4.8.7 Halaman <i>Popup</i> Mulai Tantangan	56
4.8.8 Halaman <i>Popup Top score</i>	57
4.8.9 Halaman Soal Tantangan	57
4.8.10 Halaman <i>Popup</i> Hasil Jawab Soal Tantangan	58
4.8.11 Halaman <i>Popup</i> Total Poin	58
4.9 Evaluasi <i>Prototype</i>	59
BAB 5 IMPLEMENTASI	60
5.1 Spesifikasi Sistem	60
5.1.1 Spesifikasi Perangkat Keras	60
5.1.2 Spesifikasi Perangkat Lunak	61
5.2 Batasan-batasan Implementasi	61
5.3 Implementasi Kelas dan Antarmuka	61
5.3.1 Implementasi Kelas	62
5.3.2 Implementasi Antarmuka	62
5.4 Implementasi Kode Program	69
5.4.1 Implementasi Kode Program Membuat Atau <i>Update</i> Tugas Harian	70
5.4.2 Implementasi Kode Program Membuat dan Menampilkan Soal Tantangan	73
5.4.3 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban	78
5.4.4 Implementasi Kode Program Menampilkan Total Poin Tantangan	81
5.4.5 Implementasi Kode Program Menampilkan <i>Top score</i>	83
BAB 6 PENGUJIAN DAN ANALISIS	85
6.1 Pengujian	85



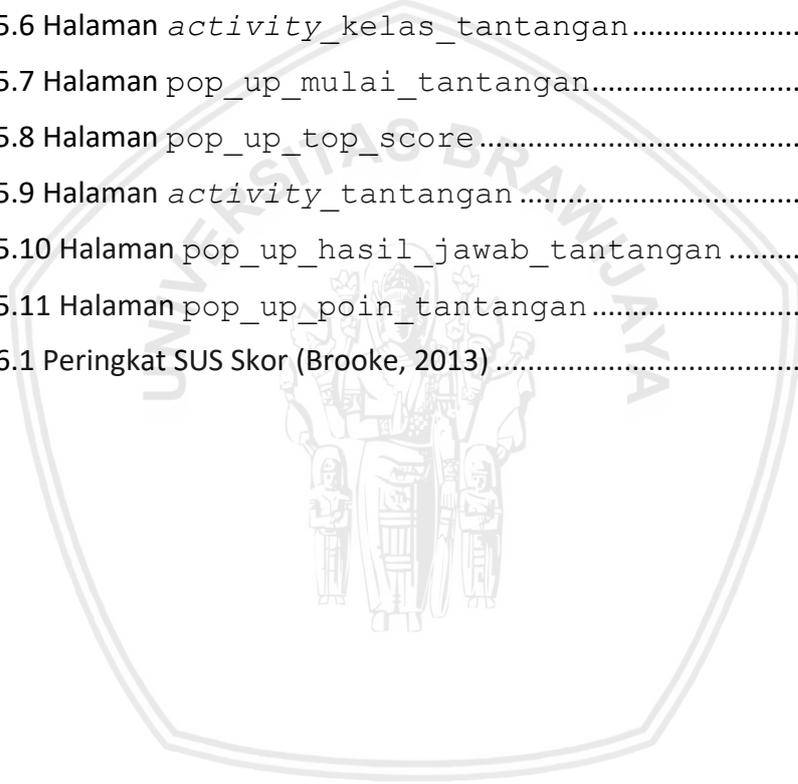
6.1.1 Pengujian <i>Black box Testing</i> Dengan Teknik <i>Requirement validation Testing</i>	85
6.1.2 Pengujian <i>Usability</i>	99
6.1.3 Pengujian Akurasi Implementasi <i>Google Speech Recognition</i> ..	100
6.2 Analisis	101
6.2.1 Analisis Hasil Pengujian <i>Requirement Validation Testing</i>	101
6.2.2 Analisis Hasil Pengujian <i>Usability</i>	101
6.2.3 Analisis Hasil Pegujian Akurasi Implementasi <i>Google Speech Recognition</i>	102
BAB 7 PENUTUP	103
7.1 Kesimpulan.....	103
7.2 Saran	104
DAFTAR REFERENSI	105
LAMPIRAN	107



DAFTAR GAMBAR

Gambar 2.1 Siklus <i>Prototyping</i> (Pressman, 2001).....	10
Gambar 3.1 Diagram Alir Metodologi.....	14
Gambar 3.2 Diagram Alir Analisis Kebutuhan dan Perancangan.....	15
Gambar 4.1 Gambaran Umum Sistem.....	19
Gambar 4.2 <i>Use case</i> Diagram.....	25
Gambar 4.3 <i>Activity</i> Diagram Belajar.....	35
Gambar 4.4 <i>Activity</i> Diagram Tantangan.....	36
Gambar 4.5 <i>Activity</i> Diagram Tugas Harian.....	37
Gambar 4.6 <i>Class</i> Diagram.....	38
Gambar 4.7 Desain <i>Database</i>	39
Gambar 4.8 <i>Screenflow</i>	41
Gambar 4.9 Halaman <i>Splashscreen</i>	42
Gambar 4.10 Tampilan Menu utama.....	43
Gambar 4.11 Tampilan Halaman Pilih Kategori Kata.....	44
Gambar 4.12 Tampilan Halaman Soal Belajar.....	45
Gambar 4.13 Tampilan Halaman Hasil Jawab Soal dan Penilaian Belajar.....	46
Gambar 4.14 Tampilan Halaman Pilih Kelas Tantangan.....	47
Gambar 4.15 Tampilan Halaman Mulai Tantangan.....	48
Gambar 4.16 Tampilan Halaman <i>Pop up Top score</i>	49
Gambar 4.17 Tampilan Halaman Soal Tantangan.....	50
Gambar 4.18 Tampilan Halaman Hasil Jawab Soal dan Penilaian Tantangan.....	51
Gambar 4.19 Tampilan Halaman Total Poin.....	52
Gambar 4.20 Halaman <i>Splashscreen</i>	53
Gambar 4.21 Tampilan Menu.....	54
Gambar 4.22 Tampilan Halaman Pilih Kategori Kata.....	54
Gambar 4.23 Tampilan Halaman Soal Belajar.....	55
Gambar 4.24 Tampilan Halaman Hasil Jawab Soal dan Penilaian Belajar.....	55
Gambar 4.25 Tampilan Halaman Pilih Kelas Tantangan.....	56
Gambar 4.26 Tampilan Halaman Mulai Tantangan.....	56
Gambar 4.27 Tampilan Halaman Mulai Tantangan.....	57

Gambar 4.28 Tampilan Halaman Soal Tantangan.....	57
Gambar 4.29 Tampilan Halaman Hasil Jawab Soal dan Penilaian Tantangan	58
Gambar 4.30 Tampilan Halaman Total Poin	58
Gambar 5.1 Halaman <i>activity_splash</i>	63
Gambar 5.2 Halaman <i>activity_main</i>	64
Gambar 5.3 Halaman <i>activity_categori_belajar</i>	65
Gambar 5.4 Halaman <i>activity_belajar</i>	65
Gambar 5.5 Halaman <i>pop_up_hasil_jawab_belajar</i>	66
Gambar 5.6 Halaman <i>activity_kelas_tantangan</i>	66
Gambar 5.7 Halaman <i>pop_up_mulai_tantangan</i>	67
Gambar 5.8 Halaman <i>pop_up_top_score</i>	67
Gambar 5.9 Halaman <i>activity_tantangan</i>	68
Gambar 5.10 Halaman <i>pop_up_hasil_jawab_tantangan</i>	68
Gambar 5.11 Halaman <i>pop_up_poin_tantangan</i>	69
Gambar 6.1 Peringkat SUS Skor (Brooke, 2013)	101



DAFTAR TABEL

Tabel 4.2 Identifikasi Aktor	22
Tabel 4.3 Kebutuhan Fungsional.....	23
Tabel 4.3 Kebutuhan Fungsional (lanjutan)	24
Tabel 4.4 Kebutuhan <i>Non</i> Fungsional	24
Tabel 4.5 <i>Use case Scenario</i> Menampilkan Soal	26
Tabel 4.6 <i>Use case Scenario</i> Mengeluarkan Suara Soal	27
Tabel 4.7 <i>Use case Scenario</i> Menjawab Soal	27
Tabel 4.7 <i>Use case Scenario</i> Menjawab Soal (lanjutan).....	28
Tabel 4.8 <i>Use case Scenario</i> Menilai Jawaban	28
Tabel 4.9 <i>Use case Scenario</i> Menampilkan Jumlah Kata Benar.....	28
Tabel 4.9 <i>Use case Scenario</i> Menampilkan Jumlah Kata Benar (lanjutan)	29
Tabel 4.10 <i>Use case Scenario</i> Menampilkan Sisa Waktu	29
Tabel 4.11 <i>Use case Scenario</i> Hitung Total Poin	30
Tabel 4.12 <i>Use case Scenario</i> Menyimpan <i>Top score</i> Baru.....	30
Tabel 4.12 <i>Use case Scenario</i> Menyimpan <i>Top score</i> Baru (lanjutan)	31
Tabel 4.13 <i>Use case Scenario</i> Menampilkan <i>Top score</i>	31
Tabel 4.14 <i>Use case Scenario</i> Menampilkan Tugas Harian.....	32
Tabel 4.15 <i>Use case Scenario</i> Memperbarui tugas harian.....	32
Tabel 4.15 <i>Use case Scenario</i> Memperbarui tugas harian (lanjutan)	33
Tabel 4.16 <i>Use case Scenario</i> Menampilkan Kategori Belajar	33
Tabel 4.17 <i>Use case Scenario</i> Menampilkan Kelas Tantangan	33
Tabel 4.17 <i>Use case Scenario</i> Menampilkan Kelas Tantangan (lanjutan).....	34
Tabel 4.18 Penjelasan Halaman Menu.....	42
Tabel 4.19 Penjelasan Halaman Menu utama	43
Tabel 4.20 Penjelasan Halaman Pilih Kategori Kata.....	44
Tabel 4.21 Penjelasan Halaman Soal Belajar	45
Tabel 4.22 Penjelasan Halaman Jawab Soal dan Penilaian Belajar	46
Tabel 4.23 Penjelasan Halaman Kelas Tantangan.....	47
Tabel 4.24 Penjelasan Halaman Mulai Tantangan.....	48
Tabel 4.25 Penjelasan Halaman <i>Popup Top score</i>	49



Tabel 4.26 Penjelasan Halaman Soal Tantangan	50
Tabel 4.27 Penjelasan Halaman Jawab Soal dan Penilaian Tantangan.....	51
Tabel 4.28 Penjelasan Halaman Total Poin	52
Tabel 5.1 Spesifikasi Perangkat Keras Laptop	60
Tabel 5.2 Spesifikasi Perangkat Keras <i>Smartphone</i>	60
Tabel 5.3 Spesifikasi Perangkat Lunak Laptop	61
Tabel 5.3 Spesifikasi Perangkat Lunak <i>Smartphone</i>	61
Tabel 5.4 Implementasi Kelas	62
Tabel 5.5 Implementasi Antarmuka	63
Tabel 5.6 Implementasi Kode Program Tugas Harian Pada Kelas <i>MainActivity</i>	70
Tabel 5.6 Implementasi Kode Program Tugas Harian Pada Kelas <i>MainActivity</i> (lanjutan)	71
Tabel 5.7 Implementasi Tugas Harian Pada Kelas <i>myDatabaseHandler</i>	71
Tabel 5.7 Implementasi Tugas Harian Pada Kelas <i>myDatabaseHandler</i> (lanjutan)	72
Tabel 5.8 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas <i>kelasTantanganActivity</i>	73
Tabel 5.8 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas <i>kelasTantanganActivity</i> (lanjutan).....	74
Tabel 5.9 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas <i>myDatabaseHandler</i>	75
Tabel 5.9 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas <i>myDatabaseHandler</i> (lanjutan).....	76
Tabel 5.10 Implementasi Kode Program Menampilkan Soal Tantangan Pada Kelas <i>tantanganActivity</i>	76
Tabel 5.10 Implementasi Kode Program Menampilkan Soal Tantangan Pada Kelas <i>tantanganActivity</i> (lanjutan).....	77
Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas <i>belajarActivity</i>	78
Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas <i>belajarActivity</i> (lanjutan)	79
Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas <i>belajarActivity</i> (lanjutan)	80

Tabel 5.12 Implementasi Kode Program Menampilkan Total Poin Tantangan Pada Kelas <i>tantanganActivity</i>	81
Tabel 5.12 Implementasi Kode Program Menampilkan Total Poin Tantangan Pada Kelas <i>tantanganActivity</i> (lanjutan)	82
5.13 Implementasi Kode Program Menampilkan <i>Top score</i> Pada Kelas kelas <i>TantanganActivity</i>	83
5.13 Implementasi Kode Program Menampilkan <i>Top score</i> Pada Kelas kelas <i>TantanganActivity</i> (lanjutan).....	84
Tabel 5.14 Implementasi Kode Program Mengambil Data <i>Top score</i> Pada Kelas <i>myDatabaseHandler</i>	84
Tabel 6.1 Kasus Uji Menampilkan Soal	85
Tabel 6.1 Kasus Uji Menampilkan Soal (lanjutan).....	86
Tabel 6.2 Kasus Uji Mengeluarkan suara Soal	86
Tabel 6.2 Kasus Uji Mengeluarkan suara Soal (lanjutan).....	87
Tabel 6.3 Kasus Uji Menjawab Soal Dan Menampilkan Hasil Jawaban	87
Tabel 6.3 Kasus Uji Menjawab Soal Dan Menampilkan Hasil Jawaban (lanjutan)	88
Tabel 6.4 Kasus Uji Menampilkan Jumlah Kata Benar	88
Tabel 6.4 Kasus Uji Menampilkan Jumlah Kata Benar (lanjutan)	89
Tabel 6.5 Kasus Uji Menampilkan Sisa Waktu Tantangan	90
Tabel 6.6 Kasus Uji Hitung Total Poin Tantangan	90
Tabel 6.6 Kasus Uji Hitung Total Poin Tantangan (lanjutan).....	91
Tabel 6.7 Kasus Uji Menyimpan <i>Top score</i> baru dan Menampilkan <i>Top score</i>	91
Tabel 6.7 Kasus Uji Menyimpan <i>Top score</i> baru dan Menampilkan <i>Top score</i> (lanjutan).....	92
Tabel 6.8 Kasus Uji Menampilkan Tugas Harian Dan Memperbarui Tugas Harian	92
Tabel 6.9 Kasus Uji Menampilkan Kategori Belajar Dan Kelas Tantangan.	93
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i>	93
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i> (lanjutan).....	94
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i> (lanjutan).....	95
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i> (lanjutan).....	96
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i> (lanjutan).....	97
Tabel 6.10 Hasil Pengujian <i>Requirement validation testing</i> (lanjutan).....	98
Tabel 6.11 SUS Kuesioner.....	99

Tabel 6.12 Hasil Rekapitulasi dan Perhitungan Skor SUS Kuesioner 100
Tabel 6.13 Hasil Pengujian 101



DAFTAR LAMIRAN

Lampiran 1 Hasil Pengujian Akurasi Implementasi Google *Speech* Recognition
.....107



BAB 1 PENDAHULUAN

1.1 Latar belakang

Saat ini telah masuk era liberalisasi kawasan ASEAN melalui pembentukan ASEAN *Economic Community* (AEC) atau Masyarakat Ekonomi ASEAN (MEA) yang telah berlaku pada bulan Januari tahun 2016. Pertanyaannya adalah apakah kita sudah siap menghadapi MEA dan bagaimana daya saing kita dibanding dengan negara-negara kawasan ASEAN lainnya. Dengan terbentuknya MEA, masyarakat Indonesia dipaksa untuk bersaing dengan masyarakat Negara-negara Asia Tenggara lainnya. Sebagai salah satu hal penting untuk bersaing yaitu bagaimana cara kita berkomunikasi dengan orang-orang dari negara ASEAN lainnya, oleh karena itu pemahaman Bahasa Internasional akan semakin dibutuhkan.

Bahasa *Inggris* yakni salah satu bahasa internasional yang telah banyak digunakan dan dipelajari oleh banyak orang, oleh karena bahasa *Inggris* dianggap penting, tetapi dalam kehidupan yang sebenarnya belajar bahasa *Inggris* secara pendidikan formal atau pendidikan *non* formal memerlukan waktu atau biaya yang tidak sedikit sehingga membuat masyarakat yang ingin belajar bahasa *Inggris* kurang tertarik untuk melakukannya. Telah banyak aplikasi yang dibuat untuk aplikasi belajar bahasa *Inggris*, seperti aplikasi yang memungkinkan pengguna dalam melakukan percakapan Bahasa *Inggris* antara pengguna dengan aplikasi android seperti pada penelitian yang berjudul “Aplikasi *English teacher* Sebagai Alat Bantu Belajar *English Conversation* Berbasis Android dengan Menerapkan *Voice Recognition*” yang disusun oleh Jefrizal, Jaroji dan Agus Tedyana pada tahun 2017.

Dari hasil pra penelitian yang telah dilakukan oleh peneliti yang bertujuan untuk mendapatkan data pendukung mengapa pengembangan aplikasi aplikasi belajar pengucapan bahasa *Inggris* diperlukan. Pra penelitian yang dilakukan yaitu dengan wawancara lima responden diantaranya dua responden memiliki latar belakang sebagai tenaga pengajar pada salah satu lembaga kursus bahasa *Inggris* yang berlokasi di kota Malang dan tiga responden merupakan tenaga pengajar pada forum komunitas bahasa *Inggris* Universitas Brawijaya. Dari hasil penelitian menunjukkan data, dari lima responden menyatakan bahwa saat ini masyarakat pada umumnya telah banyak mengetahui kosakata bahasa *Inggris* yang didapatkan dari berbagai sumber baik dari sekolah *formal* dan tidak *formal* ataupun dari media cetak dan elektronik namun pengetahuan kosakata bahasa *Inggris* yang masyarakat ketahui kebanyakan masih salah pada cara pengucapan yang benar, hal itu dikarenakan kurangnya latihan pada kehidupan sehari-hari dan pendidikan mengenai cara pengucapan yang benar selain di sekolah *formal* masih sedikit, pernyataan ini berdasarkan kesimpulan yang didapatkan dari jawaban responden yang jawaban diberikan berdasarkan dari pengalaman-pengalaman responden selama menjadi tenaga pengajar atau bergerak pada dibidang pendidikan bahasa *Inggris*.

Teknologi perangkat bergerak sudah dikenal masyarakat banyak dan berpengaruh banyak bagi kehidupan. Saat ini dunia teknologi mengalami perkembangan yang sangat pesat dan memberikan sumbangsih yang sangat besar bagi umat manusia, baik di bidang pertanian, peternakan, kedokteran, dan transportasi. Saat ini perusahaan-perusahaan yang bergerak di dunia teknologi berlomba-lomba membuat inovasi baru dan pelayanan yang lebih baik. Salah satu contoh perkembangan teknologi ini dapat dilihat dari perangkat teknologi yang dulunya memiliki ukuran yang lumayan besar, saat ini mulai bertambah kecil dan lebih praktis. Begitu pula dengan pelayanan di dunia teknologi semakin praktis dan mudah dalam menggunakan layanan. Perkembangan teknologi telah hadir di dunia pendidikan yang sejalan dengan kesempatan untuk berlatih dan belajar. Kehadirannya dalam dunia pendidikan tepat waktu dilihat sifat dari kebutuhan belajar saat ini lebih luas, akses yang cepat ke materi belajar dan kebutuhan yang terus menerus untuk komunikasi.

Voice Recognition adalah suatu sistem yang dapat mengidentifikasi suara. Pada *smartphone* android terdapat sebuah aplikasi yang dikenal dengan *Automatic Speech Recognition* yang berfungsi untuk memanfaatkan suara sebagai alat masukan dalam melakukan suatu proses (Jefrizal, Jaroji dan Tedyyana, 2017). *Automatic Speech Recognition* ada dua macam yakni *Speech to text* yang mengubah dari suara menjadi bentuk teks dan *Text to speech* mengubah dari bentuk teks menjadi suara. Aplikasi *speech recognition* dapat digunakan untuk melakukan suatu proses dalam *device* android.

Prototyping merupakan proses pengembangan *software* yang dilakukan secara berulang dengan menampilkan gambaran lengkap (*prototype*) dan menguji langsung kepada *user*. Metode *prototyping* biasanya digunakan karena *user* dapat memberikan informasi umum mengenai sistem yang akan dibangun, dengan metode ini *developer* dapat menerima masukan dari *user* untuk pengembangan aplikasi sesuai kebutuhan (Salma, Darmawan dan Al-Anshary, 2017). Dengan menggunakan pendekatan metode ini maka aplikasi yang dihasilkan akan sesuai dengan kebutuhan *user* dan metode ini juga dapat mempermudah pihak pengembangan untuk menemukan kekurangan pada aplikasi yang akan dikembangkan serta dapat mengurangi perubahan atau perbaikan yang perlu dilakukan setelah melakukan pengujian.

Saat ini aplikasi belajar bahasa Inggris sudah banyak tersedia namun aplikasi belajar bahasa Inggris pada saat ini yang mendominasi hanya aplikasi terjemahan sedangkan untuk belajar pengucapan bahasa Inggris masih sangat kurang dan belum ada aplikasi yang berfokus pada pengucapan kosakata bahasa Inggris yang memiliki cara belajar dan berlatih seperti bermain permainan. Sesuai dengan harapan untuk memberi kontribusi khususnya di dunia pendidikan dan masyarakat banyak untuk dapat belajar bahasa *Inggris* terutama keinginan untuk memberikan alternatif bagi pengguna untuk dapat belajar pengucapan kosakata bahasa *Inggris* yang benar, oleh karena itu muncul sebuah ide untuk membuat sebuah aplikasi yang menggunakan implementasi *Voice Recognition* berbasis android *smartphone* sehingga mudah digunakan kapanpun dan dimanapun untuk

memudahkan pengguna dalam belajar bahasa *Inggris* terutama dalam hal pengucapan kosakata Bahasa *Inggris* yang sesuai dengan kebutuhan pengguna aplikasi. Karena salah satu tujuan dikembangkannya aplikasi edukasi berbasis android sebagai sarana belajar pengucapan kosakata bahasa *Inggris* yang sesuai kebutuhan pengguna aplikasi maka diharapkan dengan metode *prototyping* ini dapat membantu pihak pengembangan untuk menghasilkan aplikasi yang sesuai kebutuhan pengguna untuk belajar pengucapan kosakata bahasa Inggris.

1.2 Rumusan masalah

Berdasarkan dari latar belakang, adapun rumusan masalah pada penelitian ini adalah:

1. Bagaimana rancangan aplikasi edukasi berbasis android sebagai sarana belajar pengucapan kosakata bahasa Inggris dengan menggunakan metode *prototyping*?
2. Bagaimana implementasi *Voice Recognition* pada *google speech API* pada aplikasi belajar pengucapan kosakata Bahasa Inggris?
3. Bagaimana hasil dari pengujian akurasi *google speech API*, *requirement validation* dan *usability* terhadap aplikasi android belajar pengucapan kosakata bahasa Inggris?

1.3 Tujuan

Adapun tujuan pada penelitian ini, yaitu:

1. Dapat merancang aplikasi edukasi berbasis android sebagai sarana belajar pengucapan kosakata bahasa *Inggris* dengan menggunakan metode *prototyping*.
2. Dapat mengimplementasikan *Voice Recognition* pada *google speech API* pada aplikasi belajar pengucapan kosakata Bahasa Inggris.
3. Dapat hasil dari pengujian akurasi *google speech API*, *requirement validation* dan *usability* terhadap aplikasi android belajar pengucapan kosakata bahasa Inggris.

1.4 Manfaat

Manfaat yang dapat diperoleh oleh adanya aplikasi ini akan menambah pilihan masyarakat dalam memilih sarana belajar pengucapan bahasa *Inggris* terutama dalam hal pengucapan dan menyediakan sarana belajar praktis dan menghibur.

1.5 Batasan masalah

Pada penelitian ini terdapat beberapa batasan masalah, yaitu:

1. Aplikasi hanya dapat berjalan pada *smartphone* dengan sistem operasi android.
2. Aplikasi ini hanya berfokus pada kosakata bahasa *Inggris* dan terjemahannya dalam bahasa Indonesia.
3. Aplikasi memanfaatkan teknologi *voice recognition* yaitu *google speech API*.
4. Pengujian menggunakan *black-box testing* terhadap aplikasi android dengan metode *Requirement validation testing*, pengujian *usability* dan pengujian akurasi implementasi *google speech recognition*.
5. Metode pengembangan yang digunakan yaitu metode *Prototyping*.

1.6 Sistematika Pembahasan

Bagian ini bertujuan untuk menunjukkan gambaran beserta uraian dari penelitian ini berdasarkan pokok-pokok bahasan yang terdapat pada beberapa bab, yaitu:

BAB 1 PENDAHULUAN

Bagian ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB 2 TINJAUAN PUSTAKA

Bagian menjelaskan dasar teori dan referensi yang mendasari penelitian yang berasal dari beberapa literatur yaitu seperti artikel, jurnal ataupun buku yang dapat memperkuat atau memperjelas penulisan penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Bagian ini menjelaskan mengenai tahapan metodologi untuk dilakukan pada penelitian pengembangan aplikasi sebagai sarana belajar pengucapan kosakata bahasa *Inggris*.

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Bagian ini menjelaskan mengenai tahapan menggali dan analisis kebutuhan dan perancangan aplikasi android belajar pengucapan kosakata bahasa *Inggris*.

BAB 5 IMPLEMENTASI

Bagian ini berisi mengenai tahapan implementasi aplikasi, yang mencakup teknik dan penjelasan mengenai implementasi aplikasi android belajar pengucapan kosakata Bahasa *Inggris*.

BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini menjelaskan mengenai pengujian hasil implementasi aplikasi dan analisis hasil pengujian pada aplikasi android belajar pengucapan kosakata

Bahasa *Inggris*. Pengujian pada hasil implementasi aplikasi ini terdapat tiga pengujian yang akan dilakukan.

BAB 7 PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian skripsi yang telah dilakukan dan saran-saran untuk penelitian atau pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Bagian ini menerangkan pendapat singkat dari materi terkait dengan penelitian yang berfungsi menjadi landasan dalam pengembangan aplikasi android belajar pengucapan kosakata bahasa *Inggris*.

2.1 Kajian Pustaka

Bagian ini membahas penelitian yang sudah ada dan berhubungan atau memiliki topik yang serupa dengan penelitian pada skripsi ini. Penelitian yang pertama ialah penelitian oleh Supriyanta, Pudji Widodo dan Bekti Maryuni Susanto dalam penelitian yang berjudul “Aplikasi Konversi Suara Ke Teks Berbasis Android Menggunakan *Google speech API*”, google memasang fitur pada android yang memiliki tujuan untuk menjadikan suara sebagai salah satu alternatif metode *input*, seperti penelusuran dengan menggunakan suara dan *input* suara untuk mengisi *text field* tertentu. Penelitian ini memiliki tujuan untuk membangun aplikasi yang dapat mengkonversi suara ke dalam bentuk teks dengan bahasa Indonesia. Aplikasi dibuat dengan implementasi *API Google speech*, bahasa pemrograman yang digunakan yaitu bahasa pemrograman Java. Aplikasi dibuat dengan berbasis *smartphone* Android. Manfaat dari aplikasi ini diharapkan dapat mengurangi kesalahan saat memasukkan *input* pada kolom teks.

Penelitian kedua ialah yang dilakukan oleh Jefrizal, Jaroji, dan Agus Tedyana pada tahun 2017 dalam penelitiannya yang berjudul “Aplikasi *English teacher* Sebagai Alat Bantu Belajar English Conversation Berbasis Android dengan Menerapkan *Voice Recognition*”, dengan aplikasi belajar bahasa Inggris yang telah banyak tersedia dapat memungkinkan orang-orang untuk belajar bahasa *Inggris* secara digital dengan mudah dimanapun dan kapanpun. Belajar bahasa *Inggris* dengan cara manual atau dengan cara melakukan pengetikan lebih rentan akan kesalahan dalam melakukan penulisan bahasa Inggris. Pada penelitian ini diusulkan sebuah aplikasi *english teacher* sebagai aplikasi belajar *english conversation* dengan berbasis android yang menerapkan *voice recognition*, sehingga dapat mengurangi kesalahan pada saat belajar bahasa *Inggris*. Aplikasi ini menerapkan *voice recognition text to speech* dan *speech to text*. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman java, *database* MySQL dan *API google text to speech*. Dilakukan pengujian dengan sampel 10 orang dengan hasil tingkat akurasi pada setiap percobaan adalah 95% tergantung pada pengucapan dari pengguna.

2.2 Aplikasi Belajar Pengucapan Kosakata Bahasa *Inggris*

Bahasa *Inggris* adalah salah satu bahasa internasional yang sudah banyak digunakan dan dipelajari oleh banyak orang, oleh karena itu bahasa *Inggris* dianggap cukup penting. Belajar bahasa *Inggris* secara formal atau dengan *non formal* memerlukan cukup banyak waktu atau biaya sehingga membuat orang-orang yang ingin belajar bahasa *Inggris* kurang tertarik untuk belajar bahasa Inggris, oleh sebab itu dibuatlah sebuah aplikasi *english teacher* dengan

menerapkan *Voice Recognition* berbasis android *smartphone* untuk mendukung pengguna dalam belajar bahasa *Inggris* (Jefrizal, Jaroji dan Tedyyana, 2017). Pada saat ini kebutuhan aplikasi untuk belajar bahasa *Inggris* menjadi semakin meningkat. Hal ini mendorong diperlukannya aplikasi yang dapat digunakan sebagai sarana untuk belajar bahasa *Inggris* dengan baik. Aplikasi belajar bahasa *Inggris* sudah banyak tersedia akan tetapi masih sedikit aplikasi yang dapat melatih pengucapan bahasa *Inggris*. Oleh karena itu diperlukan sebuah aplikasi yang dapat melatih keterampilan berkomunikasi yang baik dan benar dalam bahasa *Inggris* terutama pada pengucapan bahasa *Inggris* yang benar. Pada sebuah aplikasi belajar pengucapan bahasa *Inggris* yang diperlukan adalah aplikasi tidak hanya memberitahu pengucapan yang benar akan tetapi dapat melatih pengucapan pengguna dalam bahasa *Inggris*.

2.3 Android

Android merupakan salah sistem operasi perangkat bergerak yang dibuat berdasarkan modifikasi versi sistem operasi linux. Android pada awalnya dikembangkan oleh perusahaan pengembang dengan nama Android. pada tahun 2005, sebagai bagian strategi saat memasuki pasar perangkat bergerak, perusahaan Google membeli Android dan mengambil alih pengembangannya sampai saat tulisan ini dibuat. Keuntungan yang utama ketika mengadopsi sistem operasi android yakni menawarkan pendekatan terpadu pada pengembangan aplikasi. Pengembang aplikasi hanya perlu mengembangkan aplikasi mereka untuk sistem operasi android, dan aplikasi mereka dapat dijalankan pada banyak perangkat bergerak dengan tipe yang berbeda, dengan syarat perangkat yang digunakan menggunakan android. Pada dunia *smartphone* aplikasi adalah bagian rantai sukses yang paling penting, oleh karena itu pabrikan *hardware* melihat android sebagai harapan terbaiknya untuk menantang serangan hebat iPhone, yang sudah memiliki dasar aplikasi yang besar (Supriyanta, Widodo dan Susanto, 2014).

Android merupakan sistem operasi bersifat *open source* yang berarti bebas tersedia bagi pabrikan untuk melakukan penyesuaian, oleh karena itu tidak ada konfigurasi *hardware* dan *software* yang tetap atau sama. Sistem operasi android terdapat fitur-fitur berikut:

- a) *Storage*, menggunakan SQLite sebuah *database* relasional yang ringan.
- b) Konektivitas, mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, *Bluetooth*, Wifi, LTE dan WiMAX
- c) *Messaging*, mendukung SMS dan MMS.
- d) *Web Browser*, berdasarkan *open source* WebKit.
- e) Aplikasi *Support*, mendukung berbagai tipe aplikasi, MPEG, MP4, 3GP, JPEG, PNG dan lain-lain.
- f) *Hardware support*, *accelerometer* sensor, kamera, digital kompas, *proximity* sensor, dan GPS.

- g) *Multitouch, multitasking*, mendukung *flash* dan *tethering*, berbagi koneksi internet.

Sistem operasi android dapat dibagi ke dalam empat lapisan, yakni kernel linux, *libraries* dan android *runtime*, *application framework* dan *application*. Kernel linux merupakan bagian dasar dari sistem operasi android, pada lapisan di atasnya adalah lapisan *libraries*. Pada lapisan *libraries* terdapat kode yang menyediakan fitur utama pada sistem operasi android, sebagai contoh *library* SQLite yang menyediakan dukungan berupa *database* untuk sebuah aplikasi dapat menggunakannya sebagai penyimpanan data pada *storage internal*. Pada lapisan yang sama terdapat juga *Android Runtime*, yang menyediakan perangkat librari inti yang dapat memungkinkan pengembang aplikasi untuk menggunakan bahasa pemrograman java. *Android runtime* termasuk dalam *dalvik virtual machine*, yang dapat memungkinkan aplikasi android untuk menjalankan prosesnya sendiri dengan menggunakan *instance dalvik virtual machine* (aplikasi android dikompilasi ke dalam *dalvik executable*). Lapisan di atasnya ialah *application framework*, yang menampilkan berbagai kemampuan yang terdapat pada sistem operasi android kepada pengembang aplikasi sehingga mereka dapat menggunakannya pada aplikasi mereka. Lapisan yang paling atas ialah aplikasi, lapisan ini dapat kita temukan pada aplikasi yang dijalankan bersama dengan perangkat android, seperti *contacts*, *browser* serta aplikasi yang di *download* dari *Play Store*.

Sejak tahun 2009, versi sistem operasi android diberi kode nama yang dinamai dengan nama-nama makanan. Versi dirilis sesuai dengan urutan alphabet yaitu, Cupcake (1.5), Donut (1.6), Eclair (2.0 - 2.1), Frovo (2.2 - 2.2.3), Gingerbread (2.3 - 2.3.7), Honeycomb (3.0 - 3.2.6), Ice Cream Sandwich (4.0 – 4.0.4), Jelly Bean (4.1 – 4.3), Kitkat (4.4+), Lollipop (5.0+), Marshmallow(6.0+), Nougat (7.0+), Oreo (8.0+) dan selanjutnya versi terbaru adalah android Pie(9.0).

2.4 Voice Recognition

Voice Recognition adalah suatu sistem yang dapat mengidentifikasi suara dari seseorang. Pada perangkat android terdapat sistem yang dikenal dengan *automatic apeech recognition* yang memiliki fungsi untuk memanfaatkan suara sebagai alat masukan untuk melakukan suatu proses. *Automatic speech recognition* terdiri dari dua jenis yaitu *Speech to text* yang mengubah suara menjadi teks dan *Text to speech* mengubah teks menjadi suara. Penggunaan *automatic speech recognition* dapat dimanfaatkan untuk melakukan proses pencarian kata, pencarian letak suatu lokasi dan melakukan suatu perintah dalam *device* android (Jefrizal, Jaroji dan Agus Tedyyana, 2017).

Speech recognition sistem dapat dibagi menjadi beberapa bagian, yaitu *feature extraction*, *acoustic model database* yang dibangun berdasarkan data *training*, *dictionary*, model bahasa dan algoritma pengenalan suara. Sinyal suara analog yang didapatkan di analisis oleh sistem dalam interval, periode ini biasanya 20 ms dikarenakan sinyal suara pada periode tersebut dianggap statis.

Speech recognition dapat dimanfaatkan untuk pengembangan aplikasi ini, sehingga aplikasi dapat dijadikan sebagai aplikasi yang dapat memberikan penilaian terhadap kata yang diucapkan oleh pengguna. Di samping itu pengguna juga dapat memperbaiki kemampuan verbal dalam mengucapkan kata-kata bahasa *Inggris*.

2.5 Google Speech API

Google speech API atau *google voice search* diluncurkan pada tahun 2008 di Amerika Serikat namun hanya untuk beberapa tipe *smartphone*. *Google speech API* merupakan sebuah *framework* yang tentunya dikembangkan oleh Google untuk mengenali suara, kemudian dapat diubah menjadi *string* (teks) dan memasukkannya ke dalam kolom halaman pencarian google untuk pencarian berdasarkan *input* suara. Pengenalan suara dilakukan pada *server* Google menggunakan algoritme *Hidden Markov Model* (HMM). Dengan kata lain *input* suara yang diterima oleh perangkat android akan dikirim ke *server* google, yang kemudian *server* google akan melakukan pengenalan dan mengubah suara menjadi bentuk teks menggunakan algoritme HMM. Hasil konversi suara menjadi teks kemudian dimasukkan dalam halaman pencarian google kemudian *server* google akan mengirimkan hasil pencariannya tersebut ke perangkat android (Reddy & Mahender, 2013 disitasi dalam Supriyanta, Widodo dan Susanto, 2014). Namun saat ini untuk perangkat android sudah tersedia *google speech API* yang tidak perlu terkoneksi langsung ke *server* google.

Salah satu fitur pada *google speech API* yang diimplementasikan untuk pengembangan aplikasi ini yaitu memasukkan suara ke perangkat android melalui ucapan dan mengonversi ucapan menjadi teks yang tersimpan di perangkat android. TTS (*Text to speech*) merupakan sistem dalam bidang teknologi bahasa, yang dapat mengonversi teks menjadi suara sesuai dengan pembacaan teks dalam bahasa yang digunakan.

2.6 Android Studio

Android Studio merupakan sumber daya pengembangan aplikasi *mobile* yang terpadu dan terintegrasi untuk pengembangan aplikasi android, berdasarkan *Intellij IDEA* (Developer, 2019). Selain dari editor kode *Intellij* dan merupakan alat pengembangan yang memiliki daya guna, editor android studio terdapat banyak fitur untuk meningkatkan produktivitas saat pengembang membuat aplikasi android, misalnya:

- a) *Instant run* yang dapat melakukan perubahan ke aplikasi yang berjalan tanpa membuat APK yang baru.
- b) Sistem versi yang berbasis *gradle* sehingga *fleksibel*.
- c) Cocok digunakan pada semua perangkat android.
- d) Kinerja *emulator* yang cepat dan terdapat banyak fitur.

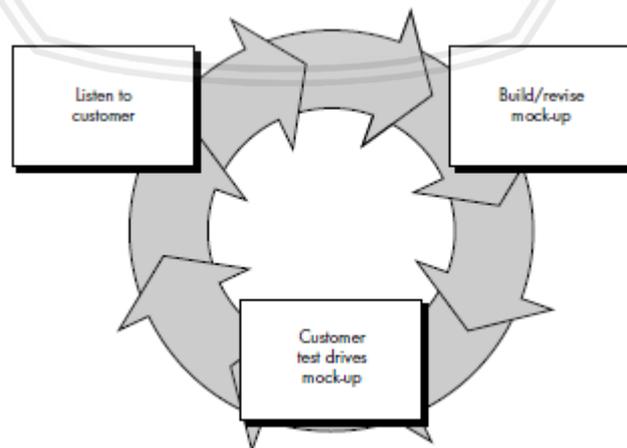
- e) *Template* kode dan dapat integrasi ke GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- f) Dukungan bahasa pemrograman C++ dan NDK.
- g) Alat untuk pengujian dan kerangka kerja yang ekstensif.
- h) Alat *lint* untuk meningkatkan kegunaan, kinerja, kompatibilitas versi, dan masalah-masalah lain.
- i) Terdapat banyak dukungan bawaan terutama untuk Google Cloud Platform yang mempermudah pengintegrasian Google Cloud Messaging dan App Engine.

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan *file* kode sumber dan *file* sumber daya. Jenis-jenis modul mencakup: modul aplikasi android, modul pustaka dan modul google app engine.

2.7 Metode Pengembangan *Prototyping*

Prototyping merupakan proses pengembangan *software* yang dilakukan secara berulang dengan menampilkan gambaran lengkap dan menguji langsung kepada calon *user*. Metode *prototyping* sering digunakan karena calon *user* hanya memberikan informasi umum mengenai sistem yang akan dibangun. Dengan metode ini *developer* dapat menerima masukan dari calon *user* untuk pengembangan aplikasi sesuai kebutuhan (Salma, Darmawan dan Al-Anshary, 2017).

Dalam metode *prototyping*, terdiri dari beberapa siklus, pada siklus akhir tim pengembang menghasilkan suatu *prototype* yang akan diujikan ke responden atau calon *user*. Kemudian responden akan mengevaluasi kekurangan *prototype* tersebut. Hasil evaluasi ini akan di analisis kembali oleh tim pengembang dan kemudian menghasilkan *prototype* yang baru. Demikian siklus ini akan berlangsung terus sampai didapat sistem yang sesuai dengan keinginan calon *user*.



Gambar 2.1 Siklus *Prototyping* (Pressman, 2001)

Dalam Gambar 2.1 tahapan awal dalam model ini adalah perencanaan yang dilakukan secara cepat kemudian dilanjutkan dengan perancangan. Setelah

tahapan perancangan dilakukan, kemudian dilanjutkan dengan melakukan pembuatan *prototype* aplikasi. Setelah itu *prototype* akan diberikan kepada *customer* untuk di-*review* dan diberikan umpan balik. Model *prototyping* memiliki keuntungan dalam hal komunikasi yang intens antara pengguna dan pengembang, membantu analis dalam menentukan kebutuhan pengguna yang sebenarnya dan meminimalkan kesalahan persepsi (Subhiyakto dan Utomo, 2017).

2.8 Pengujian *Black Box*

Pendekatan pengujian *Black box* adalah metode pengujian data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir (Perry, 1990 disitasi dalam Komarudin, 2016).

Pengujian *black box* lebih berfokus pada spesifikasi fungsional dan *non* fungsional perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program (Mustaqbal, Firdaus dan Rahmadi, 2015).

Black box testing cenderung untuk menemukan hal-hal berikut:

- a) Fungsi yang tidak benar atau tidak ada.
- b) Kesalahan antarmuka (*interface errors*).
- c) Kesalahan pada struktur data dan akses basis data.
- d) Kesalahan *performansi* (*performance errors*).
- e) Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

- a) Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
- b) *Input* seperti apa yang dapat menjadi bahan kasus uji yang baik?
- c) Apakah sistem sensitif pada *input-input* tertentu?
- d) Bagaimana sekumpulan data dapat diisolasi?
- e) Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
- f) Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metode atau teknik untuk melaksanakan *Black box testing*, antara lain:

- a) *Equivalence Partitioning*
- b) *Boundary Value Analysis/Limit Testing*
- c) *Comparison Testing*
- d) *Sample Testing*
- e) *Robustness Testing*

- f) *Behavior Testing*
- g) *Requirement validation Testing*
- h) *Performance Testing*
- i) Uji Ketahanan (*Endurance Testing*)
- j) Uji Sebab-Akibat (*Cause-Effect Relationship Testing*)
- k) *Usability testing*

Penelitian ini penulis menggunakan metode *requirement validation testing* yaitu pengujian yang untuk menguji apakah hasil implementasi aplikasi sudah sesuai dengan kebutuhan aplikasi yang didefinisikan pada proses analisis kebutuhan.

2.9 Requirement validation Testing

Proses pengujian berbasis *requirement* menangani dua masalah utama: pertama, melakukan validasi bahwa *requirement* benar, lengkap, tidak ambigu, dan konsisten secara logis; dan kedua, merancang serangkaian kasus uji yang diperlukan dan memadai dari *requirement* tersebut untuk memastikan bahwa desain dan kode sepenuhnya memenuhi *requirement* tersebut (Bender, 2009). Dalam merancang tes, dua masalah perlu diatasi: mengurangi sejumlah besar tes potensial menjadi ukuran yang masuk akal dan memastikan bahwa tes mendapatkan jawaban yang tepat untuk alasan yang tepat.

Strategi pengujian secara keseluruhan untuk mengintegrasikan pengujian di seluruh siklus hidup pengembangan dan fokus pada kualitas Spesifikasi *requirement*. Ini mengarah pada deteksi cacat dini yang telah terbukti jauh lebih murah daripada menemukan cacat selama aplikasi telah di distribusikan. Proses pengujian juga memiliki fokus pada pencegahan cacat, bukan hanya deteksi cacat.

2.10 Pengujian Usability

Pengujian *usability* adalah alat penelitian dalam metodologi eksperimen klasik yang memiliki rentang tes yang dapat dilakukan cukup banyak dari eksperimen klasik dengan ukuran sampel besar dan desain tes kompleks hingga studi kualitatif yang sangat informal dengan hanya satu peserta (Rubin dan Chisnell, 2008). Setiap pendekatan pengujian memiliki tujuan yang berbeda, serta waktu dan sumber daya yang berbeda. Aspek yang diukur dalam pengujian *usability* adalah aspek efektivitas dengan mengukur keberhasilan pengguna dalam menyelesaikan tugas yang diberikan. Aspek kedua yang diukur adalah aspek efisiensi dengan mengukur waktu yang diperlukan pengguna dalam mengerjakan tugas yang diberikan. Aspek ketiga yang diukur adalah aspek kepuasan atau satisfaction dengan mengukur tingkat kepuasan pengguna menggunakan kuesioner yang dapat merepresentasikan kepuasan pengguna terhadap suatu produk.

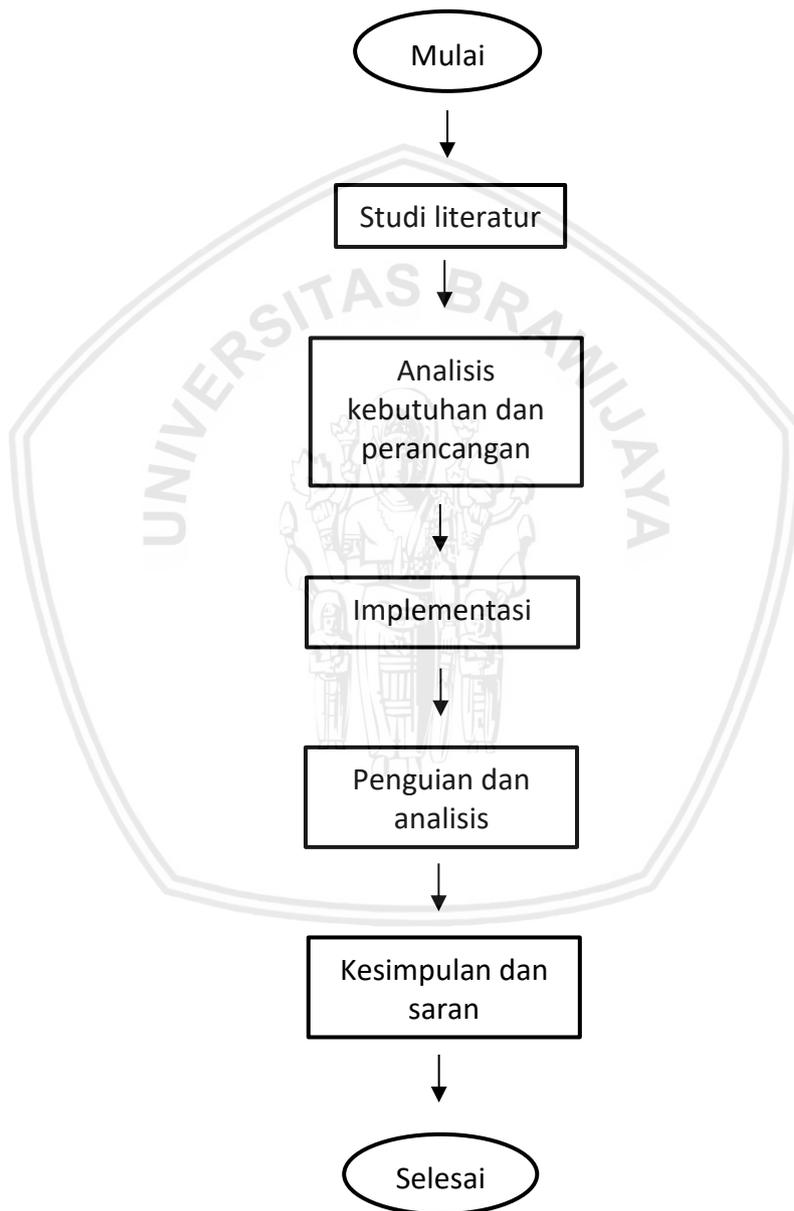
2.11 System Usability Scale

System usability scale (SUS) merupakan sebuah kuesioner yang dibuat oleh John Brooke pada tahun 1986. Pada Kuesioner SUS ini terdapat sepuluh buah pernyataan yang bertujuan untuk memberikan pandangan secara menyeluruh dan penilaian secara subjektif dari *usability* aplikasi yang diujikan. Tujuan dari kuesioner SUS untuk memberikan gambaran tentang persepsi subjektif pengguna terhadap tingkat *usability* sebuah produk, pengguna tidak perlu mengalokasikan banyak waktu dalam pengisian kuesioner ini karena kuesioner SUS dapat diisi dalam waktu yang singkat (Brooke, 2013). Selanjutnya, skala yang digunakan dalam kuesioner SUS adalah skala *likert* dengan penilaian 5 skala *likert* dimana skala 1 mengisyaratkan ketidaksetujuan yang kuat dan skala 5 menunjukkan persetujuan yang kuat (Brooke, 1986). Skor SUS dapat menunjukkan tingkat penerimaan pengguna pada aplikasi atau sistem berdasarkan tingkat *usability*. Menurut Brooke pada jurnalnya tahun 2013 sebuah sistem termasuk pada kategori “*Not Acceptable*” bila skor SUS berada pada *range* 0-50, Skor SUS pada *range* 51-70 termasuk kategori “*Marginal*”, sedangkan suatu sistem berada pada kategori “*Acceptable*” apabila memiliki skor SUS pada *range* 71-100.



BAB 3 METODOLOGI

Bagian ini menerangkan mengenai tahapan pada metodologi yang dilakukan pada penelitian skripsi pengembangan aplikasi android sebagai sarana belajar pengucapan kosakata bahasa *Inggris* menggunakan *google speech API*. Berikut dalam Gambar 3.1 merupakan diagram alir pada metodologi ini.

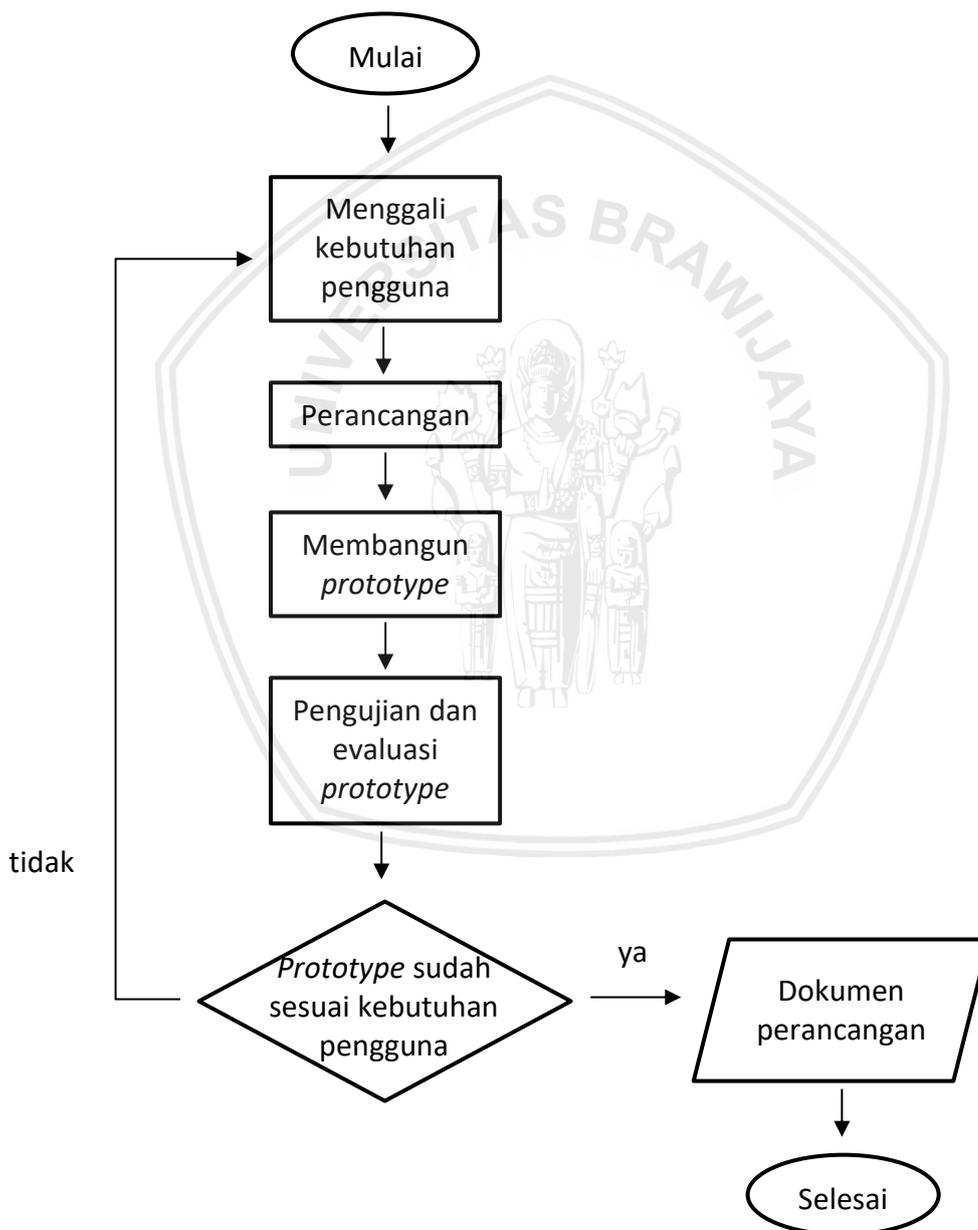


Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Literatur

Tahapan ini dilakukan pada penyusunan penelitian ini yang bertujuan untuk mengetahui teori dan metode dari penelitian sebelumnya, yang memiliki hubungan dengan penelitian yang akan dilakukan, yakni: studi literatur pengembangan aplikasi android, implementasi *voice recognition*, metode pengembangan *prototyping*, pengujian *black box* dan pengujian *usability*. Studi literatur ini didapatkan dari buku, skripsi dan jurnal tersedia.

3.2 Analisis Kebutuhan dan Perancangan



Gambar 3.2 Diagram Alir Analisis Kebutuhan dan Perancangan

Pada tahap ini terdapat siklus pengembangan metode *prototyping* yang disesuaikan dengan metodologi pada penelitian skripsi ini yang dapat dilihat dalam Gambar 3.2. Tahap pertama untuk mendapatkan gambaran kebutuhan apa saja yang diperlukan pada perangkat lunak yang akan dibangun yaitu tahap menggali dan analisis kebutuhan. Analisis kebutuhan pada sistem dilakukan dengan mengidentifikasi semua kebutuhan sistem dan juga aktor yang berinteraksi dengan sistem. Analisis kebutuhan pada aplikasi android sebagai sarana belajar pengucapan kosakata bahasa *Inggris* ini diuraikan menjadi beberapa bagian, yaitu: gambaran umum perangkat lunak, identifikasi aktor, analisis kebutuhan fungsional dan non fungsional, dan *use case diagram*.

Tahap menggali kebutuhan pada metode pengembangan *prototyping* melibatkan lima responden dalam proses penggalian kebutuhan untuk memperoleh kebutuhan fungsional beserta kebutuhan non fungsional, pemilihan responden berdasarkan kriteria tiga orang mahasiswa jurusan sastra *Inggris* dengan rentang usia 18 hingga 25 tahun dan dua orang tenaga pengajar bahasa *Inggris* dengan rentang usia 20 sampai 35 tahun. Kriteria ini dipilih berdasarkan kebutuhan aplikasi yaitu sebagai aplikasi belajar pengucapan kosakata bahasa *Inggris*.

Setelah kebutuhan aplikasi telah diperoleh maka yang dilakukan selanjutnya perancangan sistem. Tahap perancangan pada penelitian ini terdiri dari perancangan *activity diagram*, diagram kelas, desain *database*, *screenflow* dan tampilan antarmuka pengguna. Tahapan perancangan cepat merupakan tahapan proses untuk menghubungkan antara kebutuhan yang sebelumnya telah dilakukan menggunakan studi literatur terhadap aplikasi dengan tema yang sama yang diterjemahkan ke dalam diagram-diagram UML (*Unified Modeling Language*) diantaranya dalam bentuk *use case diagram*, *activity diagram* dan diagram kelas, membuat perancangan antarmuka tampilan aplikasi yang dilanjutkan pada tahapan implementasi *prototype* mencakup tahapan instalasi perangkat lunak pendukung dan kemudian tahapan evaluasi *prototype*.

Tahapan membangun *prototype* merupakan tahapan *file-file* atau data-data hasil dari tahapan pengumpulan kebutuhan perbaikan digabungkan menggunakan perangkat lunak pendukung yang menghasilkan gambaran dari penggunaan sistem. Tahapan evaluasi *prototype* merupakan tahapan pengujian *prototype* yang sudah dihasilkan dari tahapan membangun *prototype* yang pengujian melibatkan lima responden untuk melakukan penilaian terhadap *prototype*, responden merupakan lima orang yang sama yang menjadi responden pada tahap penggalian kebutuhan. Tahapan ini dilakukan dengan melakukan pengujian ke responden dan melakukan wawancara untuk mendapat *feedback*. Tahapan perbaikan *prototype* merupakan tahapan yang dilakukan perbaikan kesalahan atau kelemahan-kelemahan yang terdapat pada aplikasi yang telah dibangun. Tahapan perbaikan *prototype* dilakukan berdasarkan hasil pengujian *prototype* yang telah dilakukan sebelumnya oleh pengembang dan responden. Pada penulisan penelitian ini setiap iterasi yang terjadi tidak didokumentasikan

tetapi dokumentasi dari penelitian ini langsung hasil akhir dari iterasi yang sudah dilakukan.

3.3 Implementasi

Implementasi sistem mengacu pada hasil perancangan yang dibuat sebelumnya. Pada tahap implementasi sistem yang pertama dilakukan yaitu mendefinisikan spesifikasi sistem. Implementasi sistem dilakukan menggunakan editor android studio dengan Bahasa pemrograman java dan menerapkan/mengintegrasikan *Google speech API* ke aplikasi android untuk implementasi *Voice Recognition* sehingga aplikasi dapat mengubah suara *user* ke bentuk *text*. Kemudian pada tahapan yang terakhir pada implementasi yang dilakukan adalah implementasi/instalasi pada perangkat bergerak android yakni berupa *smartphone*.

3.4 Pengujian dan Analisis

Pengujian aplikasi dan analisis hasil pengujian dilakukan untuk mengetahui apakah aplikasi dapat berfungsi sesuai dengan perancangan yang telah dibuat. Pengujian pada aplikasi ini dilakukan dengan tiga cara, yaitu: pengujian *black box*, *usability* dan akurasi implementasi *google speech API*. Kemudian dari hasil pengujian dianalisis untuk mengetahui apakah aplikasi telah dan berfungsi sesuai dengan kebutuhan fungsional dan *non* fungsional serta tujuan pembuatan aplikasi. Pengujian *black box* berfokus pada kebutuhan fungsional, pengujian ini untuk menemukan apakah terdapat kesalahan seperti pada kesalahan kesalahan kinerja sistem dan *interface*. Metode yang digunakan yaitu *requirement validation testing* yang dilakukan untuk menguji apakah kebutuhan fungsional dari hasil implementasi aplikasi telah sesuai dengan hasil tahap analisis kebutuhan. Pengujian *usability* merupakan kebutuhan non fungsional aplikasi. Pengujian ini bertujuan untuk menilai dari tampilan antarmuka aplikasi apakah pengguna mudah dalam menggunakan aplikasi. Pengujian berfokus menilai pada tampilan antarmuka pengguna dengan menggunakan *system usability scale* kuesioner dengan skala *likert* 1 sampai 5.

Untuk menguji tingkat akurasi dari hasil implementasi *google speech API* pada aplikasi edukasi berbasis android sebagai sarana belajar pengucapan dalam pengucapan kosakata bahasa *Inggris* ini. Dalam pengujian ini melibatkan satu orang tenaga pengajar pada salah satu lembaga kursus bahasa *Inggris* yang berada di kota Malang sebagai responden. Pengujian dilaksanakan dengan cara responden diinstruksikan mengucapkan kosakata Bahasa *Inggris* yang ditampilkan pada layar aplikasi kemudian aplikasi akan mengubah suara menjadi *text* dan menjalankan proses pencocokan antara *text* yang diterima dengan *text* yang ditampilkan pada aplikasi. Setelah itu responden akan diminta mengisi *form* berdasarkan hasil percobaan yang telah dilakukan.

3.5 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan berhasil diselesaikan. Tahap ini merupakan tahapan terakhir yang berisi kesimpulan dari hasil penelitian dan saran untuk penelitian atau pengembangan selanjutnya.



BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Bab ini membahas mengenai analisis kebutuhan dan perancangan aplikasi aplikasi belajar pengucapan kosakata Bahasa *Inggris* yang dibagi menjadi delapan bagian, yaitu : Gambaran umum sistem, identifikasi aktor kebutuhan fungsional, *use case diagram*, *use case scenario*, perancangan, membangun *prototype* dan evaluasi *prototype*. Pada pengembangan aplikasi ini penulis menerapkan metode pengembangan *Prototype* yang proses penggalan kebutuhan dan evaluasi *prototype* melibatkan calon *real user* pada aplikasi yang akan dikembangkan, penulis memilih lima responden yang akan diminta untuk melakukan proses penggalan kebutuhan seperti yang dijelaskan pada bab sebelumnya.

4.1 Gambaran Umum Sistem



Gambar 4.1 Gambaran Umum Sistem

Aplikasi belajar pengucapan kosakata bahasa *Inggris* ini merupakan aplikasi dengan sistem operasi android yang diimplementasi menggunakan aplikasi editor android studio dan menggunakan bahasa pemrograman *Java* dan Extensible Markup Language(XML).

Dalam Gambar 4.1 merupakan ilustrasi dari cara kerja aplikasi dengan implementasi dari API *Google speech*. Fitur *audio* yang terdapat pada *smartphone* dimanfaatkan untuk mempermudah memahami penyebutan kosakata bahasa *Inggris*. Aplikasi akan menampilkan satu kata Bahasa *Inggris* pada layar yang akan diucapkan oleh pengguna, kemudian pengguna diminta untuk mengucapkan kata yang ditampilkan dengan jelas dan cukup keras sehingga *smartphone* dapat menangkap apa yang diucapkan oleh *user*, selanjutnya aplikasi akan mengubah kata yang diucapkan oleh *user* ke bentuk *text* dan aplikasi akan mencocokkan kata yang ditampilkan pada aplikasi dan kata yang diucapkan oleh *user*. Selain itu aplikasi juga akan mengeluarkan *audio* dari *smartphone* mengenai cara pengucapan kosakata Bahasa *Inggris* yang tepat, dengan demikian *user* dapat belajar dan melatih pengucapan kosakata Bahasa *Inggris* dengan mudah.

4.2 Skenario Aplikasi Belajar Pengucapan Kosakata Bahasa Inggris

Pada skenario belajar pengucapan ini akan membahas bagaimana aplikasi ini dapat dijadikan sebagai aplikasi belajar pengucapan kosakata Bahasa Inggris. Dengan memanfaatkan *voice recognition* aplikasi ini dapat memberitahu pengucapan kosakata Bahasa Inggris yang benar dan menilai apakah pengucapan dari pengguna sudah benar, yaitu terdapat pada kedua fitur utamanya.

4.2.1 Konten Aplikasi Belajar pengucapan Kosakata Bahasa Inggris

Pada bagian ini menjabarkan konten apa saja yang ada pada aplikasi belajar pengucapan kosakata bahasa Inggris, yaitu:

1. Konten tugas harian merupakan konten pada aplikasi ini yang bertujuan agar pengguna dapat aktif menggunakan aplikasi setiap hari untuk belajar pengucapan kosakata bahasa Inggris. Pada tugas harian terdiri dari tiga tugas harian yang dapat dikerjakan yang berupa perintah untuk mengerjakan soal belajar atau soal tantangan, jika tugas harian telah selesai dikerjakan maka aplikasi akan mencentang tugas harian yang sudah selesai dikerjakan pengguna. Daftar tugas harian akan berganti setiap hari.
2. Konten *top score* merupakan konten yang hanya ada pada konten tantangan, ketika pengguna mendapatkan *top score* baru setelah mengerjakan tantangan yang sesuai dengan kelas yang dikerjakan maka *top score* dapat di simpan dan pengguna dapat melihat daftar tugas *top score* sesuai dengan kelas tantangan yang dipilih.
3. Konten Belajar merupakan konten untuk belajar pengucapan kosakata bahasa Inggris. Pada konten ini terdapat tiga kategori soal yaitu kata kerja, kata benda dan kata sifat. Ketika mengerjakan soal pengguna dapat mendengarkan suara *audio* berupa pengucapan yang benar dari soal bahasa Inggris, pengguna juga dapat menjawab soal dengan cara menekan tombol jawab dan mengucapkan soal bahasa Inggris yang kemudian aplikasi akan menerima masukan suara sebagai jawaban, setelah itu aplikasi memeriksa apakah jawaban pengguna benar atau salah.
4. Konten Tangan merupakan konten untuk melatih pengucapan kosakata bahasa Inggris. Pada konten ini terdapat tiga kelas soal yaitu mudah, sedang dan sulit. Ketika mengerjakan soal pengguna tidak dapat mendengarkan suara *audio* berupa pengucapan yang benar dari soal bahasa Inggris hal ini merupakan pembeda dari konten belajar, sama dengan konten belajar pengguna dapat menjawab soal dengan cara menekan tombol jawab dan mengucapkan soal bahasa Inggris yang kemudian aplikasi akan menerima masukan suara sebagai jawaban, setelah itu aplikasi memeriksa apakah jawaban pengguna benar atau salah. Perbedaan lain pada konten tantangan dengan konten belajar adalah pada konten tantangan terdapat batasan waktu dalam menjawab semua soal dan terdapat perhitungan total poin setelah tantangan selesai.

4.2.2 Alur Aplikasi Belajar pengucapan Kosakata Bahasa Inggris

Pada bagian ini membahas alur dari aplikasi belajar pengucapan kosakata bahasa Inggris, alur dibahas dengan cara mendeskripsi lebih detail pada fitur atau konten yang telah dijelaskan pada sub bab konten aplikasi belajar pengucapan kosakata bahasa Inggris.

Pada aplikasi ini terdapat dua fitur utama yaitu, fitur belajar dan tantangan. Pada fitur belajar akan ditampilkan pilihan menu berupa kategori pengelompokan kosakata Bahasa Inggris berdasarkan kata benda, kata sifat dan kata kerja, pada fitur belajar diberikan jumlah soal 15 dan pengguna dapat menggunakan bantuan untuk mendengar suara *audio* berupa pengucapan yang benar dari soal kosakata bahasa Inggris, pengguna dapat melewati soal atau menjawab soal dengan cara menekan tombol jawab kemudian mengucapkan soal yang ditampilkan dan aplikasi akan menangkap suara pengguna sebagai jawaban, aplikasi akan menilai jawaban dari pengguna salah atau benar dan kemudian menampilkan hasil jawaban di layar aplikasi. Pada fitur tantangan perbedaan yang ada dengan fitur belajar berupa tidak adanya bantuan untuk mendengar pengucapan yang benar dari soal dan *user* akan diberikan soal-soal kosakata Bahasa Inggris yang harus diucapkan dalam waktu yang sudah ditentukan dan setiap soal yang benar akan diberikan poin, dan setiap soal yang dilewati maka poin akan semakin berkurang. Pada fitur tantangan ini juga diterapkan kelas-kelas yang menandakan tingkat kesulitan dari kosakata yang ditentukan berdasarkan banyaknya huruf pada kata dan waktu tantangan, pembagian kelas, yaitu:

- Mudah, berupa kata yang terdiri dari 1 – 5 huruf dengan jumlah soal 10 butir dan waktu tantangan 120 detik.
- Sedang, berupa kata yang terdiri dari 6 – 8 huruf dengan jumlah soal 10 butir dan waktu tantangan 90 detik.
- Sulit, berupa kata yang terdiri dari 9 atau lebih huruf dengan jumlah soal 10 butir dan waktu tantangan 60 detik.

Dengan dua fitur utama tersebut maka diharapkan pengguna dapat lebih mudah dalam menilai apakah pengucapan kosakata yang diucapkan telah sesuai dan tepat atau belum serta pengguna juga telah diberikan contoh pengucapan yang benar sehingga pengguna dapat mengikuti suara *audio* pengucapan yang benar.

Selain dua fitur utama yang telah dijelaskan ada dua fitur tambahan yang ada pada aplikasi belajar pengucapan kosakata bahasa Inggris, yaitu: fitur tambahan yang pertama adalah fitur tugas harian berada pada halaman utama aplikasi yang setiap harinya pengguna diberikan tiga tugas harian untuk dikerjakan agar pengguna lebih aktif dalam menggunakan aplikasi, jika tugas harian telah selesai dikerjakan maka aplikasi akan mencentang tugas harian yang sudah selesai dikerjakan pengguna, tugas harian akan berganti dengan tugas yang baru setiap pergantian tanggal dan status tugas harian belum selesai. Fitur tambahan yang

kedua adalah *top score* yang fitur ini hanya ada pada fitur utama tantangan, setiap kelas tantangan memiliki *top score* masing-masing. Ketika pengguna mendapatkan poin yang masuk ke dalam tiga besar *top score* maka pada halaman total poin pengguna bisa memasukkan nama dan menyimpan *top score* baru, daftar *top score* dapat dilihat sesuai dengan kelas yang dipilih. Dari kedua fitur tambahan yang disebutkan maka diharapkan pengguna akan lebih sering untuk menggunakan aplikasi belajar pengucapan kosakata bahasa Inggris untuk berlatih pengucapan yang benar.

4.3 Identifikasi Aktor

Identifikasi aktor menjabarkan aktor yang berinteraksi dengan sistem agar aplikasi yang akan dibangun dapat berjalan, yang dimaksud aktor yaitu sesuatu yang berada di luar sistem. Aktor pada aplikasi aplikasi belajar pengucapan pengucapan kosakata bahasa Inggris yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Semua orang yang dapat menggunakan aplikasi/sistem

Pada Tabel 4.2 aktor yang diidentifikasi hanya satu yaitu pengguna yang dimaksud adalah orang-orang yang menggunakan sistem/aplikasi aplikasi belajar pengucapan kosakata bahasa Inggris.

4.4 Kebutuhan Fungsional Dan Non Fungsional

Pada tahap menggali kebutuhan fungsional dan kebutuhan *non* fungsional akan didefinisikan daftar kebutuhan apa saja untuk membangun aplikasi belajar pengucapan kosakata Bahasa Inggris. Kebutuhan fungsional yaitu fungsi-fungsi yang sistem dapat lakukan. Penggalan kebutuhan pada aplikasi ini melibatkan lima responden untuk mendefinisikan kebutuhan fungsional dan *non* fungsional, hal tersebut dilakukan sesuai dengan metode *prototype* yang proses penggalan kebutuhan melibatkan calon pengguna aplikasi, responden yang dilibatkan sebanyak lima orang dan penggalan kebutuhan dilakukan dengan cara wawancara.

Pada Tabel 4.3 terdapat beberapa kebutuhan fungsional yang harus terdapat pada aplikasi aplikasi belajar pengucapan pengucapan kosakata Bahasa Inggris yang akan dibangun. Setiap fungsional akan dimodelkan ke bentuk *use case* diagram.

Pada proses penggalan kebutuhan aplikasi belajar pengucapan kosakata bahasa Inggris ada juga kebutuhan *non* fungsional yakni kebutuhan yang berupa perilaku yang terdapat pada aplikasi. Kebutuhan *non* fungsional tidak dimodelkan dalam bentuk *use case* diagram. Kebutuhan non fungsional dijelaskan pada Tabel 4.4.

Tabel 4.3 Kebutuhan Fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi Kebutuhan
1	F01	Menampilkan soal	Dapat menampilkan soal kosakata bahasa Inggris dan artinya dalam bahasa Indonesia.
2	F02	Mengeluarkan suara soal	Pada fitur belajar, aplikasi dapat mengeluarkan <i>audio</i> atau suara berupa pengucapan yang benar dari kosakata bahasa Inggris yang merupakan soal yang sedang ditampilkan, yaitu dengan cara mengubah <i>text</i> ke suara. Suara soal dapat di dengarkan dengan cara pengguna menekan tombol dengan <i>icon audio</i> .
3	F03	Menjawab soal	Aplikasi dapat menampilkan tombol untuk menerima masukan berupa suara dari pengguna yang kemudian masukan suara di ubah ke bentuk <i>text</i> , masukan suara yang diterima akan di anggap oleh aplikasi sebagai jawaban dari soal.
4	F04	Menilai jawaban	Aplikasi dapat menilai jawaban pengguna yaitu benar atau salah dan menampilkan hasil jawaban.
5	F05	Menampilkan jumlah kata benar	Aplikasi dapat menampilkan jumlah kata benar yang dijawab pengguna.
6	F06	Menampilkan sisa waktu	Pada fitur tantangan, aplikasi dapat menampilkan sisa waktu untuk pengguna menyelesaikan soal tantangan.
7	F07	Hitung total poin	Pada fitur tantangan, aplikasi dapat melakukan perhitungan total poin dan menampilkan total poin.
8	F08	Menyimpan <i>top score</i> baru	Pada fitur tantangan, aplikasi dapat menyimpan <i>top score</i> baru yang didapatkan pengguna sesuai dengan kelas tantangan yang dijalankan.

Tabel 4.3 Kebutuhan Fungsional (lanjutan)

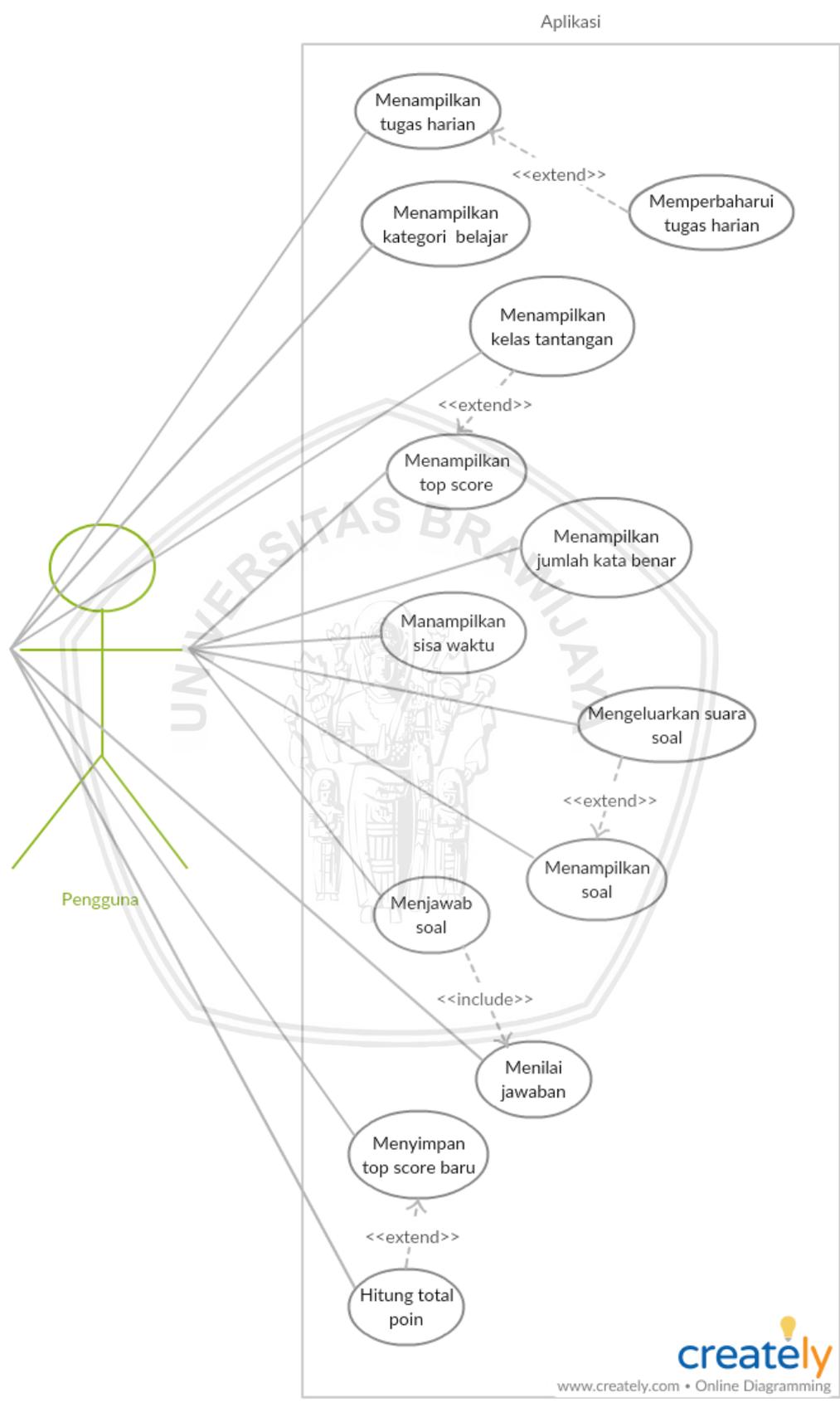
No	Kode Fungsi	Nama Fungsi	Deskripsi Kebutuhan
9	F09	Menampilkan <i>top score</i>	Pada fitur tantangan, aplikasi dapat menampilkan <i>list</i> tiga <i>top score</i> sesuai dengan kelas yang dipilih.
10	F10	Menampilkan tugas harian	Aplikasi dapat menampilkan <i>list</i> tiga tugas harian yang dapat dikerjakan oleh pengguna.
11	F11	Memperbarui tugas harian	Aplikasi dapat memperbarui tugas baru yaitu berupa memberitahukan bahwa tugas telah selesai dikerjakan.
12	F12	Menampilkan kategori belajar	Aplikasi dapat menampilkan tiga pilihan kategori pada fitur belajar.
13	F13	Menampilkan kelas tantangan	Aplikasi dapat menampilkan tiga pilihan kelas pada fitur tantangan.

Tabel 4.4 Kebutuhan Non Fungsional

No	Kode Non Fungsional	Nama Non Fungsional	Deskripsi
1	NF01	<i>Usability</i>	Aplikasi harus dapat menampilkan antarmuka yang mudah digunakan oleh pengguna dan pengguna mudah untuk memahami setiap fungsi pada aplikasi hanya dengan melihat antarmuka aplikasi.

4.5 Use Case diagram

Use case diagram merupakan gambaran dari interaksi sistem dengan aktor yang dibuat dalam bentuk diagram berdasarkan dari kebutuhan fungsional. Dalam Gambar 4.2 dapat dilihat *use case* diagram aplikasi belajar pengucapan pengucapan kosakata Bahasa *Inggris* yang memiliki satu aktor dan tigabelas *use case*. Dari *use case* yang telah dibuat kemudian akan dijelaskan masing-masing *use case* ke dalam bentuk *use case scenario*.



Gambar 4.2 Use case Diagram

4.6 Use Case Scenario

Pada bagian ini dijelaskan *scenario* dari masing-masing *use case* diagram yang telah dibuat pada bagian diagram.

4.6.1 Use case Scenario Menampilkan Soal

Use case scenario menampilkan soal dapat dilihat pada Tabel 4.5.

Tabel 4.5 Use case Scenario Menampilkan Soal

Kode <i>use case</i>	UC01
Nama <i>use case</i>	Menampilkan soal
Tujuan	Menampilkan soal kosakata bahasa Inggris dan artinya dalam bahasa Indonesia.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman untuk memilih salah satu kategori belajar atau kelas tantangan.
Main flow	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu kategori belajar atau kelas tantangan. 2. Aplikasi menampilkan satu kata acak sesuai kategori belajar atau tantangan dan artinya dalam Bahasa Indonesia. 3. Pengguna menjawab soal. 4. Aplikasi menilai hasil jawaban pengguna benar. 5. Aplikasi menampilkan soal selanjutnya. 6. Langkah 3 dan 4 diulangi sampai semua soal terjawab.
Alternative flow 1	<ol style="list-style-type: none"> 4.1. Aplikasi menilai hasil jawaban pengguna salah. 4.2. Aplikasi tetap menampilkan soal yang sama.
Alternative flow 2	<ol style="list-style-type: none"> 2.1. Pengguna melewati soal yang sedang ditampilkan. 2.2. Aplikasi menampilkan soal selanjutnya.
Post condition	Aplikasi telah menampilkan semua soal dan kembali ke halaman menampilkan kategori belajar atau kelas tantangan.

4.6.2 Use case Scenario Mengeluarkan Suara Soal

Use case scenario mengeluarkan suara soal dapat dilihat pada Tabel 4.6.

Tabel 4.6 Use case Scenario Mengeluarkan Suara Soal

Kode <i>use case</i>	UC02
Nama <i>use case</i>	Mengeluarkan suara soal
Tujuan	Aplikasi mengeluarkan <i>audio</i> atau suara berupa pengucapan yang benar dari kosakata bahasa Inggris yang merupakan soal yang sedang ditampilkan. Suara soal dapat di dengarkan dengan cara pengguna menekan tombol dengan <i>icon audio</i> .
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman soal belajar.
Main flow	<ol style="list-style-type: none"> 1. Pengguna menekan tombol dengarkan suara soal. 2. Aplikasi mengubah <i>text</i> soal ke bentuk suara.
Alternative flow	-
Post condition	Aplikasi mengeluarkan suara berupa pengucapan kosakata bahasa Inggris yang banar dari soal yang sedang ditampilkan.

4.6.3 Use case Scenario Menjawab Soal

Use case scenario menjawab soal dapat dilihat pada Tabel 4.7.

Tabel 4.7 Use case Scenario Menjawab Soal

Kode <i>use case</i>	UC03
Nama <i>use case</i>	Menjawab soal
Tujuan	Aplikasi menerima masukan berupa suara dari pengguna yang kemudian masukan suara di ubah ke bentuk <i>text</i> , masukan suara yang diterima akan di anggap oleh aplikasi sebagai jawaban dari soal.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman soal belajar atau soal tantangan.

Tabel 4.7 Use case Scenario Menjawab Soal (lanjutan)

Main flow	<ol style="list-style-type: none"> 1. Pengguna menekan tombol jawab soal. 2. Aplikasi mengeluarkan suara bel yang menandakan aplikasi siap menerima jawaban berupa masukan suara. 3. Aplikasi menerima masukan suara dari pengguna.
Alternative flow	-
Post condition	Aplikasi mengubah masukan suara pengguna ke bentuk <i>text</i> .

4.6.4 Use case Scenario Menilai Jawaban

Use case scenario menilai jawaban dapat dilihat pada Tabel 4.8.

Tabel 4.8 Use case Scenario Menilai Jawaban

Kode <i>use case</i>	UC04
Nama <i>use case</i>	Menilai jawaban
Tujuan	Aplikasi menilai jawaban pengguna yaitu benar atau salah dan menampilkan hasil jawaban.
Aktor	Pengguna
Pre-condition	Pengguna telah menjawab soal.
Main flow	<ol style="list-style-type: none"> 1. Aplikasi menerima jawaban pengguna dari masukan suara yang telah di ubah ke <i>text</i>. 2. Aplikasi menampilkan jawaban pengguna. 3. Aplikasi mencocokkan jawaban pengguna dengan soal.
Alternative flow	-
Post condition	Aplikasi menampilkan hasil jawaban pengguna berupa keterangan benar atau salah.

4.6.5 Use case Scenario Menampilkan Jumlah Kata Benar

Use case scenario menampilkan jumlah kata benar dapat dilihat pada Tabel 4.9.

Tabel 4.9 Use case Scenario Menampilkan Jumlah Kata Benar

Kode <i>use case</i>	UC05
Nama <i>use case</i>	Menampilkan Jumlah Kata Benar



Tabel 4.9 Use case Scenario Menampilkan Jumlah Kata Benar (lanjutan)

Tujuan	Aplikasi menampilkan jumlah kata benar yang dijawab pengguna
Aktor	Pengguna
Pre-condition	Aplikasi telah menilai hasil jawaban pengguna.
Main flow	<ol style="list-style-type: none"> 1. Aplikasi menilai jawaban pengguna benar. 2. Aplikasi menambah jumlah kata benar. 3. Aplikasi menampilkan menampilkan halaman soal.
Alternative flow	<ol style="list-style-type: none"> 1.1. Aplikasi menilai hasil jawaban pengguna salah. 1.2. Aplikasi tidak menambah jumlah kata benar.
Post condition	Aplikasi menampilkan jumlah kata benar yang telah dijawab pengguna pada halaman soal.

4.6.6 Use case Scenario Manampilkan Sisa Waktu

Use case scenario manampilkan sisa waktu dapat dilihat pada Tabel 4.10.

Tabel 4.10 Use case Scenario Manampilkan Sisa Waktu

Kode use case	UC06
Nama use case	Manampilkan sisa waktu
Tujuan	Pada fitur tantangan, aplikasi menampilkan sisa waktu untuk pengguna menyelesaikan soal tantangan.
Aktor	Pengguna
Pre-condition	Pengguna memulai tantangan.
Main flow	<ol style="list-style-type: none"> 1. Aplikasi menampilkan halaman soal 2. Aplikasi menampilkan hitungan mundur sisa waktu menjawab soal tantangan. 3. Sisa waktu menjawab soal telah habis.
Alternative flow	<ol style="list-style-type: none"> 1. Pengguna telah menjawab soal terakhir. 2. Hitung mundur sisa waktu dihentikan.
Post condition	Aplikasi menampilkan halaman total poin.

4.6.7 Use case Scenario Hitung Total Poin

Use case scenario hitung total poin dapat dilihat pada Tabel 4.11.

Tabel 4.11 Use case Scenario Hitung Total Poin

Kode use case	UC07
Nama use case	Hitung total poin
Tujuan	Pada fitur tantangan, aplikasi melakukan perhitungan total poin dan menampilkan total poin.
Aktor	Pengguna
Pre-condition	Pengguna telah selesai menjalankan soal tantangan.
Main flow	<ol style="list-style-type: none"> 1. Aplikasi menampilkan halaman total poin. 2. Aplikasi melakukan perhitungan total poin berdasarkan jumlah kata benar dan kata yang dilewati.
Alternative flow	-
Post condition	Aplikasi menampilkan jumlah total poin dan bintang yang didapatkan pengguna sesuai dengan total poin.

4.6.8 Use case Scenario Menyimpan Top score Baru

Use case scenario menyimpan top score baru dapat dilihat pada Tabel 4.12.

Tabel 4.12 Use case Scenario Menyimpan Top score Baru

Kode use case	UC08
Nama use case	Menyimpan top score baru
Tujuan	Pada fitur tantangan, aplikasi menyimpan top score baru yang didapatkan pengguna sesuai dengan kelas tantangan yang dijalankan.
Aktor	Pengguna
Pre-condition	Aplikasi telah menghitung total poin.

Tabel 4.12 Use case Scenario Menyimpan Top score Baru (lanjutan)

Main flow	<ol style="list-style-type: none"> 1. Aplikasi menentukan apakah total poin termasuk tiga <i>top score</i> pada kelas tantangan yang telah dijalankan. 2. Total poin termasuk tiga <i>top score</i> pada kelas tantangan yang telah dijalankan. 3. Aplikasi menampilkan kolom input nama dan tombol simpan <i>top score</i> baru.
Alternative flow	<ol style="list-style-type: none"> 2.1. Total poin tidak termasuk tiga <i>top score</i> pada kelas tantangan yang telah dijalankan. 2.2. Aplikasi tidak Aplikasi menampilkan kolom input nama dan tombol simpan <i>top score</i> baru.
Post condition	Aplikasi telah menyimpan <i>top score</i> baru sesuai kelas tantangan yang dijalankan dan menutup kolom input nama dan tombol simpan <i>top score</i> baru.

4.6.9 Use case Scenario Menampilkan Top score

Use case scenario menampilkan *top score* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Use case Scenario Menampilkan Top score

Kode <i>use case</i>	UC09
Nama <i>use case</i>	Menampilkan <i>top score</i>
Tujuan	Pada fitur tantangan, aplikasi dapat menampilkan <i>list</i> tiga <i>top score</i> sesuai dengan kelas yang dipilih.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman pilih kelas tantangan.
Main flow	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu kelas tantangan. 2. Aplikasi menampilkan halaman mulai tantangan. 3. Pengguna menekan tombol <i>top score</i>.
Alternative flow	-
Post condition	Aplikasi menampilkan <i>list</i> tiga <i>top score</i> sesuai kelas tantangan yang dipilih.

4.6.10 Use case Scenario Menampilkan Tugas Harian

Use case scenario menampilkan tugas harian dapat dilihat pada Tabel 4.14.

Tabel 4.14 Use case Scenario Menampilkan Tugas Harian

Kode use case	UC010
Nama use case	Menampilkan tugas harian
Tujuan	Aplikasi menampilkan <i>list</i> tiga tugas harian yang dapat dikerjakan oleh pengguna.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman utama aplikasi.
Main flow	<ol style="list-style-type: none"> 1. Aplikasi menampilkan halaman utama. 2. Aplikasi melakukan pengecekan apakah tanggal tugas harian sudah berganti. 3. Tanggal tugas harian telah berganti. 4. Aplikasi membuat <i>list</i> tugas harian baru.
Alternative flow	<ol style="list-style-type: none"> 3.1 Tanggal tugas harian belum berganti. 3.2 Aplikasi tidak membuat tugas harian baru.
Post condition	Aplikasi menampilkan <i>list</i> tiga tugas harian.

4.6.11 Use case Scenario Memperbarui tugas harian

Use case scenario memperbarui tugas harian dapat dilihat pada Tabel 4.15.

Tabel 4.15 Use case Scenario Memperbarui tugas harian

Kode use case	UC011
Nama use case	Memperbarui tugas harian
Tujuan	Aplikasi memperbarui tugas baru yaitu berupa memberitahukan bahwa tugas telah selesai dikerjakan.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman utama aplikasi.

Tabel 4.15 Use case Scenario Memperbarui tugas harian (lanjutan)

Main flow	<ol style="list-style-type: none"> 1. Aplikasi menampilkan halaman utama. 2. Aplikasi melakukan pengecekan apakah ada tugas harian yang telah dikerjakan. 3. Aplikasi menemukan ada tugas harian yang telah dikerjakan. 4. Aplikasi memperbarui status tugas harian menjadi telah selesai.
Alternative flow	<ol style="list-style-type: none"> 3.1 Aplikasi tidak menemukan ada tugas harian yang telah dikerjakan. 3.2 aplikasi tidak memperbarui status tugas harian.
Post condition	Aplikasi menampilkan <i>list</i> tiga tugas harian.

4.6.12 Use case Scenario Menampilkan Kategori Belajar

Use case scenario menampilkan kategori belajar dapat dilihat pada Tabel 4.16.

Tabel 4.16 Use case Scenario Menampilkan Kategori Belajar

Kode <i>use case</i>	UC012
Nama <i>use case</i>	Menampilkan kategori belajar
Tujuan	Aplikasi dapat menampilkan tiga pilihan kategori pada fitur belajar.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman utama aplikasi.
Main flow	<ol style="list-style-type: none"> 1. Pengguna mamilih fitur belajar. 2. Aplikasi menampilkan halaman pilih kategori belajar.
Alternative flow	-
Post condition	Aplikasi menampilkan tombol <i>list</i> tiga kategori soal belajar.

4.6.13 Use case Scenario Menampilkan Kelas Tantangan

Use case scenario menampilkan kelas tantangan dapat dilihat pada Tabel 4.17.

Tabel 4.17 Use case Scenario Menampilkan Kelas Tantangan

Kode <i>use case</i>	UC013
Nama <i>use case</i>	Menampilkan kelas tantangan



Tabel 4.17 *Use case Scenario* Menampilkan Kelas Tantangan (lanjutan)

Tujuan	Aplikasi dapat menampilkan tiga pilihan kelas pada fitur tantangan.
Aktor	Pengguna
Pre-condition	Pengguna berada pada halaman utama aplikasi.
Main flow	<ol style="list-style-type: none"> 1. Pengguna memilih fitur tantangan . 2. Aplikasi menampilkan halaman pilih kelas tantangan.
Alternative flow	-
Post condition	Aplikasi menampilkan tombol <i>list</i> tiga kelas soal tantangan.

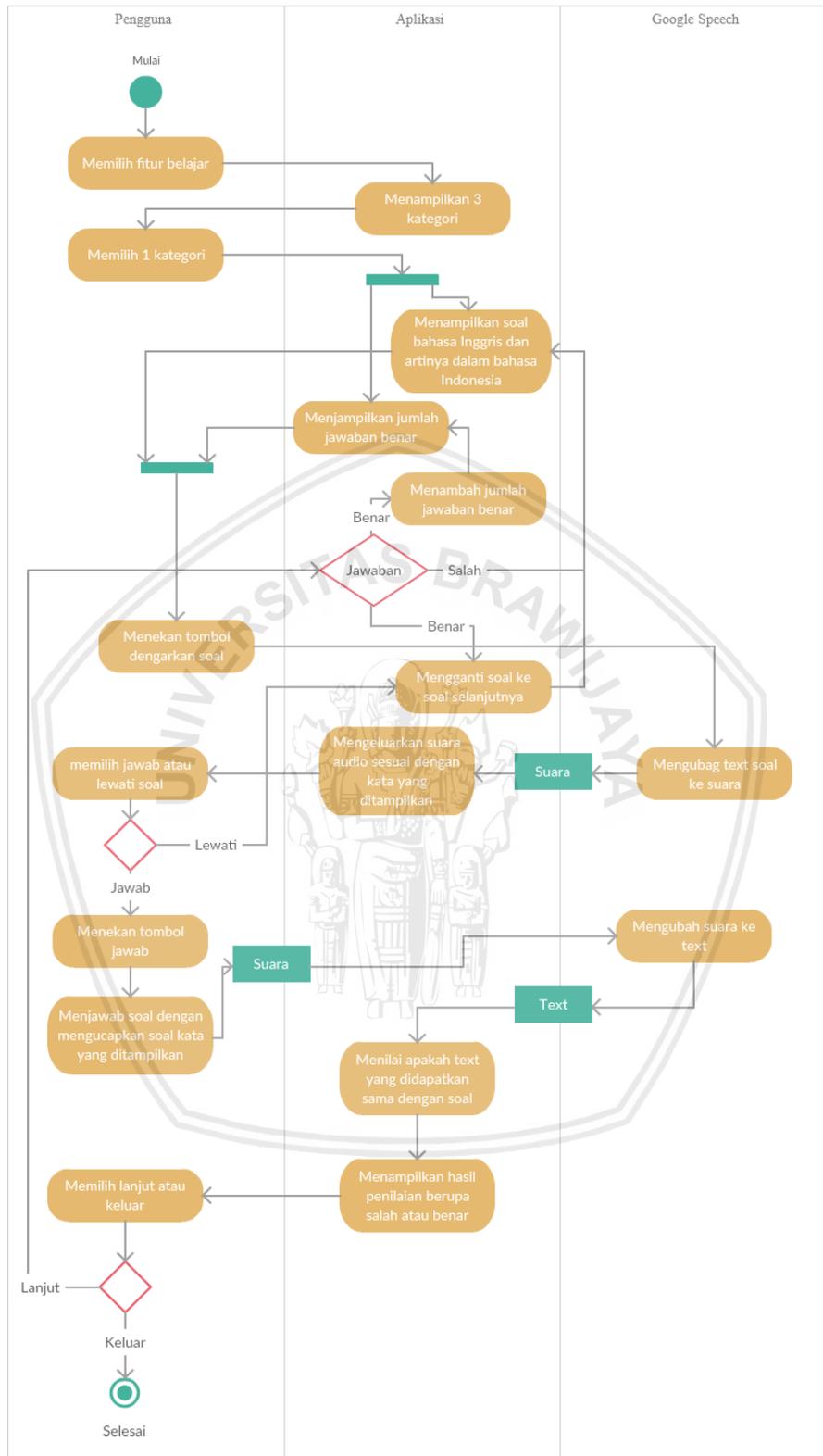
4.7 Perancangan

Pada tahap perancangan terdapat beberapa tahap dalam membangun perancangan aplikasi ini. Tahapan yang dimaksud, yaitu perancangan *activity diagram*, diagram kelas, *screenflow* dan perancangan antarmuka.

4.7.1 *Activity diagram*

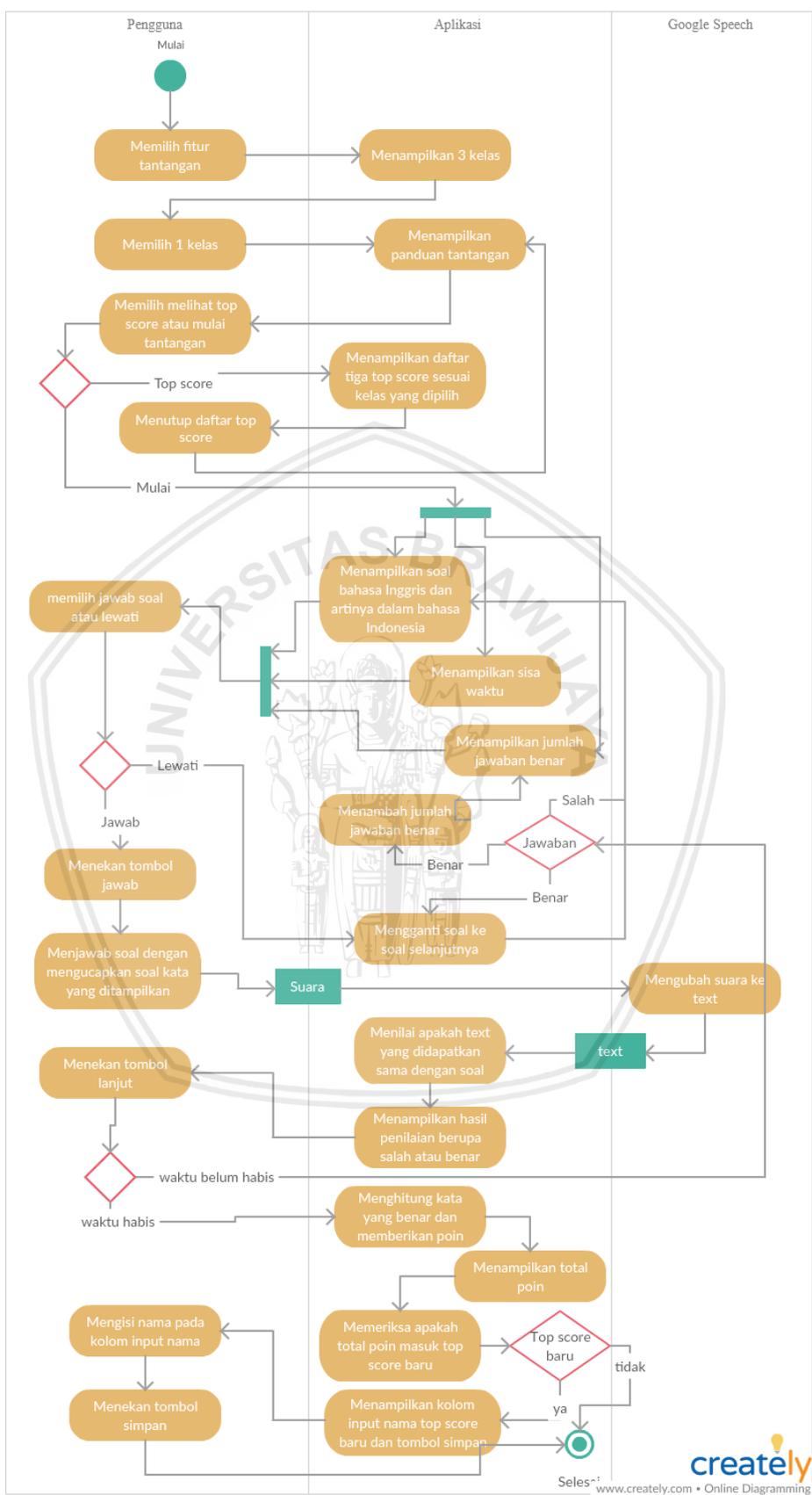
Activity diagram merupakan aktivitas yang dilakukan oleh aktor dengan aplikasi yang saling berhubungan yang dijelaskan dalam bentuk diagram. *Activity diagram* adalah Gambaran dari alur yang berurutan dari aktivitas *use case* oleh karena itu *activity diagram* dimodelkan berdasarkan *use case diagram* yang telah dibuat sebelumnya.

Dalam Gambar 4.3 menampilkan aktivitas ketika pengguna memilih fitur belajar. Kemudian aplikasi akan menampilkan beberapa kategori yaitu kategori kata kerja, kata sifat dan kata benda, pengguna diberikan instruksi untuk memilih salah satu kategori. Setelah memilih kategori maka aplikasi akan menampilkan satu kosakata bahasa *Inggris* beserta artinya dalam bahasa Indonesia, kemudian pengguna dapat menekan tombol dengarkan soal untuk mengeluarkan suara *audio* berupa pengucapan yang benar dari soal kosakata bahasa Inggris. Kemudian pengguna boleh menjawab soal kata yang diberikan dengan menekan tombol jawab atau melewati dengan menekan tombol lewati. Ketika pengguna melewati soal kata maka aplikasi akan menampilkan soal kata yang baru. Ketika pengguna menekan jawab soal, selanjutnya pengguna diminta untuk mengucapkan kata yang ditampilkan pada layar, jika pengguna salah menjawab soal kata maka aplikasi akan memberikan keterangan “salah”, namun jika pengguna menjawab soal dengan benar maka aplikasi akan menampilkan keterangan “benar” dan kemudian pengguna akan diberikan pilihan lanjut atau keluar dari halaman belajar.



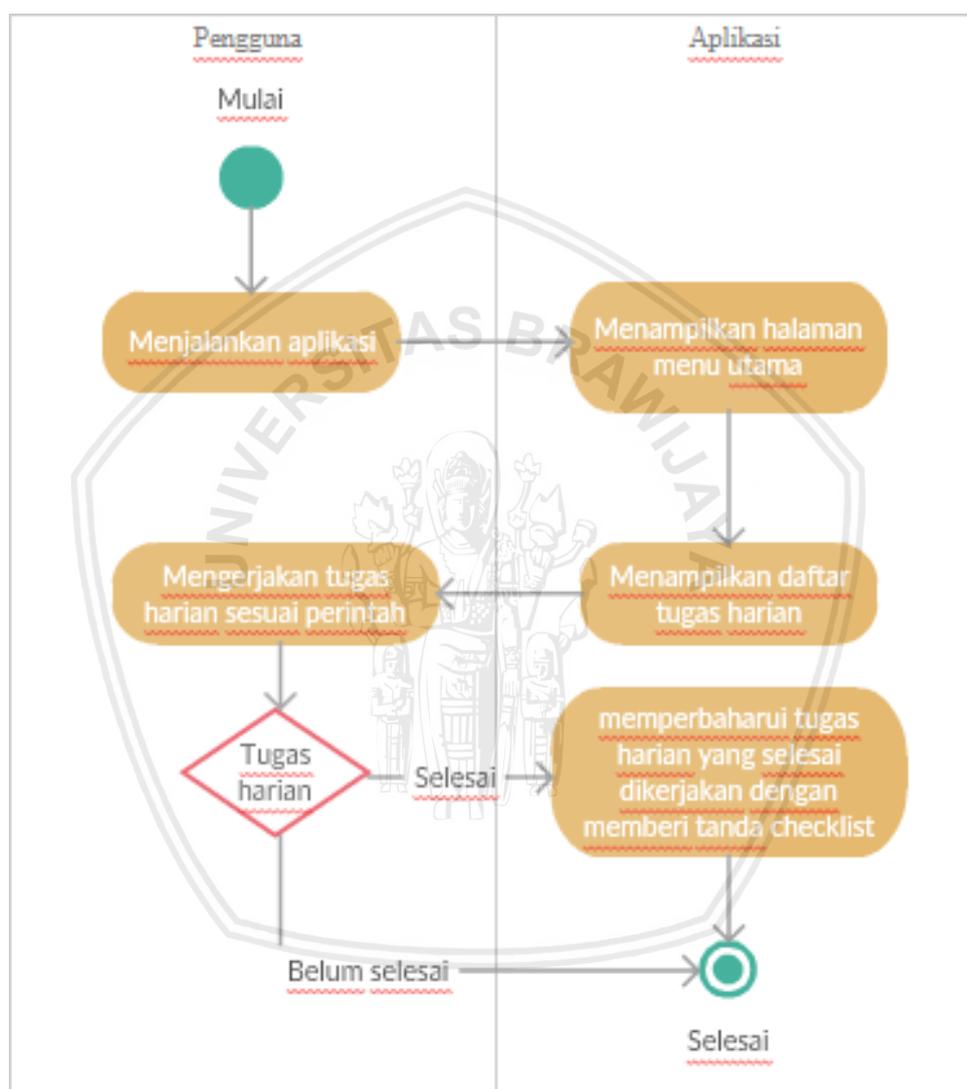
Gambar 4.3 Activity Diagram Belajar





Gambar 4.4 Activity Diagram Tantangan

Dalam Gambar 4.5 menampilkan aktivitas ketika tugas harian. Daftar tugas harian ditampilkan pada halaman menu utama aplikasi, pengguna dapat mengerjakan tugas harian sesuai dengan yang di perintahkan. Jika tugas harian selesai dikerjakan maka secara otomatis aplikasi akan memperbarui status tugas harian menjadi selesai, untuk tugas harian dengan status selesai akan di beri tanda *checklist* pada daftar tugas harian di halaman menu utama.



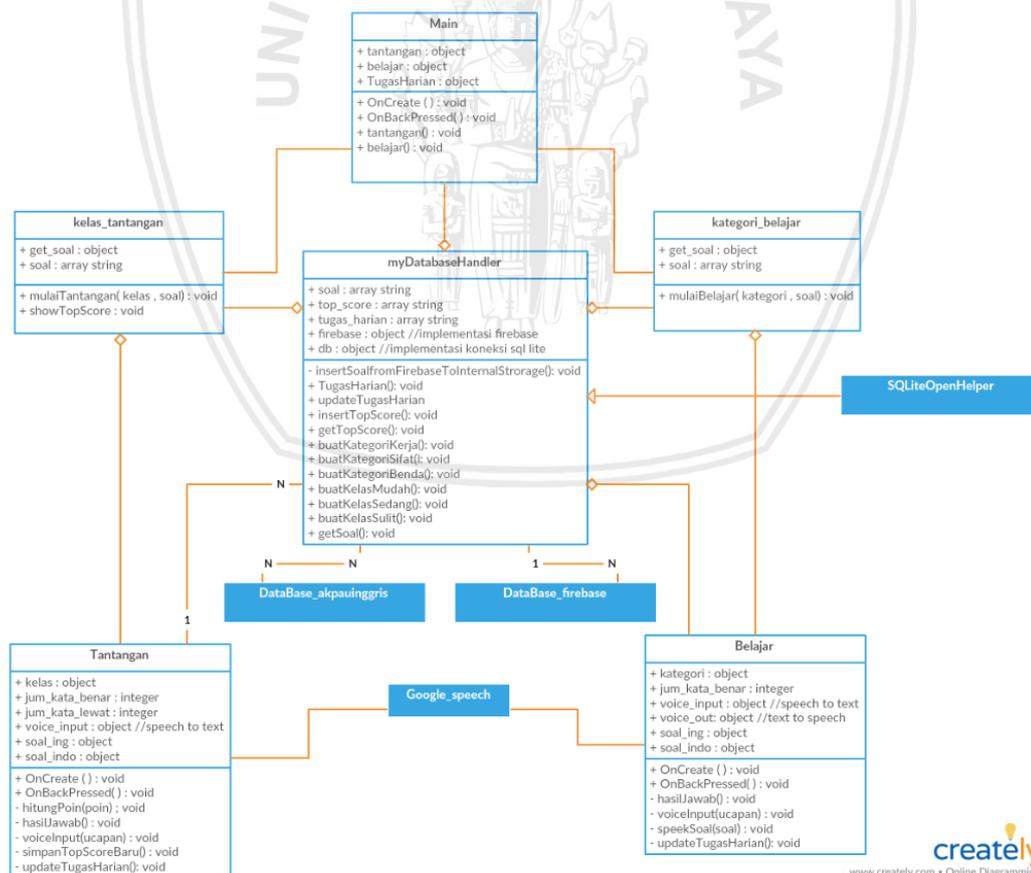
Gambar 4.5 Activity Diagram Tugas Harian

Dalam Gambar 4.4 menampilkan aktivitas ketika pengguna memilih fitur tantangan. Kemudian aplikasi akan menampilkan beberapa kelas yaitu kelas mudah, sedang dan sulit. Pengguna diberikan instruksi untuk memilih salah satu kelas. Setelah memilih kelas maka aplikasi akan menampilkan panduan tantangan dan juga menampilkan tombol mulai dan tombol *top score*. Tombol *top score* untuk melihat *top score* sesuai dengan kelas yang dipilih, jika pengguna menekan tombol untuk mulai maka aplikasi akan menampilkan halaman soal beserta satu kosakata bahasa *Inggris* beserta artinya dalam Bahasa Indonesia, menampilkan

jangka waktu tantangan berupa hitungan mundur dan menampilkan jumlah kata benar yang telah dijawab. Pengguna boleh langsung menjawab soal kata yang diberikan dengan menekan tombol jawab atau melewati dengan menekan tombol lewati, ketika pengguna melewati soal kata maka aplikasi akan menampilkan soal kata yang baru. Ketika pengguna menekan jawab soal, selanjutnya pengguna diminta untuk mengucapkan kata yang ditampilkan pada layar, jika pengguna salah menjawab soal kata maka aplikasi akan memberikan keterangan “salah”, namun jika pengguna menjawab soal dengan benar maka aplikasi akan menampilkan keterangan “benar”. Jika waktu yang diberikan telah habis atau soal yang disediakan telah habis maka aplikasi akan menghentikan tantangan dan menampilkan total poin yang didapatkan oleh pengguna. Jika total poin yang didapatkan pengguna masuk dalam tiga *top score* pada kelas yang dipilih maka aplikasi akan menampilkan kolom input nama dan tombol simpan *top score* baru yang berfungsi untuk menyimpan nama *top score* baru dan score yang didapatkan.

4.7.2 Diagram Kelas

Diagram kelas merupakan diagram yang menunjukkan kelas-kelas yang terdapat dalam sistem dan mengidentifikasi hubungan antar kelas.

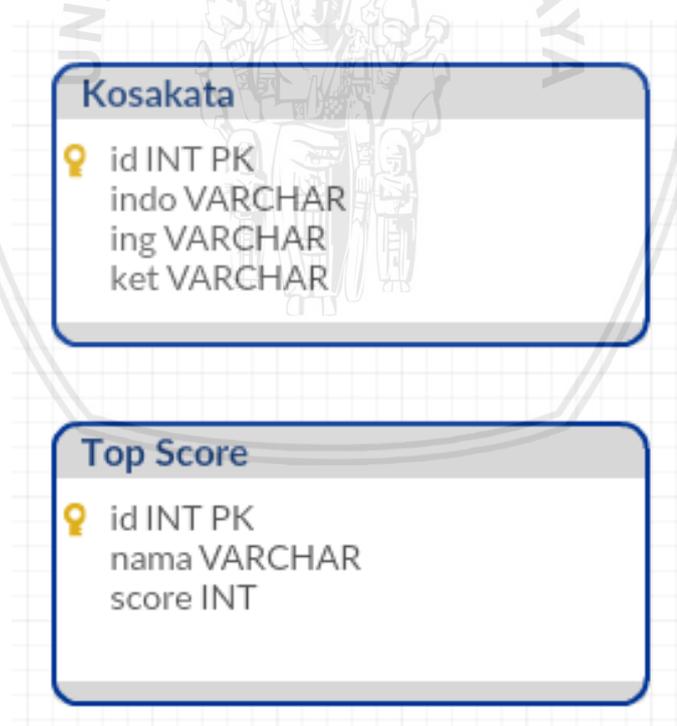


Gambar 4.6 Class Diagram

Dalam Gambar 4.6 menunjukkan kelas-kelas yang nantinya akan berinteraksi ketika aplikasi dijalankan. Kelas Main merupakan kelas utama yang pertama kali dijalankan. Setelah itu kelas Main akan memanggil kelas Kelas_tantangan untuk menampilkan tiga kelas tantangan dan kelas Kategori_belajar untuk menampilkan tiga kategori belajar. Kelas Kelas_tantangan dan kelas Kategori_belajar akan memanggil kelas myDatabaseHandler untuk membuat soal. kelas myDatabaseHandler terhubung dengan object SQLiteOpenHelper yang berfungsi untuk mengakses database Database_akpauinggris pada penyimpanan internal aplikasi dengan perintah sql lite. kelas myDatabaseHandler juga terhubung dengan object Database_firebase untuk mengakses database pada firebase. Kelas_tantangan akan memanggil kelas Tantangan untuk menampilkan soal tantangan. Kategori_belajar akan memanggil kelas Belajar untuk menampilkan soal belajar. Kelas Tantangan dan Belajar memiliki langsung hubungan dengan *object google speech* yang memiliki fungsi untuk merubah teks ke suara dan suara ke teks.

4.7.3 Desain Database

Pada bagian ini menjelaskan desain *database* yang akan diimplementasi pada pada tahap implementasi sistem.



Gambar 4.7 Desain Database

Dalam Gambar 4.7 menunjukkan terdapat dua tabel pada *database* aplikasi, tabel yang dimaksud yakni tabel kosakata dan tabel *top score*. Pada aplikasi belajar pengucapan kosakata bahasa Inggris *database* yang dibutuhkan cukup sederhana

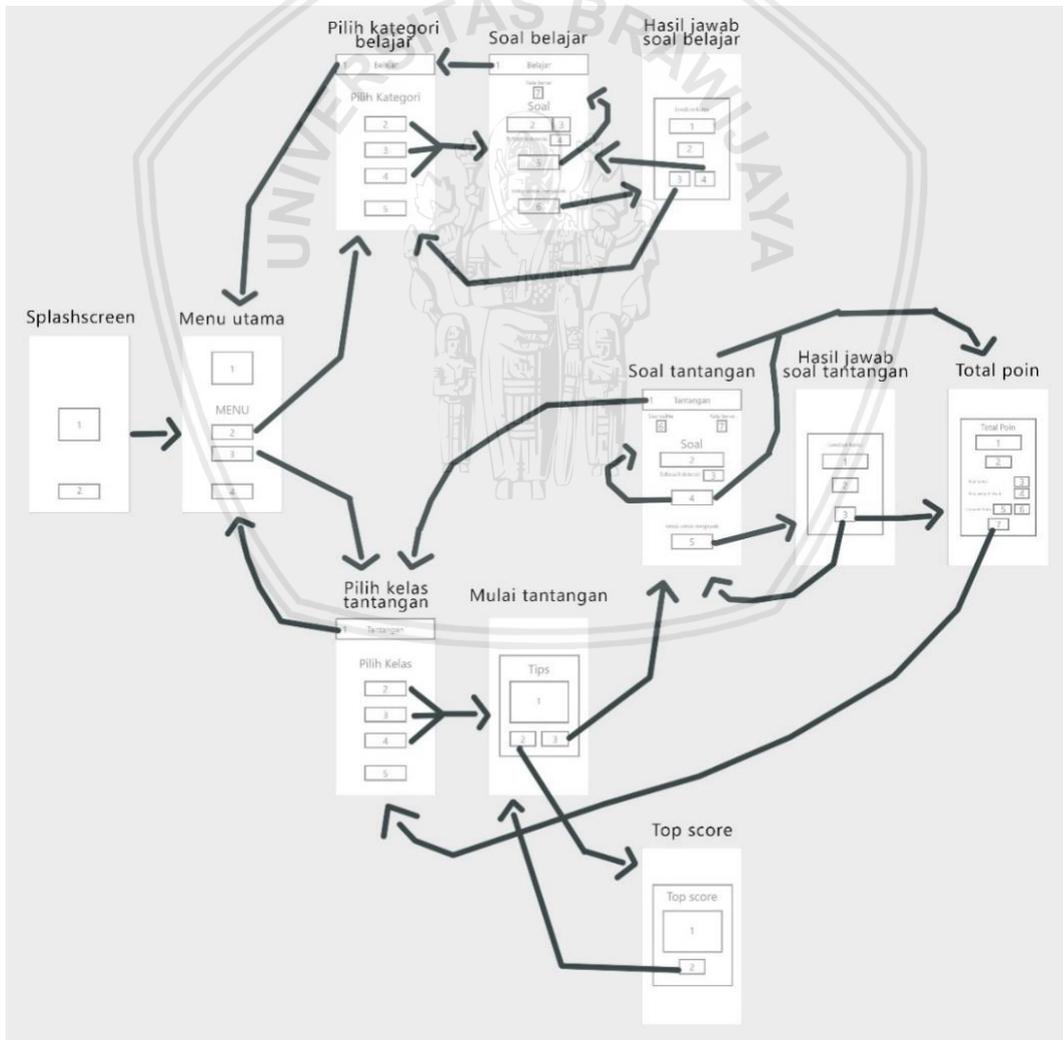
dan dari kedua tabel pada *database* tidak memiliki relasi antar tabel karena tidak dibutuhkan relasi antar tabel yang dimaksud. Desain *database* pada SQLite *database* yang disimpan pada penyimpanan internal dari perangkat yang menginstal aplikasi terdapat dua tabel seperti dalam gambar 4.7, sedangkan pada *database* pada *firebase* hanya terdapat satu tabel yakni tabel kosakata.

4.7.4 Screenflow

Pada bagian ini menjelaskan *screenflow* dan fitur yang terdapat pada aplikasi belajar kosakata bahasa Inggris.

Dalam Gambar 4.8 menunjukkan alur dari antarmuka aplikasi belajar pengucapan kosakata bahasa Inggris. Pada saat pertama kali menjalankan aplikasi halaman antarmuka yang pertama yaitu halaman *splashscreen* kemudian secara otomatis berpindah ke halaman menu utama. Pada halaman menu utama terdapat tombol belajar pada kotak nomor 2, jika pengguna menekan tombol belajar maka aplikasi akan menampilkan halaman pilih kategori belajar. Pada halaman pilih kategori belajar terdapat tombol untuk kembali ke halaman sebelumnya pada kotak nomor 1, tombol untuk memilih kategori kata kerja pada kotak nomor 2, kategori kata benda pada kotak nomor 3 dan kategori kata sifat pada kotak nomor 4. Jika pengguna memilih salah satu kategori maka aplikasi akan berpindah ke halaman soal belajar dan menampilkan soal sesuai kategori yang dipilih. Pada halaman soal belajar terdapat tombol untuk kembali ke halaman sebelumnya pada kotak nomor 1, tombol untuk melewati soal ke soal berikutnya pada kotak nomor 5 dan tombol jawab pada kotak nomor 6 untuk menjawab soal, jika pengguna menekan tombol jawab dan pengguna telah menjawab soal kemudian aplikasi akan menampilkan halaman hasil jawab soal belajar. Pada halaman hasil jawab soal belajar terdapat tombol keluar pada kotak nomor 3 untuk keluar dari halaman soal belajar kemudian menampilkan halaman pilih kategori belajar, dan terdapat juga tombol lanjut pada kotak nomor 4 untuk kembali ke halaman soal belajar, jika jawaban benar maka soal akan berganti dan jika salah maka soal tidak berganti. Pada halaman menu utama terdapat juga tombol tantangan pada kotak nomor 3, jika memilih tantangan maka aplikasi akan menampilkan halaman pilih kelas tantangan. Pada halaman pilih kelas tantangan terdapat tombol untuk kembali ke halaman sebelumnya pada kotak nomor 1, tombol untuk memilih kelas mudah pada kotak nomor 2, tombol untuk memilih kelas sedang pada kotak nomor 3 dan tombol untuk memilih kelas sulit pada kotak nomor 4. Jika pengguna memilih salah satu kelas maka aplikasi akan menampilkan halaman mulai tantangan yang terdapat panduan tantangan, tombol *top score* pada kotak nomor 2 untuk melihat tiga daftar *top score* sesuai kelas yang dipilih dan tombol mulai untuk mulai tantangan. Jika pengguna menekan tombol *top score* maka aplikasi akan menampilkan halaman *top score*, pada halaman *top score* terdapat 3 daftar *top score* dengan urutan terbesar ke terkecil dan tombol untuk menutup halaman *top score* pada kotak nomor 2 kemudian aplikasi menampilkan halaman mulai tantangan. Pada halaman mulai tantangan, jika pengguna menekan tombol mulai maka aplikasi akan menampilkan halaman soal tantangan. Pada halaman soal tantangan terdapat tombol untuk kembali ke halaman

sebelumnya pada kotak nomor 1 , tombol untuk melewati soal ke soal berikutnya pada kotak nomor 4 dan tombol jawab pada kotak nomor 5 untuk menjawab soal, jika pengguna menekan tombol jawab dan pengguna telah menjawab soal kemudian aplikasi akan menampilkan halaman hasil jawab soal tantangan. Pada halaman hasil jawab soal tantangan terdapat tombol lanjut pada kotak nomor 3 untuk kembali ke halaman soal belajar, jika jawaban benar maka soal akan berganti dan jika salah maka soal tidak berganti tapi jika soal telah habis maka aplikasi akan menampilkan halaman total poin. Pada halaman soal tantangan jika waktu habis atau soal telah habis maka secara otomatis aplikasi akan menampilkan halaman total poin. Pada halaman total poin terdapat informasi total poin yang diperoleh pengguna, jika pengguna mendapat *top score* baru maka aplikasi akan menampilkan kolom input nama dan tombol simpan *top score* baru, terdapat juga tombol keluar pada kotak nomor 7 untuk keluar dari halaman total poin dan aplikasi akan menampilkan halaman pilih kelas tantangan.



Gambar 4.8 Screenflow

4.7.5 Perancangan Antarmuka

Pada perancangan antarmuka aplikasi belajar pengucapan kosakata bahasa Inggris memiliki 11 tampilan antarmuka yaitu halaman *splashscreen*, halaman utama, halaman pilih kategori pada fitur belajar, halaman soal pada fitur belajar, halaman jawab soal dan penilaian pada fitur belajar, halaman pilih kelas pada fitur tantangan, halaman mulai tantangan, halaman *top score*, halaman soal pada fitur tantangan, halaman jawab soal dan penilaian pada fitur tantangan, dan halaman total poin pada fitur tantangan.

4.7.5.1 Halaman *Splashscreen*



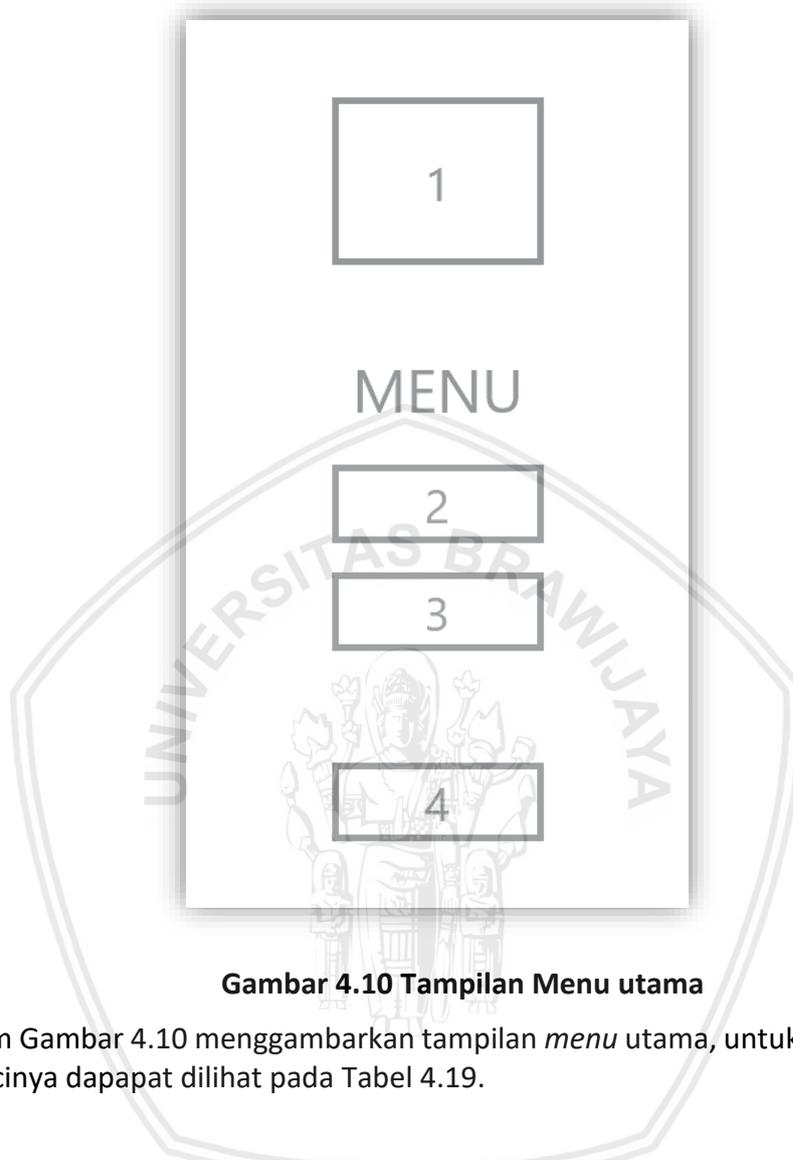
Gambar 4.9 Halaman *Splashscreen*

Dalam Gambar 4.9 menggambarkan tampilan yang pertama kali dijalankan ketika aplikasi dijalankan, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.18.

Tabel 4.18 Penjelasan Halaman Menu

No	Nama objek	Tipe	Keterangan
1	Logo aplikasi	Gambar	Logo aplikasi.
2	Nama aplikasi	Teks	Label nama aplikasi.

4.7.5.2 Halaman *Menu* Utama



Gambar 4.10 Tampilan Menu utama

Dalam Gambar 4.10 menggambarkan tampilan *menu* utama, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.19.

Tabel 4.19 Penjelasan Halaman Menu utama

No	Nama objek	Tipe	Keterangan
1	List tugas harian	Checkbox	List tiga checkbox untuk tugas harian.
2	Belajar	Tombol	Tombol untuk masuk ke fitur belajar.
3	Tantangan	Tombol	Tombol untuk masuk ke fitur tantangan.
4	Logo aplikasi	Gambar	Logo aplikasi.

4.7.5.3 Halaman Pilih Kategori



Gambar 4.11 Tampilan Halaman Pilih Kategori Kata

Dalam Gambar 4.11 menggambarkan tampilan halaman pilih kategori kata, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.20.

Tabel 4.20 Penjelasan Halaman Pilih Kategori Kata

No	Nama objek	Tipe	Keterangan
1	Kembali	Tombol	Tombol kembali ke halaman sebelumnya.
2	Kata kerja	Tombol	Tombol untuk masuk ke soal belajar kategori kata kerja.
3	Kata benda	Tombol	Tombol untuk masuk ke soal belajar kategori kata benda.
4	Kata sifat	Tombol	Tombol untuk masuk ke soal belajar kategori kata sifat.
5	Logo aplikasi	Gambar	Logo aplikasi

4.7.5.4 Halaman Soal Belajar



Gambar 4.12 Tampilan Halaman Soal Belajar

Dalam Gambar 4.12 menggambarkan tampilan halaman soal belajar, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.21.

Tabel 4.21 Penjelasan Halaman Soal Belajar

No	Nama objek	Tipe	Keterangan
1	Kembali	Tombol	Tombol kembali ke halaman sebelumnya.
2	Soal	Teks	Teks soal bahasa Inggris.
3	Dengarkan	Tombol	Tombol untuk mengeluarkan suara <i>audio</i> dari soal kosakata bahasa <i>Inggris</i> .
4	Bahasa Indonesia	Teks	Teks terjemahan Bahasa Indonesia dari soal Bahasa Inggris
5	Lewati	Tombol	Tombol untuk melewati soal.
6	Jawab	Tombol	Tombol untuk menjawab soal.
7	Jumlah kata benar	Teks	Jumlah jawaban yang benar dalam bentuk angka.

4.7.5.5 Halaman *Popup* Hasil Jawab Soal dan Penilaian Belajar



Gambar 4.13 Tampilan Halaman Hasil Jawab Soal dan Penilaian Belajar

Dalam Gambar 4.13 menggambarkan tampilan halaman jawab soal dan penilaian ucapan, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.22.

Tabel 4.22 Penjelasan Halaman Jawab Soal dan Penilaian Belajar

No	Nama objek	Tipe	Keterangan
1	Jawaban	Teks	Teks yang didapatkan dari jawaban pengguna.
2	Keterangan hasil jawab	Teks	Teks keterangan penilaian pengucapan pengguna berupa benar atau salah.
3	Lanjut	Tombol	Tombol untuk lanjut menjawab soal yang sama jika pengucapan salah dan jika pengucapan benar maka akan lanjut ke soal selanjutnya.
4	Keluar	Tombol	Tombol untuk keluar dari fitur belajar.

4.7.5.6 Halaman Pilih Kelas Tantangan



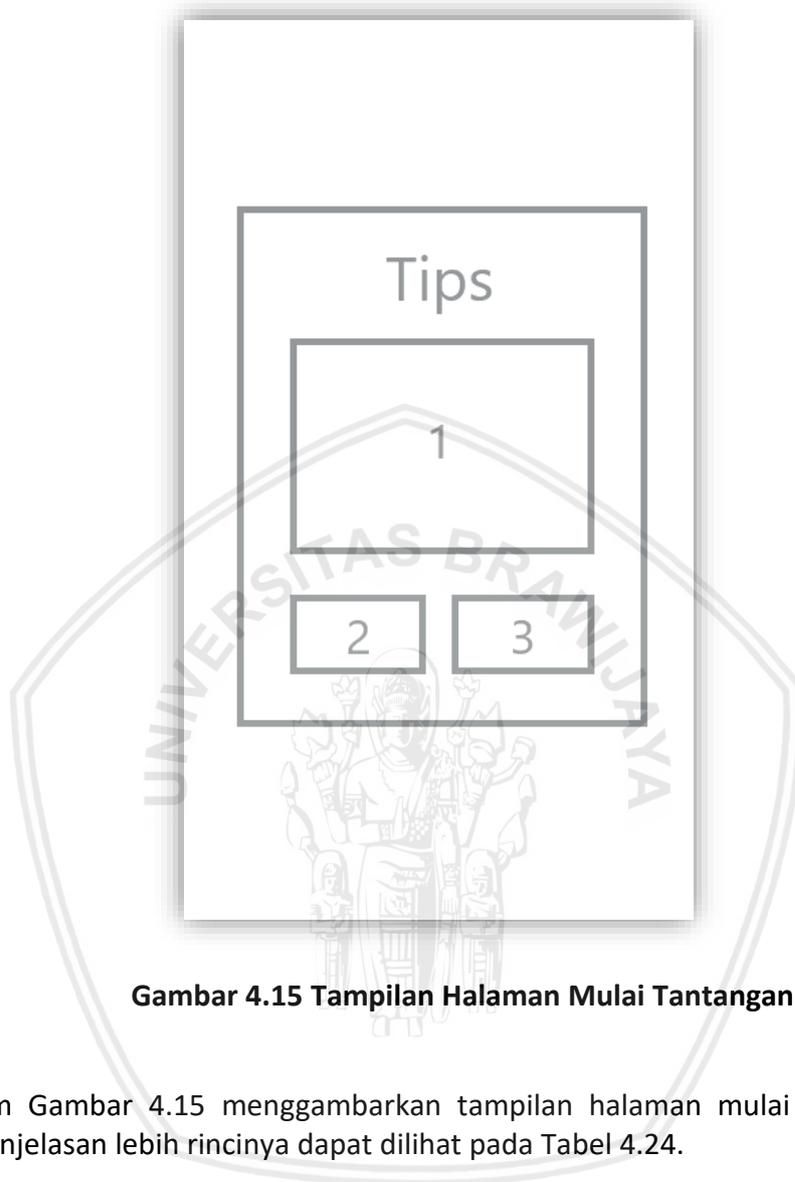
Gambar 4.14 Tampilan Halaman Pilih Kelas Tantangan

Dalam Gambar 4.14 menggambarkan tampilan pilih kelas tantangan, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.23.

Tabel 4.23 Penjelasan Halaman Kelas Tantangan

No	Nama objek	Tipe	Keterangan
1	Kembali	Tombol	Tombol untuk kembali ke halaman sebelumnya.
2	Mudah	Tombol	Tombol untuk masuk ke soal tantangan kelas mudah.
3	Sedang	Tombol	Tombol untuk masuk ke soal tantangan kelas sedang.
4	Sulit	Tombol	Tombol untuk masuk ke soal tantangan kelas sulit.
5	Logo aplikasi	Gambar	Logo aplikasi.

4.7.5.7 Halaman *Popup* Mulai Tantangan



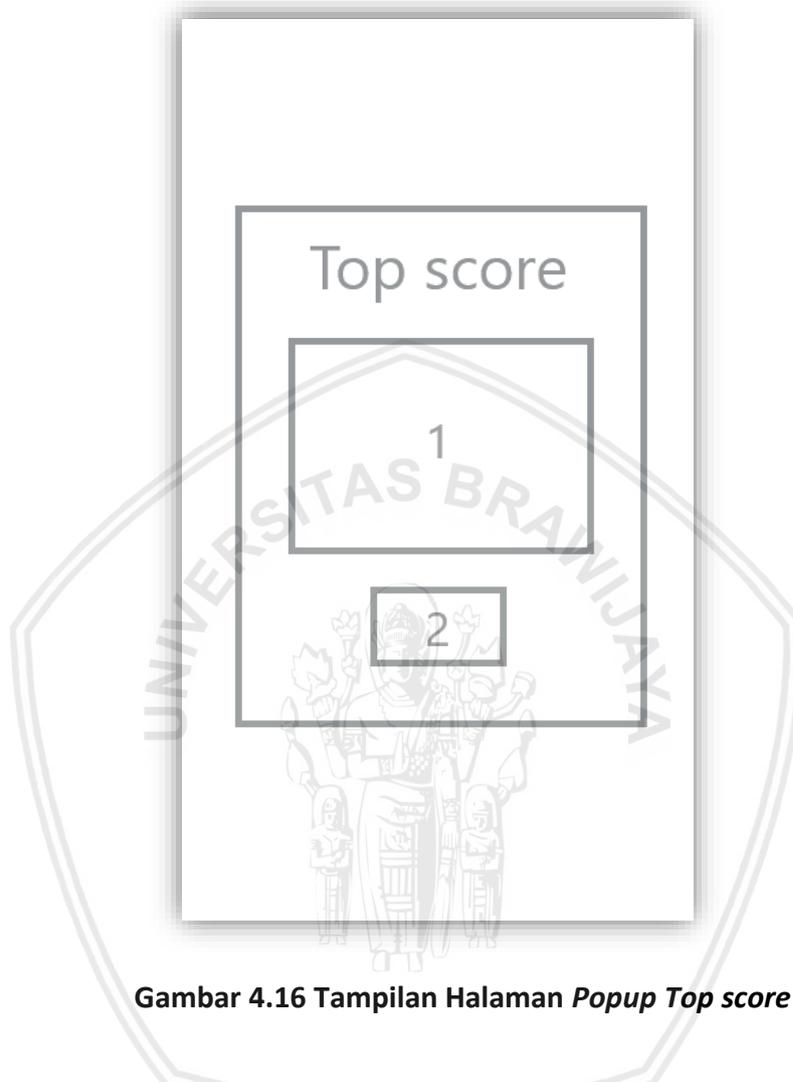
Gambar 4.15 Tampilan Halaman Mulai Tantangan

Dalam Gambar 4.15 menggambarkan tampilan halaman mulai tantangan, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.24.

Tabel 4.24 Penjelasan Halaman Mulai Tantangan

No	Nama objek	Tipe	Keterangan
1	Tips tantangan	<i>Text</i>	Pernjelasan mengerjakan soal tantangan.
2	<i>Top score</i>	Tombol	Tombol untuk menampilkan <i>top score</i> .
3	Mulai	Tombol	Tombol untuk mulai soal tantangan.

4.7.5.8 Halaman *Popup Top score*



Gambar 4.16 Tampilan Halaman *Popup Top score*

Dalam Gambar 4.16 menggambarkan tampilan halaman *list* tiga *top score* dari *point* terbesar ke terkecil, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.25.

Tabel 4.25 Penjelasan Halaman *Popup Top score*

No	Nama objek	Tipe	Keterangan
1	List top score 1, top score 2, top score 3	Teks	Urutan tiga top score dari terbesar ke terkecil.
2	Tutup	Tombol	Tombol untuk menutup popup top score.

4.7.5.9 Halaman Soal Tantangan



Gambar 4.17 Tampilan Halaman Soal Tantangan

Dalam Gambar 4.17 menggambarkan tampilan halaman soal tantangan, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.26.

Tabel 4.26 Penjelasan Halaman Soal Tantangan

No	Nama objek	Tipe	Keterangan
1	Kembali	Tombol	Tombol kembali ke halaman sebelumnya.
2	Soal	Teks	Teks soal bahasa Inggris.
3	Bahasa Indonesia	Teks	Teks terjemahan Bahasa Indonesia dari soal Bahasa Inggris
4	Lewati	Tombol	Tombol untuk melewati soal.
5	Jawab	Tombol	Tombol untuk menjawab soal.
6	Sisa waktu	Teks	Hitung mundur sisa waktu pengguna untuk menjalankan tantangan.
7	Jumlah kata benar	Teks	Jumlah jawaban yang benar dalam bentuk angka.

4.7.5.10 Halaman *Popup* Hasil Jawab Soal Tantangan



Gambar 4.18 Tampilan Halaman Hasil Jawab Soal dan Penilaian Tantangan

Dalam Gambar 4.18 menggambarkan tampilan halaman jawab soal soal tantangan dan penilaian, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.27.

Tabel 4.27 Penjelasan Halaman Jawab Soal dan Penilaian Tantangan

No	Nama objek	Tipe	Keterangan
1	Jawaban	Teks	Teks yang didapatkan dari jawaban pengguna.
2	Keterangan hasil jawab	Teks	Teks keterangan penilaian pengucapan pengguna berupa benar atau salah.
3	Lanjut	Tombol	Tombol untuk lanjut menjawab soal yang sama jika pengucapan salah dan jika pengucapan benar maka akan lanjut ke soal selanjutnya.

4.7.5.11 Halaman *Popup* Total Poin



Gambar 4.19 Tampilan Halaman Total Poin

Dalam Gambar 4.19 menggambarkan tampilan halaman total poin, untuk penjelasan lebih rincinya dapat dilihat pada Tabel 4.28.

Tabel 4.28 Penjelasan Halaman Total Poin

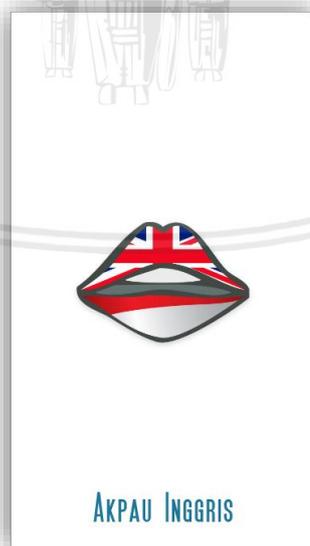
No	Nama objek	Tipe	Keterangan
1	<i>Rating</i>	<i>Rating bar</i>	Bintang yang di peroleh sesuai dengan total poin.
2	Total poin	Teks	Total poin tantangan yang didapatkan pengguna dalam bentuk angka.
3	Jumlah kata benar	Teks	Jumlah jawaban yang benar dalam bentuk angka.
4	Jumlah kata yang dilewati	Teks	Jumlah kata yang dilewati dalam bentuk angka.
5	Kolom <i>input</i> nama	<i>Text field</i>	Kolom input nama <i>top score</i> baru.
6	Simpan	Tombol	Tombol simppan <i>top score</i> baru.
7	Keluar	Tombol	Tombol keluar dari soal tantangan dan kembali ke halaman pilih kelas tantangan.

4.8 Implementasi *Prototype*

Implementasi *prototype* pada perancangan aplikasi ini berfokus pada tampilan antarmuka dengan menyesuaikan kebutuhan fungsional, *use case*, *activity diagram*, *screenflow* dan rancangan antarmuka aplikasi yang telah dijelaskan pada tahapan penggalian kebutuhan dan perancangan. Implementasi ini bertujuan untuk memberikan Gambaran secara umum bagaimana tampilan dan alir proses aplikasi saat digunakan oleh pengguna. Alir proses dari *prototype* disesuaikan dengan *screen flow* dan untuk tampilan antarmuka disesuaikan dengan perancangan antarmuka yang telah dibuat sebelumnya. Implementasi *prototype* ini menggunakan software adobe XD yang merupakan software khusus untuk implementasi *prototype* aplikasi perangkat lunak, software tersebut dapat secara langsung digunakan untuk simulasi dalam bentuk digital dari *prototype* sehingga pada saat pengujian *prototype* ke pengguna hasil pengujian akan lebih baik dikarenakan *prototype* aplikasi yang pengguna jalankan layaknya menjalankan aplikasi yang telah jadi.

Dalam implementasi *prototype* aplikasi ini memiliki sebelas tampilan antarmuka yang di implementasikan dalam bentuk *prototype* yaitu halaman *splashscreen*, halaman utama, halaman pilih kategori pada fitur belajar, halaman soal pada fitur belajar, halaman jawab soal dan penilaian pada fitur belajar, halaman pilih kelas pada fitur tantangan, halaman mulai tantangan, halaman soal pada fitur tantangan, halaman jawab soal dan penilaian pada fitur tantangan, dan halaman total poin pada fitur tantangan.

4.8.1 Halaman *Splashscreen*



Gambar 4.20 Halaman *Splashscreen*

Dalam Gambar 4.20 menggambarkan tampilan yang pertama kali dijalankan ketika aplikasi dijalankan.

4.8.2 Halaman Utama



Gambar 4.21 Tampilan Menu

Dalam Gambar 4.21 menggambarkan tampilan *menu* utama.

4.8.3 Halaman Pilih Kategori



Gambar 4.22 Tampilan Halaman Pilih Kategori Kata

Dalam Gambar 4.22 menggambarkan tampilan halaman pilih kategori kata.

4.8.4 Halaman Soal Belajar



Gambar 4.23 Tampilan Halaman Soal Belajar

Dalam Gambar 4.23 menggambarkan tampilan halaman soal belajar.

4.8.5 Halaman *Popup* Hasil Jawab Soal dan Penilaian Belajar



Gambar 4.24 Tampilan Halaman Hasil Jawab Soal dan Penilaian Belajar

Dalam Gambar 4.24 menggambarkan tampilan halaman jawab soal dan penilaian ucapan.

4.8.6 Halaman Pilih Kelas Tantangan



Gambar 4.25 Tampilan Halaman Pilih Kelas Tantangan

Dalam Gambar 4.25 menggambarkan tampilan pilih kelas tantangan.

4.8.7 Halaman *Popup* Mulai Tantangan



Gambar 4.26 Tampilan Halaman Mulai Tantangan

Dalam Gambar 4.26 menggambarkan tampilan halaman mulai tantangan.

4.8.8 Halaman *Popup Top score*



Gambar 4.27 Tampilan Halaman Mulai Tantangan

Dalam Gambar 4.27 menggambarkan tampilan halaman *list* tiga *top score* dari *point* terbesar ke terkecil.

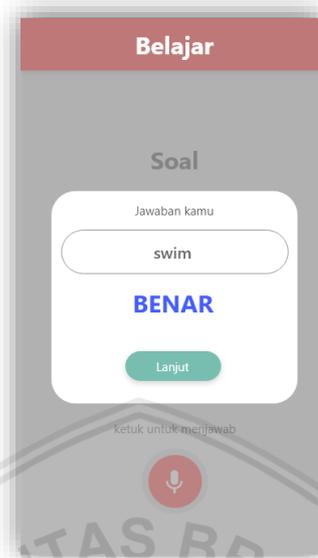
4.8.9 Halaman Soal Tantangan



Gambar 4.28 Tampilan Halaman Soal Tantangan

Dalam Gambar 4.28 menggambarkan tampilan halaman soal tantangan.

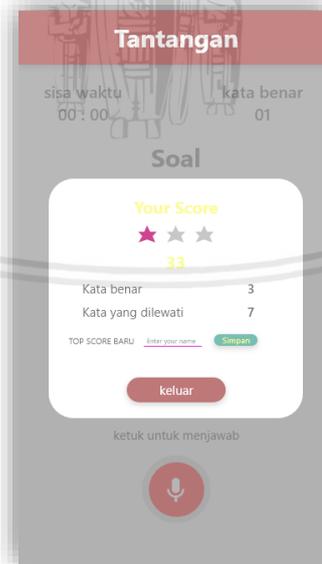
4.8.10 Halaman *Popup* Hasil Jawab Soal Tantangan



Gambar 4.29 Tampilan Halaman Hasil Jawab Soal dan Penilaian Tantangan

Dalam Gambar 4.29 menggambarkan tampilan halaman jawab soal soal tantangan dan penilaian.

4.8.11 Halaman *Popup* Total Poin



Gambar 4.30 Tampilan Halaman Total Poin

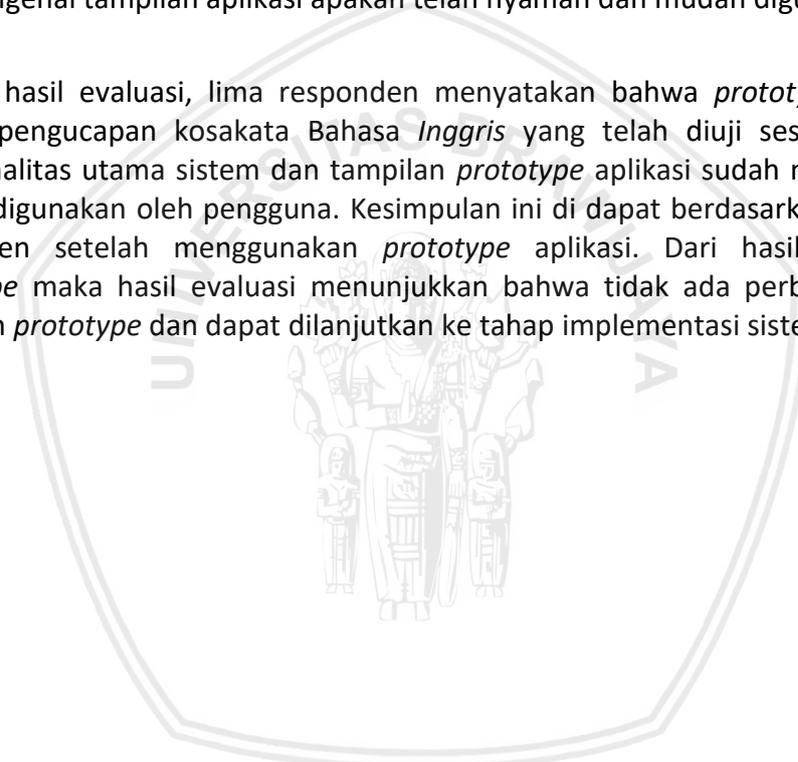
Dalam Gambar 4.30 menggambarkan tampilan halaman total poin.

4.9 Evaluasi *Prototype*

Pada metode pengembangan *prototyping* terdapat tahapan evaluasi *prototype* yang melibatkan calon pengguna untuk melakukan penilaian atau evaluasi terhadap *prototype* dari aplikasi belajar pengucapan kosakata Bahasa *Inggris*. Pada evaluasi *prototype* ini evaluator melibatkan sebanyak lima responden untuk melakukan penilaian terhadap *prototype*, responden merupakan lima orang yang sama yang menjadi responden pada tahap penggalan kebutuhan.

Evaluasi dilakukan dengan cara mengujikan *prototype* kepada responden dan kemudian penguji melakukan wawancara ke responden untuk mendapatkan *feedback* terhadap *prototype* yang telah mereka uji. Pengujian *prototype* hanya berfokus pada apakah fungsionalitas telah sesuai dengan rancangan dari aplikasi dan mengenai tampilan aplikasi apakah telah nyaman dan mudah digunakan oleh *user*.

Dari hasil evaluasi, lima responden menyatakan bahwa *prototype* aplikasi belajar pengucapan kosakata Bahasa *Inggris* yang telah diuji sesuai dengan fungsionalitas utama sistem dan tampilan *prototype* aplikasi sudah nyaman dan mudah digunakan oleh pengguna. Kesimpulan ini di dapat berdasarkan jawaban responden setelah menggunakan *prototype* aplikasi. Dari hasil pengujian *prototype* maka hasil evaluasi menunjukkan bahwa tidak ada perbaikan pada tampilan *prototype* dan dapat dilanjutkan ke tahap implementasi sistem.



BAB 5 IMPLEMENTASI

Implementasi dimulai dengan menjelaskan spesifikasi dari perangkat lunak dan keras yang digunakan untuk implementasi sistem. Dilanjutkan dengan menjabarkan tentang batasan-batasan implementasi, implementasi kelas dan antarmuka, dan implementasi kode program.

5.1 Spesifikasi Sistem

Pada bagian ini menjabarkan spesifikasi dari sistem yang digunakan untuk proses implementasi sistem.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi dari perangkat keras pada proses implementasi aplikasi belajar pengucapan kosakata Bahasa Inggris berupa laptop Asus dan *smartphone* LG G5, Untuk spesifikasi perangkat ditunjukkan pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1 Spesifikasi Perangkat Keras Laptop

Nama Komponen	Spesifikasi
Sistem Model	Laptop Asus X455LF
Prosesor	Intel(R) Core(TM) i5 CPU @ 2.20GHz (4 CPU), ~2.2GHz
Memory	8192 MB RAM
Graphic	Intel(R) HD Graphic 5500

Tabel 5.2 Spesifikasi Perangkat Keras Smartphone

Nama Komponen	Spesifikasi
Sistem Model	<i>Smartphone</i> LG G5
Prosesor	Quad-core (2x2.15 GHz Kryo & 2x1.6 GHz Kryo)
Memory	32 GB, 4 GB RAM
Graphic	1440 x 2560 pixels, 16:9 ratio (~554 ppi density)

Perangkat keras laptop Asus dengan spesifikasi pada Tabel 5.1 digunakan untuk implementasi kode program sedangkan perangkat keras *smartphone* LG G5 dengan spesifikasi seperti pada Tabel 5.2 digunakan untuk proses *debug* kode, instalasi dan pengujian sistem.

5.1.2 Spesifikasi Perangkat Lunak

Bagian ini menjelaskan spesifikasi dari perangkat lunak yang digunakan pada proses implementasi aplikasi aplikasi belajar pengucapan pengucapan kosakata bahasa *Inggris*. Pada Tabel 5.3 menjabarkan spesifikasi dari perangkat lunak yang digunakan untuk implementasi kode program sedangkan pada Tabel 5.4 menjabarkan spesifikasi dari perangkat lunak untuk melakukan instalasi dan pengujian aplikasi.

Tabel 5.3 Spesifikasi Perangkat Lunak Laptop

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10
Bahasa pemograman	Java
Editor	Android Studio

Tabel 5.3 Spesifikasi Perangkat Lunak Smartphone

Nama Komponen	Spesifikasi
Sistem Operasi	Android Nougat 7.0

5.2 Batasan-batasan Implementasi

Pada implementasi aplikasi aplikasi belajar pengucapan pengucapan kosakata Bahasa *Inggris*, mempunyai batasan-batasan sebagai berikut:

1. Pengembangan menggunakan IDE Android Studio.
2. Implementasi kode program menggunakan Bahasa pemograman Java.
3. Implementasi *voice recognition* menggunakan *Google speech API*.
4. Aplikasi dapat berjalan pada sistem operasi android minimal versi *Ice Cream Sandwich 4.0*.

5.3 Implementasi Kelas dan Antarmuka

Pada bagian ini membahas implementasi kelas dan antarmuka pada aplikasi belajar pengucapan kosakata bahasa Inggris menggunakan *google voice reconition*. Implementasi kelas dan antarmuka mengacu pada perancangan diagram kelas yang ditunjukkan pada Gambar 4.5. Kelas program di implementasikan dalam bentuk file program dengan ekstensi file *.java, sedangkan untuk antarmuka diimplementasi dalam bentuk ekstensi file *.xml.

5.3.1 Implementasi Kelas

Implementasi kelas pada aplikasi belajar pengucapan kosakata bahasa Inggris mengacu pada diagram kelas pada Gambar 4.5. Implementasi kelas pada aplikasi dijabarkan pada Tabel 5.4.

Tabel 5.4 Implementasi Kelas

Nama File Program	Keterangan
<code>splashActivity</code>	Kelas yang dijalankan pertama kali yang dijalankan untuk menampilkan halaman <code>activity_splash</code> .
<code>MainActivity</code>	Kelas yang otomatis dijalankan setelah kelas <code>splashActivity</code> , kelas ini untuk menampilkan halaman <code>activity_main</code> dan juga melakukan perubahan atau <i>update</i> tugas harian.
<code>categoriBelajarActivity</code>	Kelas untuk menampilkan halaman <code>activity_categori_belajar</code> .
<code>belajarActivity</code>	Kelas untuk menampilkan soal belajar dan implementasi <i>google voice recognition</i> untuk menjawab soal, kelas ini untuk menampilkan halaman <code>activity_belajar</code> dan <code>pop_up_hasil_jawab_belajar</code> .
<code>kelasTantanganActivity</code>	Kelas ini untuk menampilkan halaman <code>activity_kelas_tantangan</code> , <code>pop_up_mulai_tantangan</code> dan <code>pop_up_top_score</code> .
<code>tantanganActivity</code>	Kelas untuk menampilkan soal tantangan dan implementasi <i>google voice recognition</i> untuk menjawab soal, kelas ini untuk menampilkan halaman <code>activity_tantangan</code> , <code>pop_up_hasil_jawab_tantangan</code> , dan <code>pop_up_poin_tantangan</code> .
<code>myDatabaseHandler</code>	Kelas merupakan kelas untuk aktivitas yang berkaitan dengan penyimpanan <i>internal</i> aplikasi dan koneksi ke <i>firebase</i> untuk <i>download</i> data kosakata.

5.3.2 Implementasi Antarmuka

Antarmuka aplikasi belajar pengucapan kosakata bahasa Inggris diimplementasikan dalam bentuk *extention file* *.xml dan disesuaikan dengan perancangan aplikasi yang telah dibuat sebelumnya. Antarmuka pada aplikasi belajar pengucapan bahasa Inggris dijabarkan pada Tabel 5.5

Tabel 5.5 Implementasi Antarmuka

Nama File Program	Keterangan
<i>activity_splash</i>	Halaman <i>splashscreen</i> .
<i>activity_main</i>	Halaman utama aplikasi.
<i>activity_kategori_belajar</i>	Halaman pilih kategori soal belajar.
<i>activity_belajar</i>	Halaman latihan soal belajar.
<i>pop_up_hasil_jawab_belajar</i>	Halaman <i>popup</i> hasil jawaban soal belajar.
<i>activity_kelas_tantangan</i>	Halaman pilih kelas soal tantangan.
<i>pop_up_mulai_tantangan</i>	Halaman <i>popup</i> penjelasan tantangan dan untuk memulai soal tantangan.
<i>pop_up_top_score</i>	Halaman <i>list</i> tiga <i>top score</i> .
<i>activity_tantangan</i>	Halaman soal tantangan.
<i>pop_up_hasil_jawab_tantangan</i>	Halaman <i>popup</i> hasil jawaban soal tantangan.
<i>pop_up_poin_tantangan</i>	Halaman <i>popup</i> total poin tantangan.

Berikut ini penjelasan lebih detail antarmuka aplikasi belajar pengucapan kosakata bahasa Inggris, antara lain:

a. Halaman *activity_splash*



Gambar 5.1 Halaman *activity_splash*

Halaman ini ditampilkan pada saat pertama kali aplikasi dijalankan dan secara otomatis berpindah ke halaman utama, halaman ini akan menampilkan *progress bar* yang menunjukkan *loading* proses *download* data dari *firebase* ke penyimpanan internal aplikasi yang dilakukan ketika aplikasi pertama kali dijalankan setelah *install* dan untuk *progress bar* selanjutnya hanya akan menunjukkan *delay* sebelum ke halaman utama. Halaman *activity_splash* ditunjukkan dalam Gambar 5.1.

b. Halaman *activity_main*

Halaman ini merupakan halaman utama yang memunculkan fitur utama dari aplikasi belajar kosakata bahasa Inggris, yaitu: halaman belajar dan tantangan. Selain itu pada halaman ini juga ditampilkan fitur tugas harian yang berjumlah tiga. Halaman *activity_main* ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Halaman *activity_main*

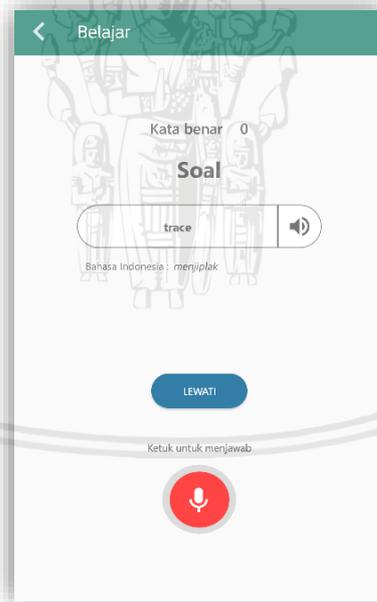
c. Halaman *activity_kategori_belajar*

Halaman ini merupakan halaman yang menampilkan tiga kategori soal untuk dipilih oleh pengguna, yaitu: kategori kata benda, kata kerja dan kata sifat. Halaman *activity_kategori_belajar* ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Halaman *activity_categori_belajar*

d. Halaman *activity_belajar*



Gambar 5.4 Halaman *activity_belajar*

Halaman ini merupakan halaman yang menampilkan soal latihan belajar yang sesuai dengan pilihan kategori pengguna. Pada halaman ini juga ditampilkan tombol untuk mendengar soal, tombol lewati soal, jawab soal, jumlah soal yang dijawab dengan benar dan bahasa Indonesia dari soal kosakata bahasa Inggris yang ditampilkan. Halaman *activity_belajar* ditunjukkan dalam Gambar 5.4.

e. Halaman `pop_up_hasil_jawab_belajar`

Halaman ini merupakan halaman *popup* yang menampilkan jawab pengguna beserta keterangan benar atau salah dari jawaban pengguna, jika jawaban benar menampilkan keterangan benar berwarna hijau dan jika jawaban salah maka menampilkan keterangan salah berwarna merah. Pada halaman ini juga menampilkan tombol keluar dan lanjut soal latihan. Halaman `pop_up_hasil_jawab_belajar` ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Halaman `pop_up_hasil_jawab_belajar`

f. Halaman `activity_kelas_tantangan`



Gambar 5.6 Halaman `activity_kelas_tantangan`

Halaman ini merupakan halaman yang menampilkan tiga kelas soal tantangan untuk dipilih oleh pengguna, yaitu: kelas mudah, sedang dan sulit. Halaman `activity_kelas_tantangan` ditunjukkan dalam Gambar 5.6.

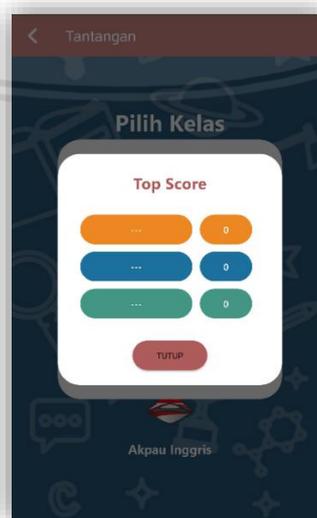
- g. Halaman `pop_up_mulai_tantangan`



Gambar 5.7 Halaman `pop_up_mulai_tantangan`

Halaman ini merupakan halaman *popup* setelah memilih kelas tantangan yang halaman ini menampilkan penjelasan tantangan. Pada halaman ini ditampilkan juga tombol untuk memulai tantangan dan tombol untuk melihat tiga *top score* dari kelas yang dipilih. Halaman `pop_up_mulai_tantangan` ditunjukkan dalam Gambar 5.7.

- h. Halaman `pop_up_top_score`



Gambar 5.8 Halaman `pop_up_top_score`

Halaman ini merupakan halaman *popup* yang menampilkan *list* tiga *top score* berdasarkan kelas yang dipilih. Halaman `pop_up_top_score` ditunjukkan dalam Gambar 5.8.

i. Halaman `activity_tantangan`

Halaman ini merupakan halaman yang menampilkan soal tantangan yang sesuai dengan pilhan kelas pengguna. Pada halaman ini juga ditampilkan tombol untuk tombol lewati soal, jawab soal, jumlah soal yang dijawab dengan benar, bahasa Indonesia dari soal kosakata bahasa Inggris yang ditampilkan, dan sisa waktu tantangan. Halaman `activity_tantangan` ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Halaman `activity_tantangan`

j. Halaman `pop_up_hasil_jawab_tantangan`

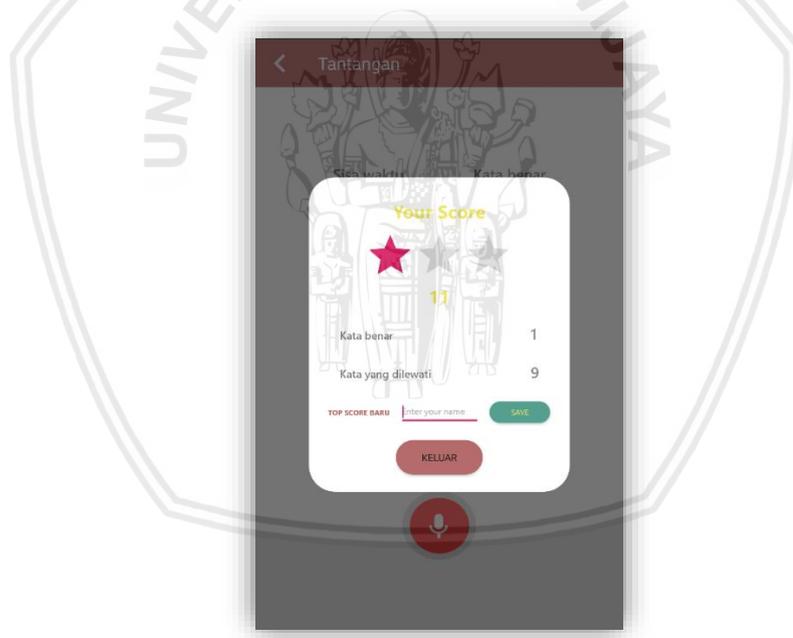


Gambar 5.10 Halaman `pop_up_hasil_jawab_tantangan`

Halaman ini merupakan halaman *popup* yang menampilkan jawab pengguna beserta keterangan benar atau salah dari jawaban pengguna, jika jawaban benar menampilkan keterangan benar berwarna hijau dan jika jawaban salah maka menampilkan keterangan salah berwarna merah. Pada halaman ini juga menampilkan tombol lanjut soal. Halaman `pop_up_hasil_jawab_tantangan` ditunjukkan dalam Gambar 5.10.

k. Halaman `pop_up_poin_tantangan`

Halaman ini merupakan halaman *popup* yang menampilkan total *poin* tantangan yang diperoleh oleh pengguna beserta menampilkan rekapitulasi jumlah soal benar dan jumlah soal yang dilewati. Halaman ini juga menampilkan bintang yang diperoleh sesuai total poin, tombol keluar, jika pengguna memperoleh total *poin* yang menjadi tiga *top score* pada kelas yang dipilih maka akan menampilkan kolom *input* nama pengguna untuk disimpan sebagai nama tiga *top score* yang baru beserta tombol simpan *top score*, namun jika total poin yang diperoleh tidak masuk *top score* baru maka halaman *popup* tidak menampilkan kolom *input* nama dan tombol simpan *top score* baru. Halaman `pop_up_poin_tantangan` ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Halaman `pop_up_poin_tantangan`

5.4 Implementasi Kode Program

Pada bagian ini menjelaskan implementasi kode program pada kelas JAVA yang dapat menjalankan fungsi tertentu. Yang dicantumkan dalam penelitian ini yaitu kode program membuat atau *update* tugas harian, mempersiapkan dan menampilkan soal kosakata bahasa Inggris pada fitur tantangan, menjawab soal dan menampilkan hasil jawaban, menampilkan total *poin* tantangan, dan menampilkan *top score*.

5.4.1 Implementasi Kode Program Membuat Atau *Update* Tugas Harian

Tugas harian bertujuan untuk memberikan tugas kepada pengguna untuk lebih aktif dalam menggunakan aplikasi belajar pengucapan kosakata bahasa Inggris. Ketika tugas telah selesai dikerjakan maka *list* tugas akan di *checklist*, tugas harian tiap pergantian tanggal akan berganti secara otomatis.

Tabel 5.6 Implementasi Kode Program Tugas Harian Pada Kelas MainActivity

1	CheckBox tugas1, tugas2, tugas3;
2	
3	myDatabaseHandler tugas_harian = new myDatabaseHandler(this);
4	
5	@Override
6	protected void onCreate(Bundle savedInstanceState) {
7	super.onCreate(savedInstanceState);
8	setContentView(R.layout.activity_main);
9	tugas1 = findViewById(R.id.tugas1);
10	tugas2 = findViewById(R.id.tugas2);
11	tugas3 = findViewById(R.id.tugas3);
12	
13	Calendar cal = Calendar.getInstance();
14	cal.clear(Calendar.HOUR_OF_DAY);
15	cal.clear(Calendar.AM_PM);
16	cal.clear(Calendar.HOUR);
17	cal.clear(Calendar.MINUTE);
18	cal.clear(Calendar.SECOND);
19	cal.clear(Calendar.MILLISECOND);
20	Date current_date = cal.getTime();
21	String temp_date = String.valueOf(current_date);
22	
23	ArrayList<ArrayList<String>> temp_tugasharian;
24	temp_tugasharian = new
25	ArrayList<ArrayList<String>>(tugas_harian.getTugasHarian());
26	
27	
28	if(!temp_date.equalsIgnoreCase(temp_tugasharian.get(0).get(4))) {
29	tugas_harian.bersihkanTugasHarian();
30	temp_tugasharian.removeAll(temp_tugasharian);
31	
32	temp_tugasharian.addAll(tugas_harian.getListTugas("belajar"));
33	for (int i = 0; i < 2; i++){
34	String deskripsi =
35	temp_tugasharian.get(i).get(0);
36	String status = "belum selesai";
37	String catoekel = temp_tugasharian.get(i).get(2);
38	int jumsoaljawab =
39	Integer.parseInt(temp_tugasharian.get(i).get(3));
40	tugas_harian.insertDataTugasHarian(deskripsi,
41	status, catoekel, jumsoaljawab, current_date);
42	}
43	temp_tugasharian.removeAll(temp_tugasharian);
44	
45	temp_tugasharian.addAll(tugas_harian.getListTugas("tantangan"));
46	String deskripsi = temp_tugasharian.get(0).get(0);
47	String status = "belum selesai";
48	String catoekel = temp_tugasharian.get(0).get(2);
49	int jumsoaljawab =
50	Integer.parseInt(temp_tugasharian.get(0).get(3));
51	tugas_harian.insertDataTugasHarian(deskripsi, status,
52	catoekel, jumsoaljawab, current_date);
53	}
54	

Tabel 5.6 Implementasi Kode Program Tugas Harian Pada Kelas MainActivity (lanjutan)

55	<code>temp_tugasharian.removeAll(temp_tugasharian);</code>
56	<code>temp_tugasharian = new</code>
57	<code>ArrayList<ArrayList<String>>(tugas_harian.getTugasHarian());</code>
58	
59	<code>tugas1.setText(temp_tugasharian.get(0).get(0));</code>
60	<code>tugas1.setChecked(false);</code>
61	<code>tugas2.setText(temp_tugasharian.get(1).get(0));</code>
62	<code>tugas2.setChecked(false);</code>
63	<code>tugas3.setText(temp_tugasharian.get(2).get(0));</code>
64	<code>tugas3.setChecked(false);</code>
65	
66	<code>if(temp_tugasharian.get(0).get(1).equalsIgnoreCase("selesai")){</code>
67	<code>tugas1.setChecked(true);</code>
68	<code>}</code>
69	
70	<code>if(temp_tugasharian.get(1).get(1).equalsIgnoreCase("selesai")){</code>
71	<code>tugas2.setChecked(true);</code>
72	<code>}</code>
73	
74	<code>if(temp_tugasharian.get(2).get(1).equalsIgnoreCase("selesai")){</code>
75	<code>tugas3.setChecked(true);</code>
76	<code>}</code>
77	
78	

Pada Tabel 5.6 merupakan kode untuk tugas harian pada kelas `MainActivity`, pada baris 16 sampai 24 merupakan kode program untuk mendapatkan tanggal saat aplikasi dijalankan. Baris 31 sampai 57 yaitu kode program untuk melakukan *update* atau mengganti tugas harian yang lama dengan yang baru jika tanggal telah berganti hari. Baris 70 sampai 80 yaitu kode program untuk melakukan *update* jika tugas telah selesai dikerjakan yaitu status *checkbox* yang sebelumnya *unchecked* ke *checked*.

Tabel 5.7 Implementasi Tugas Harian Pada Kelas myDatabaseHandler

1	<code>public void insertDataTugasHarian(String deskripsi, String</code>
2	<code>status, String catorkel, int jumsoaljawab, Date tgl_mulai) {</code>
3	<code>final SQLiteDatabase db = this.getWritableDatabase();</code>
4	<code>ContentValues contentValues = new ContentValues();</code>
5	<code>contentValues.clear();</code>
6	<code>contentValues.put("Deskripsi", deskripsi);</code>
7	<code>contentValues.put("Status", status);</code>
8	<code>contentValues.put("CatOrKel", catorkel);</code>
9	<code>contentValues.put("Jumsoaljawab", jumsoaljawab);</code>
10	<code>contentValues.put("Tgl_mulai", String.valueOf(tgl_mulai));</code>
11	<code>db.insert(TABLE_NAME_TUGASHARIAN, null, contentValues);</code>
12	<code>}</code>
13	
14	<code>public ArrayList<ArrayList<String>> getListTugas(String ket) {</code>
15	<code>final SQLiteDatabase db = this.getReadableDatabase();</code>
16	<code>String selectQuery = null;</code>
17	<code>if (ket.equalsIgnoreCase("belajar")) {</code>
18	<code>selectQuery = "SELECT Deskripsi, Ket, CatOrKel,</code>
	<code>Jumsoaljawab FROM " + TABLE_NAME_LIST_TUGASHARIAN + " WHERE Ket =</code>
	<code>'belajar' ORDER BY RANDOM() LIMIT 2";</code>
19	<code>} else if (ket.equalsIgnoreCase("tantangan")) {</code>
20	<code>selectQuery = "SELECT Deskripsi, Ket, CatOrKel,</code>
	<code>Jumsoaljawab FROM " + TABLE_NAME_LIST_TUGASHARIAN + " WHERE Ket =</code>

Tabel 5.7 Implementasi Tugas Harian Pada Kelas `myDatabaseHandler` (lanjutan)

```

21 'tantangan' ORDER BY RANDOM() LIMIT 1";
22     }
23     Cursor cursor;
24     cursor = db.rawQuery(selectQuery, null);
25     cursor.moveToFirst();
26     tugas.clear();
27     while (!cursor.isAfterLast()) {
28         tugas.add(new ArrayList<String>());
29         tugas.get(tugas.size() - 1).add(0, cursor.getString(0));
30         tugas.get(tugas.size() - 1).add(1, cursor.getString(1));
31         tugas.get(tugas.size() - 1).add(2, cursor.getString(2));
32         tugas.get(tugas.size() - 1).add(3, cursor.getString(3));
33         cursor.moveToNext();
34     }
35     cursor.close();
36     return tugas;
37 }
38 public ArrayList<ArrayList<String>> getTugasHarian() {
39     final SQLiteDatabase db = this.getReadableDatabase();
40     String selectQuery = "SELECT Deskripsi, Status, CatOrKel,
41     Jumsoaljawab, Tgl_mulai FROM " + TABLE_NAME_TUGASHARIAN;
42     Cursor cursor = db.rawQuery(selectQuery, null);
43     cursor.moveToFirst();
44     tugas_harian.clear();
45     while (!cursor.isAfterLast()) {
46         tugas_harian.add(new ArrayList<String>());
47         tugas_harian.get(tugas_harian.size() - 1).add(0,
48     cursor.getString(0));
49         tugas_harian.get(tugas_harian.size() - 1).add(1,
50     cursor.getString(1));
51         tugas_harian.get(tugas_harian.size() - 1).add(2,
52     cursor.getString(2));
53         tugas_harian.get(tugas_harian.size() - 1).add(3,
54     cursor.getString(3));
55         tugas_harian.get(tugas_harian.size() - 1).add(4,
56     cursor.getString(4));
57         cursor.moveToNext();
58     }
59     cursor.close();
60     return tugas_harian;
61 }
62 public void updateStatusTugasHarian(String catorkel, int
63     jumsoaljawab){
64     SQLiteDatabase db = this.getWritableDatabase();
65     db.execSQL("UPDATE " + TABLE_NAME_TUGASHARIAN + " SET Status
66     = 'selesai' WHERE CatOrKel = '"+ catorkel +" ' AND Jumsoaljawab =
67     '"+jumsoaljawab+" ' ");
68 }
69 }
70 public void bersihkanTugasHarian() {
71     SQLiteDatabase db = this.getWritableDatabase();
72     db.execSQL("DELETE FROM " + TABLE_NAME_TUGASHARIAN);
73 }

```

Pada Tabel 5.7 merupakan implementasi kode program tugas harian pada kelas `myDatabaseHandler` untuk mengakses penyimpanan *internal* pada aplikasi dengan nama database `akpauinggris.db`. Pada baris 1 sampai 12 merupakan kode

program dengan metode untuk *insert* data tugas harian yang baru ke tabel Tugas_harian pada database internal aplikasi. Pada baris 14 sampai 36 merupakan kode program dengan metode untuk mendapatkan tiga *list* tugas harian secara random dari tabel *List_tugas_harian*. Baris 38 sampai 55 merupakan kode program dengan metode untuk mengambil data dari tabel Tugas_harian. Baris 57 sampai 60 merupakan kode program dengan metode untuk melakukan *update* kolom status pada tabel Tugas_harian. Baris 62 sampai 65 merupakan kode program untuk menghapus data pada tabel Tugas_harian.

5.4.2 Implementasi Kode Program Membuat dan Menampilkan Soal Tantangan

Pada bagian ini menjelaskan kode program untuk membuat dan menampilkan soal tantangan yang akan ditampilkan ke pengguna sesuai dengan pilihan kelas yang dipilih. Soal tantangan dibuat pada kelas *kelasTantanganActivity* kemudian soal dikirim ke kelas *tantanganActivity*.

Tabel 5.8 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas kelasTantanganActivity

1	<code>Button button_kelas_mudah, button_kelas_c_sulit, button_kelas_sulit;</code>
2	<code>Dialog popMulai, popTS;</code>
3	<code>Intent intent;</code>
4	<code>myDatabaseHandler soal_tantangan = new myDatabaseHandler(this);</code>
5	
6	<code>@Override</code>
7	<code>protected void onCreate(Bundle savedInstanceState) {</code>
8	<code> super.onCreate(savedInstanceState);</code>
9	<code> setContentView(R.layout.activity_kelas_tantangan);</code>
10	
11	<code> soal_tantangan.setSoal_t_mudah();</code>
12	<code> soal_tantangan.setSoal_t_cukup_sulit();</code>
13	<code> soal_tantangan.setSoal_t_sulit();</code>
14	
15	<code> button_kelas_mudah = (Button)</code>
16	<code> findViewById(R.id.buttonKelasMudah);</code>
17	<code> button_kelas_mudah.setOnClickListener(new</code>
18	<code> View.OnClickListener() {</code>
19	<code> @Override</code>
20	<code> public void onClick(View v) {</code>
21	<code> if (soal_tantangan.getSoal_t_mudah().size() == 10) {</code>
22	<code> ArrayList<ArrayList<String>> temp_soal;</code>
23	<code> temp_soal = new</code>
24	<code> ArrayList<ArrayList<String>>(soal_tantangan.getSoal_t_mudah());</code>
25	<code> nextActivityTantangan("mudah", temp_soal);</code>
26	<code> } else {</code>
27	<code> Toast toast =</code>
28	<code> Toast.makeText(getApplicationContext(),</code>
29	<code> "wait a second and press again!",</code>
30	<code> Toast.LENGTH_SHORT);</code>
31	<code> toast.show();</code>
32	<code> }</code>
33	<code> }</code>
34	<code> });</code>
35	<code> button_kelas_c_sulit = (Button)</code>
	<code> findViewById(R.id.buttonCSulit);</code>
	<code> button_kelas_c_sulit.setOnClickListener(new</code>

Tabel 5.8 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas kelasTantanganActivity (lanjutan)

```

36 View.OnClickListener() {
37     @Override
38     public void onClick(View v) {
39         if (soal_tantangan.getSoal_t_cukup_sulit().size() ==
40         10) {
41             ArrayList<ArrayList<String>> temp_soal;
42             temp_soal = new
43             ArrayList<ArrayList<String>>(soal_tantangan.getSoal_t_cukup_sulit
44             ());
45             nextActivityTantangan("sedang", temp_soal);
46         } else {
47             Toast toast =
48             Toast.makeText(getApplicationContext(),
49             "wait a second and press again!",
50             Toast.LENGTH_SHORT);
51             toast.show();
52         }
53     }
54 });
55
56 button_kelas_sulit = (Button)
57 findViewById(R.id.buttonKelasSulit);
58 button_kelas_sulit.setOnClickListener(new
59 View.OnClickListener() {
60     @Override
61     public void onClick(View v) {
62         if (soal_tantangan.getSoal_t_sulit().size() == 10) {
63             ArrayList<ArrayList<String>> temp_soal;
64             temp_soal = new
65             ArrayList<ArrayList<String>>(soal_tantangan.getSoal_t_sulit());
66             nextActivityTantangan("sulit", temp_soal);
67         } else {
68             Toast toast =
69             Toast.makeText(getApplicationContext(),
70             "wait a second and press again!",
71             Toast.LENGTH_SHORT);
72             toast.show();
73         }
74     }
75 });
76
77 public void nextActivityTantangan(String kelas,
78 ArrayList<ArrayList<String>> soal) {
79     intent = new Intent(this, tantanganActivity.class);
80     if (kelas.equals("mudah")) {
81         intent.putExtra(kelas, soal);
82         intent.putExtra("status", kelas);
83     } else if (kelas.equals("sedang")) {
84         intent.putExtra(kelas, soal);
85         intent.putExtra("status", kelas);
86     } else if (kelas.equals("sulit")) {
87         intent.putExtra(kelas, soal);
88         intent.putExtra("status", kelas);
89     }
90     showPopupMulai(kelas);
91 }

```

Pada Tabel 5.8 merupakan implementasi kode program membuat soal tantangan pada kelas `kelasTantanganActivity`. Pada baris 11 sampai 13 merupakan kode program untuk mendapatkan soal tantangan dari tiga kelas yaitu mudah, sedang, dan sulit. Pada baris 15 sampai 71 merupakan kode program untuk memilih satu diantara tiga kelas menggunakan tombol dan kelas yang dipilih akan mengirimkan data soal kosakata berdasarkan kelas yang dipilih ke aktivitas yang akan dijalankan selanjutnya. Baris 74 sampai 88 merupakan kode program dengan metode untuk berpindah dan mengirimkan soal tantangan ke aktivitas selanjutnya.

Tabel 5.9 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas `myDatabaseHandler`

```

1  public void setSoal_t_mudah() {
2      SQLiteDatabase db = this.getReadableDatabase();
3      String selectQuery = "SELECT B_Ing, B_Indo FROM " +
4      TABLE_NAME_KOSAKATA + " WHERE LENGTH(B_Ing) < 6 ORDER BY RANDOM()
5      LIMIT 10";
6      Cursor cursor = db.rawQuery(selectQuery, null);
7      cursor.moveToFirst();
8
9      while (!cursor.isAfterLast()) {
10         soal_t_mudah.add(new ArrayList<String>());
11         soal_t_mudah.get(soal_t_mudah.size() - 1).add(0,
12         cursor.getString(0));
13         soal_t_mudah.get(soal_t_mudah.size() - 1).add(1,
14         cursor.getString(1));
15         cursor.moveToNext();
16     }
17     cursor.close();
18 }
19
20 public void setSoal_t_cukup_sulit() {
21     SQLiteDatabase db = this.getReadableDatabase();
22     String selectQuery = "SELECT B_Ing, B_Indo FROM " +
23     TABLE_NAME_KOSAKATA + " WHERE LENGTH(B_Ing) > 5 AND LENGTH(B_Ing)
24     < 9 ORDER BY RANDOM() LIMIT 10";
25     Cursor cursor = db.rawQuery(selectQuery, null);
26     cursor.moveToFirst();
27
28     while (!cursor.isAfterLast()) {
29         soal_t_sedang.add(new ArrayList<String>());
30         soal_t_sedang.get(soal_t_sedang.size() - 1).add(0,
31         cursor.getString(0));
32         soal_t_sedang.get(soal_t_sedang.size() - 1).add(1,
33         cursor.getString(1));
34         cursor.moveToNext();
35     }
36     cursor.close();
37 }
38
39 public void setSoal_t_sulit() {
40     SQLiteDatabase db = this.getReadableDatabase();
41     String selectQuery = "SELECT B_Ing, B_Indo FROM " +
42     TABLE_NAME_KOSAKATA + " WHERE LENGTH(B_Ing) > 8 ORDER BY RANDOM()
43     LIMIT 10";
44     Cursor cursor = db.rawQuery(selectQuery, null);
45     cursor.moveToFirst();
46
47     while (!cursor.isAfterLast()) {
48         soal_t_sulit.add(new ArrayList<String>());
49         soal_t_sulit.get(soal_t_sulit.size() - 1).add(0,

```

Tabel 5.9 Implementasi Kode Program Membuat Soal Tantangan Pada Kelas myDatabaseHandler (lanjutan)

```

40 cursor.getString(0));
    soal_t_sulit.get(soal_t_sulit.size() - 1).add(1,
41 cursor.getString(1));
    cursor.moveToNext();
42 }
43 cursor.close();
44 }
45
46 public ArrayList<ArrayList<String>> getSoal_b_kerja() {
47     return soal_b_kerja;
48 }
49
50 public ArrayList<ArrayList<String>> getSoal_b_benda() {
51     return soal_b_benda;
52 }
53
54 public ArrayList<ArrayList<String>> getSoal_b_sifat() {
55     return soal_b_sifat;
56 }
57
58 public ArrayList<ArrayList<String>> getSoal_t_mudah() {
59     return soal_t_mudah;
60 }
61
62 public ArrayList<ArrayList<String>> getSoal_t_cukup_sulit() {
63     return soal_t_sedang;
64 }
65
66 public ArrayList<ArrayList<String>> getSoal_t_sulit() {
67     return soal_t_sulit;
68 }

```

Pada Tabel 5.9 merupakan implementasi kode program membuat soal kosakata pada kelas `myDatabaseHandler` untuk mengakses penyimpanan internal pada aplikasi dengan nama database `akpauinggris.db`. Pada baris 1 sampai 44 merupakan kode program dengan metode untuk mengambil data kosakata dari tabel `Kosakata_tabel` secara acak dengan jumlah sepuluh kata sesuai dengan kelas yang dipilih yaitu jika kelas mudah maka akan memilih kosakata dengan jumlah huruf kurang dari lima, jika kelas sedang maka akan memilih kosakata dengan jumlah huruf lebih dari lima dan kurang dari sembilan, dan jika kelas sulit maka akan memilih kosakata dengan jumlah huruf lebih dari delapan.

Tabel 5.10 Implementasi Kode Program Menampilkan Soal Tantangan Pada Kelas `tantanganActivity`

```

1  TextView soalView, bhsIndoView;
2  Button  button_lewati;
3
4  Bundle kelas;
5
6  ArrayList<ArrayList<String>> soal;
7  int i = 0; //soal ke...
8
9  @Override
10 protected void onCreate(Bundle savedInstanceState) {
11     super.onCreate(savedInstanceState);
12     setContentView(R.layout.activity_tantangan);

```

Tabel 5.10 Implementasi Kode Program Menampilkan Soal Tantangan Pada Kelas `tantanganActivity` (lanjutan)

```

13
14     //soal
15     kelas = getIntent().getExtras();
16
17     if (kelas.getString("status").equalsIgnoreCase("mudah")) {
18         soal = new ArrayList<ArrayList<String>>((Collection<?
19 extends ArrayList<String>>) kelas.getSerializable("mudah"));
20     } else if
21 (kelas.getString("status").equalsIgnoreCase("sedang")) {
22     soal = new ArrayList<ArrayList<String>>((Collection<?
23 extends ArrayList<String>>) kelas.getSerializable("sedang"));
24     } else if
25 (kelas.getString("status").equalsIgnoreCase("sulit")) {
26     soal = new ArrayList<ArrayList<String>>((Collection<?
27 extends ArrayList<String>>) kelas.getSerializable("sulit"));
28     }
29
30     soalView = (TextView) findViewById(R.id.soalView);
31     soalView.setText(soal.get(i).get(0));
32
33     bhsIndoView = (TextView) findViewById(R.id.bhsIndoView);
34     bhsIndoView.setText(soal.get(i).get(1));
35
36     button_lewati = (Button) findViewById(R.id.buttonLewati);
37     button_lewati.setOnClickListener(new View.OnClickListener() {
38         @Override
39         public void onClick(View v) {
40             jumkatalewat++;
41             if (i == soal.size()-1) {
42                 showPopupHitungPoin();
43                 countdownTimer.cancel();
44             } else {
45                 i++;
46                 soalView = (TextView)
47                 findViewById(R.id.soalView);
48                 soalView.setText(soal.get(i).get(0));
49
50                 bhsIndoView = (TextView)
51                 findViewById(R.id.bhsIndoView);
52                 bhsIndoView.setText(soal.get(i).get(1));
53             }
54         }
55     });

```

Pada Tabel 5.10 merupakan implementasi kode program menampilkan soal tantangan pada kelas `tantanganActivity`. Pada baris 15 yaitu kode program untuk mendapatkan data yang dikirim dari aktivitas sebelum aktivitas kelas `tantanganActivity` dijalankan. Baris 17 sampai 23 yaitu kode program untuk menyeleksi soal yang akan ditampilkan sesuai dengan kelas yang dipilih. Baris 25 sampai 29 merupakan kode program untuk menampilkan soal kosakata bahasa Inggris dan bahasa Indonesia pada halaman `activity_tantangan`. Baris 31 sampai 49 merupakan kode program untuk melewati soal ke soal selanjutnya yang ditampilkan pada halaman `activity_tantangan`.

5.4.3 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban

Pada bagian ini membahas kode program untuk mendengarkan soal, menjawab soal dan menampilkan hasil Jawaban pengguna dengan implementasi *google voice recognition speech to text* dan *text to speech*, kode program tersebut diimplementasi pada kelas `belajarActivity`.

Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas `belajarActivity`

1	<code>Button button_jawab;</code>
2	<code>ImageButton button_speak;</code>
3	
4	<code>Dialog popuphasil;</code>
5	
6	<code>ArrayList<String> voice_input = new ArrayList<>();</code>
7	
8	<code>private TextToSpeech myTTS;</code>
9	
10	<code>private SpeechRecognizer SpeechRecognizer = null;</code>
11	<code>private Intent recognizerIntent;</code>
12	<code>@Override</code>
13	<code>protected void onCreate(Bundle savedInstanceState) {</code>
14	<code>super.onCreate(savedInstanceState);</code>
15	<code>setContentView(R.layout.activity_belajar);</code>
16	
17	<code>myTTS = new TextToSpeech(this, new</code>
18	<code>TextToSpeech.OnInitListener() {</code>
19	<code>@Override</code>
20	<code>public void onInit(int status) {</code>
21	<code>if (status == TextToSpeech.SUCCESS) {</code>
22	<code>int result =</code>
23	<code>myTTS.setLanguage(Locale.ENGLISH);</code>
24	<code>if (result == TextToSpeech.LANG_MISSING_DATA</code>
25	<code> result == TextToSpeech.LANG_NOT_SUPPORTED) {</code>
26	<code>Log.e("TTS", "Language Not Supported");</code>
27	<code>else {</code>
28	<code>myTTS.setSpeechRate((float) 0.1);</code>
29	<code>button_speak.setEnabled(true);</code>
30	<code>}</code>
31	<code>else {</code>
32	<code>Log.e("TTS", "Initialization Failed");</code>
33	<code>}</code>
34	<code>});</code>
35	
36	<code>button_speak = findViewById(R.id.speakerButton);</code>
37	<code>button_speak.setOnClickListener(new</code>
38	<code>View.OnClickListener() {</code>
39	<code>@Override</code>
40	<code>public void onClick(View v) {</code>
41	<code>speakWords(soal.get(i).get(0));</code>
42	<code>}</code>
43	<code>});</code>
44	<code>SpeechRecognizer =</code>
45	<code>SpeechRecognizer.createSpeechRecognizer(this);</code>
46	<code>SpeechRecognizer.setRecognitionListener(this);</code>
47	<code>button_jawab = (Button) findViewById(R.id.button_jawab);</code>
48	<code>button_jawab.setOnClickListener(new</code>

Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas `belajarActivity` (lanjutan)

```

49     View.OnClickListener() {
50         @Override
51         public void onClick(View v) {
52             startVoiceInput();
53         }
54     });
55
56     private void speakWords(String text) {
57         //speak straight away
58         myTTS.speak(text, TextToSpeech.QUEUE_FLUSH, null);
59     }
60
61     private void startVoiceInput() {
62         recognizerIntent = new
63         Intent(Intent.ACTION_RECOGNIZE_SPEECH);
64         recognizerIntent.putExtra(Intent.EXTRA_LANGUAGE_MODEL,
65         Intent.EXTRA_LANGUAGE_MODEL_FREE_FORM);
66         recognizerIntent.putExtra(Intent.EXTRA_LANGUAGE,
67         Locale.ENGLISH);
68         try {
69             SpeechRecognizer.startListening(recognizerIntent);
70         } catch (ActivityNotFoundException a) {
71             }
72     }
73
74     public void showPopupHasilJawab() {
75         popuphasil = new Dialog(this);
76
77         popuphasil.getWindow().requestFeature(Window.FEATURE_NO_TITLE);
78
79
80         popuphasil setContentView(R.layout.pop_up_hasil_jawab_belajar);
81         popuphasil.getWindow().setBackgroundDrawable(new
82         ColorDrawable(Color.TRANSPARENT));
83         popuphasil.setCancelable(false);
84         popuphasil.setCanceledOnTouchOutside(false);
85         popuphasil.show();
86
87         jawabanView = popuphasil.findViewById(R.id.jawabanView);
88         jawabanView.setText(voice_input.get(0));
89
90         lanjut = popuphasil.findViewById(R.id.lanjut);
91
92         ketBenarSalah = popuphasil.findViewById(R.id.ketBenarSalah);
93         if(soal.get(i).get(0).equalsIgnoreCase(voice_input.get(0))) {
94             jumkatabenar++;
95             ketBenarSalah.setText("BENAR");
96             ketBenarSalah.setTextColor(Color.GREEN);
97             soal.remove(i);
98             jumkatabenarV.setText(""+jumkatabenar);
99
100             myDatabaseHandler tugas_harian = new
101             myDatabaseHandler(getApplicationContext());
102
103             ArrayList<ArrayList<String>> temp_tugasharian;
104             temp_tugasharian = new
105             ArrayList<ArrayList<String>>(tugas_harian.getTugasHarian());

```

Tabel 5.11 Implementasi Kode Program Mendengarkan soal, Menjawab Soal dan Menampilkan Hasil Jawaban Pada Kelas `belajarActivity` (lanjutan)

```

104 for (int t_i = 0; t_i < 3; t_i++){
105     int temp_jumsoalbenar =
Integer.parseInt(temp_tugasharian.get(t_i).get(3));
106
107 if(temp_tugasharian.get(t_i).get(2).equalsIgnoreCase(kategori.get
String("status")) && temp_jumsoalbenar <= jumkatabenar){
108
109 tugas_harian.updateStatusTugasHarian(kategori.getString("status")
, temp_jumsoalbenar);
110     }
111 }
112
113 lanjut.setOnClickListener(new View.OnClickListener() {
114     @Override
115     public void onClick(View v) {
116         if (i == soal.size()-1 || i > soal.size()-1 ) {
117             i = -1;
118         }
119         i++;
120         soalView = (TextView)
findViewById(R.id.soalView);
121         soalView.setText(soal.get(i).get(0));
122
123         bhsIndoView = (TextView)
findViewById(R.id.bhsIndoView);
124         bhsIndoView.setText(soal.get(i).get(1));
125
126         puphampil.dismiss();
127     }
128 });
129 } else {
130     ketBenarSalah.setText("SALAH");
131     ketBenarSalah.setTextColor(Color.RED);
132     lanjut.setOnClickListener(new View.OnClickListener() {
133         @Override
134         public void onClick(View v) {
135             puphampil.dismiss();
136         }
137     });
138 }
139
140 Button keluar;
141 keluar = puphampil.findViewById(R.id.keluar);
142 keluar.setOnClickListener(new View.OnClickListener() {
143     @Override
144     public void onClick(View v) {
145         puphampil.dismiss();
146         Intent intent = new Intent(getApplicationContext(),
categoriBelajarActivity.class);
147         startActivity(intent);
148         finish();
149     }
150 });
151 }
152
153 @Override
154 public void onResults(Bundle results) {
155     ArrayList<String> result =
156     results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
157     voice_input.add(0, result.get(0));
158     showPopupHasilJawab();
159 }

```

Pada Tabel 5.11 merupakan implementasi kode program *google voice recognition speech to text* dengan implementasi *speech recognizer intent* dan *text to speech* pada kelas *belajarActivity*. Pada baris 17 sampai 34 merupakan kode program untuk inialisasi *text to speech*. Baris 36 sampai 42 dan baris 56 sampai 59 merupakan kode program untuk mendengar soal sesuai dengan soal yang ditampilkan dengan menggunakan *text to speech*. Baris 44 sampai 45 merupakan kode program untuk inialisasi *speech recognizer* yang digunakan untuk menerima masukan suara. Baris 47 sampai 54 merupakan kode program untuk mengaktifkan tombol jawab dan memanggil metode untuk menerima masukan suara dan metode mengubah suara ke teks. Pada baris 61 sampai 72 merupakan kode program untuk menerima masukan suara dan mengubah ke bentuk teks dengan menggunakan *recognizer intent*. Baris 153 sampai 159 merupakan kode program untuk mendapatkan hasil masukan suara sebagai jawaban pengguna dalam bentuk *text*, kemudian memanggil metode untuk memberikan penilaian dari jawaban pengguna. Pada baris 75 sampai 151 merupakan kode program untuk menampilkan hasil jawaban pengguna dalam bentuk *text* pada halaman *pop_up_hasil_jawab_belajar* dan memberikan penilaian pada jawaban apakah benar atau salah.

5.4.4 Implementasi Kode Program Menampilkan Total Poin Tantangan

Pada bagian ini menjelaskan implementasi kode program untuk menampilkan total poin yang diperoleh pengguna setelah menjawab soal tantangan. Kode program di implementasikan pada kelas *tantanganActivity*.

Tabel 5.12 Implementasi Kode Program Menampilkan Total Poin Tantangan Pada Kelas *tantanganActivity*

1	Dialog <i>popuppoin</i> ;
2	
3	int jumkatabenar = 0;
4	int jumkatalewat = 0;
5	int jummakskata = 10;
6	
7	public void showPopupHitungPoin() {
8	<i>popuppoin</i> = new Dialog(this);
9	
10	<i>popuppoin</i> .getWindow().requestFeature(Window. FEATURE_NO_TITLE);
11	<i>popuppoin</i> .setContentView(R.layout. pop_up_poin_tantangan);
12	<i>popuppoin</i> .getWindow().setBackgroundDrawable(new
	ColorDrawable(Color. TRANSPARENT));
13	<i>popuppoin</i> .setCancelable(false);
14	<i>popuppoin</i> .setCanceledOnTouchOutside(false);
15	<i>popuppoin</i> .show();
16	
17	<i>jumkatabenarV</i> = <i>popuppoin</i> .findViewById(R.id. kataBenar);
18	<i>jumkatabenarV</i> .setText(""+ jumkatabenar);
19	<i>jumkatalewatV</i> = <i>popuppoin</i> .findViewById(R.id. katalewat);
20	<i>jumkatalewatV</i> .setText(""+ jumkatalewat);
21	<i>jumpoinV</i> = <i>popuppoin</i> .findViewById(R.id. jumlahPoin);
22	final int jumpoin = (int) (jummakskata * jumkatabenar + 10 -
	jumkatalewat);
23	
24	top_score .setTop3(kelas .getString("status"));
25	ArrayList<ArrayList<String>> temp_top3;
26	temp_top3 = new

Tabel 5.12 Implementasi Kode Program Menampilkan Total Poin Tantangan Pada Kelas `tantanganActivity` (lanjutan)

```

27 ArrayList<ArrayList<String>>(top_score.getTop3());
28     int lasttop3 = Integer.parseInt(temp_top3.get(2).get(1));
29     final LinearLayout newtopscoreLayout =
    popuppoin.findViewById(R.id.newtopscorelayout);
30     Button savenewtop =
    popuppoin.findViewById(R.id.saveTopScore);
31     final EditText namanewtop =
    popuppoin.findViewById(R.id.namaNewTop);
32     if(lasttop3 < jumpoin){
33         newtopscoreLayout.setVisibility(View.VISIBLE);
34         savenewtop.setEnabled(true);
35         namanewtop.setEnabled(true);
36         savenewtop.setOnClickListener(new View.OnClickListener()
37     {
38         @Override
39         public void onClick(View v) {
40             String namaTopBaru =
    namanewtop.getText().toString().trim();
41             if (namaTopBaru.isEmpty() || namaTopBaru.length()
    > 10){
42                 Toast.makeText(getApplicationContext(),
    "Enter name, max 10 character", Toast.LENGTH_SHORT).show();
43             } else {
44                 top_score.insertDataTopScore(namaTopBaru,
    jumpoin, kelas.getString("status"));
45             }
46             top_score.deleteScoreNoTop3(kelas.getString("status"));
47             newtopscoreLayout.setVisibility(View.INVISIBLE);
48         }
49     });
50     } else {
51         newtopscoreLayout.setVisibility(View.INVISIBLE);
52         savenewtop.setEnabled(false);
53         namanewtop.setEnabled(false);
54     }
55     ratingBar = popuppoin.findViewById(R.id.ratingBar);
56     if(jumpoin > 90){
57         ratingBar.setRating(3);
58     } else if (jumpoin > 50){
59         ratingBar.setRating(2);
60     } else if (jumpoin <= 50 && jumpoin != 0){
61         ratingBar.setRating(1);
62     }
63     jumpoinV.setText(""+jumpoin);
64
65     Button keluar;
66     keluar = popuppoin.findViewById(R.id.keluar);
67     keluar.setOnClickListener(new View.OnClickListener() {
68     @Override
69     public void onClick(View v) {
70         popuppoin.dismiss();
71         Intent intent = new Intent(getApplicationContext(),
    kelasTantanganActivity.class);
72         startActivity(intent);
73         finish();
74     }
75     });
76 }

```

Pada Tabel 5.12 merupakan implementasi kode program menghitung dan menampilkan total poin tantangan pada halaman `pop_up_poin_tantangan`. Pada baris 17 sampai 22 merupakan kode program untuk menghitung total poin berdasarkan jumlah kata yang dapat dijawab dengan benar dan jumlah kata yang di lewati oleh pengguna. Baris 24 sampai 56 merupakan kode program untuk memeriksa apakah total poin yang didapatkan masuk dalam tiga *top score* atau tidak, jika total poin masuk dalam *top score* maka akan menampilkan kolom inputan nama *top score* baru dan tombol simpan *top score* baru dengan memanggil kelas `myDatabaseHandler`. Baris 58 sampai 66 merupakan kode program untuk menampilkan hasil perhitungan total poin dan bintang yang didapatkan oleh pengguna sesuai dengan total poin yang jika total poin lebih dari 90 maka akan mendapatkan tiga bintang, jika total poin lebih dari 50 maka mendapatkan dua bintang dan jika total poin kurang atau sama dengan 50 maka mendapatkan satu bintang.

5.4.5 Implementasi Kode Program Menampilkan *Top score*

Pada bagian ini menjelaskan implementasi kode program untuk melihat *list* tiga *top score* sesuai dengan kelas tantangan yang dipilih. Kode program ini diimplementasi pada kelas `kelasTantanganActivity`.

5.13 Implementasi Kode Program Menampilkan *Top score* Pada Kelas `kelasTantanganActivity`

```

1  Dialog popTS;
2
3  public void showPopupTopScore(String kelas) {
4      popTS = new Dialog(this);
5      popTS.getWindow().requestFeature(Window.FEATURE_NO_TITLE);
6      popTS setContentView(R.layout.pop_up_top_score);
7      popTS.getWindow().setBackgroundDrawable(new
8  ColorDrawable(Color.TRANSPARENT));
9      popTS.setCanceledOnTouchOutside(false);
10     popTS.show();
11
12     myDatabaseHandler top_score = new myDatabaseHandler(this);
13     top_score.setTop3(kelas);
14
15     TextView namatop1 = popTS.findViewById(R.id.namatop1);
16     TextView namatop2 = popTS.findViewById(R.id.namatop2);
17     TextView namatop3 = popTS.findViewById(R.id.namatop3);
18     TextView scoretop1 = popTS.findViewById(R.id.scoretop1);
19     TextView scoretop2 = popTS.findViewById(R.id.scoretop2);
20     TextView scoretop3 = popTS.findViewById(R.id.scoretop3);
21
22     ArrayList<ArrayList<String>> temp_top3;
23     temp_top3 = new
24     ArrayList<ArrayList<String>>(top_score.getTop3());
25
26     namatop1.setText(temp_top3.get(0).get(0));
27     scoretop1.setText(temp_top3.get(0).get(1));
28     namatop2.setText(temp_top3.get(1).get(0));
29     scoretop2.setText(temp_top3.get(1).get(1));
30     namatop3.setText(temp_top3.get(2).get(0));
31     scoretop3.setText(temp_top3.get(2).get(1));
32
33     Button tutup;
34     tutup = popTS.findViewById(R.id.tutup);

```

5.13 Implementasi Kode Program Menampilkan *Top score* Pada Kelas *kelasTantanganActivity* (lanjutan)

```

32  tutup.setOnClickListener(new View.OnClickListener() {
33      @Override
34      public void onClick(View v) {
35          popTS.dismiss();
36      }
37  });
38  }

```

Pada Tabel 5.13 merupakan implementasi kode program untuk menampilkan *top score* pada halaman `pop_up_top_score`. Pada baris 11 sampai 12 merupakan kode program untuk mengambil data *top score* dengan memanggil kelas `myDatabaseHandler`. Baris 23 sampai 28 merupakan kode program untuk menampilkan *list top score* sesuai urutan terbesar ke terkecil.

Tabel 5.14 Implementasi Kode Program Mengambil Data *Top score* Pada Kelas *myDatabaseHandler*

```

1  public void setTop3(String kelas) {
2      final SQLiteDatabase db = this.getReadableDatabase();
3      String selectQuery = "SELECT Nama, Score FROM " +
4      TABLE_NAME_TOPSCORE + " WHERE Kelas LIKE '%" + kelas + "%' ORDER
5      BY Score DESC LIMIT 3";
6      Cursor cursor = db.rawQuery(selectQuery, null);
7      cursor.moveToFirst();
8      while (!cursor.isAfterLast()) {
9          top3.add(new ArrayList<String>());
10         top3.get(top3.size() - 1).add(0, cursor.getString(0));
11         top3.get(top3.size() - 1).add(1, cursor.getString(1));
12         cursor.moveToNext();
13     }
14     cursor.close();
15 }

```

Pada Tabel 5.14 merupakan implementasi kode program untuk mendapatkan data *top score* dari database `akpauinggris.db` pada tabel `Top_score`. Pada baris 3 merupakan kode program untuk seleksi data *top score* pada tabel `Top_score` dengan mengurutkan data dari *top score* terbesar ke terkecil kemudian pada baris 4 merupakan kode program untuk mengeksekusi kode program pada baris 3.

BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini menjelaskan pengujian aplikasi dan analisis hasil pengujian yang dilakukan berdasarkan hasil implementasi yang sudah dilakukan, untuk mengetahui apakah hasil implementasi aplikasi sudah sesuai dengan harapan atau belum.

6.1 Pengujian

Pada penelitian ini pengujian yang dilakukan yaitu dengan *black box testing* menggunakan teknik *requirement validation testing*, Pengujian *usability* dan pengujian akurasi implementasi *google voice recognition* pada aplikasi belajar pengucapan kosakta bahasa Inggris. Hasil terbaik berasal dari pengujian tidak lebih dari lima responden dan menjalankan tes kecil sebanyak yang anda mampu (Nielsen, 2000), pengujian yang lebih dari lima responden hanya akan mendapatkan hasil yang cenderung sama dengan pengujian dengan maksimal 5 responden. Oleh karena itu penentuan jumlah partisipan/responden dalam pengujian *usability* dan pengujian *black box* berdasarkan pendapat Jacob Nielsen pada jurnalnya tahun 2000 yang berjudul “Why You Only Need to Test with 5 Users”.

6.1.1 Pengujian *Black box Testing* Dengan Teknik *Requirement validation Testing*

Pengujian *black box* dengan teknik *requirement validation* ialah pengujian yang fokus utama pada kebutuhan fungsional aplikasi. Yang dilakukan pada pengujian ini adalah membuat kasus uji untuk masing-masing kebutuhan fungsional, kemudian dilakukan pengujian pada masing-masing kasus uji untuk mendapatkan data berupa hasil pengujian yang selanjutnya akan dianalisis.

6.1.1.1 Kasus Uji *Requirement validation Testing*

Pada bagian ini mejabarkan prosedur menguji fungsionalitas pada aplikasi yang telah di implementasi. Kasus uji merupakan prosedur atau langkah-langkah untuk menguji kebutuhan fungsional pada hasil implementasi aplikasi. Pada kasus uji yang pertama adalah menampilkan soal pada semua kategori kelas belajar dan soal pada semua kelas tantangan yang dijelaskan pada Tabel 6.1.

Tabel 6.1 Kasus Uji Menampilkan Soal

Kode kasus uji	TES01
Nama kasus uji	Menampilkan soal
Objek kasus uji	Kebutuhan fungsional (F01)
Tujuan pengujian	Untuk memastikan aplikasi dapat menampilkan soal pada semua kategori belajar dan kelas tantangan.

Tabel 6.1 Kasus Uji Menampilkan Soal (lanjutan)

Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu belajar. 3. Memilih kategori kata kerja. 4. Melewati soal sampai kembali ke soal yang pertama. 5. Kembali ke halaman sebelumnya. 6. Memilih kategori kata benda dan melakukan langkah 4 sampai 5. 7. Memilih kategori kata sifat dan melakukan langkah 4 dan 5. 8. Kembali ke halaman menu utama. 9. Memilih menu tantangan. 10. Memilih kelas mudah. 11. Melewati semua soal sampai menampilkan pop up total poin. 12. Keluar dari halaman soal. 13. Memilih kelas sedang dan melakukan langkah 11 sampai 12. 14. Memilih kelas sulit dan melakukan langkah 11 sampai 12.
Hasil yang diharapkan	Aplikasi dapat menampilkan semua soal kosakata bahasa Inggris dan artinya dalam bahasa Indoseia sesuai pilihan fitur atau menu.

Kasus uji yang selanjutnya adalah mengeluarkan suara soal yang artinya aplikasi dapat mengeluarkan suara *audio* berupa pengucapan yang benar dari soal kosakata bahasa Inggris yang ditampilkan, fitur ini hanya terdapat pada menu belajar. Kasus uji mengeluarkan suara soal dijelaskan pada Tabel 6.2.

Tabel 6.2 Kasus Uji Mengeluarkan suara Soal

Kode kasus uji	TES02
Nama kasus uji	Mengeluarkan suara soal
Objek kasus uji	Kebutuhan fungsional (F02)
Tujuan pengujian	Untuk memastikan aplikasi dapat mengeluarkan suara <i>audio</i> berupa pengucapan yang benar dari soal kosakata bahasa Inggris.

Tabel 6.2 Kasus Uji Mengeluarkan suara Soal (lanjutan)

Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu belajar. 3. Memilih kategori kata kerja. 4. Menekan tombol dengarkan <i>audio</i> soal. 5. Melewati soal sampai kembali ke soal yang pertama dan melakukan langkah 4 6. Kembali ke halaman sebelumnya. 7. Memilih kategori kata benda dan melakukan langkah 4 sampai 6. 8. Memilih kategori kata sifat dan melakukan langkah 4 sampai 6.
Hasil yang diharapkan	Aplikasi dapat mengeluarkan suara <i>audio</i> berupa pengucapan yang benar dari soal kosakata bahasa Inggris yang ditampilkan.

Kasus uji selanjutnya adalah menjawab soal dan menampilkan hasil jawaban yang artinya aplikasi dapat menerima masukan suara dari pengguna sebagai jawaban dan menampilkan apakah hasil jawaban benar atau salah yang dijelaskan pada Tabel 6.3.

Tabel 6.3 Kasus Uji Menjawab Soal Dan Menampilkan Hasil Jawaban

Kode kasus uji	TES03
Nama kasus uji	Menjawab soal dan menampilkan hasil jawaban
Objek kasus uji	Kebutuhan fungsional (F03 dan F04)
Tujuan pengujian	Untuk memastikan aplikasi menerima masukan suara dari pengguna sebagai jawaban dan menampilkan apakah hasil jawaban.
Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu belajar. 3. Memilih kategori kata kerja. 4. Menekan tombol jawab. 5. Menjawab soal dengan cara menyebutkan soal yang ditampilkan. 6. Melihat hasil jawaban. 7. Menekan tombol lanjut.



Tabel 6.3 Kasus Uji Menjawab Soal Dan Menampilkan Hasil Jawaban (lanjutan)

	<ol style="list-style-type: none"> 8. Kembali ke halaman sebelumnya. 9. Memilih kategori kata benda dan melakukan langkah 4 sampai 8. 10. Memilih kategori kata sifat dan melakukan langkah 4 dan 8. 11. Kembali ke halaman menu utama. 12. Memilih menu tantangan. 13. Memilih kelas mudah. 14. Menekan tombol jawab. 15. Menjawab soal dengan cara menyebutkan soal yang ditampilkan. 16. Melihat hasil jawaban. 17. Menekan tombol lanjut. 18. Kembali ke halaman sebelumnya. 19. Memilih kelas sedang dan melakukan langkah 14 sampai 18. <p>Memilih kelas sulit dan melakukan langkah 14 sampai 18.</p>
Hasil yang diharapkan	Aplikasi dapat menerima masukan suara dari pengguna sebagai jawaban dan menampilkan apakah hasil jawaban benar atau salah beserta jawaban dari pengguna dalam bentuk teks.

Kasus uji selanjutnya adalah menampilkan jumlah kata benar yang artinya aplikasi dapat menampilkan jumlah kata benar yang telah dijawab. Kasus uji ini dijelaskan pada Tabel 6.4.

Tabel 6.4 Kasus Uji Menampilkan Jumlah Kata Benar

Kode kasus uji	TES04
Nama kasus uji	Menampilkan jumlah kata benar
Objek kasus uji	Kebutuhan fungsional (F05)
Tujuan pengujian	Untuk memastikan aplikasi dapat menampilkan jumlah kata benar yang telah dijawab.

Tabel 6.4 Kasus Uji Menampilkan Jumlah Kata Benar (lanjutan)

Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu belajar. 3. Memilih kategori kata kerja. 4. Menjawab soal dengan benar dan lanjut ke soal selanjutnya. 5. Melakukan langkah 4 sampai minimal 5 soal terjawab dengan benar. 6. Kembali ke halaman sebelumnya. 7. Memilih kategori kata benda dan melakukan langkah 4 sampai 6. 8. Memilih kategori kata sifat dan melakukan langkah 4 dan 6. 9. Kembali ke halaman menu utama. 10. Memilih menu tantangan. 11. Memilih kelas mudah. 12. Menjawab soal dengan benar dan lanjut ke soal selanjutnya. 13. Melakukan langkah 4 sampai minimal 5 soal terjawab dengan benar. 14. Kembali ke halaman sebelumnya. 15. Memilih kelas sedang dan melakukan langkah 12 sampai 14. 16. Memilih kelas sulit dan melakukan langkah 12 sampai 14.
Hasil yang diharapkan	Aplikasi dapat dapat menampilkan jumlah kata benar yang telah dijawab oleh pengguna secara <i>real time</i> .

Kasus uji selanjutnya adalah menampilkan sisa waktu tantangan yang artinya aplikasi dapat menampilkan sisa waktu pengguna dalam mengerjakan soal tantangan yang dijelaskan pada Tabel 6.5.

Tabel 6.5 Kasus Uji Menampilkan Sisa Waktu Tantangan

Kode kasus uji	TES05
Nama kasus uji	Menampilkan sisa waktu tantangan
Objek kasus uji	Kebutuhan fungsional (F06)
Tujuan pengujian	Untuk memastikan aplikasi dapat menampilkan sisa waktu pengguna dalam mengerjakan soal tantangan.
Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu tantangan. 3. Memilih kelas mudah. 4. Melihat sisa waktu. 5. Kembali ke halaman sebelumnya. 6. Memilih kelas sedang dan melakukan langkah 4 sampai 5. 7. Memilih kelas sulit dan melakukan langkah 4 sampai 5.
Hasil yang diharapkan	Aplikasi dapat menampilkan sisa waktu hitung mundur dalam mengerjakan soal tantangan.

Kasus uji yang selanjutnya adalah hitung total poin tantangan artinya aplikasi dapat melakukan perhitungan total poin dan menampilkan total poin yang didapatkan oleh pengguna setelah mengerjakan soal tantangan. Kasus uji ini dijelaskan pada Tabel 6.6.

Tabel 6.6 Kasus Uji Hitung Total Poin Tantangan

Kode kasus uji	TES06
Nama kasus uji	Hitung total poin tantangan
Objek kasus uji	Kebutuhan fungsional (F07)
Tujuan pengujian	Untuk memastikan aplikasi dapat melakukan perhitungan total poin dan menampilkan total poin tantangan.

Tabel 6.6 Kasus Uji Hitung Total Poin Tantangan (lanjutan)

Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu tantangan. 3. Memilih kelas mudah. 4. Mengerjakan soal tantangan sampai selesai. 5. Melihat total poin. 6. Keluar dari soal tantangan. 7. Memilih kelas sedang dan melakukan langkah 4 sampai 6. 8. Memilih kelas sulit dan melakukan langkah 4 sampai 6.
Hasil yang diharapkan	Aplikasi dapat melakukan perhitungan total poin dan menampilkan total poin tantangan pada setiap kelas tantangan yang dikerjakan.

Kasus uji selanjutnya adalah menyimpan *top score* baru dan menampilkan *top score* yang artinya aplikasi dapat menyimpan *top score* baru dan menampilkan *list* tiga *top score* pada setiap kelas tantangan. Kasus uji ini dijelaskan pada Tabel 6.7.

Tabel 6.7 Kasus Uji Menyimpan *Top score* baru dan Menampilkan *Top score*

Kode kasus uji	TES07
Nama kasus uji	Menyimpan <i>top score</i> baru dan menampilkan top
Objek kasus uji	Kebutuhan fungsional (F08 dan F09)
Tujuan pengujian	Untuk memastikan aplikasi dapat menyimpan <i>top score</i> baru dan menampilkan <i>list</i> tiga <i>top score</i> .
Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu tantangan. 3. Memilih kelas mudah. 4. Menekan tombol mulai. 5. Mengerjakan soal tantangan sampai selesai. 6. Memasukkan nama <i>top score</i> baru pada kolom input nama. 7. Menekan tombol simpan <i>top score</i> baru. 8. Keluar dari soal tantangan. 9. Memilih kelas mudah.

Tabel 6.7 Kasus Uji Menyimpan *Top score* baru dan Menampilkan *Top score* (lanjutan)

	<ol style="list-style-type: none"> 10. Menekan tombol <i>top score</i>. 11. Memilih kelas sedang dan melakukan langkah 4 sampai 8. 12. Memilih kelas sedang. 13. Menekan tombol <i>top score</i>. 14. Memilih kelas sulit dan melakukan langkah 4 sampai 8. 15. Memilih kelas sulit. 16. Menekan tombol <i>top score</i>.
Hasil yang diharapkan	Aplikasi dapat menyimpan <i>top score</i> baru dan menampilkan <i>list</i> tiga <i>top score</i> pada setiap kelas tantangan.

Kasus uji selanjutnya adalah menampilkan tugas harian dan memperbarui tugas harian yang artinya aplikasi dapat menampilkan daftar tugas harian dan memperbarui tugas harian jika tugas harian telah dikerjakan. Kasus uji ini dijelaskan pada Tabel 6.8.

Tabel 6.8 Kasus Uji Menampilkan Tugas Harian Dan Memperbarui Tugas Harian

Kode kasus uji	TES08
Nama kasus uji	Menampilkan tugas harian dan memperbarui tugas harian
Objek kasus uji	Kebutuhan fungsional (F10 dan F11)
Tujuan pengujian	Untuk memastikan aplikasi dapat menampilkan daftar tugas harian dan memperbarui tugas harian.
Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Melihat daftar tugas harian. 3. Mengerjakan tugas harian. 4. Kembali ke halaman menu utama. 5. Melihat tugas harian yang sudah dikerjakan
Hasil yang diharapkan	Aplikasi dapat menampilkan daftar tugas harian dan memperbarui tugas harian jika tugas harian telah dikerjakan dengan memberikan tanda <i>check/list</i> pada tugas harian.

Kasus uji selanjutnya adalah menampilkan kategori belajar dan kelas tantangan yang artinya aplikasi dapat menampilkan daftar kategori belajar dan kelas tantangan yang dijelaskan pada Tabel 6.9.

Tabel 6.9 Kasus Uji Menampilkan Kategori Belajar Dan Kelas Tantangan.

Kode kasus uji	TES09
Nama kasus uji	Menampilkan kategori belajar dan kelas tantangan
Objek kasus uji	Kebutuhan fungsional (F12 dan F13)
Tujuan pengujian	Untuk memastikan aplikasi dapat menampilkan daftar kategori belajar dan kelas tantangan.
Prosedur pengujian	<ol style="list-style-type: none"> 1. Aplikasi dijalankan. 2. Memilih menu belajar. 3. Kembali ke halaman menu utama. 4. Memilih menu tantangan.
Hasil yang diharapkan	Aplikasi dapat menampilkan tiga daftar kategori belajar dan kelas tantangan.

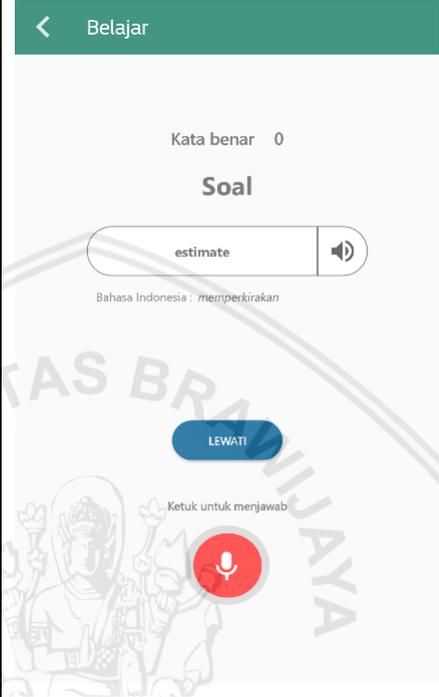
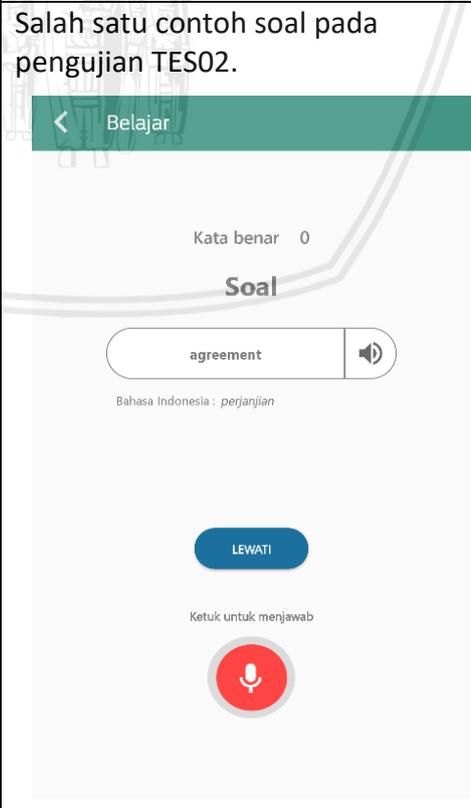
6.1.1.2 Hasil Pengujian *Requirement validation testing*

Hasil pengujian *requirement validation testing* merupakan hasil pengujian dari keseluruhan kasus uji yang sudah di buat pada sub bab kasus uji *requirement validation testing*. Berikut hasil pengujian yang ditunjukkan pada Tabel 6.10.

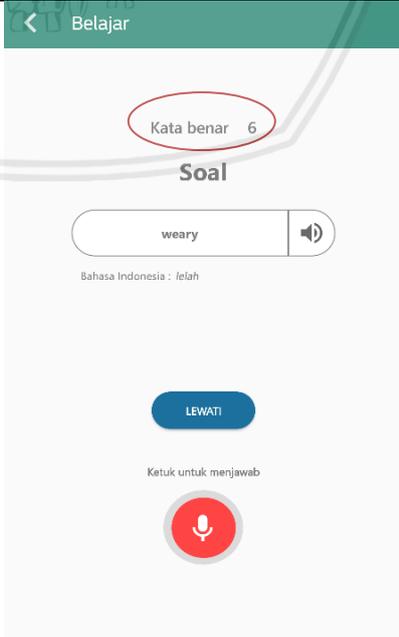
Tabel 6.10 Hasil Pengujian *Requirement validation testing*

Kode kasus uji	Hasil yang diperoleh		Status validasi
TES01	Aplikasi dapat menampilkan semua soal kosakata bahasa Inggris dan artinya dalam bahasa Indonesia sesuai pilihan fitur atau menu yang dipilih.	Salah satu contoh soal pada menu belajar kategori kata kerja.	Valid

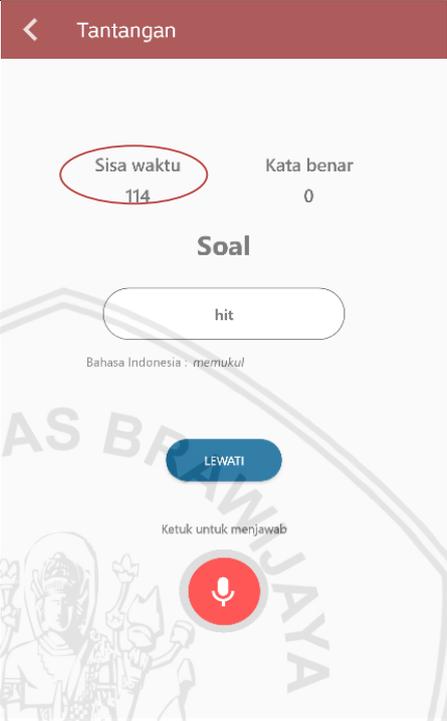
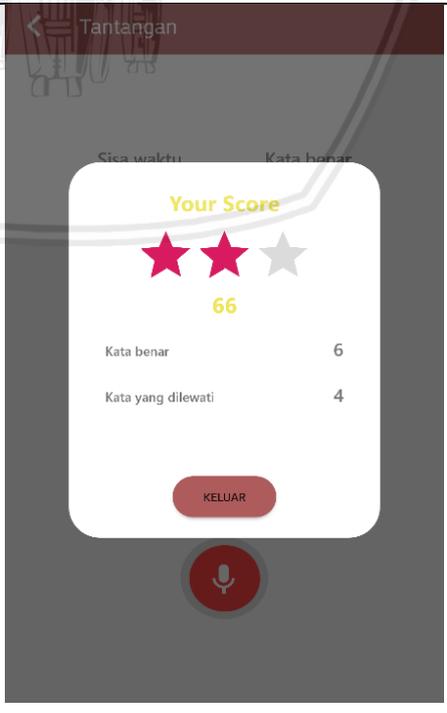
Tabel 6.10 Hasil Pengujian *Requirement validation testing* (lanjutan)

Kode kasus uji	Hasil yang diperoleh	Status validasi
		
<p>TES02</p>	<p>Aplikasi dapat mengeluarkan suara <i>audio</i> berupa pengucapan yang benar dari soal kosakata bahasa Inggris yang ditampilkan setelah menekan tombol <i>audio</i>.</p> 	<p>Valid</p>

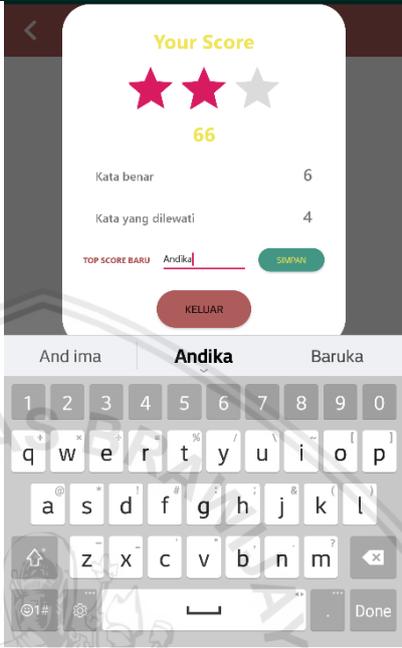
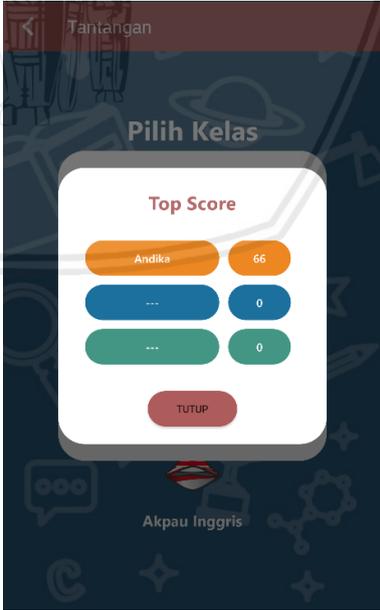
Tabel 6.10 Hasil Pengujian *Requirement validation testing* (lanjutan)

Kode kasus uji	Hasil yang diperoleh	Status validasi
TES03	<p>Aplikasi dapat menerima masukan suara dari pengguna sebagai jawaban dan menampilkan apakah hasil jawaban benar atau salah beserta jawaban dari pengguna dalam bentuk teks.</p>  <p>Contoh hasil jawaban benar pengguna.</p>	Valid
TES04	<p>Aplikasi dapat menampilkan jumlah kata benar yang telah dijawab oleh pengguna secara <i>real time</i> yaitu berjumlah 6.</p> 	Valid

Tabel 6.10 Hasil Pengujian Requirement validation testing (lanjutan)

Kode kasus uji	Hasil yang diperoleh	Status validasi
<p>TES05</p> <p>Aplikasi dapat menampilkan sisa waktu hitung mundur dalam mengerjakan soal tantangan yang sesuai dengan kelas yang dipilih.</p>	 <p>Contoh sisa waktu pada halaman kelas mudah.</p>	<p>Valid</p>
<p>TES06</p> <p>Aplikasi dapat menampilkan total poin tantangan pada setiap kelas tantangan yang dikerjakan.</p>		<p>Valid</p>

Tabel 6.10 Hasil Pengujian *Requirement validation testing* (lanjutan)

Kode kasus uji	Hasil yang diperoleh		Status validasi
<p>TES07</p> <p>Aplikasi dapat menampilkan kolom input nama <i>top score</i> baru dan dapat menyimpan <i>top score</i> baru serta menampilkan <i>list</i> tiga <i>top score</i> pada setiap kelas tantangan sesuai dengan kelas yang dipilih.</p>		 <p>Contoh tampilan ketika pengguna mendapatkan <i>top score</i> baru pada kelas mudah.</p>  <p>Contoh tampilan <i>list</i> tiga <i>top score</i> pada kelas mudah.</p>	<p>Valid</p>

Tabel 6.10 Hasil Pengujian Requirement validation testing (lanjutan)

Kode kasus uji	Hasil yang diperoleh	Status validasi
TES08	<p>Aplikasi dapat menampilkan daftar tugas harian dan memperbarui tugas harian jika tugas harian telah dikerjakan dengan memberikan tanda <i>checklist</i> pada tugas harian.</p> 	Valid
TES09	<p>Aplikasi dapat menampilkan tiga daftar kategori belajar dan tiga daftar kelas tantangan.</p> 	Valid

6.1.2 Pengujian *Usability*

Pada pengujian *usability* dilakukan untuk menguji kebutuhan non fungsional kode NF01 dengan menggunakan SUS kuesioner dan metode skala *Likert* digunakan untuk mendapatkan feedback dari para responden. Sesuai dengan SUS kuesioner maka ada 10 butir pertanyaan yang diajukan kepada 5 responden. Melalui pengujian *usability*, setiap responden melakukan pengujian dengan tahapan yang sama dengan kasus uji pada *requirement validation testing*, dengan demikian responden akan menggunakan semua fitur pada aplikasi. Kemudian responden diminta melakukan penilaian dengan memberi tanda checklist pada salah satu angka penilaian skala *likert* pada tiap pertanyaan untuk mengukur perilaku setiap responden dengan memberikan penilaian pada salah satu dari 5 poin pilihan pada setiap butir pertanyaan, sangat setuju (SS) dengan skala 5, setuju (S) dengan skala 4, tidak memutuskan/netral (N) dengan skala 3, tidak setuju (TS) dengan skala 2, dan sangat tidak setuju (STS) dengan skala 1. Pada pengujian ini menggunakan pertanyaan asli dari SUS kuesioner yang masih menggunakan bahasa Inggris, pada pertanyaan nomor 1, 3, 5, 7 dan 9 merupakan pertanyaan yang bernilai positif mengenai aplikasi sedangkan untuk pertanyaan 2, 4, 6, 8 dan 10 merupakan pertanyaan bernilai negatif untuk aplikasi. Untuk lebih jelasnya dapat dilihat pada Tabel 6.11.

Tabel 6.11 SUS Kuesioner

No	Pertanyaan	1	2	3	4	5
1	<i>I think that I would like to use this system frequently.</i>					
2	<i>I found the system unnecessarily complex</i>					
3	<i>I thought the system was easy to use</i>					
4	<i>I think that I would need the support of a technical person to be able to use this system</i>					
5	<i>I found the various functions in this system were well integrated</i>					
6	<i>I thought there was too much inconsistency in this system</i>					
7	<i>I would imagine that most people would learn to use this system very quickly</i>					
8	<i>I found the system very cumbersome to use</i>					
9	<i>I felt very confident using the system</i>					
10	<i>I needed to learn a lot of things before I could get going with this system</i>					

6.1.2.1 Hasil Pengujian *Usability*

Pada bagian ini merupakan hasil pengujian dari seluruh responden yang telah di rekapitulasi dan hasil perhitungan skor pada SUS kuesioner. Hasil pengujian dan perhitungan skor yang telah didapatkan dari semua responden dapat dilihat pada Tabel 6.12. Untuk menghitung skor SUS, pertama menghitung jumlah skor dari setiap pertanyaan dari masing-masing responden. Skor setiap pertanyaan akan berkisar dari 0 hingga 4 dikarenakan perhitungan SUS score untuk pertanyaan 1,3,5,7, dan 9 perhitungan skor adalah posisi skala di kurang 1, untuk pertanyaan 2,4,6,8 dan 10 perhitungan score adalah 5 di kurang posisi skala, kemudian hitung jumlah skor setiap pertanyaan dan kali jumlah skor dengan 2,5 untuk mendapatkan nilai SUS secara keseluruhan dari satu responden, kemudian lakukan tahapan yang sama untuk penilaian dari responden lainnya. Setelah mendapatkan skor SUS dari semua responden, selanjutnya menghitung rata-rata dari skor SUS untuk mendapatkan peringkat SUS skor pada aplikasi belajar pengucapan kosakata bahasa Inggris yang akan dijadikan acuan pada tahap analisis hasil pengujian *usability*.

Tabel 6.12 Hasil Rekapitulasi dan Perhitungan Skor SUS Kuesioner

Responden	Pertanyaan										Skor SUS
	1	2	3	4	5	6	7	8	9	10	
Responden 1	5	1	5	1	5	2	5	1	5	1	97.5
Responden 2	3	2	4	1	5	3	3	2	4	2	72.5
Responden 3	4	1	5	1	5	3	4	1	5	2	87.5
Responden 4	4	1	5	1	5	2	4	1	5	1	92.5
Responden 5	3	2	4	1	5	2	3	2	4	2	75.0
Rata-rata skor SUS											85

6.1.3 Pengujian Akurasi Implementasi *Google Speech Recognition*

Pengujian ini bertujuan untuk menguji secara langsung tingkat akurasi google *speech* API yang telah diimplementasi pada sistem dan menentukan apakah API yang digunakan sudah sesuai dengan yang diharapkan atau belum. Pada pengujian ini melibatkan seorang tenaga pengajar bahasa Inggris sebagai penguji yaitu salah satu tenaga pengajar pada tempat kursus bahasa Inggris di kota Malang untuk menguji akurasi *google speech to text* dan *text to speech* yang di implementasikan pada aplikasi belajar kosakata bahasa Inggris. Pengujian ini dilakukan dengan cara penguji menggunakan salah satu kategori belajar pada aplikasi dan menggunakan fitur *text* soal ke suara dan kemudian menjawab soal dengan salah kemudian menjawab dengan benar, untuk satu soal yang ditampilkan penguji diberikan tiga kali percobaan untuk menjawab soal dengan benar, setelah itu penguji mengisi form yang berisi kesesuaian soal dan jawaban yang di uji. Pengujian ini dilakukan

sampai seluruh soal pada fitur belajar terjawab. Hasil dari pengujian ini ditunjukkan pada Lampiran 1.

6.2 Analisis

Bagian ini menjelaskan analisis dari pengujian yang telah dilakukan pada sub bab sebelumnya. Analisis dibagi menjadi tiga bagian, yaitu: analisis hasil pengujian *requirement validation testing*, analisis hasil pengujian *usability* dan analisis hasil pengujian akurasi implementasi *google speech* recognition.

Tabel 6.13 Hasil Pengujian

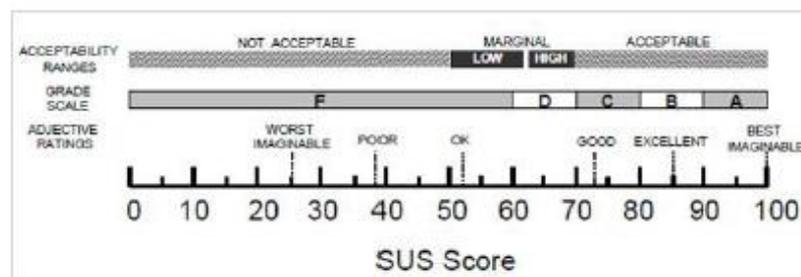
Pengujian	Hasil
Pengujian <i>requirement validation testing</i>	100% valid
Pengujian <i>usability</i>	Rata-tara skor SUS = 85
Pengujian akurasi implementasi <i>google speech</i> recognition	Akurasi 100%

6.2.1 Analisis Hasil Pengujian *Requirement Validation Testing*

Berdasarkan pada Tabel 6.13 hasil pengujian pada *requirement validation testing* yang telah dilaksanakan oleh peneliti selama dua hari dengan jumlah penguji sebanyak 5 orang diperoleh hasil bahwa hasil implementasi aplikasi belajar pengucapan kosakata bahasa Inggris 100% valid atau telah sesuai dengan semua kebutuhan fungsional yang telah dideskripsikan pada bab analisis kebutuhan dan perancangan. Angka 100% di dapat dari perhitungan jumlah kasus uji valid yaitu 9 dibagi jumlah kasus uji yaitu 9 kemudian dikali 100. Dengan demikian hasil implementasi aplikasi dapat disimpulkan telah sukses dan bekerja sesuai dengan kebutuhan fungsional yang diharapkan.

6.2.2 Analisis Hasil Pengujian *Usability*

Berdasarkan pada Tabel 6.13 rata-rata skor SUS dari pengujian *usability* aplikasi belajar pengucapan kosakata bahasa Inggris adalah 85, nilai rata-rata skor SUS tersebut akan digunakan sebagai acuan untuk analisis *usability* aplikasi. Analisis dilakukan dengan menyesuaikan hasil pengujian dengan kategori peringkat SUS skor dapat dilihat dalam Gambar 6.1.



Gambar 6.1 Peringkat SUS Skor (Brooke, 2013)

Dalam Gambar 6.1 menunjukkan hasil pengujian *usability* dengan rata-rata skor SUS yaitu 85 termasuk dalam tingkat kategori penerimaan pengguna pada peringkat “Acceptable”, untuk kelas skala termasuk dalam peringkat “B” dan untuk adjective rating berada pada tingkat “Good”. Berdasarkan dari hasil pengujian *usability* aplikasi belajar pengucapan kosakata bahasa Inggris dapat disimpulkan bahwa tampilan antarmuka pengguna aplikasi dapat dengan mudah digunakan dan dipahami sehingga pengguna dapat menerima tampilan antarmuka pengguna aplikasi.

6.2.3 Analisis Hasil Pegujian Akurasi Implementasi *Google Speech Recognition*

Berdasarkan hasil yang ditunjukkan pada form pengujian pada Lampiran 1. Pada salah satu kategori belajar yang dipilih responden yaitu kategori kata kerja, saat pengujian berlangsung semua soal ketika penguji menjawab salah dengan sengaja hasil yang didapatkan sesuai yaitu hasil jawaban salah dan untuk pengujian pengucapan yang benar diperoleh hasil 100% soal dapat dijawab dengan benar walaupun beberapa soal tidak dapat dijawab dengan benar pada percobaan pertama namun untuk percobaan selanjutnya penguji dapat menjawab dengan benar dikarenakan kurang jelasnya penekanan huruf dari penguji dan kecepatan penguji dalam mengucapkan jawaban dapat berpengaruh. Perhitungan akurasi kata benar dilakukan dengan cara jumlah kata benar dibagi dengan jumlah soal kemudian dikali 100. Sedangkan untuk hasil audio pengucapan yang benar dari soal, penguji menyatakan bahwa pengucapan suara kosakata bahasa Inggris yang dikeluarkan oleh aplikasi telah sesuai dengan pengucapan yang benar. Berdasarkan pada Tabel 6.13 hasil akurasi implementasi *google speech recognition* pada aplikas belajar pengucapan kosakata bahasa Inggris bernilai 100% sehingga dapat disimpulkan implementasi *google speech recognition* telah sesuai dengan yang diharapkan.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan seluruh tahapan yang sudah dilakukan pada penelitian ini maka dapat disimpulkan bahwa:

1. Aplikasi belajar pengucapan kosakata bahasa Inggris berbasis android dirancang dengan menggunakan metode *prototyping*, sebelum masuk tahapan utama metode *prototyping* tahapan yang dilakukan adalah studi literatur dengan mempelajari penelitian yang serupa sebelumnya, tahap perancangan pada metode *prototyping* terdiri dari tiga tahapan utama, yaitu: tahapan pertama menggali dan analisis kebutuhan, tahapan kedua membangun *prototype* dari aplikasi yang disesuaikan dengan hasil dari analisis kebutuhan yang dilakukan pada tahap sebelumnya, dan tahapan yang selanjutnya pengujian dan evaluasi *prototype*. Setelah tahapan utama metode *prototyping* selesai, tahapan selanjutnya adalah tahapan implementasi sistem kemudian tahapan pengujian sistem untuk menguji apakah sistem sudah sesuai dengan kebutuhan dan *prototype* atau belum.
2. *Google speech recognition* diimplementasi pada dua fitur utama yaitu fitur belajar dan fitur tantangan. Pada fitur belajar diimplementasikan *google text to speech* API yang berfungsi untuk mengeluarkan *audio* dari aplikasi berupa pengucapan yang benar dari soal bahasa Inggris dan *google speech to text* API yang bertujuan untuk menerima masukan suara dari pengguna yang kemudian diubah ke *text* dan diaggap sebagai jawaban oleh aplikasi yang akan dinilai oleh aplikasi apakah sesuai dengan soal atau tidak. Pada fitur tantangan hanya *speech to text* saja yang diimplementasi dikarenakan pada fitur tantangan tidak dibutuhkan bantuan *audio* pengucapan yang benar dari soal yang ditampilkan.
3. Hasil pengujian aplikasi edukasi belajar pengucapan kosakata bahasa Inggris terbagi menjadi tiga, yaitu:
 - a. Hasil pengujian *requirement validation* dalam aplikasi belajar pengucapan bahasa Inggris diperoleh hasil 100% valid dengan kebutuhan fungsional yang artinya sesuai dengan yang diharapkan.
 - b. Hasil pengujian *usability* aplikasi belajar pengucapan kosakata bahasa Inggris mendapatkan rata-rata SUS skor dengan nilai 85 dan berada pada tingkat penerimaan pengguna "*acceptable*" sehingga dapat disimpulkan bahwa tampilan antarmuka pengguna aplikasi dapat dengan mudah digunakan dan dipahami sehingga pengguna dapat menerima tampilan antarmuka pengguna aplikasi.

- c. Hasil akurasi implementasi *google speech recognition* pada aplikasi belajar pengucapan kosakata bahasa Inggris bernilai 100% sehingga dapat disimpulkan implementasi *google speech recognition* telah sesuai dengan yang diharapkan.

7.2 Saran

Saran yang dapat diberikan untuk penelitian berikutnya berdasarkan hasil penelitian yang telah dilakukan oleh penulis, yaitu:

1. Melakukan evaluasi *user experience* dan *user interface* serta perbaikan antarmuka untuk meningkatkan penerimaan aplikasi oleh pengguna pengguna.
2. Melakukan pengujian *compatibility* pada aplikasi karena belum dilakukan pengujian tersebut pada hasil implementasi aplikasi belajar pengucapan kosakata Bahasa Inggris.
3. Menambahkan fitur tambahan pada fitur tantangan yang lebih menarik seperti fitur bagikan *point* tantangan ke sosial media, fitur untuk melakukan tantangan dengan melawan pengguna lain dan fitur *top score* yang dibuat online sehingga *top score* dapat bersaing dengan pengguna lain.
4. Menambah pengelompokan kategori kosakata pada fitur belajar dan pada fitur tantangan kosakata yang dimasukkan pada kelas mudah, sedang dan sulit ditambahkan seleksi berdasarkan kriteria yang lebih sesuai seperti seberapa sering kosakata digunakan pada kehidupan sehari-hari dan tingkat kesulitan pengucapan kosakata yang ditentukan dari individu atau organisasi yang memiliki sertifikasi dibidang bahasa Inggris atau berdasarkan penelitian.
5. Menambahkan fitur interaktif tanya jawab antara pengguna dengan aplikasi menggunakan bahasa Inggris.

DAFTAR REFERENSI

- Bender, R., 2009. *Requirements Based Testing Process Overview*. New York: Bender RBT Inc.
- Brooke, J. 1986. SUS – A Quick And Dirty *Usability* Scale. [online] tersedia di :< <https://hell.meiert.org/core/pdf/sus.pdf>>.
- Brooke, J., 2013. *SUS: A Retrospective*. *Journal of Usability Studies*, [e-journal] 8, 29-40. Tersedia melalui *Journal Of Usability Studies* <http://uxpajournal.org/wp-content/uploads/pdf/JUS_Brooke_February_2013.pdf>.
- Developer. (2019). Mengenal Android Studio. [online] Tersedia di: <<https://developer.android.com/studio/intro?hl=ID>>.
- Jefrizal., Jaroji. dan Tedyyana, A., 2017. Aplikasi *English teacher* Sebagai Alat Bantu Belajar English Conversation Berbasis Android dengan Menerapkan *Voice Recognition*. *Inovtek Polbeng*, 2(2).
- Komarudin, M., 2016. Pengujian Perangkat Lunak Metode *Black-box* Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Mikrotik*, 6(3).
- Mustaqbal, M.S., Firdaus, R.F. dan Rahmadi, H., 2015. Pengujian Aplikasi Menggunakan *Black box testing* Boundary Value Analysis. *Ilmiah Teknologi Informasi Terapan*, 1(3).
- Nielsen, j., 2000. Why You Only Need to Test with 5 Users, [online] Tersedia di: <<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>>.
- Pressman, R. S. 2001. *Software Engineering A Practitioners* 5th Edition. New York : McGraw-Hill.
- Rifa'atunnisa., Satria, E. dan Cahyana, R., 2014. Pengembangan Aplikasi Zakat Berbasis Android Menggunakan Metode *Prototype*. *Algoritme*, 11(1).
- Rubin, J. & Chisnell, D. 2008. *Handbook of Usability testing : how to plan , design, and conduct effective test* (2nd ed.). Canada : Wiley Publishing.
- Salma, A., Darmawan, I. dan Al-Anshary, F.M., 2017. Perancangan Aplikasi Callme Berbasis Android Menggunakan Metode *Prototyping* (Modul Administrasi *Customer* Dan Admin). *e-Proceeding of Engineering*, 4(2).
- Supriyanta., Widodo, P. dan Susanto, B.K., 2014. Aplikasi Konversi Suara Ke Teks Berbasis Android Menggunakan *Google speech API*. *Bianglala Informatika*, 2(2).
- Subhiyakto, E.R. dan Utomo, D.W., 2017. Analisis Dan Perancangan Aplikasi Pemodelan Kebutuhan Perangkat Lunak Menggunakan Metode *Prototyping*. *Prosiding Seminar Nasional Multi Disiplin Ilmu dan Call For Papers Unisbank*, 3(3).

Suryadharma, I.K., Budiman, G. dan Irawan, B., 2014. Perancangan Aplikasi *Speech to text* Bahasa *Inggris* Ke Bahasa Bali Menggunakan Pocketsphinx Berbasis Android. e-Proceeding of Engineering, 1(1).

